```
                .TITLE  CNDMEA0 DMV11 LINE UNIT DIAG3
                .SBTTL  PROGRAM DOCUMENT
                .REM @

                                IDENTIFICATION
                                --------------

PRODUCT CODE:   AC-T834A-MC

PRODUCT NAME:   CNDMEA0 DMV-11 LINE UNIT STATIC DIAGNOSTIC PART 03

PRODUCT DATE:   APRIL 1984

MAINTAINER:     ISS DIAGNOSTICS

AUTHORS:        CHRIS BRIENEN
                DAVE HOFFMAN
                RAY MARSHALL

MODIFIED BY:    JAKI BERG       9-APR-1984

PURPOSE:        THIS DIAGNOSTIC IS DESIGNED TO PERFORM STATIC LOGIC TESTS FOR
                THE M8053 OR M8064 (HEREAFTER REFERRED TO AS THE DMV OR DMV-11)


        THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
        NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
        EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT CORPORATION ASSUMES NO
        RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

        NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
        SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
        AFFILIATED COMPANIES.

        COPYRIGHT (C) 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

        DIGITAL         PDP             UNIBUS          MASSBUS
        DEC             DECUS           DECTAPE
```

```
47
48
49                                  ***** MODIFICATION HISTORY *****
50
51      REV A: ORIGINAL RELEASE                  BRIENEN, HOFFMAN, MARSHALL      14-JAN-81
52
53      REV B: INSTALLED OUTSTANDING PATCHES                                     11-JUL-83
54
55      CVDME6 => CNDMEA                                    JAKI BERG            9-APR-84
56      CHANGES WERE MADE TO  CVDME8  TO PRODUCE  CNDMEA  FOR THE  FALCON-PLUS PROJECT
57      (SBC 11/21+).   CHANGES, MARKED BY ";JB REV A-0", ARE:
58       - SET THE  ODT BREAK  VECTOR (LOCATION 140) TO THE STARTING ADDRESS OF
59         FALCON'S ODT ROM (170000-OCTAL).
60
```

PROGRAM DOCUMENT

CONTENTS

1.0 INTRODUCTION

THE M8053 AND M8064 ARE SINGLE-LINE SYNCHRONOUS, MICRO-PROCESSOR BASED COMMUNICATIONS INTERFACES WHICH CAN SUPPORT BOTH CHARACTER-ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED (SDLC, HDLC, ETC.) PROTOCOLS. THE PURPOSE OF THIS PROGRAM IS TO PERFORM STATIC DIAGNOSTIC TESTING OF THE VIA, FIFO, USYRT (BCP/BOP MODES), AND LINE DRIVERS ON THESE BOARDS. NOTE THAT ALL EXTERNAL LOOPBACK (XLB) TESTS ARE CONTAINED HERE. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: MODEM LOOPBACK AND ASSORTED EXTERNAL LOOPBACK TESTS (INCLUDING BCP:CRC-16/ODD VRC/EVEN VRC; BOP:CRC-CCITT-1'S/0'S).

THE STATIC LOGIC TESTS WILL PROVIDE EXTENSIVE TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS, SWITCH OPTIONS, AND ABILITY TO "LOCK" ONTO INTERMITTENT ERRORS. IN ADDITION TESTS ARE DESIGNED AND STRUCTURED TO ACHIEVE MAXIMUM FAULT RESOLUTION AND FACILITATE REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.

THIS PROGRAM IS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR AND A STRUCTURED PROGRAMMING APPROACH. BECAUSE THE DESIGN CONFORMS TO THE SUPERVISOR (STANDALONE VERSION) THE PROGRAM IS COMPATIBLE WITH ACT, APT, XXDP+, AND SLIDE.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM ALLOWS MODIFICATION OF DEVICE PARAMETERS, SUCH AS LSI-BUS ADDRESS, VECTOR ADDRESSES AND DEVICE PRIORITY. IN ADDITION, THE OPERATOR CAN SPECIFY PARTICULAR TESTS TO BE RUN AND A VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.

DEVICE ERRORS WILL BE REPORTED AS THEY OCCUR. THE REPORT WILL INCLUDE A TEST NUMBER AND DESCRIPTION OF THE ERROR, GOOD AND BAD TEST DATA, AND APPLICABLE DEVICE REGISTER CONTENTS.

2.0 HARDWARE REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8053/8064 STATIC LOGIC TESTS:

SBC-11/21+
16K WORDS OF MEMORY
CONSOLE TERMINAL
M8053 OR M8064 COMMUNICATIONS INTERFACE

THE FOLLOWING HARDWARE IS REQUIRED TO FULLY TEST THE DMV-11 LINE DRIVERS:

H3254, H3255 LOOPBACK CONNECTORS

3.0 PRELIMINARY PROGRAM REQUIREMENTS

PROGRAM DOCUMENT

```
    167                         THIS  PROGRAM  (CNDME)  SHOULD  BE THE LAST OF  THE FIVE DMV-11
    168                         STATIC DIAGNOSTICS TO BE RUN.
```

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE MAXIMUM TIME REQUIRED TO RUN THIS PROGRAM IS ABOUT 15 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM.

4.7 ERROR LOGGING

AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR RESTART COMMAND.

5.0 PROGRAM LOAD MEDIA

THIS PROGRAM CAN BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY

THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+, WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
B) RECEIVE DIAGNOSTIC SUPERVISOR IDENTIFICATION AND PROMPT (DRS-C>)
C) ENTER STA<CR>
D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
E) GET END OF PASS MESSAGES OR ERROR MESSAGES
F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED, THE FOLLOWING IDENTIFICATION IS TYPED :

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CNDME A-0
DMV-11 LINE UNIT TESTS - PART 3 OF 3
UNIT IS M8053 OR M8064
DR>
```

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3. (FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

PROGRAM DOCUMENT

6.3.1 START COMMAND

```
**********************************************************************
STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
   <FLAG-LIST>/SOP:<INCR>
**********************************************************************
```

6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)

<TEST-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. TESTS WILL BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF THE ORDER OF SPECIFICATION. THE DEFAULT IS TO EXECUTE ALL TESTS. ON THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION USED IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE OPERATOR. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.2 PASS SWITCH (/PASS:<PASS CNT>)

<PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER OF PASSES. A PASS IS DEFINED AS THE EXECUTION OF THE FULL DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG LIST>)

<FLAG-LIST> IS A SEQUENCE OF ELEMENTS OF THE FORM <FLAG>, <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS ONE OF THE FOLLOWING VALUES:

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR
IER	INHIBIT ERROR REPORTING
IBE	INHIBIT BASIC ERROR REPORTS
IXE	INHIBIT EXTENDED ERROR REPORTS
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER
PNT	PRINT NUMBER OF TEST BEING EXECUTED
BOE	BELL ON ERROR
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION TESTS
ISR	INHIBIT STATISTICAL REPORTS
IDU	INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
LOT	LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF 6.3.1.5.

6.3 1.4 END OF PASS SWITCH (/EOP:<INCR>)

<INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF PASSES) IT IS DESIRED THAT THE END OF PASS MESSAGE BE PRINTED. THE DEFAULT IS AT THE END OF EVERY PASS. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE PARAMETER DIALOGUE, THE SOFTWARE PARAMETER DIALOGUE, AND THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "# UNITS?" TO WHICH THE OPERATOR REPLIES WITH A DECIMAL NUMBER N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE DEVICE TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL BE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING ALL THE HARDWARE INFORMATION FOR ONE UNIT. THE OPERATOR MUST SUPPLY N (NUMBER OF UNITS) VALUES FOR EACH QUESTION. HE MAY DO THIS BY GIVING ONE ANSWER TO EACH QUESTION (IN WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR BY GIVING N VALUES, SEPARATED BY COMMAS, TO EACH QUESTION (SERIES WILL BE POSED ONCE). EACH QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT VALUE AFTER THE PARENTHESES.

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION "# UNITS?" IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

STA/TESTS:1:2 4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST

THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

```
**********************************************************************
RES(TART)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
   <FLAG-LIST>/UNITS:<UNIT-LIST>
**********************************************************************
```

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

<TEST-LIST>, <PASS-CNT>, AND <FLAG-LIST> ARE AS IN THE START COMMAND.

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

6.3.2.3 EFFECT OF RESTART COMMAND

THE RESTART COMMAND DIFFERS FROM THE START COMMAND IN THAT THE P-TABLES FROM THE PREVIOUS START COMMAND (THERE MUST HAVE BEEN ONE) ARE USED, INSTEAD OF NEW ONES BEING BUILT. THE UNITS SWITCH GIVES THE ABILITY TO SELECT A SUBSET OF THESE. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

```
**********************************************************************
CON(TINUE)/PASS:<PASS CNT/FLAGS:<FLAG LIST>
**********************************************************************
```

6.3.3.1 PASS SWITCH (/PASS:<PASS CNT>)

<PASS CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS CNT FROM THE PREVIOUS START OR RESTART.

IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.3.3 EFFECT OF CONTINUE COMMAND

CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.

6.3.4 PROCEED COMMAND

****
PRO(CEED)/FLAGS:<FLAG-LIST>
****

6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)

<FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED FLAGS RETAIN THEIR CURRENT VALUE.

6.3.4.2 EFFECT OF PROCEED COMMAND

PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE PARAMETERS MAY BE ALTERED.

6.3.5 ADD COMMAND

****
ADD/UNITS:<UNIT LIST>
****

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER

HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

****
DRO(P)/UNITS:<UNIT-LIST>
****

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 EFFECT OF DROP COMMAND

THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING. THE UNITS WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR START COMMAND. THE UNITS SWITCH MUST BE ENTERED. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.

6.3.7 PRINT COMMAND

****
PRI(NT)
****

6.3.7.1 EFFECT OF PRINT COMMAND

THE TOTAL NUMBER OF ERRORS FOR EACH UNIT SINCE THE LAST START OR RESTART COMMAND ARE PRINTED. THE ISR (INHIBIT STATISTICAL REPORTING) FLAG IS CLEARED.

6.3.8 DISPLAY COMMAND

****
DIS(PLAY)/UNITS:<UNIT-LIST>
****

6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO

DESIGNATED.

6.3.9 FLAGS COMMAND

****
FLA(GS)
****

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

****
ZFL(AGS)
****

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- HARD CORE QUESTIONS (SEE 6.2),HARDWARE DIALOGUE (SEE 6.3.1.5) OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING 3 QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

1. DEVICE CSR ADDRESS : (O) 160020?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE LSI BUS. THE ALLOWABLE RANGE IS 160020-177760 (OCTAL), AND THE DEFAULT VALUE IS 160020.

2. DEVICE VECTOR ADDRESS : (O) 300 ?

THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE DEFAULT VALUE IS 300.

3. DEVICE PRIORITY LEVEL : (0) 4 ?

THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF THIS DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7, AND THE DEFAULT VALUE IS 4.

4. BOARD TYPE (0=M8064, 1=M8053-V35, 2=M8053-EIA) : (0) 0 ?

THIS IS THE TYPE OF DMV-11 CURRENTLY INSTALLED. NOTE THAT THE M8053 IS SWITCH SELECTABLE BETWEEN V.35 AND EIA.

5. TURNAROUND CONNECTOR TYPE :
(0=M3254&M3255, 1=INTEGRAL MODEM CABLE, 2=EIA CABLE,
3=V.35 CABLE, 4=NONE) : (0) 0 ?

THIS IS THE TYPE OF EXTERNAL LOOPBACK CONNECTOR BEING USED. IF NO LOOPBACK CONNECTOR IS PRESENT (4), THE EXTERNAL LOOP-BACK TESTS WILL ALL BE RUN USING TTL-INTERNAL LOOPBACK.

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY THIS TEST.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
@ UNITS (D) ? 16
UNIT 0
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 7
<QUESTION 1> ?
<QUESTION 2> ? 7-11,,13-15
<QUESTION 3> ? 77
```

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

```
                        7.0 TEST DESCRIPTIONS


                        ;*****************************************************************************
                        ;*      TEST 1  <RX DATA FLUSHING TEST>
                        ;*
                        ;*  IN BCP MODE/HALF DUPLEX IT IS DESIRABLE TO HAVE THE ABILITY TO FLUSH
                        ;*  THE USYRT OF ITS CRC CHARACTERS. THIS FLUSHING IS ACCOMPLISHED BY WRITING
                        ;*  TO THE VIA SHIFT REGISTER.
                        ;*  THIS TEST VERIFIES THAT WHEN THE VIA SR IS WRITTEN INTO, 8 PULSES WILL
                        ;*  BE GENERATED AT THE CB1 PIN (WHICH DIRECTLY FEEDS THE CHARACTER FIFO).
                        ;*
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      TEST 2  <INTEGRAL MODEM INTERFACE TEST>
                        ;*
                        ;*  THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM AND A MESSAGE IS
                        ;*  TRANSMITTED,  RECEIVED,  AND CHECKED USING A TURNAROUND CONNECTOR ON
                        ;*  THE  BOARD  OR  AT THE END OF A CABLE. THE FOLLOWING MESSAGE WILL BE
                        ;*  SENT IN BCP MODE WITH CRC-16 SPECIFIED:
                        ;*
                        ;*      SYNC SYNC 000 125 252 377 000 CRC1 CRC2 SYNC
                        ;*
                        ;*  IF  THE  P-TABLE  FOR  THE  CURRENT  UNIT INDICATES THAT NO EXTERNAL
                        ;*  TURNAROUND IS PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      TEST 3  <DATA TEST -- BCP XLB CRC-16>
                        ;*
                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                        ;*  RECEIVE  IN  BCP  MODE  WITH  CRC-16  ERROR  DETECTION THE FOLLOWING
                        ;*  MESSAGE:
                        ;*
                        ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
                        ;*     357 337 277 177
                        ;*
                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
                        ;*  THREE  TIMES WITH CRC'S FOLLOWING EACH ONE. THE LAST TRANSMISSION OF
                        ;*  THE  CRC  WILL BE FOLLOWED BY SEVERAL SYNC CHARACTERS BEFORE DROPPING
                        ;*  TXE & RXE. 8-BIT CHARACTER LENGTHS ARE ALSO UTILIZED.
                        ;*
                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      TEST 4  <DATA TEST -- BCP XLB ODD VRC>
                        ;*
                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                        ;*  RECEIVE  IN  BCP  MODE  WITH  ODD  VRC ERROR DETECTION THE FOLLOWING
```

```
                        ;*  MESSAGE:
                        ;*
                        ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
                        ;*     357 337 277 177
                        ;*
                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
                        ;*  THREE  TIMES.  AFTER  THE  LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
                        ;*  SENT  BEFORE  DROPING  TXE  &  RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
                        ;*  UTILIZED.
                        ;*
                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                        ;*********************************************************************


                        ;*********************************************************************
                        ;*      TEST 5  <DATA TEST -- BCP XLB EVEN VRC>
                        ;*
                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                        ;*  RECEIVE  IN  BCP  MODE  WITH  EVEN VRC ERROR DETECTION THE FOLLOWING
                        ;*  MESSAGE:
                        ;*
                        ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
                        ;*     357 337 277 177
                        ;*
                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
                        ;*  THREE  TIMES.  AFTER  THE  LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
                        ;*  SENT  BEFORE  DROPING  TXE  &  RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
                        ;*  UTILIZED.
                        ;*
                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                        ;*********************************************************************


                        ;*********************************************************************
                        ;*      TEST 6  <DATA TEST -- BOP XLB CRC-CCITT-1>
                        ;*
                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                        ;*  RECEIVE  IN  BOP MODE WITH CRC-CCITT-1 ERROR DETECTION THE FOLLOWING
                        ;*  SHORT MESSAGE:  125  252  000 377 001
                        ;*
                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY FLAG CHARACTERS AND REPEATED
                        ;*  THREE  TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
                        ;*  LENGTHS ARE ALSO UTILIZED.
                        ;*
                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                        ;*********************************************************************


                        ;*********************************************************************
                        ;*      TEST 7  <DATA TEST -- BOP XLB CRC-CCITT-0>
                        ;*
                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                        ;*  RECEIVE  IN  BOP MODE WITH CRC-CCITT-0 ERROR DETECTION THE FOLLOWING
                        ;*  SHORT MESSAGE:  125  252  000  377  001
```

```
855                             ;*
856                             ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY FLAG CHARACTERS AND REPEATED
857                             ;*  THREE  TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
858                             ;*  LENGTHS ARE ALSO UTILIZED.
859                             ;*
860                             ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
861                             ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
862                             ;*********************************************************************************
863
864
865                             ;*********************************************************************************
866                             ;*      TEST 8  <MODEM CONTROL SIGNAL LOOPBACK TEST>
867                             ;*
868                             ;*  FIRST, THE DMV-11 IS INITIALIZED. THEN, TTL LOOPBACK IS SELECTED,
869                             ;*  AND THE FOLLOWING CHECKS ARE PERFORMED INVOLVING THE MODEM STATUS
870                             ;*  REGISTER :
871                             ;*  - RING, CARRIER, MODEM READY, TEST MODE, CTS ARE CHECKED FOR 1 STATE.
872                             ;*  - RTS IS DE-ASSERTED AND CTS IS CHECKED FOR 0.
873                             ;*  - RTS IS ASSERTED AND CTS IS CHECKED FOR 1.
874                             ;*
875                             ;*  NEXT, IF THE OPTION IS AN M8053 WITH AN H3254 TEST CONNECTOR INSTALLED,
876                             ;*  THE DMV-11 IN INITIALIZED AGAIN, (TTL LOOPBACK IS CLEARED), AND
877                             ;*  THE FOLLOWING CHECKS ARE PERFORMED :
878                             ;*  - RING, CARRIER, MODEM READY, CTS ARE CHECKED FOR 1, TEST MODE IS CHECKED
879                             ;*    FOR 0.
880                             ;*  - RTS IS DE-ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 0.
881                             ;*  - RTS IS ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 1.
882                             ;*  - DTR IS DE-ASSERTED, AND MODEM READY IS CHECKED FOR 0.
883                             ;*  - DTR IS ASSERTED, AND MODEM READY IS CHECKED FOR 1.
884                             ;*********************************************************************************
885
886
887                             ;*********************************************************************************
888                             ;*      TEST 9  <DDCMP MESSAGE TEST>
889                             ;*
890                             ;*  THIS  TEST  WILL  USE XLB IF IT IS ENABLED -- OTHERWISE TTL LOOPBACK
891                             ;*  WILL  BE  UTILIZED.  THIS  ASSURES  THAT  IT  CAN ALWAYS BE RUN AS A
892                             ;*  GENERAL "RINGOUT" OF THE M8053.
893                             ;*
894                             ;*  INITIALIZATION:  BCP  MODE, CRC-16, IDLE = 0, SYNC (S/AR) = 226 OCT.
895                             ;*  (96 HEX.), RXCL & TXCL = 0 (CHAR. LENGTH = 8).
896                             ;*
897                             ;*  THE FOLLOWING SAMPLE DDCMP MESSAGE IS TRANSMITTED & RECEIVED AND ALL
898                             ;*  DATA AND CRC CHARACTERS ARE CHECKED FOR ERRORS;
899                             ;*
900                             ;*                    ... HEADER ...  ...  ... DATA (PATTERN K) ...
901                             ;*  SYNC SYNC 201 000 075 003 002 001 CRC CRC 000 377 ... 252 000 CRC CRC
902                             ;*
903                             ;*  THE  ATTEMPT  HERE  IS TO PROVIDE A TEST JUST BELOW THE LEVEL OF THE
904                             ;*  FUNCTIONAL  DIAGNOSTIC.  THE  USTRT  WILL BE RESPONSIBLE FOR ALL CRC
905                             ;*  GENERATION  AND  VERIFICATION BUT THE CRC'S WILL ALSO BE VERIFIED BY
906                             ;*  SOFTWARE.
907                             ;*********************************************************************************
908
909
```

```
911
912                          8.0 ERROR INFORMATION
913
914
915                          8.1 ERROR REPORTING
916
917                          ERRORS  ARE  REPORTED  BY  THE PROGRAM AS THEY OCCUR (IF NOT
918                          INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR
919                          ERROR  REPORT  FORMAT,  AND CONSISTS OF A DESCRIPTION OF THE
920                          ERROR, THE TEST NUMBER, SUBTEST  NUMBER,  PC  OF  THE  ERROR
921                          CALL,  DEVICE  ADDRESS,  AND  BASIC  AND  EXTENDED  ERROR
922                          INFORMATION.
923
924                          THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH
925                          DESCRIBES A "MASTER CLEAR FAILURE" ERROR, AND PROVIDES THE PC
926                          OF THE ERROR CALL AND THE DEVICE REGISTER CONTENTS :
927
928                            CNDMB DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
929                            MASTER CLEAR FAILURE
930
931                              THE CONTENTS OF ALL BYTE SELECT REG'S ARE:
932                            BSEL0   BSEL1   BSEL2   BSEL3
933                             000     000     000     000
934                                BSEL4   BSEL5   BSEL6   BSEL7
935                                 000     000     121     000
936                            BSEL10  BSEL11  BSEL12  BSEL13
937                             000     000     000     000
938                                BSEL14  BSEL15  BSEL16  BSEL17
939                                 000     000     000     000
940
941                          FOR OTHER ERRORS, THE REPORT  MAY  BE  MORE  EXTENSIVE,  AND
942                          REQUIRE ADDITIONAL DATA TO BE REPORTED.
943
944                          IF  EXTENDED  ERROR INFORMATION HAD BEEN INHIBITED USING THE
945                          IXE FLAG PRIOR TO RUNNING THE TEST, THE  ABOVE  ERROR  WOULD
946                          HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :
947
948                            CNDMB DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
949                            MASTER CLEAR FAILURE
950
951                                  a
952
```

```
                                .SBTTL GENERAL EQUATES AND DS INVOCATION & SETUP


000000                          HELP=0          ; CONTROL LISTING OF HELP INFORMATION
                                                ;
                                                ; HELP=0   NO LIST
                                                ; HELP=1   LIST

002000                                  .=2000

                                         MCALL  SVC
002000                                  SVC     ; INITIALIZE SUPERVISOR MACROS


002000                                  BGNMOD  LU1MOD


000001                          $LSTIN= 1
000001                          $LSTTAG= 1
000001                          SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
000001                          SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
000001                          SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
000001                          SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
000001                          SVCTAG= 1       ; LIST OTHER TAGS, SHIFTED RIGHT

                                ;       CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
                                ;       TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
                                ;       SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
                                ;       CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
```

PROGRAM HEADER

```
     989                            .SBTTL  PROGRAM HEADER
     990                            ;++
     991                            ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
     992                            ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
     993                            ;--
     994
     995 002000                             POINTER BGNAU,BGNDU,ERRTBL
     996
    1004
    1005 002000                             HEADER  CNDME,A,0,15.,0
         002000                     L$NAME::
         002000     103                             .ASCII  /C/
         002001     116                             .ASCII  /N/
         002002     104                             .ASCII  /D/
         002003     115                             .ASCII  /M/
         002004     105                             .ASCII  /E/
         002005     000                             .BYTE   0
         002006     000                             .BYTE   0
         002007     000                             .BYTE   0
         002010                     L$REV::
         002010     101                             .ASCII  /A/
         002011                     L$DEPO::
         002011     060                             .ASCII  /0/
         002012                     L$UNIT::
         002012  000000                             .WORD   0
         002014                     L$TIML::
         002014  000017                             .WORD   15.
         002016                     L$HPCP::
         002016  033700                             .WORD   L$HARD
         002020                     L$SPCP::
         002020  000000                             .WORD   0
         002022                     L$HPTP::
         002022  002150                             .WORD   L$HW
         002024                     L$SPTP::
         002024  000000                             .WORD   0
         002026                     L$LADP::
         002026  034542                             .WORD   L$LAST
         002030                     L$STA::
         002030  000000                             .WORD   0
         002032                     L$CO::
         002032  000000                             .WORD   0
         002034                     L$DTYP::
         002034  000000                             .WORD   0
         002036                     L$APT::
         002036  000000                             .WORD   0
         002040                     L$DTP::
         002040  002124                             .WORD   L$DISPATCH
         002042                     L$PRIO::
         002042  000000                             .WORD   0
         002044                     L$ENVI::
         002044  000000                             .WORD   0
         002046                     L$EXPI::
         002046  000000                             .WORD   0
         002050                     L$MREV::
         002050     003                             .BYTE   C$REVISION
         002051     003                             .BYTE   C$EDIT
         002052                     L$EF::
```

PROGRAM HEADER

```
002052  000000                          .WORD   0
002054  000000                          .WORD   0
002056                          L$SPC::
002056  000000                          .WORD   0
002060                          L$DEVP::
002060  003264                          .WORD   L$DVTYP
002062                          L$REPP::
002062  000000                          .WORD   0
002064                          L$EXP4::
002064  000000                          .WORD   0
002066                          L$EXP5::
002066  000000                          .WORD   0
002070                          L$AUT::
002070  023256                          .WORD   L$AU
002072                          L$DUT::
002072  023252                          .WORD   L$DU
002074                          L$LUN::
002074  000000                          .WORD   0
002076                          L$DESP::
002076  003304                          .WORD   L$DESC
002100                          L$LOAD::
002100  104035                          EMT     E$LOAD
002102                          L$ETP::
002102  002172                          .WORD   L$ERRTBL
002104                          L$ICP::
002104  022554                          .WORD   L$INIT
002106                          L$CCP::
002106  023250                          .WORD   L$CLEAN
002110                          L$ACP::
002110  023124                          .WORD   L$AUTO
002112                          L$PRT::
002112  022546                          .WORD   L$PROT
002114                          L$TEST::
002114  000000                          .WORD   0
002116                          L$DLY::
002116  000000                          .WORD   0
002120                          L$HIME::
002120  000000                          .WORD   0
1006
1012
1013                            .EVEN
```

```
1015                                    .SBTTL  DISPATCH TABLE
1016
1017 002122                             SLASH
                                        ;;//////////////////////////////////////////////////////////////////////
1018                                    ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1019                                    ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1020 002122                             SLASH
                                        ;;//////////////////////////////////////////////////////////////////////
1021
1022 002122                                     DISPATCH 9.
     002122  000011                                                          .WORD   9
     002124                                                        L$DISPATCH::
     002124  023260                                                 .WORD   T1
     002126  023560                                                 .WORD   T2
     002130  026264                                                 .WORD   T3
     002132  027142                                                 .WORD   T4
     002134  027704                                                 .WORD   T5
     002136  030446                                                 .WORD   T6
     002140  031130                                                 .WORD   T7
     002142  031612                                                 .WORD   T8
     002144  032556                                                 .WORD   T9
1023
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

                                ;////////////////////////////////////////////////////////////////////////////
                                ;/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ;/ THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ;/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
                                ;////////////////////////////////////////////////////////////////////////////

002146                                  BGNHW   DFPTBL
002146  000010                                                                  .WORD   L10000-L$HW/2
002150                                                                  L$HW::
002150                                                                  DFPTBL::

002150  160020                          .WORD   160020          ;DMV11 CSR UNIBUS ADDRESS
002152  000300                          .WORD      300          ;DMV11 INTERRUPT VECTOR
002154  004000                          .WORD     4000          ;DMV11 INTERRUPT PRIORITY LEVEL = 4
002156  000000                          .WORD      000          ;SWITCH REG. #1 (BOOT ADDRESS)
002160  000000                          .WORD      000          ;SWITCH REG. #2 (DDCMP ADDRESS)
002162  000000                          .WORD        0          ;MODULE IS M8064
002164  000000                          .WORD        0          ;H3254&H3255 USED
002166  000001                          .WORD        1          ;BAUD RATE = 56 K
                                                                ;       0 = 19.2 K
                                                                ;       1 = 56 K

002170                                  ENDHW
002170                                                          L10000:
```

```
1054                                    .SBTTL  SOFTWARE P TABLE
1055
1056                                    ;////////////////////////////////////////////////////////////////////
1057                                    ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1058                                    ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1059                                    ;////////////////////////////////////////////////////////////////////
1060
1061 002170                                    BGNSW   SFPTBL
     002170  000000                                                                    .WORD   L10001-L$SW/2
     002172                                                                     L$SW::
     002172                                                                     SFPTBL::
1062
1063 002172                                    ENDSW
     002172                                                                     L10001:
```

GLOBAL EQUATES SECTION -- BASIC EQUATES

```
                                .SBTTL  GLOBAL EQUATES SECTION -- BASIC EQUATES


                                ;////////////////////////////////////////////////////////////////////////////////
                                ;/      THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
                                ;/      ARE USED IN MORE THAN ONE TEST.
                                ;////////////////////////////////////////////////////////////////////////////////

002172                                  EQUALS
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
                                ;                                       ; BIT POSITION IN SECOND STATUS WORD
000040                          EF.START==      32.                     ; (100000) START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.                     ; (040000) RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.                     ; (020000) CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.                     ; (010000) A NEW PASS HAS BEEN STARTED
000034                          EF.PWR==        28.                     ; (004000) A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
000340                          PRI07== 340
000300                          PRI06== 300
000240                          PRI05== 240
000200                          PRI04== 200
```

GLOBAL EQUATES SECTION -- BASIC EQUATES

```
000140                  PRIO3== 140
000100                  PRIO2== 100
000040                  PRIO1== 40
000000                  PRIO0== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==     4
000010                  LOT==    10
000020                  ADR==    20
000040                  IDU==    40
000100                  ISR==   100
000200                  UAM==   200
000400                  BOE==   400
001000                  PNI==  1000
002000                  PRI==  2000
004000                  IXE==  4000
010000                  IBE== 10000
020000                  IER== 20000
040000                  LOE== 40000
100000                  HOE== 100000
```

REGISTER DEFINITIONS -- MAINTENANCE REGISTERS -- SELN & BSELN

```
                        .SBTTL  REGISTER DEFINITIONS -- MAINTENANCE REGISTERS -- SELN & BSELN

                        ;;*****************************************************************************
                        ;* MAINTENANCE REGISTER # 0 - BSEL0
                        ;;*****************************************************************************
000020                  IEO     = BIT4          ;"INTERRUPT ENABLE OUT"
000001                  IEI     = BIT0          ;"INTERRUPT ENABLE IN"

                        ; BIT 7 IS ALSO USED BY THE MICROCODE.  ITS LABEL IS "RQI" WHICH STANDS FOR
                        ; "REQUIST IN".  IT'S PART OF THE HANDSHAKING FOR USING THE SEL & BSEL REG'S.
                        ; HOWEVER, THE MAINT. LOOP DOES NOT MAKE USE OF THIS BIT AND IT IS THEREFORE
                        ; UNNECESSARY TO DEFINE IT HERE.

                        ;;*****************************************************************************
                        ;* MAINTENANCE REGISTER # 1 - BSEL1
                        ;;*****************************************************************************
000200                  RUN     = BIT7          ;"RUN" & ALSO CONTROLS 6502 MICROPROCESSOR'S RDY STATE
000100                  MCLR    = BIT6          ;MASTER CLEAR
000001                  MREQ    = BIT0          ;M-LOOP ACCESS
000301                  STRTMLOP= RUN!MCLR!MREQ  ;INITIATE M-LOOP

                        ;;*****************************************************************************
                        ;* MAINTENANCE REGISTER # 2 - BSEL2
                        ;;*****************************************************************************
000200                  MRDY    = BIT7          ;M-LOOP READY

                        ;*****************************************************************************
                        ;* MAINTENANCE LOOP COMMAND DEFINITIONS
                        ;*****************************************************************************
000001                  REDLOC  = 1             ;READ LOC. W/IN DMV-11 ---- (SEL4) ***> BSEL6
000002                  WRILOC  = 2             ;WRITE LOC. W/IN DMV-11 --- BSEL6 ***> (SEL4)
000003                  REDPAG  = 3             ;READ BLOCK W/IN DMV-11 --- (SEL6) ***> (SEL4)
000004                  WRIPAG  = 4             ;WRITE BLOCK W/IN DMV-11 -- (SEL4) ***> (SEL6)
000005                  EXECUT  = 5             ;SET 6502'S PC AND EXECUTE  - SEL6 ***> PC
000007                  DOTBMT  = 7             ;SET MAINTENANCE INTERRUPT DISABLE IN PROCESSOR
                                                ;STATUS --- (KB7) ***> BSEL3
```

```
                        .SBTTL  REGISTER DEFINITIONS -- USYRT


        120400          USYRT = 120400          ;USYRT BASE ADDRESS = A100 (HEX)

                        ;;*****************************************************************************
                        ;* USYRT "RECEIVER DATA BUFFER" REGISTER -- READ ONLY
                        ;;*****************************************************************************

        120400          RDSRL   = 120400        ;ADDRESS OF THIS REG

                        ;;*****************************************************************************
                        ;* USYRT "RECEIVER STATUS" REGISTER -- READ ONLY
                        ;;*****************************************************************************

        120401          RDSRH   = 120401        ;ADDRESS OF THIS REG

                        ;BIT DEFINITIONS ON BYTE BASIS :
        000200          RERR    = BIT7          ;ERROR CHECK
        000160          ABC     = BIT6!BIT5!BIT4 ;ASSEMBLED BIT COUNT
        000010          ROR     = BIT3          ;RECEIVER OVER RUN
        000004          RABGA   = BIT2          ;RECEIVED ABORT/GA CHARACTER
        000002          REOM    = BIT1          ;RECEIVED END-OF-MESSAGE
        000001          RSOM    = BIT0          ;RECEIVED START-OF-MESSAGE

                        ;BIT DEFINITIONS ON WORD BASIS :
        100000          RXERR   = BIT15         ;RECEIVED CRC/VRC ERROR
        004000          RXOR    = BIT11         ;RECEIVER OVER RUN
        002000          RXABGA  = BIT10         ;RECEIVED ABORT/GO AHEAD CHARACTER
        001000          RXEOM   = BIT9          ;RECEIVED END-OF-MESSAGE
        000400          RXSOM   = BIT8          ;RECEIVED START-OF-MESSAGE

        000001          RERCHK  = BIT0          ;FLAG TO INVOKE RERR CHK IN SUBROUTINE RXCHAR

                        ;;*****************************************************************************
                        ;* USYRT "TRANSMITTER DATA BUFFER" REGISTER
                        ;;*****************************************************************************

        120402          TDSRL   = 120402        ;ADDRESS OF THIS REG

                        ;;*****************************************************************************
                        ;* USYRT "TX STATUS AND CONTROL" REGISTER
                        ;;*****************************************************************************

        120403          TDSRH   = 120403        ;ADDRESS OF THIS REG

                        ;BIT DEFINITIONS ON BYTE BASIS :
        000200          TERR    = BIT7          ;TRANSMITTER UNDERRUN ERROR
        000010          TGA     = BIT3          ;TRANSMIT GO AHEAD
        000004          TAB     = BIT2          ;TRANSMIT ABORT
        000002          TEOM    = BIT1          ;TRANSMIT END-OF-MESSAGE
        000001          TSOM    = BIT0          ;TRANSMIT START-OF-MESSAGE

                        ;BIT DEFINITIONS ON WORD BASIS :
        100000          TXERR   = BIT15         ;TRANSMITTER UNDERRUN ERROR
        004000          TXGA    = BIT11         ;TRANSMIT GO AHEAD
        002000          TXAB    = BIT10         ;TRANSMIT ABORT
```

REGISTER DEFINITIONS -- USYRT

```
1170        001000              TXEOM   = BIT9          ;TRANSMIT END-OF-MESSAGE
1171        000400              TXSOM   = BIT8          ;TRANSMIT START-OF-MESSAGE
1172
1173                            ;;*****************************************************************************
1174                            ;* USYRT "SYNC/SECONDARY ADDRESS" REGISTER
1175                            ;;*****************************************************************************
1176
1177        120404              PCSARL  = 120404        ;ADDRESS OF THIS REG
1178        000226              SYNCH   = 226           ;STANDARD SYNCH CHARACTER
1179
1180                            ;;*****************************************************************************
1181                            ;* USYRT "MODE CONTROL"
1182                            ;;*****************************************************************************
1183
1184        120405              PCSARH  = 120405        ;ADDRESS OF THIS REG
1185
1186                                    ;BIT DEFINITIONS ON BYTE BASIS:
1187
1188        000200              APA     = BIT7          ;"ALL PARTIES ADDRESS" ENABLE
1189        000100              PROTO   = BIT6          ;SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
1190        000040              STRIP   = BIT5          ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1191        000020              SECAD   = BIT4          ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1192        000010              IDLE    = BIT3          ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1193        000007              XYZ     = BIT2!BIT1!BIT0 ;CRC/PARITY SELECTION CONTROL
1194
1195                                    ;BIT DEFINITIONS ON WORD BASIS:
1196
1197        100000              APA0    = BIT15         ;"ALL PARTIES ADDRESS" ENABLE
1198        040000              DDCMP   = BIT14         ;CODE FOR DDCMP MODE
1199        020000              STRIPS  = BIT13         ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1200        010000              SECADR  = BIT12         ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1201        004000              IDLES   = BIT11         ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1202        000400              CRCOS   = BIT8          ;CODE FOR CRC-CCITT-0 SELECTION
1203        001400              CRC16   = BIT9!BIT8     ;CODE FOR CRC-16 SELECTION
1204        003400              NOCHK   = BIT10!BIT9!BIT8 ;CODE FOR NO ERROR CHECKING
1205        002400              EVRC    = BIT10!BIT8    ;CODE FOR VRC EVEN CHECK
1206        002000              OVRC    = BIT10         ;CODE FOR VRC ODD CHECK
1207
1208                            ;;*****************************************************************************
1209                            ;* USYRT "DATA LENGTH SELECT" REGISTER
1210                            ;;*****************************************************************************
1211
1212        120407              PCR     = 120407        ;ADDRESS OF THIS REG
1213
1214                                    ;BIT DEFINITIONS:
1215
1216        000340              TXDL    = BIT7!BIT6!BIT5 ;TRANSMIT DATA LENGTH SELECTION
1217        000020              EXADD   = BIT4          ;EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
1218        000010              EXCON   = BIT3          ;EXTENDED CONTROL FIELD -- NOT USED OR TESTED
1219        000007              RXDL    = BIT2!BIT1!BIT0 ;RECEIVER DATA LENGTH SELECTION
1220
1221                            ;;*****************************************************************************
1222                            ;* USYRT STATUS REGISTER (ADDR. A400)
1223                            ;;*****************************************************************************
1224        122000              USTATR = 122000         ;USYRT STATUS REGISTER ADDRESS = A400 (HEX)
1225
1226                                    ;BIT DEFINITIONS:
```

REGISTER DEFINITIONS -- USYRT

```
1227
1228          000200                  RDA     = BIT7          ;RECEIVER DATA AVAILABLE
1229          000100                  TBMT    = BIT6          ;TRANSMITTER BUFFER EMPTY
1230          000040                  RXACT   = BIT5          ;RECEIVER ACTIVE
1231          000020                  RSA     = BIT4          ;RECEIVER STATUS AVAILABLE
1232          000010                  TSO     = BIT3          ;TRANSMITTER SERIAL OUTPUT
1233          000004                  TXACT   = BIT2          ;TRANSMITTER ACTIVE
1234          000002                  TXU     = BIT1          ;TRANSMITTER UNDERRUN
1235          000001                  SFR     = BIT0          ;SYNC/FLAG RECEIVED
```

REGISTER DEFINITIONS -- 6522 VIA CHIP

```
                        .SBTTL  REGISTER DEFINITIONS -- 6522 VIA CHIP

120000          VIA     = 120000                ;VIA BASE ADDRESS = A000 (HEX)

                ;;*************************************************************************
                ;* MODEM & MAINTENANCE CONTROL -- "ORB" 8 BIT PORT B -- WRITE ONLY
                ;;*************************************************************************

120000          VIAORB  = 120000                ;ADDRESS OF THIS REGISTER -- HEX = A0X0

000200          NULCLK  = BIT7                  ;"NULL CLK L" -- NULL CLOCK
000100          RXEN    = BIT6                  ;"RXENL" -- USYRT RECEIVER ENABLE
000040          TXEN    = BIT5                  ;"TXENL" -- USYRT TRANSMITTER ENABLE
000020          DTR     = BIT4                  ;"DTR" -- DATA TERMINAL READY
000010          RTSND   = BIT3                  ;"RTSND" -- REQUEST TO SEND
000004          HDX     = BIT2                  ;"HDX" -- HALF DUPLEX
000002          TTLOOP  = BIT1                  ;"SELECT TTL LEVEL LOOPBACK"
000001          PRESET  = BIT0                  ;"PRESET H" --
000000          DTRL    = 0                     ;DTR IS ASSERTED LOW

                ;;*************************************************************************
                ;* MODEM STATUS REGISTER -- "ORA" 8 BIT PORT A -- READ ONLY
                ;;*************************************************************************

120001          VIAMS   = 120001                ;ADDRESS OF THIS REGISTER -- HEX = A0X1

000200          RING    = BIT7                  ;"RING H" --
000100          CARIER  = BIT6                  ;"CARRIER H" --
000040          MDMRDY  = BIT5                  ;"MODEM RDY H" --
000020          SPEED   = BIT4                  ;"BAUD RATE SWITCH -- (19.2K/56K)
000010          CTS     = BIT3                  ;"CTS H -- CLEAR TO SEND
000004          TM      = BIT2                  ;"TEST MODE H" --
000002          RCVDAT  = BIT1                  ;"RCV DATA H" --
000001          UMAINT  = BIT0                  ; SELECT USYRT INT LOOPBACK **SELECT BIT**


                ;;*************************************************************************
                ;* DATA DIRECTION FOR PORT B -- "DDRB" -- READ/WRITE
                ;;*************************************************************************

120002          VIADPB  = 120002                ;ADDRESS OF THIS REGISTER -- HEX = A0X2

                ;  ALL BITS ARE DEFINED THE SAME:
                ;       THE BIT SETTING DEFINED THE DIRECTION OF ITS RELATED BIT IN BIT PORT B

                ;       INITIALIZED TO 377 (HEX = FF) -- PORT B IS READ/WRITE


                ;;*************************************************************************
                ;* DATA DIRECTION FOR PORT A    "DDRA" -- READ/WRITE
                ;;*************************************************************************

120003          VIADPA  = 120003                ;ADDRESS OF THIS REGISTER -- HEX = A0X3

                ;  ALL BITS ARE DEFINED THE SAME:
                ;       THE BIT SETTING DEFINED THE DIRECTION OF ITS RELATED BIT IN BIT PORT A
```

REGISTER DEFINITIONS -- 6522 VIA CHIP

```
1294                            ;       INITIALIZED TO 001 (HEX = 01) -- PORT A IS READ ONLY (EXCEPT FOR
1295                            ;                                       BIT0 WHICH ENABLES USYRT INTERNAL LOOPBACK).
1296
1297
1298
1299                            ;;*****************************  ***************************************************
1300                            ;* TIMER 1 LOW ORDER (LATCH & COUNTER) -- "T1L-L" & "T1C-L" -- WRITE & READ
1301                            ;;*****************************************************************************
1302
1303    120004                  VIAT1A  = 120004                ;ADDRESS OF THIS REGISTER -- HEX = A0X4
1304
1305                            ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1306                            ;  WHEN READING, LOW ORDER COUNTER IS READ.
1307
1308
1309
1310                            ;;*****************************************************************************
1311                            ;* TIMER 1 HIGH ORDER COUNTER & TRIGGER -- "T1L-H AND TRIGGER" & "T1C-H"
1312                            ;*       -- WRITE & READ
1313                            ;;*****************************************************************************
1314
1315    120005                  VIAT1B  = 120005                ;ADDRESS OF THIS REGISTER -- HEX = A0X5
1316
1317                            ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1318                            ;                 ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1319
1320                            ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1321
1322
1323
1324                            ;;*****************************************************************************
1325                            ;* TIMER 1 LOW ORDER LATCH -- "T1L-L"  - READ/WRITE
1326                            ;;*****************************************************************************
1327
1328    120006                  VIAT1C  = 120006                ;ADDRESS OF THIS REGISTER -- HEX = A0X6
1329
1330                            ;  THE LOW ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1331                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1332
1333
1334
1335                            ;;*****************************************************************************
1336                            ;* TIMER 1 HIGH ORDER LATCH -- "T1L-H" -- READ/WRITE
1337                            ;;*****************************************************************************
1338
1339    120007                  VIAT1D  = 120007                ;ADDRESS OF THIS REGISTER -- HEX = A0X7
1340
1341                            ;  THE HIGH ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1342                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1343
1344
1345
1346                            ;;*****************************************************************************
1347                            ;* TIMER 2 LOW ORDER (LATCH & COUNTER) -- "T2L-L" & "T2C-L" -- WRITE & READ
1348                            ;;*****************************************************************************
1349
1350    120010                  VIAT2A  = 120010                ;ADDRESS OF THIS REGISTER -- HEX = A0X8
```

REGISTER DEFINITIONS -- 6522 VIA CHIP

```
1351
1352                            ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1353                            ;  WHEN READING, LOW ORDER COUNTER IS READ.
1354
1355
1356
1357                            ;;*****************************************************************************
1358                            ;* TIMER 2 HIGH ORDER COUNTER & TRIGGER -- "T2L-H AND TRIGGER" & "T2C-H"
1359                            ;*       -- WRITE & READ
1360                            ;;*****************************************************************************
1361
1362    120011                  VIAT2B  = 120011                ;ADDRESS OF THIS REGISTER -- HEX = A0X9
1363
1364                            ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1365                            ;                ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1366
1367                            ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1368
1369                            ;;*****************************************************************************
1370                            ;* SHIFT REGISTER -- "SR" -- READ/WRITE
1371                            ;;*****************************************************************************
1372
1373    120012                  VIASR   = 120012                ;ADDRESS OF THIS REGISTER -- HEX = A0XA
1374
1375                            ;  SHIFTING IS CONTROLLED BY THE SETTING OF VIASRC (ACR2 ---> ACR4) IN VIAACR
1376
1377
1378
1379                            ;;*****************************************************************************
1380                            ;* AUXILIARY CONTROL REGISTER -- "ACR" -- READ/WRITE
1381                            ;;*****************************************************************************
1382
1383    120013                  VIAACR  = 120013                ;ADDRESS OF THIS REGISTER -- HEX = A0XB
1384
1385    000300                  T1MODE  = BIT7!BIT6             ;CONTROL THE MODE OF TIMER # 1
1386
1387                                    ;BIT 7:
1388                                    ; 0     PB7 DISABLED -- ONLY T1TO IN VIAIFR REFLECTS TIMEOUT
1389                                    ; 1     PB7 & T1TO REFLECT TIMEOUT
1390
1391                                    ;BIT 6:
1392                                    ; 0     TIMER 1 IN ONE-SHOT MODE
1393                                    ; 1     TIMER 1 IN CONTINUOUS SQUARE WAVE MODE
1394
1395    000040                  T2MODE  = BIT5                  ;CONTROLS THE MODE OF TIMER # 1
1396
1397                                    ; 0     PULSE COUNTING MODE
1398                                    ; 1     INTERVAL TIMER MODE
1399
1400    000034                  SRMODE  = BIT4!BIT3!BIT2        ;CONTROLS THE MODE OF THE SHIFT REGISTER
1401
1402                                    ; 0     SR DISABLED
1403                                    ; 1     SHIFT IN UNDER CONTROL OF T2, SHFT PULSES GEN'D ON CB1
1404                                    ; 2     SHIFT IN AT SYS. CLOCK RATE, SHFT PULSES GEN'D ON CB1
1405                                    ; 3     SHIFT IN UNDER CONTROL OF EXTERNAL INPUT PULSES
1406                                    ; 4     SHIFT OUT -- FREE RUNNING -- RATE CONTROLLED BY T2
1407                                    ; 5     SHIFT OUT -- RATE CONTROLLED BY T2 -- PULSES ON CB1
```

REGISTER DEFINITIONS -- 6522 VIA CHIP

```
                                        ; 6     SHIFT OUT -- SYS. CLOCK RATE -- PULSES ON CB1
                                        ; 7     SHIFT OUT -- UNDER CONTROL OF PULSES APPLIED TO CB1

000002                  PBLENB  = BIT1          ;PB LATCH CONTROL -- 1 ENABLES LATCH
000001                  PALENB  = BIT0          ;PA LATCH CONTROL -- 1 ENABLES LATCH



                        ;;*************************************************************************
                        ;* PERIPHERAL CONTROL REGISTER -- "PCR" -- READ/WRITE
                        ;;*************************************************************************

120014                  VIAPCR  = 120014        ;ADDRESS OF THIS REGISTER -- HEX = A0XC

000340                  CB2CTL  = BIT7!BIT6!BIT5        ;CB2 MODE SELECT
000020                  CB1CTL  = BIT4                  ;CB1 MODE SELECT
000016                  CA2CTL  = BIT3!BIT2!BIT1        ;CA2 MODE SELECT
000001                  CA1CTL  = BIT0                  ;CA1 MODE SELECT



                        ;;*************************************************************************
                        ;* INTERRUPT FLAG REGISTER -- "IFR" -- READ ONLY
                        ;;*************************************************************************

120015                  VIAIFR  = 120015        ;ADDRESS OF THIS REGISTER -- HEX = A0XD

000200                  FLGIRQ  = BIT7          ;SET WHEN A FLAG IN THIS REG. GOES HIGH AND
                                                ;ITS CORRESPONDING BIT IN VIAIER IS SET.
                                                ;(I.E. VIAIER IS THE ENABLE REGISTER FOR THE
                                                ;FOR THE SETTING OF IRQ AND THE ISSUEANCE OF
                                                ;AN INTERRUPT TO THE 6502 WHEN IRQ IS SET.)

000100                  FLGT1   = BIT6          ;TIMEOUT OF TIMER 1
000040                  FLGT2   = BIT5          ;TIMEOUT OF TIMER 2
000020                  FLGCB1  = BIT4          ;ACTIVE TRANSITION OF PIN 18 (CB1)
000010                  FLGCB2  = BIT3          ;ACTIVE TRANSITION OF PIN 19 (CB2)
000004                  FLGSR   = BIT2          ;COMPLETION OF 8 SHIFTS
000002                  FLGCA1  = BIT1          ;ACTIVE TRANSITION OF PIN 40 (CA1)
000001                  FLGCA2  = BIT0          ;ACTIVE TRANSITION OF PIN 39 (CA2)



                        ;;*************************************************************************
                        ;* INTERRUPT ENABLE REGISTER -- "IER" -- READ/WRITE
                        ;;*************************************************************************

120016                  VIAIER  = 120016        ;ADDRESS OF THIS REGISTER -- HEX = A0XE

000200                  INTSC   = BIT7          ;CONTROLS THE SETTING OR CLEARING OF BITS IN
                                                ;THE REST OF IER.  IF = 0  THE OTHER BITS IN
                                                ;THIS REG., IF SET, WILL CLEAR THEIR RESPECTIVE
                                                ;BITS IN THE INT. ENAB. REG.,  IF = 1, THE
                                                ;RESPECTIVE BITS WILL BE SET.

                        ;  WHEN WRITING THIS REG., THE COMMENT ABOVE HOLDS.
```

REGISTER DEFINITIONS -- 6522 VIA CHIP

```
1465
1466                            ;  WHEN READING THIS REG., THE CURRENT STATE OF THE INT. ENABLE REG. IS RETURNED.
1467
1468                            ;  THE BIT ASSIGNMENTS ARE THE SAME AS FOR VIAIFR AS DEFINED ABOVE.
1469
1470
1471
1472                            ;;*****************************************************************************
1473                            ;* OUTPUT REGISTER A -- "ORA" -- READ ONLY (OR READ/WRITE UNDER CONTROL OF "DDPA")
1474                            ;;*****************************************************************************
1475
1476    120017                  VIAORA  = 120017                ;ADDRESS OF THIS REGISTER -- HEX = A0XF
1477
1478                            ;  THIS ADDRESS ACCESSES THE SAME DATA AS "VIAMS" EXCEPT THAT NO "HANDSHAKING"
1479                            ;  WILL TAKE PLACE (I.E. THERE IS NO CHANGE IN IRQ OR CA2 AS A RESULT OF
1480                            ;  READING ORA THROUGH THIS ADDRESS)
1481
1482                            ;THE BIT ASSIGNMENTS ARE THE SAME AS FOR "VIAMS" ABOVE.
1483
1484
1485
```

REGISTER DEFINITIONS -- MISC

```
                                .SBTTL  REGISTER DEFINITIONS -- MISC

                                ;;*****************************************************************************
                                ;* SWITCH PACKS
                                ;;*****************************************************************************

        121000                  SWPBOT  = 121000                ;"BOOT ADDRESS" SWITCH PACK [A200]
        121400                  SWPDDCMP = 121400               ;"DDCMP ADDRESS" SWITCH PACK [A300]

                                ;MISCELLANEOUS EQUATES

        100000                  TCCHEK  = BIT15         ;FLAG TO REQUEST H3254,5 CHECK
        001000                  RAMADR  = 001000        ;STARTING ADRS OF RAM PAGE 2 (ADRS 0200 HEX)

        000002                  EIAV35  = BIT1                  ;SELECT V.35 OR EIA 423/232C
        000001                  INTGRL  = BIT0                  ;SELECT INTEGRAL MODEM

        040000                  NORXEN  = BIT14         ;KILL RXEN DURING "INITRN"
        001000                  NOLOOP  = BIT9          ;KILL TTLOOP DURING "INITRN"

        000200                  NCTBMT  = BIT7          ;DISABLE INITIAL TBMT=0 CHECK IN TXCHAR

        100000                  NOCRDA  = BIT15         ;DISABLE INITIAL RDA=0 CHECK IN RXCHAR
        040000                  NFCRDA  = BIT14         ;DISABLE FINAL RDA=1 CHECK IN RXCHAR
        020000                  NCRACT  = BIT13         ;DISABLE RXACT=1 CHECK AFTER CLOCKING (RXCHAR)
```

GLOBAL DATA SECTION

```
1772                                    .SBTTL  GLOBAL DATA SECTION
1773
1774                                    ;'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
1775                                    ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1776                                    ;/      IN MORE THAN ONE TEST.
1777                                    ;'''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
1778
1779                                    ;+*********************************************************************************
1780                                    ; CONTROL BLOCK FOR STACKED ERROR MESSAGES
1781                                    ;--*********************************************************************************
1782
1783 002172                                     ERRTBL
     002172                                                                    L$ERRTBL::
     002172  000000                     ERRTYP::        .WORD   0
     002174  000000                     ERRNBR::        .WORD   0
     002176  000000                     ERRMSG::        .WORD   0
     002200  000000                     ERRBLK::        .WORD   0
1784
1785                                    ;;*********************************************************************************
1786                                    ;* STORAGE FOR DEVICE REGISTERS
1787                                    ;;*********************************************************************************
1788 002202                             WSR0:                           ;STORAGE FOR DEVICE CSR REGISTERS
1789 002202  000000                     BSR0:   .WORD   0
1790 002204                             WSR2:
1791 002204  000000                     BSR1:   .WORD   0
1792 002206                             WSR4:
1793 002206  000000                     BSR2:   .WORD   0
1794 002210                             WSR6:
1795 002210  000000                     BSR3:   .WORD   0
1796 002212                             WSR10:
1797 002212  000000                     BSR4:   .WORD   0
1798 002214                             WSR12:
1799 002214  000000                     BSR5:   .WORD   0
1800 002216                             WSR14:
1801 002216  000000                     BSR6:   .WORD   0
1802 002220                             WSR16:
1803 002220  000000                     BSR7:   .WORD   0
1804 002222  000000                     BSR10:  .WORD   0
1805 002224  000000                     BSR11:  .WORD   0
1806 002226  000000                     BSR12:  .WORD   0
1807 002230  000000                     BSR13:  .WORD   0
1808 002232  000000                     BSR14:  .WORD   0
1809 002234  000000                     BSR15:  .WORD   0
1810 002236  000000                     BSR16:  .WORD   0
1811 002240  000000                     BSR17:  .WORD   0
1812
1813 002242                             UREGS:  .BLKW   8.              ;THE FIRST 7 ARE FOR THE USYRT'S ACTUAL
1814                                                                    ;REGISTERS.  THE LAST ONE IS FOR THE STATUS
1815                                                                    ;REG. (USTATR).
1816 002262                             VREGS:  .BLKW   16.             ;STORAGE FOR VIA REGISTERS FOR PRINTOUT
```

GLOBAL DATA SECTION

```
1818                            ;*******************************************************************************
1819                            ;* MISCELLANEOUS STORAGE
1820                            ;*******************************************************************************
1821 002322  000000  TDATA:  .WORD   0          ;TEST DATA
1822 002324  000000  GDATA:  .WORD   0          ;GUOD DATA
1823 002326  000000  BDATA:  .WORD   0          ;BAD DATA
1824 002330  000000  XDATA:  .WORD   0          ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
1825 002332  000000  SCRACH: .WORD   0          ;GEN'L PURPOSE SCRATCH WORD
1826 002334  000000  LOGDEV: .WORD   0          ;LOGICAL DEVICE NUMBER
1827 002336  000000  REGNUM: .WORD   0          ;CONTAINS A DEVICE REGISTER NUMBER
1828 002340  000000  PSTACK: .WORD   0          ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
1829 002342  000000  PRIOR:  .WORD   0          ;CPU PRIORITY FOR PRINTOUT
1830 002344  000000  SUBRPC: .WORD   0          ;PC OF SUBR CALL FOR ERROR REPORTS
1831 002346  000000  INTFLG: .WORD   0          ;INTERRUPT RECEIVED FLAGS
1832                                            ;  BIT 0 FOR TX, BIT 1 FOR RCV
1833 002350  000000  ERRFLG: .WORD   0          ;SUBROUTINE ERROR FLAG
1834 002352  000000  TIMFLG: .WORD   0          ;EVENT TIME-OUT FLAG
1835 002354  000000  RETADR: .WORD   0          ;SUBR ERROR RETURN ADDRESS
1836 002356  000000  REDBYT: .WORD   0          ;LO BYTE CONTAINS BYTE READ FROM LU REG
1837 002360  000000  WRIBYT: .WORD   0          ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
1838 002362  000000  LOADAT: .WORD   0          ;CONTAINS TEST DATA LOADED INTO REG
1839 002364  000000  GOODAT: .WORD   0          ;STORAGE FOR EXPECTED DATA
1840 002366  000000  BADDAT: .WORD   0          ;STORAGE FOR ACTUAL DATA
1841 002370  000000  FRSTIM: .WORD   0          ;FLAG=0 IF PROGRAM JUST LOADED
1842 002372  000000  SAVE4:  .WORD   0          ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
1843 002374  000000  SAVE6:  .WORD   0          ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
1844 002376  000000  ERROR1: .WORD   0          ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
1845 002400  000000  CHPTYP: .WORD   0          ;USYRT CHIP TYPE. =0 FOR SMC, ELSE =1
1846 002402  000000  SAVLEN: .WORD   0          ;SAVED TX AND RCV CHAR LENGTHS
1847 002404  000000  DEVMAP: .WORD   0          ;BIT MAP OF ACTIVE DEVICES
1848 002406  000000  DEVPTR: .WORD   0          ;DEVICE MAP BIT POINTER
1849 002410  000000  UNIT:   .WORD   0          ;CONTAINS UNIT NO. (1 TO N)
1850 002412  000000  STARES: .WORD   0          ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
1851 002414  000000  TSTNUM: .WORD   0          ;NO. OF CURRENT TEST (FOR SOME TESTS)
1852
```

GLOBAL DATA SECTION

```
1854                                    ;;**********  CURRENT DEVICE PARAMETERS  **************
1855 002416                             BSEL0:
1856 002416                             SEL0:
1857 002416  160020                     MPCSR:   .WORD   160020          ;POINTER TO DMV11 CSR'S
1858 002420  160021                     BSEL1:   .WORD   160021          ;POINTER TO BSEL1
1859 002422                             BSEL2:
1860 002422  160022                     SEL2:    .WORD   160022          ;POINTER TO SEL2
1861 002424  160023                     BSEL3:   .WORD   160023          ;POINTER TO BSEL3
1862 002426                             BSEL4:
1863 002426  160024                     SEL4:    .WORD   160024          ;POINTER TO SEL4
1864 002430  160025                     BSEL5:   .WORD   160025          ;POINTER TO BSEL5
1865 002432                             BSEL6:
1866 002432  160026                     SEL6:    .WORD   160026          ;POINTER TO SEL6
1867 002434  160027                     BSEL7:   .WORD   160027          ;POINTER TO BSEL7
1868 002436                             BSEL10:
1869 002436  160030                     SEL10:   .WORD   160030          ;POINTER TO SEL10
1870 002440  160031                     BSEL11:  .WORD   160031          ;POINTER TO BSEL11
1871 002442                             BSEL12:
1872 002442  160032                     SEL12:   .WORD   160032          ;POINTER TO SEL12
1873 002444  160033                     BSEL13:  .WORD   160033          ;POINTER TO BSEL13
1874 002446                             BSEL14:
1875 002446  160034                     SEL14:   .WORD   160034          ;POINTER TO SEL14
1876 002450  160035                     BSEL15:  .WORD   160035          ;POINTER TO BSEL15
1877 002452                             BSEL16:
1878 002452  160036                     SEL16:   .WORD   160036          ;POINTER TO SEL16
1879 002454  160037                     BSEL17:  .WORD   160037          ;POINTER TO BSEL17
1880
1881 002456  000300                     MPIVEC:  .WORD      300          ;DMV11 INPUT INTERRUPT VECTOR
1882 002460  000304                     MPOVEC:  .WORD      304          ;DMV11 OUTPUT INTERRUPT VECTOR
1883 002462  000240                     MPRIOR:  .WORD      240          ;DMV11 DEVICE PRIORITY
1884 002464  000000                     LUSWI1:  .WORD        0          ;LINE UNIT SWITCH PACK #1
1885 002466  000000                     LUSWI2:  .WORD        0          ;LINE UNIT SWITCH PACK #2
1886 002470  000000                     BRDTYP:  .WORD        0          ;0=M8064, 1=M8053/V.35,2=M8053/EIA
1887 002472  000000                     TSTCON:  .WORD        0          ;TEST CONNECTOR INDICATOR
1888 002474  000001                     BDRATE:  .WORD        1          ;BAUD RATE = 56 K
1889                                                                     ;          0 = 19.2 K
1890                                                                     ;          1 = 56 K
```

GLOBAL DATA SECTION

```
1892
1893 002476  120400                ;TABLE OF USYRT REGISTER ADDRESSES
1894 002500  120401                USYREG: .WORD   120400          ;ADDRESS OF RDSRL
1895 002502  120402                        .WORD   120401          ;ADDRESS OF RDSRH
1896 002504  120403                        .WORD   120402          ;ADDRESS OF TDSRL
1897 002506  120404                        .WORD   120403          ;ADDRESS OF TDSRH
1898 002510  120405                        .WORD   120404          ;ADDRESS OF PCSARL
1899 002512  120407                        .WORD   120405          ;ADDRESS OF PCSARH
1900 002514  122000                        .WORD   120407          ;ADDRESS OF PCR
1901                                       .WORD   122000          ;ADDRESS OF USYRT STATUS REG
1902
1903 002516                        ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
1904                               REDDAT: .BLKB   8.
1905
1906 002526  000000                ;;*****  GEN'L PURPOSE SCRATCH STORAGE  *****
1907 002530  000000                REG0:   .WORD   0
1908 002532  000000                REG1:   .WORD   0
1909 002534  000000                REG2:   .WORD   0
1910 002536  000000                REG3:   .WORD   0
1911 002540  000000                REG4:   .WORD   0
1912 002542  000000                REG5:   .WORD   0
1913 002544  000000                REG6:   .WORD   0
1914                               REG7:   .WORD   0
1915
1916 002546  000000                ;;*****  SCRATCH STORAGE FOR MESSAGE REPORTING  *****
1917 002550  000000                TMP0:   .WORD   0
1918 002552  000000                TMP1:   .WORD   0
1919 002554  000000                TMP2:   .WORD   0
1920 002556  000000                TMP3:   .WORD   0
1921 002560  000000                TMP4:   .WORD   0
1922 002562  000000                TMP5:   .WORD   0
1923 002564  000000                TMP6:   .WORD   0
1924                               TMP7:   .WORD   0
1925
1926 002566                        ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
1927 002566      377               UPBITS:
1928 002567      000                       .BYTE   377             ;MASK FOR RDBR
1929 002570      000                       .BYTE   000             ;MASK FOR RDSR
1930 002571      360                       .BYTE   000             ;MASK FOR TDBR
1931 002572      000                       .BYTE   360             ;MASK FOR TDSR
1932 002573      000                       .BYTE   000             ;MASK FOR SSAR
1933 002574      347                       .BYTE   000             ;MASK FOR PCSAR
1934                                       .BYTE   347             ;MASK FOR PCR
1935 002575      200               TDSRNRW: .BYTE  200             ;TDSR NON-R/W BITS
```

DATA TEST PATTERNS

```
                                        .SBTTL  DATA TEST PATTERNS
                                        ;***** DATA PATTERN E *****
002576                                  PATE:
002576  377                                     .BYTE   377
002577  377                                     .BYTE   377
002600  377                                     .BYTE   377
002601  377                                     .BYTE   377
002602  377                                     .BYTE   377
002603  377                                     .BYTE   377
002604  377                                     .BYTE   377
002605  366                                     .BYTE   366

                                        ;***** DATA PATTERN F *****
002606                                  PATF:
002606  000                                     .BYTE   000
002607  000                                     .BYTE   000
002610  000                                     .BYTE   000
002611  000                                     .BYTE   000
002612  000                                     .BYTE   000
002613  000                                     .BYTE   000
002614  000                                     .BYTE   000
002615  110                                     .BYTE   110

                                        ;***** DATA PATTERN G *****
002616                                  PATG:
002616  000                                     .BYTE   000
002617  001                                     .BYTE   001
002620  003                                     .BYTE   003
002621  004                                     .BYTE   004
002622  005                                     .BYTE   005
002623  007                                     .BYTE   007
002624  100                                     .BYTE   100
002625  101                                     .BYTE   101
002626  103                                     .BYTE   103
002627  104                                     .BYTE   104
002630  105                                     .BYTE   105
002631  107                                     .BYTE   107
002632  000                                     .BYTE   000
002633  017                                     .BYTE   017
002634  027                                     .BYTE   027
002635  041                                     .BYTE   041
002636  200                                     .BYTE   200
002637  277                                     .BYTE   277
002640  103                                     .BYTE   103
002641  144                                     .BYTE   144
002642  115                                     .BYTE   115
002643  157                                     .BYTE   157
002644  000                                     .BYTE   000

                                        ;***** DATA PATTERN X *****
002645                                  PATX:
002645  125                                     .BYTE   125
002646  252                                     .BYTE   252
002647  000                                     .BYTE   000
002650  377                                     .BYTE   377
002651  001                                     .BYTE   001
002652  002                                     .BYTE   002
```

DATA TEST PATTERNS

```
1994 002653  004                        .BYTE   004
1995 002654  010                        .BYTE   010
1996 002655  020                        .BYTE   020
1997 002656  040                        .BYTE   040
1998 002657  100                        .BYTE   100
1999 002660  200                        .BYTE   200
2000 002661  376                        .BYTE   376
2001 002662  375                        .BYTE   375
2002 002663  373                        .BYTE   373
2003 002664  367                        .BYTE   367
2004 002665  357                        .BYTE   357
2005 002666  337                        .BYTE   337
2006 002667  277                        .BYTE   277
2007 002670  177                        .BYTE   177
2008
2009 002671  125                        .BYTE   125
2010 002672  252                        .BYTE   252
2011 002673  000                        .BYTE   000
2012 002674  377                        .BYTE   377
2013 002675  001                        .BYTE   001
2014 002676  002                        .BYTE   002
2015 002677  004                        .BYTE   004
2016 002700  010                        .BYTE   010
2017 002701  020                        .BYTE   020
2018 002702  040                        .BYTE   040
2019 002703  100                        .BYTE   100
2020 002704  200                        .BYTE   200
2021 002705  376                        .BYTE   376
2022 002706  375                        .BYTE   375
2023 002707  373                        .BYTE   373
2024 002710  367                        .BYTE   367
2025 002711  357                        .BYTE   357
2026 002712  337                        .BYTE   337
2027 002713  277                        .BYTE   277
2028 002714  177                        .BYTE   177
2029
2030 002715  125                        .BYTE   125
2031 002716  252                        .BYTE   252
2032 002717  000                        .BYTE   000
2033 002720  377                        .BYTE   377
2034 002721  001                        .BYTE   001
2035 002722  002                        .BYTE   002
2036 002723  004                        .BYTE   004
2037 002724  010                        .BYTE   010
2038 002725  020                        .BYTE   020
2039 002726  040                        .BYTE   040
2040 002727  100                        .BYTE   100
2041 002730  200                        .BYTE   200
2042 002731  376                        .BYTE   376
2043 002732  375                        .BYTE   375
2044 002733  373                        .BYTE   373
2045 002734  367                        .BYTE   367
2046 002735  357                        .BYTE   357
2047 002736  337                        .BYTE   337
2048 002737  277                        .BYTE   277
2049 002740  177            LPAIX:      .BYTE   177
2050
```

DATA TEST PATTERNS

```
2051                                    ;***** DATA PATTERN I *****
2052 002741                             PATI:
2053 002741     000                             .BYTE   000
2054 002742     041                             .BYTE   041
2055 002743     102                             .BYTE   102
2056 002744     143                             .BYTE   143
2057 002745     204                             .BYTE   204
2058 002746     245                             .BYTE   245
2059 002747     306                             .BYTE   306
2060 002750     347                             .BYTE   347
2061 002751     000                             .BYTE   000
2062 002752     001                             .BYTE   001
2063 002753     002                             .BYTE   002
2064 002754     004                             .BYTE   004
2065 002755     040                             .BYTE   040
2066 002756     100                             .BYTE   100
2067 002757     200                             .BYTE   200
2068 002760     000                             .BYTE   000
2069 002761     346                             .BYTE   346
2070 002762     345                             .BYTE   345
2071 002763     343                             .BYTE   343
2072 002764     307                             .BYTE   307
2073 002765     247                             .BYTE   247
2074 002766     147                             .BYTE   147
2075 002767     347                             .BYTE   347
2076 002770     242                             .BYTE   242
2077 002771     105                             .BYTE   105
2078 002772     347                             .BYTE   347
2079 002773     010                             .BYTE   010
2080 002774     020                             .BYTE   020
2081 002775     367                             .BYTE   367
2082 002776     357                             .BYTE   357
2083 002777     030                             .BYTE   030
2084 003000     027                             .BYTE   027
2085 003001     377                             .BYTE   377
2086
2087                                    ;***** DATA PATTERN J *****
2088 003002                             PATJ:
2089 003002     000                             .BYTE   000
2090 003003     000                             .BYTE   000
2091 003004     001                             .BYTE   001
2092 003005     002                             .BYTE   002
2093 003006     004                             .BYTE   004
2094 003007     020                             .BYTE   020
2095 003010     040                             .BYTE   040
2096 003011     010                             .BYTE   010
2097
2098                                    ;***** DATA PATTERN K *****
2099 003012                             PATK:
2100 003012     000                             .BYTE   000
2101 003013     377                             .BYTE   377
2102 003014     376                             .BYTE   376
2103 003015     375                             .BYTE   375
2104 003016     373                             .BYTE   373
2105 003017     376                             .BYTE   376
2106 003020     177                             .BYTE   177
2107 003021     377                             .BYTE   377
```

DATA TEST PATTERNS

```
2108 003022  000                    .BYTE   000
2109 003023  001                    .BYTE   001
2110 003024  002                    .BYTE   002
2111 003025  004                    .BYTE   004
2112 003026  010                    .BYTE   010
2113 003027  200                    .BYTE   200
2114 003030  125                    .BYTE   125
2115 003031  252                    .BYTE   252
2116 003032  000                    .BYTE   000
2117
2118                            ;***** DATA PATTERN L *****
2119 003033                     PATL:
2120 003033  000                    .BYTE   000
2121 003034  017                    .BYTE   017
2122 003035  016                    .BYTE   016
2123 003036  015                    .BYTE   015
2124 003037  013                    .BYTE   013
2125 003040  016                    .BYTE   016
2126 003041  017                    .BYTE   017
2127 003042  017                    .BYTE   017
2128 003043  000                    .BYTE   000
2129 003044  001                    .BYTE   001
2130 003045  002                    .BYTE   002
2131 003046  004                    .BYTE   004
2132 003047  010                    .BYTE   010
2133 003050  000                    .BYTE   000
2134 003051  005                    .BYTE   005
2135 003052  012                    .BYTE   012
2136 003053  000                    .BYTE   000
```

DATA TEST PATTERNS

```
2138
2139                                    ;***** DATA PATTERN Q *****
2140 003054      000                    PATQ:    .BYTE   000
2141 003055      002                             .BYTE   002
2142 003056      014                             .BYTE   014
2143 003057      060                             .BYTE   060
2144 003060      001                             .BYTE   001
2145 003061      007                             .BYTE   007
2146 003062      037                             .BYTE   037
2147 003063      177                             .BYTE   177
2148
2149
2150 003064                             ENDPAT:
2151                                    .EVEN
```

DATA TEST PATTERNS

```
2153
2154
2155
2156                                    ;*** RECEIVED DATA BUFFER (64. WORDS) ***
2157 003064                             RCVBUF: .BLKW   64.
2158
2159
2160
2161
```

GLOBAL TEXT SECTION

```
                                .SBTTL  GLOBAL TEXT SECTION

                                ;*******************************************************************************
                                ;*      THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ;*      MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ;*      MORE THAN ONE TEST.
                                ;*******************************************************************************

                                ;;**************************************************************************
                                ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;;**************************************************************************
003264                                  DEVTYP  <M8053 OR M8064>
003264                                                                  L$DVTYP::
003264  115     070     060                                                     .ASCIZ  %M8053 OR M8064%
003267  065     063     040
003272  117     122     040
003275  115     070     060
003300  066     064     000
                                                                        .EVEN

                                ;;**************************************************************************
                                ;* TITLE OF PROGRAM
                                ;;**************************************************************************

000012                          .RADIX  10.
003304                                  DESCRIPT        <DMV-11 LINE UNIT TESTS - PART 3 OF 3>
003304                                                                  L$DESC::
003304  104     115     126                                             .ASCIZ  /DMV-11 LINE UNIT TE
STS - PART 3 OF 3/
003307  055     061     061
003312  040     114     111
003315  116     105     040
003320  125     116     111
003323  124     040     124
003326  105     123     124
003331  123     040     055
003334  040     120     101
003337  122     124     040
003342  063     040     117
003345  106     040     063
003350  000
                                                                        .EVEN
000010                          .RADIX  8.
```

```
                                .SBTTL  GLOBAL SUBROUTINE SECTION


                                .SBTTL ....M-LOOP -- MSTCLR -- MASTER CLEAR AND ENTER M-LOOP
                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;  MSTCLR -- MASTER CLEAR & ENTER M-LOOP
                                ;
                                ;  CALLING SEQUENCE:
                                ;
                                ;       JSR     PC,MSTCLR
                                ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
                                ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
                                ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
                                ;
                                ; N$:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

003352  112777  000301  177040  MSTCLR: MOVB    #RUN!MCLR!MREQ,@#SEL1   ;INITIATE M-LOOP

003360  010346                          MOV     R3,-(SP)
003362  012703  000030                  MOV     #24.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003366  077301                  1$:     SOB     R3,1$
003370  012603                          MOV     (SP)+,R3

003372  132777  000200  177022          BITB    #MRDY,@#SEL2    ;DID THE M-LOOP FINISH
003400  001023                          BNE     5$              ;YES,  GOOD.  RETURN
003402  004737  004166                  JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003406  012737  000301  002324          MOV     #RUN!MCLR!MREQ,@#DATA   ;IDENTIFY REQUESTED FUNCTION
003414                                  GTDF    EM3,ERR4        ;"MRDY" TIMEOUT
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 1
003414  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
003422  012737  000001  002174                                                  MOV     #1,ERRNBR
003430  012737  014263  002176                                                  MOV     #EM3,ERRMSG
003436  012737  020120  002200                                                  MOV     #ERR4,ERRBLK
003444  000261                          SEC                     ;SET CARRY TO INDICATE ERROR
003446  000401                          BR      9$              ;EXIT WITH THE "ERROR" FLAG (CARRY BIT) SET
003450  000241                  5$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
003452  000207                  9$:     RTS     PC              ;RETURN
```

```
                                .SBTTL ....M-LOOP -- READ
                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ; READ - READ THE SPECIFIED ADDRESS WITHIN THE DMV-11 (M8053)
                                ;
                                ;  CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,READ
                                ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
                                ;       .WORD   <DESTINATION ADDRESS WITHIN LSI-11>
                                ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
                                ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
                                ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
                                ;
                                ; N$:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

003454  012577  176746        READ:   MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003460  112777  000001  176734        MOVB    #REDLOC,@BSEL2  ;TELL M-LOOP TO GIVE US THE REQUESTED DATA

003466  010346                        MOV     R3,-(SP)
003470  012703  000050                MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003474  077301                1$:     SOB     R3,1$
003476  012603                        MOV     (SP)+,R3

003500  132777  000200  176714        BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
003506  001023                        BNE     5$              ;YES,  GOOD.  RETURN

003510  004737  004166                JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003514  012737  000001  002324        MOV     #REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
003522                                GTDF    EM4,ERR4        ;"MRDY" TIMEOUT
                                        ;       QUEUE "DEVICE FATAL" ERROR # 2
        003522  012737  000001  002172                                MOV     #T.EDF,ERRTYP
        003530  012737  000002  002174                                MOV     #2,ERRNBR
        003536  012737  014307  002176                                MOV     #EM4,ERRMSG
        003544  012737  020120  002200                                MOV     #ERR4,ERRBLK
003552  000261                        SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
003554  000401                        BR      6$              ;RETURN WITH THAT INDICATION

003556  000241                5$:     CLC                     ;INDICATE "NO ERROR"
003560  117735  176646        6$:     MOVB    @BSEL6,@(R5)+   ;PUT DATA WHERE CALLER WANTS IT
003564  000205                        RTS     R5              ;RETURN
```

....M-LOOP -- READ IMMEDIATE

```
                                .SBTTL ....M-LOOP -- READ IMMEDIATE
                                ;++*************************************************************************
                                ; READI - READ IMMEDIATE THE SPECIFIED ADDRESS WITHIN THE DMV-11 (M8053)
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,READI
                                ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
                                ;       .WORD   <DESTINATION -- CONTENTS OF REG. IS PUT HERE>
                                ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
                                ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED;  PRINT IT
                                ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
                                ;
                                ; N$:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;--*************************************************************************

003566                          READI:
003566  012577  176634                MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003572  112777  000001  176622        MOVB    #REDLOC,@BSEL2  ;TELL M-LOOP TO GIVE US THE REQUESTED DATA

003600  010346                        MOV     R3,-(SP)
003602  012703  000050                MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003606  077301                1$:     SOB     R3,1$
003610  012603                        MOV     (SP)+,R3

003612  132777  000200  176602        BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
003620  001023                        BNE     5$              ;YES,  GOOD.  RETURN

003622  004737  004166                JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003626  012737  000001  002324        MOV     #REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
003634                                GTDF    EM4,ERR4        ;"MRDY" TIMEOUT
                                                              ;       QUEUE  DEVICE FATAL" ERROR # 3
003634  012737  000001  002172                                                MOV     #T.EDF,ERRTYP
003642  012737  000003  002174                                                MOV     #3,ERRNBR
003650  012737  014307  002176                                                MOV     #EM4,ERRMSG
003656  012737  020120  002200                                                MOV     #ERR4,ERRBLK
003664  000261                        SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
003666  000401                        BR      6$              ;RETURN WITH THAT INDICATION

003670  000241                5$:     CLC                     ;INDICATE "NO ERROR"
003672  017725  176534        6$:     MOV     @SEL6,(R5)+     ;PUT DATA WHERE CALLER WANTS IT
003676  000205                        RTS     R5              ;RETURN
```

```
                                .SBTTL ....M-LOOP -- WRITE
                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ; WRITE - WRITE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
                                ;
                                ;  CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,WRITE
                                ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
                                ;       .WORD   <ADDRESS OF DATA BYTE>
                                ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
                                ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
                                ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
                                ;
                                ; N$:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;---------------------------------------------------------------------------------------
                                WRITE:  MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003700  012577  176522                MOVB    @(R5)+,@SEL6    ;MAKE DATA AVAILABLE TO M-LOOP
003704  113577  176522                BR      MLWRI           ;THE REST OF THIS ROUTINE IS THE SAME AS "WRITEI"
003710  000404
```

```
                                .SBTTL ....M-LOOP -- WRITE IMMEDIATE
                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ; WRITEI - WRITE IMMEDIATE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,WRITEI
                                ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
                                ;       .WORD   <DATA FIELD -- DATA TO BE WRITTEN IN DMV-11>
                                ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
                                ;       ERRCR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
                                ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
                                ;
                                ; N$:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;---++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                WRITEI:
003712  012577  176510                MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003716  012577  176510                MOV     (R5)+,@SEL6     ;MAKE DATA AVAILABLE TO M-LOOP
003722  112777  000002  176472  MLWRI: MOVB    #WRILOC,@BSEL2  ;TELL M-LOOP TO WRITE THE DATA

003730  010346                        MOV     R3,-(SP)
003732  012703  000050                MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003736  077301                1$:     SOB     R3,1$
003740  012603                        MOV     (SP)+,R3

003742  132777  000200  176452        BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
003750  001023                        BNE     5$              ;YES,  GOOD.  RETURN
003752  004737  004166                JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003756  012737  000002  002324        MOV     #WRILOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
003764                                GTDF    EM4,ERR4        ;"MRDY" TIMEOUT
                                                              ;       QUEUE "DEVICE FATAL" ERROR # 4
        003764  012737  000001  002172                                MOV     #T.EDF,ERRTYP
        003772  012737  000004  002174                                MOV     #4,ERRNBR
        004000  012737  014307  002176                                MOV     #EM4,ERRMSG
        004006  012737  020120  002200                                MOV     #ERR4,ERRBLK
004014  000261                        SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
004016  000401                        BR      6$              ;RETURN WITH THAT INDICATION

004020  000241                5$:     CLC                     ;INDICATE "NO ERROR"
004022  000205                6$:     RTS     R5              ;RETURN
```

```
                                .SBTTL ....GETBSR -- GET BYTE SELECT REGISTERS
                                ;*****************************************************************************
                                ;
                                ;       GET THE CONTENTS OF ALL CONTROL AND STATUS REGISTERS
                                ;
                                ;       FUNCTION - THIS SUBROUTINE COLLECTS THE CONTENTS OF THE
                                ;               BYTE SELECT REGISTERS FOR THE PURPOSE OF DISPLAY.
                                ;
                                ;       ENTRY CONDITIONS - NONE
                                ;
                                ;        EXIT CONDITIONS - NONE
                                ;
                                ;    REGISTERS DESTROYED - NONE
                                ;
                                ;-----------------------------------------------------------------------------

004024  117737  176366  002202  GETBSR: MOVB    @#SEL0,BSR0     ;PUT THE CURRENT CSR VALUES INTO THE PRINT-OUT
004032  117737  176362  002204          MOVB    @#SEL1,BSR1     ;TABLE
004040  117737  176356  002206          MOVB    @#SEL2,BSR2
004046  117737  176352  002210          MOVB    @#SEL3,BSR3
004054  117737  176346  002212          MOVB    @#SEL4,BSR4
004062  117737  176342  002214          MOVB    @#SEL5,BSR5
004070  117737  176336  002216          MOVB    @#SEL6,BSR6
004076  117737  176332  002220          MOVB    @#SEL7,BSR7
004104  117737  176326  002222          MOVB    @#SEL10,BSR10
004112  117737  176322  002224          MOVB    @#SEL11,BSR11
004120  117737  176316  002226          MOVB    @#SEL12,BSR12
004126  117737  176312  002230          MOVB    @#SEL13,BSR13
004134  117737  176306  002232          MOVB    @#SEL14,BSR14
004142  117737  176302  002234          MOVB    @#SEL15,BSR15
004150  117737  176276  002236          MOVB    @#SEL16,BSR16
004156  117737  176272  002240          MOVB    @#SEL17,BSR17
004164  000207                          RTS     PC              ;RETURN TO CALLER

                                .SBTTL ....GETWSR -- GET WORD SELECT REGISTERS
                                ; "WORD" VERSION OF ABOVE SUBROUTINE

004166  017737  176224  002202  GETWSR: MOV     @#SEL0,WSR0     ;MOVE THE 4 WORD REGISTERS TO THE OTHERWISE
004174  017737  176222  002204          MOV     @#SEL2,WSR2     ;BYTE TABLE
004202  017737  176220  002206          MOV     @#SEL4,WSR4
004210  017737  176216  002210          MOV     @#SEL6,WSR6
004216  017737  176214  002212          MOV     @#SEL10,WSR10
004224  017737  176212  002214          MOV     @#SEL12,WSR12
004232  017737  176210  002216          MOV     @#SEL14,WSR14
004240  017737  176206  002220          MOV     @#SEL16,WSR16
004246  000207                          RTS     PC              ;RETURN TO CALLER
```

```
                                .SBTTL ....STUREG -- STATIC TEST OF SPECIFIED USYRT REGISTER
                                ;*****************************************************************************
                                ; STUREG -- PERFORM A STATIC TEST OF THE SPECIFIED USYRT REGISTER
                                ;
                                ; CALLING SEQUENCE:
                                ;
                                ;       <R0 CONTAINS THE ADDRESS OF THE REGISTER TO BE TESTED>
                                ;       <"TDATA" CONTAINS THE TEST BYTE>
                                ;       <"GDATA" CONTAINS THE EXPECTED DATA>
                                ;       <"REGNUM" CONTAINS REG INDEX FOR POSSIBLE ERRORS>
                                ;
                                ;       JSR     PC,STUREG
                                ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
                                ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
                                ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
                                ;
                                ; N$:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;-----------------------------------------------------------------------------

004250  010037  004264        STUREG: MOV     R0,2$           ;PUT SPECIFIED REGISTER'S ADDRESS IN I/O CALLS
004254  010037  004302                MOV     R0,4$
                              
004260  004537  003700                JSR     R5,WRITE        ;WRITE IT
004264  000000                2$:     .WORD   0               ;*** MODIFIED FROM ABOVE ***
004266  002322                        .WORD   TDATA           ;
004270  103431                        BCS     10$             ;ON ERROR, EXIT
                              
004272  005037  002326                CLR     BDATA           ;CLEAR BOTH BYTES -- JUST IN CASE....
004276  004537  003454                JSR     R5,READ         ;READ IT BACK AGAIN
004302  000000                4$:     .WORD   0               ;*** MODIFIED FROM ABOVE ***
004304  002326                        .WORD   BDATA
004306  103422                        BCS     10$             ;ON ERROR, EXIT
                              
004310  123737  002324  002326        CMPB    GDATA,BDATA     ;DID WE READ WHAT WE WROTE?
004316  000241                        CLC                     ; (THIS ISN'T NEEDED FOR THE ERROR TEST BUT
                                                              ; MUST BE CLEARED ON EXIT IF NO ERROR OCCURED)
004320  001415                        BEQ     10$             ;YES, EXIT FROM SUBTEST
004322                                GTDF    EM25,ERR7A      ;REPORT READ/WRITE ERROR
                                                              ;       QUEUE "DEVICE FATAL" ERROR # 5
004322  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
004330  012737  000005  002174                                                  MOV     #5,ERRNBR
004336  012737  014360  002176                                                  MOV     #EM25,ERRMSG
004344  012737  020244  002200                                                  MOV     #ERR7A,ERRBLK
004352  000261                        SEC                     ;INDICATE THAT AN ERROR WAS DETECTED
004354  000207                10$:    RTS     PC
                              
                              
                              .SBTTL ....STALL -  DELAY FOR 10.5 MICRO SEC'S (ON LSI 11)
                              ;*****************************************************************************
                              ; STALL -- THIS SUBROUTINE STALLS FOR ABOUT 10.5 MICRO-SECONDS
                              ;*****************************************************************************
                              
004356  000207               STALL:  RTS     PC
```

```
                                .SBTTL
                                ;*******************************************************************************
                                ;* GETURS - LOAD INTO THE 8 WORD STORAGE AREA (UREGS) THE CONTENTS OF THE
                                ;*      VARIOUS USYRT REGISTERS
                                ;*
                                ;*        CALLING SEQUENCE:
                                ;*
                                ;*******************************************************************************
004360  012737  002242  004422  GETURS: MOV     #UREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
004366  012737  120400  004420          MOV     #USYRT,4$       ;INIT POINTER TO REGISTER ADDRESSES

004374  005037  002260                  CLR     UREGS+14.       ;CLEAR STORAGE WORD
004400  004537  003454                  JSR     R5,READ         ;READ THE USYRT STATUS REGISTER
004404  122000                          .WORD   USTATR          ;STATUS REGISTER'S ADDRESS WITHIN DMV-11
004406  002260                          .WORD   UREGS+14.       ;ADDRESS ALLOCATED TO THAT REG. W/IN "UREGS"

004410  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
004414  004537  003454                  JSR     R5,READ         ;READ A LINE UNIT REG
004420  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
004422  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE

004424  005237  004420          6$:     INC     4$              ;INCREMENT REG NO.
004430  023727  004420  120406          CMP     4$,#USYRT+6     ;THIS IS NOT A VALID REGISTER ADDRESS
004436  001772                          BEQ     6$              ;SO IT MUST BE BYPASSED

004440  062737  000002  004422          ADD     #2,5$           ;ADVANCE ADDRESS OF STORAGE AREA POINTER
004446  023727  004420  120410          CMP     4$,#USYRT+10    ;SEE IF ALL REGS READ YET
004454  001355                          BNE     3$              ;BR IF NOT

004456  000207                          RTS     PC              ;RETURN



                                ;*******************************************************************************
                                ;* GETVRS: - LOAD INTO THE 16 WORD STORAGE AREA (VREGS) THE CONTENTS OF THE
                                ;*      VARIOUS VIA REGISTERS.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*******************************************************************************
004460  012737  002262  004506  GETVRS: MOV     #VREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
004466  012737  120000  004504          MOV     #VIA,4$         ;INIT POINTER TO REGISTER ADDRESSES
004474  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
004500  004537  003454                  JSR     R5,READ         ;READ A VIA REG
004504  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
004506  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
004510  005237  004504          6$:     INC     4$              ;INCREMENT REG NO.
004514  062737  000002  004506          ADD     #2,5$           ;INCREMENT STORAGE ADRS
004522  023727  004504  120020          CMP     4$,#VIA+16.     ;SEE IF ALL VIA REGS READ YET
004530  001361                          BNE     3$              ;BR IF NOT
004532  000207                          RTS     PC              ;RETURN
```

....INITT1 -- INITIALIZE TIMER #1

```
                        .SBTTL ....INITT1 -- INITIALIZE TIMER #1
                        ;*****************************************************************************
                        ;* INITT1 - INITIALIZE TIMER # 1
                        ;*
                        ;*        CALLING SEQUENCE:
                        ;*
                        ;*              JSR     R5,INITT1
                        ;*              .WORD   <VALUE LOADED INTO THE T1 LATCH @ VIAT1C & VIAT1D>
                        ;*              .WORD   <VALUE LOADED INTO "T1L-L" & "T1C-H">
                        ;*              .BYTE   <BITS 6 & 7 WILL BE LOADED INTO "ACR", BIT 5 WILL BE
                        ;*                       USED TO SET OR CLEAR BIT 6 ("T1") OF THE INTERRUPT
                        ;*                       ENABLE REGISTER ("IER")>
                        ;*              .BYTE   <UNUSED>
                        ;*
                        ;*
                        ;* NOTE:
                        ;*
                        ;* BEFORE LOADING AND STARTING THE COUNTER, THE LATCH REGISTER (ACCESSED THRU
                        ;* "VIAT1C") IS LOADED.  THEN, T1L-L IS LOADED AND NEXT, T1C-H.  THIS LAST
                        ;* LOAD WILL RESET THE TIMEOUT BIT AND COUNTER LOGIC.  IT IS EXPECTED AT THIS
                        ;* TIME (5/25/79) THAT THE INTERRUPT FACILITY OF THE VIA CHIP WILL NOT BE USED
                        ;* -- HOWEVER, ACCESS TO THE INTERRUPT ENABLE BIT IS GIVEN THROUGH THE THIRD
                        ;* PARAMETER IN THE CALLING SEQUENCE (BIT 5 = 0 WILL CAUSE THIS ROUTINE TO
                        ;* CLEAR THE ENABLE BIT ("T1") IN "IER".)
                        ;*
                        ;*****************************************************************************

004534  010146          INITT1: MOV     R1,-(SP)        ;SAVE THE REGISTER WE WILL BE USING
004536  012537  004660          MOV     (R5)+,7$        ;SETUP VALUE TO BE WRITTEN IN LATCH
004542  012537  004706          MOV     (R5)+,10$       ;SETUP VALUE TO BE WRITTEN IN COUNTER
004546  111501                  MOVB    (R5),R1         ;GET & PROCESS BITS FOR ACR 6 & 7
004550  143701  000077          BICB    #77,R1
004554  010137  004650          MOV     R1,4$           ;SETUP CALL SET ACR'S BITS 6 & 7
004560  112501                  MOVB    (R5)+,R1        ;NOW, GET THE BIT TO BE USED IN SETTING OR
                                                        ;CLEARING BIT 6 OF "IER"
004562  106301                  ASLB    R1              ;THE PASSED BIT IS IN THE WRONG POSITION
004564  106301                  ASLB    R1              ;BUT, THE PASSED BIT SHOULD CONTROL THE OPERATION.
                                                        ;WE KNOW WE ARE SETTING OR CLEARING BIT 6 --
                                                        ;THUS, THE PASSED BIT WILL BECOME THE CONTROLLING
                                                        ;BIT 7 AND WE WILL "OR" IN THE BIT WE WISH TO
                                                        ;BE CONTROLLED (BIT 6).
004566  143701  000177          BICB    #177,R1         ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
004572  153701  000100          BISB    #100,R1         ;THEN SET BIT 6
004576  010137  004610          MOV     R1,2$           ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE

004602  004537  003712          JSR     R5,WRITEI       ;WRITE TO
004606  120016                  VIAIER                  ;THE VIA'S IER
004610  000000          2$:     .WORD   0               ;INTERRUPT ENABLE/DISABLE INFORMATION

004612  004537  003566          JSR     R5,READI        ;READ THE CURRENT SETTING OF
004616  120013                  VIAACR                  ;THE VIA'S ACR
004620  000000          3$:     .WORD   0               ;INTO "3$"

004622  013701  004620          MOV     3$,R1           ;GET THAT VALUE
004626  143701  000300          BICB    #300,R1         ;CLEAR THE CURRENT SETTING OF BITS 6 & 7
004632  053701  004650          BIS     4$,R1           ;SET THEM ACCORDING TO THE PASSED VALUES
004636  010137  004650          MOV     R1,4$           ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
```

....INITT1 -- INITIALIZE TIMER #1

```
2595
2596 004642  004537  003712                JSR     R5,WRITEI       ;WRITE TO
2597 004646  120013                        VIAACR                  ;THE VIA'S ACR
2598 004650  000000                4$:     .WORD   0               ;THE NEW REGISTER SETTING
2599
2600 004652  004537  003712                JSR     R5,WRITEI       ;WRITE TO
2601 004656  120006                        VIAT1C                  ;LOW ORDER LATCH REGISTER (T1L-L)
2602 004660  000000                7$:     .WORD   0               ;THE VALUE PASSED
2603
2604 004662  113737  004661  004676        MOVB    7$+1,8$         ;SETUP FOR AND
2605 004670  004537  003712                JSR     R5,WRITEI       ;WRITE TO
2606 004674  120007                        VIAT1D                  ;HIGH ORDER LATCH REGESTER (T1L-H)
2607 004676  000000                8$:     .WORD   0               ;THE VALUE PASSED
2608
2609 004700  004537  003712                JSR     R5,WRITEI       ;WRITE TO
2610 004704  120004                        VIAT1A                  ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2611 004706  000000                10$:    .WORD   0               ;THE VALUE PASSED
2612
2613 004710  113737  004707  004724        MOVB    10$+1,11$       ;SETUP FOR AND
2614 004716  004537  003712                JSR     R5,WRITEI       ;WRITE TO
2615 004722  120005                        VIAT1B                  ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2616 004724  000000                11$:    .WORD   0               ;THE VALUE PASSED
2617
2618                                       ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2619
2620 004726  012601                        MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2621 004730  005205                        INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2622                                                               ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2623
2624 004732  000205                        RTS     R5              ;NOW, RETURN
2625
2626
```

```
                        .SBTTL ....INIT2 -- INITIALIZE TIMER #2
                        ;**************************************************************************
                        ;* INIT2 - INITIALIZE TIMER # 2
                        ;*
                        ;*        CALLING SEQUENCE:
                        ;*
                        ;*              JSR     R5,INIT2
                        ;*              .WORD   <VALUE LOADED INTO "T2L-L" & "T2C-H">
                        ;*              .BYTE   <BIT 5 WILL BE LOADED INTO "ACR", BIT 4 WILL BE USED
                        ;*                      TO SET OR CLEAR BIT 5 ("T2") OF THE INTERRUPT ENABLE
                        ;*                      REGISTER ("IER")>
                        ;*              .BYTE   <UNUSED>
                        ;*
                        ;*
                        ;* NOTE:
                        ;*
                        ;* FIRST T2L-L IS LOADED, THEN T2C-H.  THIS SECOND LOAD WILL RESET THE TIMEOUT
                        ;* BIT AND COUNTER LOGIC.  IT IS EXPECTED AT THIS TIME (5/25/79) THAT THE
                        ;* INTERRUPT FACILITY OF THE VIA CHIP WILL NOT BE USED -- HOWEVER, ACCESS TO
                        ;* THE INTERRUPT ENABLE BIT IS GIVEN THROUGH THE SECOND PARAMETER IN THE
                        ;* CALLING SEQUENCE (BIT 4 = 0 WILL CAUSE THIS ROUTINE TO CLEAR THE ENABLE BIT
                        ;* ("T2") IN "IER".)
                        ;*
                        ;**************************************************************************
004734  010146          INIT2: MOV      R1,-(SP)        ;SAVE THE REGISTER WE WILL BE USING
004736  012537  005056          MOV     (R5)+,10$       ;SETUP VALUE TO BE WRITTEN IN COUNTER
004742  111501                MOVB    (R5),R1         ;GET & PROCESS BIT FOR ACR 5
004744  143701  000337          BICB    337,R1
004750  010137  005046          MOV     R1,4$           ;SETUP CALL TO SET OR CLEAR ACR'S BIT 5
004754  112501                MOVB    (R5)+,R1        ;NOW, GET THE BIT TO BE USED IN SETTING OR
                                                        ;CLEARING BIT 5 OF "IER"
004756  106301                ASLB    R1              ;THE PASSED BIT IS IN THE WRONG POSITION
004760  106301                ASLB    R1              ;BUT, THE PASSED BIT SHOULD CONTROL THE
004762  106301                ASLB    R1              ;OPERATION.
                                                        ;WE KNOW WE ARE SETTING OR CLEARING BIT 5 --
                                                        ;THUS, THE PASSED BIT WILL BECOME THE CONTROLLING
                                                        ;BIT 7 AND WE WILL "OR" IN THE BIT WE WISH TO
                                                        ;BE CONTROLLED (BIT 5).
004764  143701  000177          BICB    177,R1          ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
004770  153701  000040          BISB    040,R1          ;THEN SET BIT 5
004774  010137  005006          MOV     R1,2$           ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE
005000  004537  003712          JSR     R5,WRITEI       ;WRITE TO
005004  120016                VIAIER                  ;THE VIA'S IER
005006  000000          2$:     .WORD   0               ;INTERRUPT ENABLE/DISABLE INFORMATION
005010  004537  003566          JSR     R5,READI        ;READ THE CURRENT SETTING OF
005014  120013                VIAACR                  ;THE VIA'S ACR
005016  000000          3$:     .WORD   0               ;INTO "3$"
005020  013701  005016          MOV     3$,R1           ;GET THAT VALUE
005024  143701  000040          BICB    040,R1          ;CLEAR THE CURRENT SETTING OF BIT 5
005030  053701  005046          BIS     4$,R1           ;SET IT ACCORDING TO THE PASSED VALUE
005034  010137  005046          MOV     R1,4$           ;PASS NEW REG. SETTING TO APPROPRIATE CALL
005040  004537  003712          JSR     R5,WRITEI       ;WRITE TO
```

....INIT12 -- INITIALIZE TIMER #2

```
2685 005044  120013                    VIAACR                  ;THE VIA'S ACR
2686 005046  000000              4$:   .WORD   0               ;THE NEW REGISTER SETTING
2687
2688 005050  004537  003712             JSR     R5,WRITEI      ;WRITE TO
2689 005054  120010                     VIAT2A                 ;LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
2690 005056  000000              10$:  .WORD   0               ;THE VALUE PASSED
2691
2692 005060  113737  005057  005074     MOVB    10$+1,11$      ;SETUP FOR AND
2693 005066  004537  003712             JSR     R5,WRITEI      ;WRITE TO
2694 005072  120011                     VIAT2B                 ;HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
2695 005074  000000              11$:  .WORD   0               ;THE VALUE PASSED
2696
2697                              ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2698
2699 005076  012601                     MOV     (SP)+,R1       ;BUT FIRST RESTORE R1
2700 005100  005205                     INC     R5             ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2701                                                            ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2702
2703 005102  000205                     RTS     R5             ;THEN RETURN
2704
```

```
                                .SBTTL ....RSTCHK -- RESET USYRT/VERIFY ALL USYRT REGS @ RESET STATE
                                ;*****************************************************************
                                ; RSTCHK - MANUALLY RESET THE USYRT AND VERIFY THAT ALL USYRT REGISTERS
                                ;         ARE IN THEIR RESET STATE. AN ERROR MESSAGE IDENTIFYING THE
                                ;         FAILING REGISTER IS STACKED IF ONE IS ENCOUNTERED.
                                ;
                                ;       CALLING SEQUENCE:
                                ;               JSR     R5,RSTCHK
                                ;*****************************************************************

005104                          RSTCHK:
005104  010146                         MOV     R1,-(SP)        ;SAVE R1
005106  010246                         MOV     R2,-(SP)        ;SAVE R2

005110  004537  003712                  JSR     R5,WRITEI       ;SET PROGRAM RESET BIT IN VIA ORB REG
005114  120000                          VIAORB
005116  000031                          DTR!RTSND!PRESET
005120  004537  003712                  JSR     R5,WRITEI       ;CLEAR PROGRAM RESET BIT IN VIA ORB REG
005124  120000                          VIAORB
005126  000030                          DTR!RTSND

005130  005001                          CLR     R1              ;INIT USYRT REG ADRS PTR
005132  012702  002606                  MOV     #PATF,R2        ;INIT DATA PATTERN POINTER
005136  016137  002476  005150  6$:     MOV     USYREG(R1),7$   ;SET USYRT READ ADDRESS
005144  004537  003566                  JSR     R5,READI        ;READ A USYRT REG
005150  000000                  7$:     .WORD   0               ;USYRT REG ADRS GOES HERE
005152  000000                  8$:     .WORD   0               ;DATA READ IS RETURNED HERE
005154  123722  005152                  CMPB    8$,(R2)+        ;SEE IF REG CONTAINS EXPECTED DATA
005160  001432                          BEQ     9$              ;BR IF MATCH

005162  010137  002336                  MOV     R1,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
005166  006237  002336                  ASR     REGNUM          ;GET WORD OFFSET
005172  005037  002324                  CLR     GDATA           ;GET EXPECTED DATA
005176  116237  177777  002324          MOVB    -1(R2),GDATA
005204  013737  005152  002326          MOV     8$,BDATA        ;GET ACTUAL DATA

                        ;STACK "USYRT NOT CLEARED BY PROGRAM RESET" MSG
005212                          GTDF    EM2,ERR10
                                ;       QUEUE "DEVICE FATAL" ERROR # 6
005212  012737  000001  002172                  MOV     #T.EDF,ERRTYP
005220  012737  000006  002174                  MOV     #6,ERRNBR
005226  012737  014214  002176                  MOV     #EM2,ERRMSG
005234  012737  020364  002200                  MOV     #ERR10,ERRBLK
005242  000261                          SEC                     ;SET C BIT TO FLAG ERROR
005244  000406                          BR      10$             ;TAKE ERROR EXIT

005246  062701  000002          9$:     ADD     #2,R1           ;INCR USYRT REG ADRS PTR
005252  020127  000020                  CMP     R1,#16.         ;SEE IF ALL REGS READ YET
005256  002727                          BLT     6$              ;OR IF NOT
005260  000241                          CLC                     ;** CLEAR C BIT FOR NO ERRORS
005262  012602                  10$:    MOV     (SP)+,R2        ;RESTORE R2
005264  012601                          MOV     (SP)+,R1        ;RESTORE R1
005266  000205                          RTS     R5              ;** RETURN
```

```
2756                                    ;*************************************************************************
2757                                    ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
2758                                    ;*************************************************************************
2759 005270  010146                    WAIT50: MOV     R1,-(SP)        ;SAVE R1
2760 005272  012701  000005                    MOV     #5.,R1          ;INIT COUNTER
2761 005276  077101                    3$:     SOB     R1,3$           ;DELAY HERE FOR 23.8 MICRO-SEC'S
2762 005300  012601                            MOV     (SP)+,R1        ;RESTORE R1
2763 005302  000207                            RTS     PC              ;RETURN
2764
2765                                    ;       OVERHEAD (JSR, MOV, MOV, MOV, & RTS) ADD UP TO 25.25 MICRO-SEC'S
2766
2767                                    ;       THEREFORE, ACTUAL TOTAL DELAY IS 49.35 MICRO-SECONDS
2768
2769
2770
2771
2772                                    .SBTTL ....SETVIA -- SET UP VIA REGISTERS
2773                                    ;*************************************************************************
2774                                    ;* SETVIA - SET UP THE VIA REGISTERS
2775                                    ;*
2776                                    ;*      THIS SUBROUTINE PROGRAMS THE VIA REGISTERS FOR NORMAL OPERATION, BY
2777                                    ;*      LOADING THE DDRB, DDRA, ORB, ACR, PCR, IER.
2778                                    ;*
2779                                    ;*      CALLING SEQUENCE :
2780                                    ;*         JSR  PC,SETVIA
2781                                    ;*************************************************************************
2782 005304                            SETVIA:
2783 005304  004537  003712                    JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
2784 005310  120002                            VIADPB
2785 005312  000377                            377
2786 005314  004537  003712                    JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
2787 005320  120003                            VIADPA                  ; (BIT0 IS ONLY OUTPUT BIT)
2788 005322  000001                            001
2789 005324  004537  003712                    JSR     R5,WRITEI       ;DISABLE USYRT INTERNAL LOOPBACK
2790 005330  120017                            VIAORA
2791 005332  000000                            000
2792 005334  004537  003712                    JSR     R5,WRITEI       ;INIT PORT B
2793 005340  120000                            VIAORB
2794 005342  000030                            DTR!RTSND
2795 005344  004537  003712                    JSR     R5,WRITEI       ;SET ACR FOR :  T1 SQUARE WAVE OUTPUT MODE,
2796 005350  120013                            VIAACR                  ;               T2 ONE-SHOT OUTPUT MODE,
2797 005352  000350                            350                     ;               SR AT SYS CLOCK RATE ON CB1
2798 005354  004537  003712                    JSR     R5,WRITEI       ;SET PCR FOR :  CB1 NEG TRANS INPUT MODE,
2799 005360  120014                            VIAPCR                  ;               CA2 NEG TRANS INPUT MODE,
2800 005362  000022                            022                     ;               CA1 NEG TRANS INPUT MODE
2801 005364  004537  003712                    JSR     R5,WRITEI       ;DISABLE ALL MICRO-INTRPTS
2802 005370  120016                            VIAIER
2803 005372  000177                            177
2804 005374  000207                            RTS     PC              ;RETURN
2805
2806
```

```
                                .SBTTL ....INIDMV -- INIT DMV (MCLR, VIA SETUP)
                                ;*************************************************************************
                                ;* INIDMV - THIS SUBROUTINE INITIALIZES THE DMV-11, BY DOING A MASTER CLEAR,
                                ;*      ENTERING THE M-LOOP, AND PROGRAMMING THE VIA REGS FOR DEFAULT
                                ;*      OPERATION.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*         JSR  PC,INIDMV
                                ;*************************************************************************
005376  004737  003352        INIDMV: JSR     PC,MSTCLR       ;MASTER CLR, M-LOOP
005402  004737  005304                JSR     PC,SETVIA       ;PROGRAM VIA
005406  000207                        RTS     PC              ;RETURN


                                .SBTTL ....CKUSTS -- CHECK USYRT STATUS REGISTERS
                                ;*************************************************************************
                                ;* CKUSTS - THIS SUBROUTINE CHECKS THE USYRT STATUS BY READING THE USYRT
                                ;*      STATUS REGISTER AND COMPARING IT TO THE LOW BYTE OF THE WORD FOLLOWING
                                ;*      THE CALL. IF THERE IS A MISMATCH, THE SUBROUTINE STACKS THE ERROR
                                ;*      INFORMATION, AND SETS THE "C" BIT AND RETURNS.
                                ;*************************************************************************
005410                          CKUSTS:
005410  004537  003566                JSR     R5,READI        ;READ USYRT STATUS REGISTER
005414  122000                        USTATR
005416  000000                1$:     .WORD   0
005420  122537  005416                CMPB    (R5)+,1$        ;SEE IF STATUS MATCHES EXPECTED
005424  000241                        CLC                     ;CLEAR C BIT
005426  001430                        BEQ     2$              ;BR IF STATUS OK
005430  012737  000007  002336        MOV     #7,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
005436  016537  177777  002324        MOV     -1(R5),GDATA    ;GET EXPECTED DATA
005444  005037  002326                CLR     BDATA           ;GET ACTUAL DATA
005450  113737  005416  002326        MOVB    1$,BDATA
                                ;STACK "USYRT STATUS INCORRECT" ERROR
005456                                  GTDF    EM68,ERR10
                                ;       QUEUE "DEVICE FATAL" ERROR # 7
005456  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
005464  012737  000007  002174                                          MOV     #7,ERRNBR
005472  012737  014650  002176                                          MOV     #EM68,ERFMSG
005500  012737  020364  002200                                          MOV     #ERR10,ERRBLK
005506  000261                        SEC                     ;SET C BIT FOR ERROR
005510  005205                2$:     INC     R5              ;INCREMENT R5 PAST ARGUMENT
005512  000205                        RTS     R5              ;RETURN
```

```
                                        .SBTTL ....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)
                                        ;**************************************************************************
                                        ;* CKTACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TXACT IN THE USYRT
                                        ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                        ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*      JSR     R5,CKTACT
                                        ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TXACT>
                                        ;**************************************************************************
005514                                  CKTACT:
005514  012737  000007  002336                MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
005522  004537  003566                        JSR     R5,READI        ;READ USYRT STATUS
005526  122000                                USTATR
005530  000000                        1$:     .WORD   0
005532  032725  000001                        BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TXACT
005536  001422                                BEQ     2$              ;BR IF EXPECTED TXACT = 0
005540  132737  000004  005530                BITB    #TXACT,1$       ;SEE IF TXACT = 1
005546  001040                                BNE     3$              ;BR IF TXACT = 1
                                        ;STACK "TXACT NOT SET" MSG
005550                                          GTDF    EM69,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 8
005550  012737  000001  002172                                                    MOV     #T.EDF,ERRTYP
005556  012737  000010  002174                                                    MOV     #8,ERRNBR
005564  012737  014677  002176                                                    MOV     #EM69,ERRMSG
005572  012737  020714  002200                                                    MOV     #ERR12,ERRBLK
005600  000261                                SEC                     ;SET C BIT TO FLAG ERROR
005602  000423                                BR      4$              ;TAKE ERROR EXIT
005604  132737  000004  005530        2$:     BITB    #TXACT,1$       ;SEE IF TXACT = 0
005612  001416                                BEQ     3$              ;BR IF TXACT = 0
                                        ;STACK "TXACT NOT CLEARED" MSG
005614                                          GTDF    EM70,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 9
005614  012737  000001  002172                                                    MOV     #T.EDF,ERRTYP
005622  012737  000011  002174                                                    MOV     #9,ERRNBR
005630  012737  014715  002176                                                    MOV     #EM70,ERRMSG
005636  012737  020714  002200                                                    MOV     #ERR12,ERRBLK
005644  000261                                SEC                     ;SET C BIT TO FLAG ERROR
005646  000401                                BR      4$              ;TAKE ERROR EXIT
005650  000241                        3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
005652  000205                        4$:     RTS     R5              ;RETURN
```

```
                                        .SBTTL ....CKRACT -- CHECK RECEIVER ACTIVE (RXACT)
                                        ;*************************************************************************
                                        ;* CKRACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RXACT IN THE USYRT
                                        ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                        ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*      JSR     R5,CKRACT
                                        ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF RXACT>
                                        ;*************************************************************************
005654                                  CKRACT:
005654  012737  000007  002336                MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
005662  004537  003566                        JSR     R5,READI        ;READ USYRT STATUS
005666  122000                                USTATR
005670  000000                        1$:     .WORD   0
005672  032725  000001                        BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RXACT
005676  001422                                BEQ     2$              ;BR IF EXPECTED RXACT = 0
005700  132737  000040  005670                BITB    #RXACT,1$       ;SEE IF RXACT = 1
005706  001040                                BNE     3$              ;BR IF RXACT = 1
                                        ;STACK "RXACT NOT SET" MSG
005710                                          GTDF    EM71,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 10
005710  012737  000001  002172                                                    MOV     #T.EDF,ERRTYP
005716  012737  000012  002174                                                    MOV     #10,ERRNBR
005724  012737  014737  002176                                                    MOV     #EM71,ERRMSG
005732  012737  020714  002200                                                    MOV     #ERR12,ERRBLK
005740  000261                                SEC                     ;SET C BIT TO FLAG ERROR
005742  000423                                BR      4$              ;TAKE ERROR EXIT
005744  132737  000040  005670        2$:     BITB    #RXACT,1$       ;SEE IF RXACT = 0
005752  001416                                BEQ     3$              ;BR IF RXACT = 0
                                        ;STACK "RXACT NOT CLEARED" MSG
005754                                          GTDF    EM72,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 11
005754  012737  000001  002172                                                    MOV     #T.EDF,ERRTYP
005762  012737  000013  002174                                                    MOV     #11,ERRNBR
005770  012737  014755  002176                                                    MOV     #EM72,ERRMSG
005776  012737  020714  002200                                                    MOV     #ERR12,ERRBLK
006004  000261                                SEC                     ;SET C BIT TO FLAG ERROR
006006  000401                                BR      4$              ;TAKE ERROR EXIT
006010  000241                        3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006012  000205                        4$:     RTS     R5              ;RETURN
```

```
2924                                    .SBTTL ....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY
2925                                    ;*****************************************************************************
2926                                    ;* CKTBMT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TBMT IN THE USYRT
2927                                    ;*       STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
2928                                    ;*       STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
2929                                    ;*
2930                                    ;*       CALLING SEQUENCE :
2931                                    ;*       JSR     R5,CKTBMT
2932                                    ;*       .WORD   <BIT 0 IS EXPECTED VALUE OF TBMT>
2933                                    ;*****************************************************************************
2934  006014                            CKTBMT:
2935  006014  012737  000007  002336            MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
2936  006022  004537  003566                    JSR     R5,READT        ;READ USYRT STATUS
2937  006026  122000                            USTATR
2938  006030  000000                    1$:     .WORD   0
2939  006032  032725  000001                    BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TBMT
2940  006036  001422                            BEQ     2$              ;BR IF EXPECTED TBMT = 0
2941  006040  132737  000100  006030            BITB    #TBMT,1$        ;SEE IF TBMT = 1
2942  006046  001040                            BNE     3$              ;BR IF TBMT = 1
2943                                    ;STACK "TBMT NOT SET" MSG
2944  006050                                    GTDF    EM73,ERR12
                                                ;       QUEUE "DEVICE FATAL" ERROR # 12
      006050  012737  000001  002172                                            MOV     #1,EDF,ERRTYP
      006056  012737  000014  002174                                            MOV     #12,ERRNBR
      006064  012737  014777  002176                                            MOV     #EM73,ERRMSG
      006072  012737  020714  002200                                            MOV     #ERR12,ERRBLK
2945  006100  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2946  006102  000423                            BR      4$              ;TAKE ERROR EXIT
2947  006104  132737  000100  006030    2$:     BITB    #TBMT,1$        ;SEE IF TBMT = 0
2948  006112  001416                            BEQ     3$              ;BR IF TBMT = 0
2949                                    ;STACK "TBMT NOT CLEARED" MSG
2950  006114                                    GTDF    EM74,ERR12
                                                ;       QUEUE "DEVICE FATAL" ERROR # 13
      006114  012737  000001  002172                                            MOV     #1,EDF,ERRTYP
      006122  012737  000015  002174                                            MOV     #13,ERRNBR
      006130  012737  015014  002176                                            MOV     #EM74,ERRMSG
      006136  012737  020714  002200                                            MOV     #ERR12,ERRBLK
2951  006144  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2952  006146  000401                            BR      4$              ;TAKE ERROR EXIT
2953  006150  000241                    3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
2954  006152  000205                    4$:     RTS     R5              ;RETURN
2955
2956
2957
2958
```

```
                                        .SBTTL ....CKRDA -- CHECK RECEIVE DATA AVAILABLE
                                        ;*****************************************************************************
                                        ;* CKRDA - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RDA IN THE USYRT
                                        ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                        ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*      JSR     R5,CKRDA
                                        ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF RDA>
                                        ;*****************************************************************************
006154                                  CKRDA:
006154  012737  000007  002336                MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006162  004537  003566                        JSR     R5,READI        ;READ USYRT STATUS
006166  122000                               USTATR
006170  000000                         1$:   .WORD   0
006172  032725  000001                        BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RDA
006176  001422                               BEQ     2$              ;BR IF EXPECTED RDA = 0
006200  132737  000200  006170              BITB    #RDA,1$         ;SEE IF RDA = 1
006206  001040                               BNE     3$              ;BR IF RDA = 1
                                        ;STACK "RDA NOT SET" MSG
006210                                        GTDF    EM75,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 14
006210  012737  000001  002172                                                MOV     #T.EDF,ERRTYP
006216  012737  000016  002174                                                MOV     #14,ERRNBR
006224  012737  015035  002176                                                MOV     #EM75,ERRMSG
006232  012737  020714  002200                                                MOV     #ERR12,ERRBLK
006240  000261                               SEC                     ;SET C BIT TO FLAG ERROR
006242  000423                               BR      4$              ;TAKE ERROR EXIT
006244  132737  000200  006170        2$:   BITB    #RDA,1$         ;SEE IF RDA = 0
006252  001416                               BEQ     3$              ;BR IF RDA = 0
                                        ;STACK "RDA NOT CLEARED" MSG
006254                                        GTDF    EM76,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 15
006254  012737  000001  002172                                                MOV     #T.EDF,ERRTYP
006262  012737  000017  002174                                                MOV     #15,ERRNBR
006270  012737  015051  002176                                                MOV     #EM76,ERRMSG
006276  012737  020714  002200                                                MOV     #ERR12,ERRBLK
006304  000261                               SEC                     ;SET C BIT TO FLAG ERROR
006306  000401                               BR      4$              ;TAKE ERROR EXIT
006310  000241                         3$:   CLC                     ;CLEAR C BIT FOR NO ERRORS
006312  000205                         4$:   RTS     R5              ;RETURN
```

```
                                        .SBTTL ....CKRSA -- CHECK RECEIVER STATUS AVAILABLE
                                        ;**************************************************************************
                                        ;* CKRSA - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RSA IN THE USYRT
                                        ;*       STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                        ;*       STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                        ;*
                                        ;*       CALLING SEQUENCE :
                                        ;*       JSR     R5,CKRSA
                                        ;*       .WORD   <BIT 0 IS EXPECTED VALUE OF RSA>
                                        ;**************************************************************************
006314                                  CKRSA:
006314  012737  000007  002336                MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006322  004537  003566                       JSR     R5,READI        ;READ USYRT STATUS
006326  122000                              USTATR
006330  000000                        1$:   .WORD   0
006332  032725  000001                       BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RSA
006336  001422                              BEQ     2$              ;BR IF EXPECTED RSA = 0
006340  132737  000020  006330              BITB    #RSA,1$         ;SEE IF RSA = 1
006346  001040                              BNE     3$              ;BR IF RSA = 1
                                        ;STACK "RSA NOT SET" MSG
006350                                  GTDF    EM77,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 16
006350  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006356  012737  000020  002174                                                  MOV     #16,ERRNBR
006364  012737  015071  002176                                                  MOV     #EM77,ERRMSG
006372  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006400  000261                              SEC                     ;SET C BIT TO FLAG ERROR
006402  000423                              BR      4$              ;TAKE ERROR EXIT
006404  132737  000020  006330        2$:   BITB    #RSA,1$         ;SEE IF RSA = 0
006412  001416                              BEQ     3$              ;BR IF RSA = 0
                                        ;STACK "RSA NOT CLEARED" MSG
006414                                  GTDF    EM78,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 17
006414  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006422  012737  000021  002174                                                  MOV     #17,ERRNBR
006430  012737  015105  002176                                                  MOV     #EM78,ERRMSG
006436  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006444  000261                              SEC                     ;SET C BIT TO FLAG ERROR
006446  000401                              BR      4$              ;TAKE ERROR EXIT
006450  000241                        3$:   CLC                     ;CLEAR C BIT FOR NO ERRORS
006452  000205                        4$:   RTS     R5              ;RETURN
```

```
                                .SBTTL ....CKROR -- CHECK RECEIVER OVERRUN
                                ;*************************************************************************
                                ;* CKROR - THIS SUBROUTINE CHECKS FOR THE OCCURANCE OF RECEIVER OVERRUN IN THE
                                ;*       USYRT RECEIVER STATUS REGISTER (RDSRH), AND REPORTS AN ERROR IF IT IS
                                ;*        NOT PROPERLY SET TO THE STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                ;*
                                ;*       CALLING SEQUENCE :
                                ;*       JSR     R5,CKROR
                                ;*       .WORD   <BIT 0 IS EXPECTED VALUE OF ROR>
                                ;*************************************************************************
006454                          CKROR:
006454  012737  000001  002336        MOV     #1,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006462  004537  003566              JSR     R5,READI        ;READ RECEIVER STATUS
006466  120401                      RDSRH
006470  000000                1$:   .WORD   0
006472  032725  000001              BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF ROR
006476  001422                      BEQ     2$              ;BR IF EXPECTED ROR = 0
006500  132737  000010  006470      BITB    #ROR,1$         ;SEE IF ROR = 1
006506  001040                      BNE     3$              ;BR IF ROR = 1
                                ;STACK "RECEIVER OVRN NOT SET" MSG
006510                          GTDF    EM90,ERR12
                                                        ;       QUEUE "DEVICE FATAL" ERROR # 18
006510  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006516  012737  000022  002174                                                  MOV     #18,ERRNBR
006524  012737  015450  002176                                                  MOV     #EM90,ERRMSG
006532  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006540  000261                      SEC                     ;SET C BIT TO FLAG ERROR
006542  000423                      BR      4$              ;TAKE ERROR EXIT
006544  132737  000010  006470  2$: BITB    #ROR,1$         ;SEE IF ROR = 0
006552  001416                      BEQ     3$              ;BR IF ROR = 0
                                ;STACK "ROR NOT CLEARED" MSG
006554                          GTDF    EM91,ERR12
                                                        ;       QUEUE "DEVICE FATAL" ERROR # 19
006554  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006562  012737  000023  002174                                                  MOV     #19,ERRNBR
006570  012737  015501  002176                                                  MOV     #EM91,ERRMSG
006576  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006604  000261                      SEC                     ;SET C BIT TO FLAG ERROR
006606  000401                      BR      4$              ;TAKE ERROR EXIT
006610  000241                3$:   CLC                     ;CLEAR C BIT FOR NO ERRORS
006612  000205                4$:   RTS     R5              ;RETURN
```

```
                                        .SBTTL ....CKSEOM -- CHECK RSOM, REOM
                                        ;**************************************************************************
                                        ;* CKSEOM - THIS SUBROUTINE CHECKS FOR THE PROPER STATES OF RSOM, REOM IN THE
                                        ;*      USYRT RECEIVER STATUS REG (RDSRH) AND REPORTS AN ERROR IF THEY ARE NOT
                                        ;*      PROPERLY SET TO THE STATES OF BITS 0,1 IN THE WORD FOLLOWING THE CALL.
                                        ;*      IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                        ;*      IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                        ;*      DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*         JSR  R5,CKSEOM
                                        ;*         <BIT 0 IS EXPECTED VALUE OF RSOM, BIT 1 IS VALUE OF REOM>
                                        ;**************************************************************************
006614                                  CKSEOM:
006614  012737  000007  002336                  MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006622  004537  003566                          JSR     R5,READI        ;READ USYRT RECEIVER STATUS
006626  120401                                  RDSRH
006630  000000                          1$:     .WORD   0
006632  032725  000001                          BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RSOM
006636  001422                                  BEQ     2$              ;BR IF EXPECTED RSOM = 0
006640  132737  000001  006630                  BITB    #RSOM,1$        ;SEE IF RSOM = 1
006646  001040                                  BNE     3$              ;BR IF RSOM = 1
                                        ;STACK "RSOM NOT SET" MSG
006650                                          GTDF    EM29,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 20
006650  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006656  012737  000024  002174                                                  MOV     #20,ERRNBR
006664  012737  014427  002176                                                  MOV     #EM29,ERRMSG
006672  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006700  000261                                  SEC                     ;SET C BIT TO FLAG ERROR
006702  000473                                  BR      6$              ;TAKE ERROR EXIT
006704  132737  000001  006630          2$:     BITB    #RSOM,1$        ;SEE IF RSOM = 0
006712  001416                                  BEQ     3$              ;BR IF RSOM = 0
                                        ;STACK "RSOM NOT CLEARED" MSG
006714                                          GTDF    EM28,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 21
006714  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006722  012737  000025  002174                                                  MOV     #21,ERRNBR
006730  012737  014406  002176                                                  MOV     #EM28,ERRMSG
006736  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006744  000261                                  SEC                     ;SET C BIT TO FLAG ERROR
006746  000451                                  BR      6$              ;TAKE ERROR EXIT
006750  032765  000002  177776          3$:     BIT     #BIT1,-2(R5)    ;GET EXPECTED STATE OF REOM
006756  001422                                  BEQ     4$              ;BR IF EXPECTED REOM = 0
006760  132737  000002  006630                  BITB    #REOM,1$        ;SEE IF REOM = 1
006766  001040                                  BNE     5$              ;BR IF REOM = 1
                                        ;STACK "REOM NOT SET" MSG
006770                                          GTDF    EM31,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 22
006770  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006776  012737  000026  002174                                                  MOV     #22,ERRNBR
007004  012737  014465  002176                                                  MOV     #EM31,ERRMSG
007012  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
007020  000261                                  SEC                     ;SET C BIT TO FLAG ERROR
007022  000423                                  BR      6$              ;TAKE ERROR EXIT
007024  132737  000002  006630          4$:     BITB    #REOM,1$        ;SEE IF REOM = 0
007032  001416                                  BEQ     5$              ;BR IF REOM = 0
```

....CKSEOM -- CHECK RSOM, REOM

```
3107                                        ;STACK "REOM NOT CLEARED" MSG
3108 007034                                         GTDF    EM30,ERR12
                                                    ;       QUEUE "DEVICE FATAL" ERROR # 23
     007034  012737  000001  002172                                       MOV     #T.EDF,ERRTYP
     007042  012737  000027  002174                                       MOV     #23,ERRNBR
     007050  012737  014444  002176                                       MOV     #EM30,ERRMSG
     007056  012737  020714  002200                                       MOV     #ERR12,ERRBLK
3109 007064  000261                         SEC                ;SET C BIT TO FLAG ERROR
3110 007066  000401                         BR      6$         ;TAKE ERROR EXIT
3111 007070  000241                 5$:     CLC                ;CLEAR C BIT FOR NO ERRORS
3112 007072  000205                 6$:     RTS     R5         ;RETURN
3113
3114
```

```
                                        .SBTTL ....CHKTSO -- CHECK TRANSMIT SERIAL OUT BIT
                                        ;*************************************************************************
                                        ;* CHKTSO - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TSO IN THE USYRT
                                        ;*      STATUS REGISTER, AND SETS THE "C" BIT IF IT IS NOT SET TO THE STATE
                                        ;*      OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*      JSR     R5,CHKTSO
                                        ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TSO>
                                        ;*************************************************************************
007074                                  CHKTSO:
007074  012737  000007  002336          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
007102  004537  003566                  JSR     R5,READI        ;READ USYRT STATUS
007106  122000                          USTATR
007110  000000                  1$:     .WORD   0
007112  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TSO
007116  001422                          BEQ     2$              ;BR IF EXPECTED TSO = 0
007120  132737  000010  007110          BITB    #TSO,1$         ;SEE IF TSO = 1
007126  001040                          BNE     3$              ;BR IF TSO = 1
                                        ;*** STACK "TSO NOT SET" ERROR ***
007130                                  GTDF    EM100,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 24
007130  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
007136  012737  000030  002174                                                  MOV     #24,ERRNBR
007144  012737  015536  002176                                                  MOV     #EM100,ERRMSG
007152  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
007160  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007162  000423                          BR      4$              ;TAKE ERROR EXIT
                                        
007164  132737  000010  007110  2$:     BITB    #TSO,1$         ;SEE IF TSO = 0
007172  001416                          BEQ     3$              ;BR IF TSO = 0
                                        ;*** STACK "TSO NOT CLEARED" ERROR ***
007174                                  GTDF    EM101,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 25
007174  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
007202  012737  000031  002174                                                  MOV     #25,ERRNBR
007210  012737  015556  002176                                                  MOV     #EM101,ERRMSG
007216  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
007224  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007226  000401                          BR      4$              ;TAKE ERROR EXIT
007230  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
007232  000205                  4$:     RTS     R5              ;RETURN
```

```
                                .SBTTL ....INITRN -- INIT TRANSMISSION OF A MESSAGE
                                ;**************************************************************************
                                ;* INITRN - THIS SUBROUTINE INITIATES TRANSMISSION OF A MESSAGE, BY LOADING
                                ;*          THE USYRT PCSARL,H AND THE PCR WITH THE DATA PASSED IN THE 2 WORDS
                                ;*          FOLLOWING THE CALL ; LOADING AND CLOCKING 1 SOM UNTIL THE FIRST
                                ;*          SYNCH OR FLAG HAS BEEN SERIALIZED IN THE USYRT. THE PROGRAM MONITORS
                                ;*          ALL THE FLAGS IN THE USYRT STATUS REGISTER THROUGHOUT THE PROCESS.
                                ;*          IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION IS STACKED
                                ;*          AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE DISCRETION
                                ;*          OF THE CALLING ROUTINE OR SUBROUTINE.
                                ;*
                                ;*       CALLING SEQUENCE :
                                ;*          JSR  R5,INITRN
                                ;*          .WORD     <VALUE TO LOAD INTO USYRT PCSARL,H>
                                ;*          .WORD     <VALUE TO LOAD INTO USYRT PCR (PASSED IN LO BYTE)>
                                ;*                    <SPECIAL VIAORB MASKING VALUE (PASSED IN HI BYTE)>
                                ;**************************************************************************
007234                          INITRN:
007234  010146                          MOV     R1,-(SP)        ;SAVE R1
007236  004537  003712                  JSR     R5,WRITEI       ;RESET THE USYRT
007242  120000                          VIAORB
007244  000031                          RTSND!DTR!PRESET
007246  004537  003712                  JSR     R5,WRITEI       ;CLEAR USYRT RESET BIT
007252  120000                          VIAORB
007254  000030                          RTSND!DTR
007256  112537  007270                  MOVB    (R5)+,1$        ;GET VALUE TO LOAD INTO USYRT PCSARL
007262  004537  003712                  JSR     R5,WRITEI       ;LOAD USYRT PCSARL
007266  120404                          PCSARL
007270  000000                  1$:     .WORD   0
007272  112537  007304                  MOVB    (R5)+,2$        ;GET VALUE TO LOAD INTO PCSARH
007276  004537  003712                  JSR     R5,WRITEI       ;LOAD USYRT PCSARH
007302  120405                          PCSARH
007304  000000                  2$:     .WORD   0
007306  112537  007332                  MOVB    (R5)+,3$        ;GET VALUE TO LOAD INTO PCR
007312  005037  002402                  CLR     SAVLEN
007316  113737  007332  002402          MOVB    3$,SAVLEN       ;SAVE CHAR LENGTH BITS
007324  004537  003712                  JSR     R5,WRITEI       ;LOAD USYRT PCR
007330  120407                          PCR
007332  000000                  3$:     .WORD   0
007334  004537  003712                  JSR     R5,WRITEI       ;SET ACR FOR T1 ONE-SHOT MODE
007340  120013                          VIAACR
007342  000200                          200
007344  004537  003712                  JSR     R5,WRITEI       ;LOAD VIA T1L-L
007350  120006                          VIAT1C
007352  000300                          300
007354  004537  003712                  JSR     R5,WRITEI       ;LOAD VIA T1L-H
007360  120007                          VIAT1D
007362  000000                          000
007364  004537  005410                  JSR     R5,CKUSTS       ;CHK USYRT STATUS FOR INIT'D STATE
007370  000110                          110                     ; TBMT = 1, TSO = 1
007372  103454                          BCS     7$              ;IF ERROR, EXIT SUBROUTINE
                                
007374  013737  007530  007414          MOV     20$,13$         ;* SET UP DEFAULT VIAORB PARAMETERS
007402  142537  007414                  BICB    (R5)+,13$       ;* CLEAR ANY SPECIFIED VIAORB BITS.
                                
007406  004537  003712                  JSR     R5,WRITEI       ;SET UP USYRT
007412  120000                          VIAORB
```

....INITRN -- INIT TRANSMISSION OF A MESSAGE

```
3207 007414  000142                  13$:   TXEN!RXEN!TTLOOP        ;* THIS VALUE MIGHT BE MODIFIED ABOVE
3208
3209 007416  004537  003712                 JSR     R5,WRITEI       ;SET TSOM IN USYRT
3210 007422  120403                         TDSRH
3211 007424  000001                         TSOM
3212 007426  004537  003712                 JSR     R5,WRITEI       ;LOAD SYNCH CHAR INTO TX BUF
3213 007432  120402                         TDSRL
3214 007434  000226                         SYNCH
3215 007436  004537  006014                 JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
3216 007442  000000                         0
3217 007444  103427                         BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3218 007446  005001                         CLR     R1              ;INIT CYCLE COUNTER
3219 007450  004537  012072          4$:    JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3220 007454  000001                         1
3221 007456  004537  003566                 JSR     R5,READI        ;READ USYRT STATUS REG
3222 007462  122000                         USTATR
3223 007464  000000                  5$:    .WORD   0
3224 007466  132737  000100  007464         BITB    #TBMT,5$        ;SEE IF TBMT IS SET YET
3225 007474  001010                         BNE     6$              ;BR IF YES
3226 007476  005201                         INC     R1              ;INCR CYCLE COUNTER
3227 007500  020127  000003                 CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
3228 007504  002761                         BLT     4$              ;BR IF LESS THAN 3 CYCLES
3229 007506  004537  006014                 JSR     R5,CKTBMT       ;GO STACK "TBMT NOT SET" MSG
3230 007512  000001                         1
3231 007514  103403                         BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3232 007516  004537  005514          6$:    JSR     R5,CKTACT       ;CHK FOR TXACT = 1
3233 007522  000001                         1
3234 007524  012601                  7$:    MOV     (SP)+,R1        ;RESTORE R1
3235 007526  000205                         RTS     R5              ;RETURN (IF C = 1, WE HAD AN ERROR)
3236
3237 007530  000142                  20$:   TXEN!RXEN!TTLOOP        ;DEFAULT VALUE FOR VIAORB: ENABLE
3238                                                                 ;TX AND RX ON USYRT, ASSERT RTS, DTR
3239
```

```
                                        .SBTTL ....CKLPBK -- DETERMINE IF TEST CAN BE RUN
                                        ;*************************************************************************
                                        ;* CKLPBK - THIS SUBROUTINE DETERMINES IF THE TEST CALLING IT CAN BE RUN. THE
                                        ;*      TEST PASSES THE DESIRED MODEM INTERFACE TYPE IN THE WORD FOLLOWING THE
                                        ;*      CALL, AND IF A PROPER EXTERNAL LOOPBACK HAS BEEN PROVIDED BY THE
                                        ;*      OPERATOR FOR THAT INTERFACE, AND IF THE BAUD RATE IS CORRECT, A RETURN
                                        ;*      IS MADE WITH THE C BIT CLEARED, TO RUN THE TEST. IF NOT, A RETURN IS
                                        ;*      MADE WITH THE C BIT SET TO 1, SO THAT THE TEST CAN BE SKIPPED.
                                        ;*
                                        ;*      IF BIT 15 IS SET IN THE WORD FOLLOWING THE CALL, THE TEST WILL NOT
                                        ;*      BE RUN UNLESS THE H3254 AND H3255 TEST CONNECTORS ARE INSTALLED.
                                        ;*
                                        ;*      IF THE PROGRAM PASSES '0' IN THE WORD FOLLOWING THE CALL, THE SUBRTN
                                        ;*      WILL ATTEMPT TO RUN WHICHEVER MODEM INTERFACE IS SELECTED BY CABLE
                                        ;*      OR TEST CONNECTOR.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*         JSR  R5,CKLPBK
                                        ;*         .WORD   <DESIRED MODEM INTERFACE INFO>
                                        ;*
                                        ;*************************************************************************
007532                                  CKLPBK:
007532  032725  100000                          BIT     #TCCHEK,(R5)+   ;SEE IF H3254,5 CHECK IS DESIRED
007536  001407                                 BEQ     2$              ;BR IF NOT
007540  005737  002472                         TST     TSTCON          ;SEE IF H3254,5 INSTALLED
007544  001002                                 BNE     1$              ;BR IF NOT
007546  000137  010072                         JMP     46$             ;BR TO RUN TEST
007552  000137  010076                 1$:     JMP     48$             ;GO TO SKIP TEST
                                        ;IF NO EXTERNAL LPBK, SKIP TEST
007556  023727  002472  000004         2$:     CMP     TSTCON,#4       ;SEE IF NO LPBK
007564  001002                                 BNE     3$              ;BR IF LOOPBACK
007566  000137  010076                         JMP     48$             ;GO TO SKIP TEST

                                        ;*** SEE IF AN INTERFACE IS REQUESTED ***

007572  026527  177776  000001         3$:     CMP     -2(R5),#INTGRL  ;SEE IF INTEGRAL MODEM REQUESTED
007600  001406                                 BEQ     8$              ;BR IF INTGRL MODEM REQUESTED
007602  026527  177776  000002                 CMP     -2(R5),#EIAV35  ;SEE IF V.35 OR EIA REQUESTED
007610  001422                                 BEQ     16$             ;BR IF V.35 REQUESTED
007612  000137  007742                         JMP     32$             ;NONE REQUESTED, FIND AN INTERFACE TO TEST

                                        ;SEE IF INTEGRAL MODEM CAN BE RUN
007616  005737  002470                 8$:     TST     BRDTYP          ;SEE IF M8064
007622  001402                                 BEQ     10$             ;BR IF M8064
007624  000137  010040                         JMP     42$             ;WRONG OPTION, GO TO SKIP TEST
007630  005737  002472                 10$:    TST     TSTCON          ;SEE IF H3254, H3255 USED
007634  001406                                 BEQ     12$             ;BR IF YES
007636  023727  002472  000001         15$:    CMP     TSTCON,#1       ;SEE IF OPERATOR SPEC'D INTEGRAL MODEM
007644  001402                                 BEQ     12$             ;BR IF YES, TO RUN TEST
007646  000137  010006                         JMP     40$             ;WRONG INTERFACE, GO SKIP TEST
007652  000137  010072                 12$:    JMP     46$             ;GO TO RUN TEST

                                        ;SEE IF V.35 OR EIA CAN BE RUN
007656  005737  002470                 16$:    TST     BRDTYP          ;SEE IF M8053 BOARD
007662  001002                                 BNE     18$             ;BR IF M8053
007664  000137  010040                         JMP     42$             ;WRONG OPTION, GO TO SKIP TEST
007670  005737  002472                 18$:    TST     TSTCON          ;SEE IF H3254, H3255 USED
```

....CKLPBK -- DETERMINE IF TEST CAN BE RUN

```
3298 007674  001002                          BNE     23$             ;BR IF NOT
3299 007676  000137  010072          20$:     JMP     46              ;GO RUN THE TEST
3300 007702  023727  002472  000003  23$:     CMP     TSTCON,#3       ;SEE IF OPERATOR SPEC'D V.35
3301 007710  001006                          BNE     28$             ;BR IF NO
3302 007712  023727  002470  000001          CMP     BRDTYP,#1       ;TSTCOM MATCH BRDTYP?
3303 007720  001766                          BEQ     20$             ;YES; RUN TEST
3304 007722  000137  010006                  JMP     40$             ;WRONG INTERFACE, GO SKIP TEST
3305
3306 007726  023727  002472  000002  28$:     CMP     TSTCON,#2       ;SEE IF OPERATOR SPEC'D EIA
3307 007734  001760                          BEQ     20$             ;BR IF YES, TO RUN EIA
3308 007736  000137  010006                  JMP     40$             ;WRONG INTERFACE, GO SKIP TEST
3309
3310                                  ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
3311
3312 007742  005737  002470          32$:     TST     BRDTYP          ;SEE IF INTEGRAL MODEM SELECTED
3313 007746  001343                          BNE     16$             ;BR IF NOT (TEST FOR V35/EIA)
3314 007750  000137  007652                  JMP     12$             ;SEE IF INTEGRAL MODEM CAN BE RUN
3315
3316                                  ;PRINT "FOR BAUD RATE SPECIFIED,"
3317 007754                           38$:
3318 007754  023727  002412  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3319 007762  001063                          BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3320 007764                                  PRINTF  #FMT30
     007764  012746  013762                                                  MOV     #FMT30,-(SP)
     007770  012746  000001                                                  MOV     #1,-(SP)
     007774  010600                                                          MOV     SP,R0
     007776  104417                                                          TRAP    C$PNTF
     010000  062706  000004                                                  ADD     #4,SP
3321 010004  000434                          BR      48$             ;GO TO PRINT "TEST NOT RUN"
3322                                  ;PRINT "IMPROPER CONNECTOR TYPE SPECIFIED"
3323 010006                           40$:
3324 010006  023727  002412  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3325 010014  001046                          BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3326 010016                                  PRINTF  #FMT31          ; *********
     010016  012746  014017                                                  MOV     #FMT31,-(SP)
     010022  012746  000001                                                  MOV     #1,-(SP)
     010026  010600                                                          MOV     SP,R0
     010030  104417                                                          TRAP    C$PNTF
     010032  062706  000004                                                  ADD     #4,SP
3327 010036  000417                          BR      48$             ;GO TO PRINT "TEST NOT RUN"
3328
3329                                  ;PRINT "FOR OPTION SPECIFIED,"
3330 010040  023727  002412  000001  42$:     CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3331 010046  001031                          BNE     50$             ;BR IF NOT TO SKIP PRINTING
3332 010050                                  PRINTF  #FMT32
     010050  012746  014065                                                  MOV     #FMT32,-(SP)
     010054  012746  000001                                                  MOV     #1,-(SP)
     010060  010600                                                          MOV     SP,R0
     010062  104417                                                          TRAP    C$PNTF
     010064  062706  000004                                                  ADD     #4,SP
3333 010070  000402                          BR      48$             ;GO TO PRINT "TEST NOT RUN"
3334
3335                                  ;*** BRANCH HERE TO RUN TEST ***
3336 010072  000241                  46$:     CLC                     ;CLEAR C BIT TO RUN TEST
3337 010074  000417                          BR      52$             ;EXIT
3338
3339                                  ;*** BRANCH HERE TO SKIP TEST ***
```

M6

....CKLPBK -- DETERMINE IF TEST CAN BE RUN

```
3340                                        ;PRINT "TEST XX NOT RUN"
3341 010076                                 48$:
3342 010076  023727  002412  000001                CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3343 010104  001012                                BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3344 010106                                        PRINTF  #FMT19,TSTNUM
     010106  013746  002414                                                                MOV     TSTNUM,-(SP)
     010112  012746  013452                                                                MOV     #FMT19,-(SP)
     010116  012746  000002                                                                MOV     #2,-(SP)
     010122  010600                                                                        MOV     SP,R0
     010124  104417                                                                        TRAP    C$PNTF
     010126  062706  000006                                                                ADD     #6,SP
3345 010132  000261                         50$:    SEC                     ;SET C BIT TO SKIP TEST
3346 010134  000205                         52$:    RTS     R5              ;RETURN
3347
3348
3349
```

```
                                .SBTTL ....TXCHAR -- TRANSMIT A CHARACTER
                                ;*********************************************************************
                                ;* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHAR BY LOADING
                                ;*       THE USYRT TDSRL WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
                                ;*       FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
                                ;*       PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
                                ;*       MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
                                ;*       IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                ;*       IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                ;*       DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                ;*
                                ;*       CALLING SEQUENCE :
                                ;*       JSR     R5,TXCHAR
                                ;*       .WORD   <DATA FOR TDSRL IN LO BYTE>
                                ;*       .WORD   <NUMBER OF CYCLES TO CLOCK (IN LO BYTE)>
                                ;*               <SWITCH TO DISABLE INITIAL TBMT=0 CHECK (MSB IN HI BYTE)>
                                ;*********************************************************************
010136                          TXCHAR:
010136  010146                          MOV     R1,-(SP)        ;SAVE R1
010140  010246                          MOV     R2,-(SP)        ;SAVE R2
010142  012537  010154                  MOV     (R5)+,1$        ;GET DATA FOR TDSRL
010146  004537  003712                  JSR     R5,WRITEI       ;LOAD DATA INTO TDSRL
010152  120402                          TDSRL
010154  000000                  1$:     .WORD   0
010156  005001                          CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
010160  005002                          CLR     R2              ;CLEAR REQ'D CYCLE COUNT
010162  112502                          MOVB    (R5)+,R2        ;GET DESIRED NO. OF CYCLES
010164  001425                          BEQ     6$              ;BR IF NO CLOCKING DONE
010166  004537  005514          3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
010172  000001                          1
010174  103421                          BCS     6$              ;BR TO EXIT IF ERROR
010176  020102                          CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
010200  001414                          BEQ     5$              ;BR IF YES

010202  131527  000200                  BITB    (R5),#NCTBMT    ;* CHECK FOR "TBMT=0 CHECK" DISABLE
010206  001004                          BNE     7$              ;* BR IF MSB IS NOT SET

010210  004537  006014                  JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
010214  000000                          0
010216  103410                          BCS     6$              ;BR TO EXIT IF ERROR
010220  004537  012072          7$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
010224  000001                          1
010226  005201                          INC     R1              ;INCR CYCLE COUNT
010230  000756                          BR      3$              ;KEEP CLOCKING
010232  004537  006014          5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
010236  000001                          1
010240  012602                  6$:     MOV     (SP)+,R2        ;RESTORE R2
010242  012601                          MOV     (SP)+,R1        ;RESTORE R1
010244  005205                          INC     R5              ;ADJUST R5 FOR SANE RETURN
010246  000205                          RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

```
                                .SBTTL ....TXCTRL -- CONTROL MESSAGE TRANSMISSION (TDSRH)
                                ;*************************************************************************
                                ;* TXCTRL - THIS SUBROUTINE ALLOWS CONTROL OF MESSAGE TRANSMISSION BY LOADING
                                ;*      THE USYRT TDSRH WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
                                ;*      FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
                                ;*      PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
                                ;*      MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
                                ;*      IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                ;*      IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                ;*      DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,TXCTRL
                                ;*      .WORD   <DATA FOR TDSRH IN LO BYTE>
                                ;*      .WORD   <NUMBER OF CYCLES TO CLOCK>
                                ;*************************************************************************
010250                          TXCTRL:
010250  010146                         MOV     R1,-(SP)        ;SAVE R1
010252  010246                         MOV     R2,-(SP)        ;SAVE R2
010254  012537  010266                 MOV     (R5)+,2$        ;GET DATA FOR TDSRH
010260  004537  003712                 JSR     R5,WRITEI       ;LOAD DATA INTO TDSRH
010264  120403                         TDSRH
010266  000000                 2$:     .WORD   0
010270  005001                         CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
010272  012502                         MOV     (R5)+,R2        ;GET DESIRED NO. OF CYCLES
010274  001422                         BEQ     6$              ;BR IF NO CLOCKING DONE
010276  004537  005514         3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
010302  000001                         1
010304  103416                         BCS     6$              ;BR TO EXIT IF ERROR
010306  020102                         CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
010310  001411                         BEQ     5$              ;BR IF YES
010312  004537  006014                 JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
010316  000000                         0
010320  103410                         BCS     6$              ;BR TO EXIT IF ERROR
010322  004537  012072                 JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
010326  000001                         1
010330  005201                         INC     R1              ;INCR CYCLE COUNT
010332  000761                         BR      3$              ;KEEP CLOCKING
010334  004537  006014         5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
010340  000001                         1
010342  012602                 6$:     MOV     (SP)+,R2        ;RESTORE R2
010344  012601                         MOV     (SP)+,R1        ;RESTORE R1
010346  000205                         RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

```
3452                                    .SBTTL ....RXCHAR -- RECEIVE A CHARACTER
3453                                    ;*********************************************************************
3454                                    ;* RXCHAR - THIS SUBROUTINE READS THE USYRT RDSR AND CHECKS THE CONTENTS
3455                                    ;*       AGAINST THE DATA PASSED IN THE WORD FOLLOWING THE CALL.
3456                                    ;*       IF BIT0 = 0 IN THE SECOND WORD FOLLOWING THE CALL, THE RERR BIT IS
3457                                    ;*       NOT CHECKED AGAINST THE EXPECTED VALUE. THEN, IT CLOCKS
3458                                    ;*       THE LINE UNIT FOR THE NO. OF CYCLES PASSED IN THE THIRD WORD
3459                                    ;*       FOLLOWING THE CALL. THE PROGRAM CONTINUALLY MONITORS RDA AND RXACT.
3460                                    ;*       IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
3461                                    ;*       IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
3462                                    ;*       DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
3463                                    ;*
3464                                    ;*       CALLING SEQUENCE :
3465                                    ;*       JSR     R5,RXCHAR
3466                                    ;*       .WORD   <EXPECTED RDSRL IN LO BYTE, RDSRH IN HI BYTE>
3467                                    ;*       .WORD   <=0 FOR NO RERR CHK, =1 FOR RERR CHK>
3468                                    ;*       .WORD   <NUMBER OF CYCLES TO CLOCK (IN LO BYTE)>
3469                                    ;*               <SPECIAL DISABLE SWITCHES: NOCRDA,NFCRDA,NCRACT(IN HI BYTE)>
3470                                    ;*********************************************************************
3471 010350                            RXCHAR:
3472 010350  010146                            MOV     R1,-(SP)        ;SAVE R1
3473 010352  010246                            MOV     R2,-(SP)        ;SAVE R2
3474 010354  004537  003566                    JSR     R5,READI        ;READ RDSRH
3475 010360  120401                            RDSRH
3476 010362  000000                    2$:     .WORD   0
3477 010364  004537  003566                    JSR     R5,READI        ;READ RDSRL
3478 010370  120400                            RDSRL
3479 010372  000000                    1$:     .WORD   0
3480 010374  111501                            MOVB    (R5),R1         ;GET EXPECTED RDSRL
3481 010376  042701  177400                    BIC     #177400,R1      ;MASK OFF UNUSED BITS
3482 010402  023727  002402  000347            CMP     SAVLEN,#TXDL!RXDL ;SEE IF 7-BIT CHARS BEING USED
3483 010410  001005                            BNE     3$              ;BR IF NOT 7-BIT CHARS
3484 010412  142737  000200  010372            BICB    #BIT7,1$        ;CLEAR 8TH BIT FOR COMPARE
3485 010420  142701  000200                    BICB    #BIT7,R1
3486 010424  123701  010372            3$:     CMPB    1$,R1           ;COMPARE RCV'D CHAR TO EXPECTED
3487 010430  001462                            BEQ     6$              ;BR IF MATCH
3488 010432  004537  003566                    JSR     R5,READI        ;READ USYRT STATUS REG
3489 010436  122000                            USTATR
3490 010440  000000                    4$:     .WORD   0
3491 010442  132737  000002  010440            BITB    #TXU,4$         ;SEE IF TX UNDERRUN OCCURRED
3492 010450  001421                            BEQ     5$              ;BR IF NOT
3493 010452  012737  000007  002336            MOV     #7,REGNUM       ;SET USYRT REG NO. FOR STATUS REG
3494                                    ;STACK "TX UNDERRUN" ERROR
3495 010460                                    GTDF    EM54,ERR12
                                                        ;       QUEUE "DEVICE FATAL" ERROR # 26
     010460  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
     010466  012737  000032  002174                                                  MOV     #26,ERRNBR
     010474  012737  014626  002176                                                  MOV     #EM54,ERRMSG
     010502  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3496 010510  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3497 010514  005037  002336            5$:     CLR     REGNUM          ;SET USYRT REG NO. FOR RDSRL
3498 010520  005037  002324                    CLR     GDATA           ;SET EXPECTED DATA
3499 010524  110137  002324                    MOVB    R1,GDATA
3500 010530  005037  002326                    CLR     BDATA           ;SET ACTUAL DATA
3501 010534  113737  010372  002326            MOVB    1$,BDATA
3502                                    ;STACK "RCV'D DATA MISCOMPARE" ERROR
3503 010542                                    GTDF    EM34,ERR10
```

```
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 27
     010542  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
     010550  012737  000033  002174                                                  MOV     #27,ERRNBR
     010556  012737  014502  002176                                                  MOV     #EM34,ERRMSG
     010564  012737  020364  002200                                                  MOV     #ERR10,ERRBLK
3504 010572  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
3505 010576  116501  000001          6$:     MOVB    1(R5),R1        ;GET RDSRH
3506 010602  042701  177400                  BIC     #177400,R1      ;MASK OFF UNUSED BITS
3507 010606  123701  010362                  CMPB    2$,R1           ;COMPARE RCV'D STATUS TO EXPECTED
3508 010612  001016                          BNE     7$              ;BR IF MISMATCH
3509 010614  000137  011474                  JMP     17$             ;CONTINUE
3510 010620  012737  000001  002336          MOV     #1,REGNUM       ;SET USYRT REG NO. FOR RDSRH
3511 010626  005037  002324                  CLR     GDATA           ;SET EXPECTED DATA
3512 010632  110137  002324                  MOVB    R1,GDATA
3513 010636  005037  002326                  CLR     BDATA           ;SET ACTUAL DATA
3514 010642  113737  010362  002326          MOVB    2$,BDATA
3515 010650  012737  000001  002336  7$:     MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
3516 010656  032765  000001  000002          BIT     #RERCHK,2(R5)   ;SEE IF RCV ERROR BIT SHOULD BE IGNORED
3517 010664  001447                          BEQ     9$              ;BR IF YES
3518                                 ;CHECK RERR BIT
3519 010666  132701  000200                  BITB    #RERR,R1        ;SEE IF EXPECTED BIT = 1
3520 010672  001022                          BNE     8$              ;BR IF YES
3521 010674  132737  000200  010362          BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 0
3522 010702  001440                          BEQ     9$              ;BR IF YES
3523                                 ;STACK "RERR NOT CLEARED" MSG
3524 010704                                  GTDF    EM35,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 28
     010704  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
     010712  012737  000034  002174                                                  MOV     #28,ERRNBR
     010720  012737  014530  002176                                                  MOV     #EM35,ERRMSG
     010726  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3525 010734  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
3526 010740  132737  000200  010362  8$:     BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 1
3527 010746  001016                          BNE     9$              ;BR IF YES
3528                                 ;STACK "RERR NOT SET" MSG
3529 010750                                  GTDF    EM36,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 29
     010750  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
     010756  012737  000035  002174                                                  MOV     #29,ERRNBR
     010764  012737  014551  002176                                                  MOV     #EM36,ERRMSG
     010772  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3530 011000  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
3531                                 ;CHECK ROR BIT
3532 011004  132701  000010          9$:     BITB    #ROR,R1         ;SEE IF EXPECTED BIT = 1
3533 011010  001022                          BNE     10$             ;BR IF YES
3534 011012  132737  000010  010362          BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 0
3535 011020  001440                          BEQ     11$             ;BR IF YES
3536                                 ;STACK "ROR NOT CLEARED" MSG
3537 011022                                  GTDF    EM16,ERR12
                                                                ;       QUEUE "DEVICE FATAL" ERROR # 30
     011022  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
     011030  012737  000036  002174                                                  MOV     #30,ERRNBR
     011036  012737  014340  002176                                                  MOV     #EM16,ERRMSG
     011044  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3538 011052  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
3539 011056  132737  000010  010362  10$:     BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 1
3540 011064  001016                          BNE     11$             ;BR IF YES
```

....RXCHAR -- RECEIVE A CHARACTER

```
3541                                            ;STACK "ROR NOT SET" MSG
3542 011066                                             GTDF    EM14,ERR12
                                                ;       QUEUE "DEVICE FATAL" ERROR # 31
     011066  012737  000001  002172                                      MOV     #T.EDF,ERRTYP
     011074  012737  000037  002174                                      MOV     #31,ERRNBR
     011102  012737  014324  002176                                      MOV     #EM14,ERRMSG
     011110  012737  020714  002200                                      MOV     #ERR12,ERRBLK
3543 011116  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3544                                            ;CHECK RABGA BIT
3545 011122  132701  000004            11$:    BITB    #RABGA,R1       ;SEE IF EXPECTED BIT = 1
3546 011126  001022                            BNE     12$             ;BR IF YES
3547 011130  132737  000004  010362            BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 0
3548 011136  001440                            BEQ     13$             ;BR IF YES
3549                                            ;STACK "RABGA NOT CLEARED" MSG
3550 011140                                             GTDF    EM39,ERR12
                                                ;       QUEUE "DEVICE FATAL" ERROR # 32
     011140  012737  000001  002172                                      MOV     #T.EDF,ERRTYP
     011146  012737  000040  002174                                      MOV     #32,ERRNBR
     011154  012737  014566  002176                                      MOV     #EM39,ERRMSG
     011162  012737  020714  002200                                      MOV     #ERR12,ERRBLK
3551 011170  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3552 011174  132737  000004  010362    12$:    BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 1
3553 011202  001016                            BNE     13$             ;BR IF YES
3554                                            ;STACK "RABGA NOT SET" MSG
3555 011204                                             GTDF    EM40,ERR12
                                                ;       QUEUE "DEVICE FATAL" ERROR # 33
     011204  012737  000001  002172                                      MOV     #T.EDF,ERRTYP
     011212  012737  000041  002174                                      MOV     #33,ERRNBR
     011220  012737  014610  002176                                      MOV     #EM40,ERRMSG
     011226  012737  020714  002200                                      MOV     #ERR12,ERRBLK
3556 011234  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3557                                            ;CHECK REOM BIT
3558 011240  132701  000002            13$:    BITB    #REOM,R1        ;SEE IF EXPECTED BIT = 1
3559 011244  001022                            BNE     14$             ;BR IF YES
3560 011246  132737  000002  010362            BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 0
3561 011254  001440                            BEQ     15$             ;BR IF YES
3562                                            ;STACK "REOM NOT CLEARED" MSG
3563 011256                                             GTDF    EM30,ERR12
                                                ;       QUEUE "DEVICE FATAL" ERROR # 34
     011256  012737  000001  002172                                      MOV     #T.EDF,ERRTYP
     011264  012737  000042  002174                                      MOV     #34,ERRNBR
     011272  012737  014444  002176                                      MOV     #EM30,ERRMSG
     011300  012737  020714  002200                                      MOV     #ERR12,ERRBLK
3564 011306  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3565 011312  132737  000002  010362    14$:    BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 1
3566 011320  001016                            BNE     15$             ;BR IF YES
3567                                            ;STACK "REOM NOT SET" MSG
3568 011322                                             GTDF    EM31,ERR12
                                                ;       QUEUE "DEVICE FATAL" ERROR # 35
     011322  012737  000001  002172                                      MOV     #T.EDF,ERRTYP
     011330  012737  000043  002174                                      MOV     #35,ERRNBR
     011336  012737  014465  002176                                      MOV     #EM31,ERRMSG
     011344  012737  020714  002200                                      MOV     #ERR12,ERRBLK
3569 011352  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3570                                            ;CHECK RSOM BIT
3571 011356  132701  000001            15$:    BITB    #RSOM,R1        ;SEE IF EXPECTED BIT = 1
3572 011362  001022                            BNE     16$             ;BR IF YES
```

```
3573 011364  132737  000001  010362          BITB    @RSOM,2$        ;SEE IF ACTUAL BIT = 0
3574 011372  001440                          BEQ     17$             ;BR IF YES
3575                                ;STACK "RSOM NOT CLEARED" MSG
3576 011374                                  GTDF    EM28,ERR12
                                             ;       QUEUE "DEVICE FATAL" ERROR # 36
     011374  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
     011402  012737  000044  002174                                          MOV     #36,ERRNBR
     011410  012737  014406  002176                                          MOV     #EM28,ERRMSG
     011416  012737  020714  002200                                          MOV     #ERR12,ERRBLK
3577 011424  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
3578 011430  132737  000001  010362  16$:    BITB    @RSOM,2$        ;SEE IF ACTUAL BIT = 1
3579 011436  001016                          BNE     17$             ;BR IF YES
3580                                ;STACK "RSOM NOT SET" MSG
3581 011440                                  GTDF    EM29,ERR12
                                             ;       QUEUE "DEVICE FATAL" ERROR # 37
     011440  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
     011446  012737  000045  002174                                          MOV     #37,ERRNBR
     011454  012737  014427  002176                                          MOV     #EM29,ERRMSG
     011462  012737  020714  002200                                          MOV     #ERR12,ERRBLK
3582 011470  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
3583
3584 011474  116502  000004  17$:    MOVB    4(R5),R2        ;GET DESIRED NO. OF CYCLES
3585 011500  005001                  CLR     R1              ;INIT CYCLE COUNT
3586
3587 011502  136527  000005  000040  18$:    BITB    5(R5),@BIT5     ;* IS RXACT CHECK TO BE DISABLED ?
3588 011510  001004                  BNE     31$             ;* BR IF YES
3589 011512  004537  005654          JSR     R5,CKRACT       ;CHK FOR RACT = 1
3590 011516  000001                  1
3591 011520  103433                  BCS     20$             ;BR TO EXIT IF ERROR
3592
3593 011522  020102          31$:    CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3594 011524  001415                  BEQ     19$             ;BR IF YES
3595
3596 011526  136527  000005  000200          BITB    5(R5),@BIT7     ;* SEE IF INITIAL RDA CHECK DESIRED
3597 011534  001004                  BNE     22$             ;* BR IF NO
3598 011536  004537  006154          JSR     R5,CKRDA        ;CHK FOR RDA = 0
3599 011542  000000                  0
3600 011544  103421                  BCS     20$             ;BR TO EXIT IF ERROR
3601
3602 011546  004537  012072  22$:    JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3603 011552  000001                  1
3604 011554  005201                  INC     R1              ;INCR CYCLE COUNT
3605 011556  000751                  BR      18$             ;CONTINUE CLOCKING
3606
3607 011560  136527  000005  000100  19$:    BITB    5(R5),@BIT6     ;* IS FINAL RDA CHECK TO BE SKIPPED ?
3608 011566  001004                  BNE     30$             ;* BR IF YES
3609 011570  004537  006154          JSR     R5,CKRDA        ;CHK RDA = 1
3610 011574  000001                  1
3611 011576  103404                  BCS     20$             ;BR IF ERROR
3612
3613 011600  062705  000006  30$:    ADD     @6,R5           ;FIX UP RETURN ADRS
3614 011604  000241                  CLC                     ;SET C = 0 FOR NO ERROR
3615 011606  000403                  BR      21$             ;TAKE ERROR-FREE EXIT
3616 011610  062705  000006  20$:    ADD     @6,R5           ;FIX UP RETURN ADDRESS
3617 011614  000261                  SEC                     ;SET C BIT FOR ERROR
3618 011616  012602          21$:    MOV     (SP)+,R2        ;RESTORE R2
3619 011620  012601                  MOV     (SP)+,R1        ;RESTORE R1
```

....RXCHAR -- RECEIVE A CHARACTER

```
3620 011622  000205                          RTS     R5              ;RETURN
3621
```

```
                                        .SBTTL ....RCV1ST -- RECEIVE FIRST CHARACTER OF MESSAGE
                                        ;*************************************************************************
                                        ;* RCV1ST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE AND MONITORS
                                        ;*       THE STATUS OF THE RECEIVER. FIRST, A CHECK IS MADE FOR RXACT = 0,
                                        ;*       RDA = 0, RSA = 0, RSOM = 0. THEN, THE LINE UNIT IS CLOCKED UNTIL
                                        ;*       RDA = 1. THE PROGRAM CHECKS FOR THIS TO OCCUR WITHIN 3 CYCLES AFTER
                                        ;*       THE NO. OF CYCLES PASSED IN THE SECOND WORD FOLLOWING THE CALL.
                                        ;*       IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                        ;*       IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                        ;*       DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                        ;*
                                        ;*       CALLING SEQUENCE :
                                        ;*       JSR     R5,RCV1ST
                                        ;*       .WORD   <EXPECTED RECEIVER CYCLE COUNT>
                                        ;*************************************************************************
011624                                  RCV1ST:
011624  010146                                  MOV     R1,-(SP)        ;SAVE R1
011626  010246                                  MOV     R2,-(SP)        ;SAVE R2
011630  005001                                  CLR     R1              ;INIT CYCLE COUNT
011632  012502                                  MOV     (R5)+,R2                ;GET CYCLE COUNT LIMIT
011634  062702  000003                         ADD     #3,R2
011640  004537  005654                         JSR     R5,CKRACT       ;CHK FOR RXACT = 0
011644  000000                                  0
011646  103446                                  BCS     6$              ;BR TO EXIT IF ERROR
011650  004537  006154                         JSR     R5,CKRDA        ;CHK FOR RDA = 0
011654  000000                                  0
011656  103442                                  BCS     6$              ;BR TO EXIT IF ERROR
011660  004537  006614                         JSR     R5,CKSEOM       ;CHK FOR RSOM = 0, REOM = 0
011664  000000                                  0
011666  103436                                  BCS     6$              ;BR TO EXIT IF ERROR
011670  004537  012072          1$:             JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
011674  000001                                  1
011676  005201                                  INC     R1              ;INCREMENT CYCLE COUNT
011700  004537  003566                         JSR     R5,READI        ;READ USYRT STATUS REG
011704  122000                                  USTATR
011706  000000                  2$:             .WORD   0
011710  132737  000200  011706                  BITB    #RDA,2$         ;SEE IF RDA SET YET
011716  001006                                  BNE     3$              ;BR IF YES
011720  020102                                  CMP     R1,R2           ;SEE IF LIMIT EXCEEDED
011722  002762                                  BLT     1$              ;BR IF NOT YET
011724  004537  006154                         JSR     R5,CKRDA        ;GO STACK "RDA NOT SET" MSG
011730  000001                                  1
011732  103414                                  BCS     6$              ;BR TO EXIT IF ERROR
011734  020165  177776          3$:             CMP     R1,-2(R5)       ;SEE IF LESS THAN REQUIRED CYCLES
011740  002004                                  BGE     4$              ;BR IF NOT
011742  004537  006154                         JSR     R5,CKRDA        ;GO STACK "RDA NOT CLEARED" MSG
011746  000000                                  0
011750  103405                                  BCS     6$              ;BR TO EXIT IF ERROR
011752  004537  005654          4$:             JSR     R5,CKRACT       ;CHK FOR RXACT = 1
011756  000001                                  1
011760  103401                                  BCS     6$              ;BR TO EXIT IF ERROR
011762  000241                  5$:             CLC                     ;CLEAR C BIT FOR NO ERRORS
011764  012602                  6$:             MOV     (SP)+,R2        ;RESTORE R2
011766  012601                                  MOV     (SP)+,R1        ;RESTORE R1
011770  000205                                  RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

```
                                        .SBTTL ....ENTRAN -- SHUT DOWN TRANSMITTER/RECEIVER
                                        ;*************************************************************************
                                        ;* ENTRAN - THIS SUBROUTINE TERMINATES A MESSAGE BY CLEARING TXEN AND RXEN,
                                        ;*      CLOCKING THE LINE UNIT FOR THE NUMBER OF CYCLES PASSED IN THE WORD
                                        ;*      FOLLOWING THE CALL, AND CHECKING FOR THE USYRT TRANSMITTER AND
                                        ;*      RECEIVER TO BE SHUT DOWN.
                                        ;*      IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                        ;*      IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                        ;*      DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*        JSR   R5,ENTRAN
                                        ;*        MSB SET=NO TTLOOP ! LOWER BYTE = <NO. OF CYCLES TO CLOCK>
                                        ;*************************************************************************
011772                                  ENTRAN:
011772  012737  000002  012042                  MOV     #TTLOOP,1$      ;INIT DEFAULT VIAORB (TTLOOP=1)
012000  112537  012050                          MOVB    (R5)+,2$        ;GET DESIRED # OF TICKS (LOWER BYTE)
012004  105725                                  TSTB    (R5)+           ;SEE IF MSB SET (TTLOOP DISABLE BIT)
012006  100002                                  BPL     4$              ;IS IT?
012010  005037  012042                          CLR     1$              ;IF YES: CLEAR VIAORB VALUE
012014  004537  005514          4$:             JSR     R5,CKTACT       ;CHK FOR TXACT = 1
012020  000001                                  1
012022  103422                                  BCS     6$              ;BR IF ERROR
012024  004537  005654                          JSR     R5,CKRACT       ;CHK FOR RXACT = 1
012030  000001                                  1
012032  103416                                  BCS     6$
012034  004537  003712                          JSR     R5,WRITEI       ;CLEAR TXEN AND RXEN IN USYRT
012040  120000                                  VIAORB                  ;
012042  000002          1$:             TTLOOP                          ;** HOLE FOR ACTUAL VIAORB WORD **
012044  004537  012072                          JSR     R5,STEPLU       ;CLOCK LU FOR DESIRED NO. OF CYCLES
012050  000000          2$:             .WORD   0
012052  004537  005514                          JSR     R5,CKTACT       ;CHK FOR TXACT = 0
012056  000000                                  0
012060  103403                                  BCS     6$              ;BR IF ERROR
012062  004537  005654                          JSR     R5,CKRACT       ;CHK FOR RXACT = 0
012066  000000                                  0
012070  000205          6$:             RTS     R5
```

```
3721                                    .SBTTL ....STEPLU -- CLOCK THE USYRT N TIMES
3722                                    ;*****************************************************************************
3723                                    ;* STEPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NUMBER OF CYCLES
3724                                    ;*       PASSED IN THE WORD FOLLOWING THE CALL. THE VIA ACR MUST BE PREVIOUSLY
3725                                    ;*       SET UP FOR T1 ONE-SHOT MODE, AND THE T1 LATCHES MUST BE PREVIOUSLY SET
3726                                    ;*       TO CONTROL THE WIDTH OF THE CLOCK PULSE. ALL THAT THIS SUBROUTINE
3727                                    ;*       DOES IS TO LOAD 000 INTO THE HI BYTE OF THE T1 COUNTER, FOR THE
3728                                    ;*       DESIRED NUMBER OF TIMES.
3729                                    ;*
3730                                    ;*       CALLING SEQUENCE :
3731                                    ;*       JSR     R5,STEPLU
3732                                    ;*       .WORD   <NUMBER OF CYCLES TO CLOCK>
3733                                    ;*****************************************************************************
3734 012072                            STEPLU:
3735 012072  010146                             MOV     R1,-(SP)        ;SAVE R1
3736 012074  012501                             MOV     (R5)+,R1        ;INIT CYCLE COUNTER
3737 012076  004537  003712             1$:     JSR     R5,WRITEI       ;LOAD T1C-H, START COUNTER, CLOCK 1 CYCLE
3738 012102  120005                             VIAT1B
3739 012104  000000                             000
3740 012106  005301                             DEC     R1              ;DECR CYCLE COUNTER
3741 012110  001372                             BNE     1$              ;BR IF ALL CYCLES NOT DONE YET
3742 012112  012601                             MOV     (SP)+,R1        ;RESTORE R1
3743 012114  000205                             RTS     R5              ;RETURN
3744
3745
3746
3747
3748
3749
```

GLOBAL ERROR REPORT SECTION

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;////////////////////////////////////////////////////////////////
                                ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
                                ;/      THAT ARE USED IN MORE THAN ONE TEST.
                                ;////////////////////////////////////////////////////////////////

                                .NLIST  BEX
012116  045  116  045  ENDEMB: .ASCIZ  /%N%N/
012123  045  116  000  NEWLIN: .ASCIZ  /%N/              ;USED TO TERMINATE ERROR MESSAGES

012126  045  116  045  FMT2:   .ASCIZ  /%N%AFAILING REG = %T%ASEL%01/
012163  045  116  045  FMT3:   .ASCIZ  /%N%A  EXPECTED:  %03%A   ACTUAL:  %03%A   XOR:  %03/
012247  045  116  045  FMT4:   .ASCIZ  /%N%ATHE CONTENTS OF ALL%T%N%T/
012305  045  116  045  FMT4A:  .ASCIZ  /%N%S1%03%S5%03%S5%03%S5%03/
012340  045  116  045  FMT4B:  .ASCIZ  /%N%T/
012345  045  116  045  FMT4C:  .ASCIZ  /%N%S5%03%S5%03%S5%03%S5%03/
012400  045  116  045  FMT5:   .ASCIZ  /%N%A  WHEN %03%A LOADED INTO BSEL1/
012443  045  116  045  FMT5A:  .ASCIZ  /%N%A  ATTEMPTING "M-LOOP" FUNCTION CODE %02%A (%T%A)/
012530  045  116  045  FMT7:   .ASCIZ  /%N%AMDIAG %%03%A FAILED/

012560  045  116  045  FMT10:  .ASCIZ  /%N%A  EXPECTED:%08%A  ACTUAL:%08%A  XOR:%08/
012634  045  101  040  FMT10A: .ASCIZ  /%A  LSI ADDR:%08/
012655  045  116  045  FMT11:  .ASCIZ  /%N%08%08%08%08/
012674  045  116  045  FMT12:  .ASCIZ  /%N%N%T/
012703  045  116  045  FMT13:  .ASCIZ  /%N%T%03%S2%03%S2%03%S2%03%S2%03%S2%03/
012751  045  123  062  FMT14:  .ASCIZ  /%S2%03%S2%03/
012766  045  101  040  FMT15:  .ASCIZ  /%A  DETECTED IN %T%T%A . ./
013020  045  101  040  FMT15A: .ASCIZ  /%A  DETECTED @ TEST PATTERN ELEMENT # %D2/

013072  045  116  045  FMT16:  .ASCIZ  /%N%T%03%S4%03%S%03/
013115  045  116  045  FMT16A: .ASCIZ  /%N%T%03%S%03%S%03%S4%03%S%03%S%03/
013157  045  101  040  FMT17:  .ASCIZ  /%A     VALUE SENT TO MPR CONTROL REGISTER:  %03/
013236  045  116  045  FMT17A: .ASCIZ  /%N%A     VALUE READ FROM CONTROL REGISTER:    %03/
013317  045  116  045  FMT17B: .ASCIZ  /%N%A     LSI-11 MEMORY ADDRESS ACCESSED:%08/
013372  045  116  045  FMT17C: .ASCIZ  /%N%A  INFORMATION ON THE FIRST OF %D5%A ERRORS:/

013452  045  116  045  FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
013503  045  124  045  FMT21:  .ASCIZ  /%T%06%N/
013513  045  116  045  FMT22:  .ASCIZ  /%N%AFAILING REG: /
013535  045  101  105  FMT23:  .ASCIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%S5%AXOR: %03%N/
013614  045  116  045  FMT24:  .ASCIZ  /%N%T%N%T%N/
013627  045  117  063  FMT25:  .ASCIZ  /%03%S5%03%S5%03%S5%03%N/
013657  045  123  064  FMT26:  .ASCIZ  /%S4%03%S5%03%S5%03%S5%03%N/
013712  045  124  045  FMT27:  .ASCIZ  /%T%T%N/
013721  045  101  105  FMT28:  .ASCIZ  /%AEXTENDED REG AX%01%A-%T%N/
013755  045  124  045  FMT29:  .ASCIZ  /%T%N/
013762  045  116  045  FMT30:  .ASCIZ  /%N%AFOR BAUD RATE SPECIFIED,/
014017  045  116  045  FMT31:  .ASCIZ  /%N%AIMPROPER CONNECTOR TYPE SPECIFIED/
014065  045  116  045  FMT32:  .ASCIZ  /%N%AFOR OPTION SPECIFIED,/
014117  045  116  045  FMT39:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
014150  045  116  045  FMT40:  .ASCIZ  /%N%AFAILING RAM ADRS: %06%A (OCT)%N/

014214  125  123  131  EM2:    .ASCIZ  /USYRT NOT INITIALIZED BY PROGRAM RESET/
014263  115  111  103  EM3:    .ASCIZ  /MICRO-DIAG. FAILURE/
014307  115  122  104  EM4:    .ASCIZ  /MRDY TIMEOUT/
014324  122  117  122  EM14:   .ASCIZ  /ROR NOT SET/
014340  122  117  122  EM16:   .ASCIZ  /ROR NOT CLEARED/
```

GLOBAL ERROR REPORT SECTION

```
3808 014360      122      105      101  EM25:   .ASCIZ  'READ/WRITE DATA ERROR'
3809 014406      122      123      117  EM28:   .ASCIZ  /RSOM NOT CLEARED/
3810 014427      122      123      117  EM29:   .ASCIZ  /RSOM NOT SET/
3811 014444      122      105      117  EM30:   .ASCIZ  /REOM NOT CLEARED/
3812 014465      122      105      117  EM31:   .ASCIZ  /REOM NOT SET/
3813 014502      122      103      126  EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
3814 014530      122      105      122  EM35:   .ASCIZ  /RERR NOT CLEARED/
3815 014551      122      105      122  EM36:   .ASCIZ  /RERR NOT SET/
3816 014566      122      101      102  EM39:   .ASCIZ  /RABGA NOT CLEARED/
3817 014610      122      101      102  EM40:   .ASCIZ  /RABGA NOT SET/
3818 014626      124      130      040  EM54:   .ASCIZ  /TX UNDERRUN ERROR/
3819 014650      125      123      131  EM68:   .ASCIZ  /USYRT STATUS INCORRECT/
3820 014677      124      130      101  EM69:   .ASCIZ  /TXACT NOT SET/
3821 014715      124      130      101  EM70:   .ASCIZ  /TXACT NOT CLEARED/
3822 014737      122      130      101  EM71:   .ASCIZ  /RXACT NOT SET/
3823 014755      122      130      101  EM72:   .ASCIZ  /RXACT NOT CLEARED/
3824 014777      124      102      115  EM73:   .ASCIZ  /TBMT NOT SET/
3825 015014      124      102      115  EM74:   .ASCIZ  /TBMT NOT CLEARED/
3826 015035      122      104      101  EM75:   .ASCIZ  /RDA NOT SET/
3827 015051      122      104      101  EM76:   .ASCIZ  /RDA NOT CLEARED/
3828 015071      122      123      101  EM77:   .ASCIZ  /RSA NOT SET/
3829 015105      122      123      101  EM78:   .ASCIZ  /RSA NOT CLEARED/
3830 015125      122      101      115  EM79:   .ASCIZ  /RAM ERROR LOADING MICROCODE/
3831 015161      103      101      122  EM80:   .ASCIZ  /CARRIER NOT SET/
3832 015201      103      101      122  EM81:   .ASCIZ  /CARRIER NOT CLEARED/
3833 015225      111      116      126  EM82:   .ASCIZ  /INVALID ERROR CODE FROM 6502/
3834 015262      115      117      104  EM83:   .ASCIZ  /MODEM STATUS INCORRECT/
3835 015311      103      124      123  EM84:   .ASCIZ  /CTS NOT CLRD/
3836 015326      103      124      123  EM85:   .ASCIZ  /CTS NOT SET/
3837 015342      103      101      122  EM86:   .ASCIZ  /CARRIER NOT CLRD/
3838 015363      103      101      122  EM87:   .ASCIZ  /CARRIER NOT SET/
3839 015403      115      117      104  EM88:   .ASCIZ  /MODEM RDY NOT CLRD/
3840 015426      115      117      104  EM89:   .ASCIZ  /MODEM RDY NOT SET/
3841 015450      122      105      103  EM90:   .ASCIZ  /RECEIVER OVERRUN NOT SET/
3842 015501      122      105      103  EM91:   .ASCIZ  /RECEIVER OVERRUN NOT CLEARED/
3843 015536      124      123      117  EM100:  .ASCIZ  /TSO BIT NOT SET/
3844 015556      124      123      117  EM101:  .ASCIZ  /TSO BIT NOT CLEARED/
3845
3846
3847                                   .SBTTL ....TEXT STRINGS FOR ERROR HANDLERS -- "TXT___"
3848                                   ;-------------------------------------------------------------------------------
3849                                   ;------------- TEXT USED BY ERROR HANDLERS ---------------------------------
3850                                   ;-------------------------------------------------------------------------------
3851
3852 015602      102      123      105  TXT1:   .ASCIZ  /BSEL0   BSEL1   BSEL2   BSEL3/
3853 015640      040      040      040  TXT2:   .ASCIZ  /    BSEL4   BSEL5   BSEL6   BSEL7/
3854 015702      102      123      105  TXT2A:  .ASCIZ  /BSEL10  BSEL11  BSEL12  BSEL13/
3855 015741      040      040      040  TXT2B:  .ASCIZ  /    BSEL14  BSEL15  BSEL16  BSEL17/
3856 016004      040      102      131  TXT3:   .ASCIZ  / BYTE SELECT REG'S ARE:/
3857 016034      040      040      040  TXT4:   .ASCIZ  /   SEL0    SEL2    SEL4    SEL6/
3858 016074      040      040      040  TXT4A:  .ASCIZ  /   SEL10   SEL12   SEL14   SEL16/
3859 016135      102      000           TXT5:   .ASCIZ  /B/
3860 016137      040      123      105  TXT6:   .ASCIZ  / SELECT REG'S ARE:/
3861 016162      040      122      105  TXT7:   .ASCIZ  / REGISTERS     ORB  ORA  DDRB DDRA T1CL T1CH T1LL T1LH /
3862 016252      040      040      040  TXT7A:  .ASCIZ  /                T2CL T2CH SR   ACR  PCR  IFR  IER  ORA  /
3863 016342      040      105      130  TXT8:   .ASCIZ  / EXPECTED:     /
3864 016362      040      101      103  TXT9:   .ASCIZ  / ACTUAL:       /
```

....TEXT STRINGS FOR ERROR HANDLERS -- "TXT___"

```
3865 016402  040  130  117  TXT10:  .ASCIZ  / XOR:          /
3866 016422  040  040  116  TXT11:  .ASCIZ  /  N  P  R      R  E  G  I  S  T  E  R  S:/
3867 016474  040  040  040  TXT11A: .ASCIZ  /              CONTROL  D A T A/
3868 016532  040  040  040  TXT11B: .ASCIZ  /                 OUT ADDR.      IN ADDR./
3869 016602  104  105  126  TXT12:  .ASCIZ  /DEVICE CSR ADDRESS : /
3870 016630  125  123  131  TXT13:  .ASCIZ  /USYRT REGS :/
3871 016645  122  104  123  TXT14:  .ASCIZ  /RDSRL   RDSRH   TDSRL   TDSRH/
3872 016703  040  040  040  TXT15:  .ASCIZ  /    PCSARL  PCSARH  PCR     USTAT/
3873 016745  126  111  101  TXT16:  .ASCIZ  /VIA REGS :/
3874 016760  117  122  102  TXT17:  .ASCIZ  /ORB     ORA     DDRB    DDRA/
3875 017015  040  040  040  TXT18:  .ASCIZ  /    T1CL    T1CH    T1LL    T1LH/
3876 017056  124  062  103  TXT19:  .ASCIZ  /T2CL    T2CH    SR      ACR/
3877 017112  040  040  040  TXT20:  .ASCIZ  /    PCR     IFR     IER     ORA/
3878
3879 017152  021  000       TXTNUL: .BYTE   21,0                    ;CTL-Q -- THIS (WE HOPE) IS HARMLESS
3880
3881 017154  116  117  120  TXTML0: .ASCIZ  /NOP/
3882 017160  122  105  101  TXTML1: .ASCIZ  /READ 1 BYTE/
3883 017174  127  122  111  TXTML2: .ASCIZ  /WRITE 1 BYTE/
3884 017211  116  120  122  TXTML3: .ASCIZ  /NPR-OUT 256 BYTES/
3885 017233  116  120  122  TXTML4: .ASCIZ  /NPR-IN 256 BYTES/
3886 017254  123  105  124  TXTML5: .ASCIZ  /SET MICROPROCESSOR'S PC/
3887 017304  125  116  104  TXTML6: .ASCIZ  /UNDEFINED/
3888 017316  101  114  114  TXTML7: .ASCIZ  /ALLOW U-PROCESSOR INTERRUPTS/
3889
3890 017353  126  111  101  TXTVR:  .ASCIZ  /VIA REGISTER /
3891 017371  117  122  102  TXTVR0: .ASCIZ  /ORB/
3892 017375  117  122  101  TXTVR1: .ASCIZ  /ORA/
3893 017401  104  104  122  TXTVR2: .ASCIZ  /DDRB/
3894 017406  104  104  122  TXTVR3: .ASCIZ  /DDRA/
3895 017413  124  061  103  TXTVR4: .ASCIZ  /T1CL/
3896 017420  124  061  103  TXTVR5: .ASCIZ  /T1CH/
3897 017425  124  061  114  TXTVR6: .ASCIZ  /T1LL/
3898 017432  124  061  114  TXTVR7: .ASCIZ  /T1LH/
3899 017437  124  062  103  TXTVR8: .ASCIZ  /T2CL/
3900 017444  124  062  103  TXTVR9: .ASCIZ  /T2CH/
3901 017451  123  122  000  TXTVRA: .ASCIZ  /SR/
3902 017454  101  103  122  TXTVRB: .ASCIZ  /ACR/
3903 017460  120  103  122  TXTVRC: .ASCIZ  /PCR/
3904 017464  111  106  122  TXTVRD: .ASCIZ  /IFR/
3905 017470  111  105  122  TXTVRE: .ASCIZ  /IER/
3906 017474  117  122  101  TXTVRF: .ASCIZ  /ORA/
3907
3908 017500  116  120  122  TXTNP:  .ASCIZ  /NPR /
3909 017505  103  117  116  TXTNP0: .ASCIZ  /CONTROL/
3910 017515  104  101  124  TXTNP1: .ASCIZ  /DATA HI/
3911 017525  104  101  124  TXTNP2: .ASCIZ  /DATA LO/
3912 017535  101  104  104  TXTNP3: .ASCIZ  /ADDR. OUT EX/
3913 017552  101  104  104  TXTNP4: .ASCIZ  /ADDR. OUT HI/
3914 017567  101  104  104  TXTNP5: .ASCIZ  /ADDR. OUT LO/
3915 017604  101  104  104  TXTNP6: .ASCIZ  /ADDR. IN EX/
3916 017620  101  104  104  TXTNP7: .ASCIZ  /ADDR. IN HI/
3917 017634  101  104  104  TXTNP8: .ASCIZ  /ADDR. IN LO/
3918
3919 017650  125  123  131  TXTUR:  .ASCIZ  /USYRT REG /
3920 017663  122  104  123  TXTUR0: .ASCIZ  /RDSRL/
3921 017671  122  104  123  TXTUR1: .ASCIZ  /RDSRH/
```

```
....TEXT STRINGS FOR ERROR HANDLERS -- "TXT___"

3922 017677     124     104     123  TXTUR2: .ASCIZ  /TDSRL/
3923 017705     124     104     123  TXTUR3: .ASCIZ  /TDSRH/
3924 017713     120     103     123  TXTUR4: .ASCIZ  /PCSARL/
3925 017722     120     103     123  TXTUR5: .ASCIZ  /PCSARH/
3926 017731     120     103     122  TXTUR6: .ASCIZ  /PCR/
3927 017735     125     123     124  TXTUR7: .ASCIZ  /USTAT/
3928                                  .LIST   BEX
3929                                  .EVEN
3930
3931
3932                                  .SBTTL ....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- "TXT__T"
3933                                  ;-----------------------------------------------------------------
3934                                  ;------------ TEXT ADDRESS TABLES USED BY ERROR HANDLERS ---------
3935                                  ;-----------------------------------------------------------------
3936
3937 017744  017154  017160  017174  TXTMLT: .WORD   TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7
     017752  017211  017233  017254
     017760  017304  017316
3938
3939 017764  017353                          .WORD   .XTVR
3940 017766  017371  017375  017401  TXTVRT: .WORD   TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7
     017774  017406  017413  017420
     020002  017425  017432
3941 020006  017437  017444  017451          .WORD   TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF
     020014  017454  017460  017464
     020022  017470  017474
3942
3943 020026  017500                          .WORD   TXTNP
3944 020030  017505  017515  017525  TXTNPT: .WORD   TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8
     020036  017535  017552  017567
     020044  017604  017620  017634
3945 020052  017663  017671  017677  TXTURT: .WORD   TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7
     020060  017705  017713  017722
     020066  017731  017735
3946
3947
3948
```

....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- "TXT__T"

```
3950                                    ;-------------------------------------------------------------------------------
3951                                    .SBTTL ....ERROR HANDLER  - ERR3 -- DUMP THE BYTE SELECT REGISTERS
3952                                    ;-------------------------------------------------------------------------------
3953 020072                                     BGNMSG  ERR3
     020072                                                                                ERR3::
3954 020072  004737  021306                     JSR     PC,ERR4$
3955 020076                                     PRINTB  #ENDEMB
     020076  012746  012116                                                                     MOV     #ENDEMB,-(SP)
     020102  012746  000001                                                                     MOV     #1,-(SP)
     020106  010600                                                                             MOV     SP,R0
     020110  104414                                                                             TRAP    C$PNTB
     020112  062706  000004                                                                     ADD     #4,SP
3956 020116                                     ENDMSG
     020116                                                                                L10002:
     020116  104423                                                                             TRAP    C$MSG
3957                                    ;-------------------------------------------------------------------------------
3958                                    .SBTTL ....ERROR HANDLER -- ERR4 -- M-LOOP TIMEOUT ERROR HANDLING
3959                                    ;-------------------------------------------------------------------------------
3960 020120                                     BGNMSG  ERR4
     020120                                                                                ERR4::
3961 020120  105037  002325                     CLRB    GDATA+1                 ;MAKE SURE BIT 8 DOESN'T PRINT
3962 020124  010146                             MOV     R1,-(SP)                ;SAVE THE WORKING REGISTER
3963 020126  013701  002324                     MOV     GDATA,R1                ;SAVE THIS FOR LATER
3964 020132  022701  000017                     CMP     #17,R1                  ;WAS THIS AN M-LOOP REQUEST?
3965 020136  002012                             BGE     5$                      ;YES, THEN REPORT THE FUNCTION CODE
3966 020140                                     PRINTX  #FMT5,R1                ;NO,  THEN IT MUST BE A BSEL1 SETTING
     020140  010146                                                                             MOV     R1,-(SP)
     020142  012746  012400                                                                     MOV     #FMT5,(SP)
     020146  012746  000002                                                                     MOV     #2,-(SP)
     020152  010600                                                                             MOV     SP,R0
     020154  104415                                                                             TRAP    C$PNTX
     020156  062706  000006                                                                     ADD     #6,SP
3967 020162  000424                             BR      20$
3968
3969 020164  001001                     5$:     BNE     6$                      ;IF IT WAS A 17, THIS IS A "NOP" AND
3970 020166  005001                             CLR     R1                      ;  THE TEXT POINTER MUST SO REFLECT.
3971 020170  022701  000007             6$:     CMP     #7,R1                   ;IS FUNCTION CODE > 7?
3972 020174  002002                             BGE     7$                      ;NO,  THEN WE CAN HANDLE IT
3973 020176  012701  000006                     MOV     #6,R1                   ;YES, THEN IT'S UNDEFINED -- SAY SO
3974 020202  006301                     7$:     ASL     R1                      ;CONVERT TO A WORD OFFSET
3975 020204                                     PRINTX  #FMT5A,GDATA,TXTMLT(R1) ;REPORT THE FAILING FUNCTION
     020204  016146  017744                                                                     MOV     TXTMLT(R1),-(SP)
     020210  013746  002324                                                                     MOV     GDATA,(SP)
     020214  012746  012443                                                                     MOV     #FMT5A,-(SP)
     020220  012746  000003                                                                     MOV     #3,-(SP)
     020224  010600                                                                             MOV     SP,R0
     020226  104415                                                                             TRAP    C$PNTX
     020230  062706  000010                                                                     ADD     #10,SP
3976
3977 020234  012601                     20$:    MOV     (SP)+,R1                ;RESTORE THE WORKING REGISTER
3978 020236  004737  021634                     JSR     PC,ERR5$                ;DUMP THE SELECT REGISTERS
3979 020242                                     ENDMSG
     020242                                                                                L10003:
     020242  104423                                                                             TRAP    C$MSG
3980                                    ;-------------------------------------------------------------------------------
3981                                    .SBTTL ....ERROR HANDLER  - ERR7A -- USART REGISTER ERRORS
3982                                    ;-------------------------------------------------------------------------------
```

....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS

```
3983 020244                            BGNMSG  ERR7A
     020244                                                                ERR7A::
3984 020244  113701  002336            MOVB    REGNUM,R1
3985 020250  006301                    ASL     R1              ;AS PASSED, THIS WAS A BYTE OFFSET
3986 020252                            PRINTB  #FMT15,#TXTUR,TXTURT(R1)
     020252  016146  020052                                                 MOV     TXTURT(R1),-(SP)
     020256  012746  017650                                                 MOV     #TXTUR,-(SP)
     020262  012746  012766                                                 MOV     #FMT15,-(SP)
     020266  012746  000003                                                 MOV     #3,-(SP)
     020272  010600                                                        MOV     SP,R0
     020274  104414                                                        TRAP    C$PNTB
     020276  062706  000010                                                ADD     #10,SP
3987 020302  004737  021262            JSR     PC,XORGB
3988 020306                            PRINTB  #FMT3,GDATA,BDATA,XDATA
     020306  013746  002330                                                 MOV     XDATA,-(SP)
     020312  013746  002326                                                 MOV     BDATA,-(SP)
     020316  013746  002324                                                 MOV     GDATA,-(SP)
     020322  012746  012163                                                 MOV     #FMT3,(SP)
     020326  012746  000004                                                 MOV     #4,-(SP)
     020332  010600                                                        MOV     SP,R0
     020334  104414                                                        TRAP    C$PNTB
     020336  062706  000012                                                ADD     #12,SP
3989 020342                            PRINTB  #ENDEMB
     020342  012746  012116                                                 MOV     #ENDEMB,-(SP)
     020346  012746  000001                                                 MOV     #1,-(SP)
     020352  010600                                                        MOV     SP,R0
     020354  104414                                                        TRAP    C$PNTB
     020356  062706  000004                                                ADD     #4,SP
3990 020362                            ENDMSG
     020362                                                                1.10004:
     020362  104423                                                        TRAP    C$MSG
3991                                   ;-------------------------------------------------------------------------
3992                                   .SBTTL ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)
3993                                   ;-------------------------------------------------------------------------
3994
3995 020364                            BGNMSG  ERR10
     020364                                                                ERR10::
3996 020364                            PRINTB  #FMT21,#TXT12,MPCSR
     020364  013746  002416                                                 MOV     MPCSR,-(SP)
     020370  012746  016602                                                 MOV     #TXT12,-(SP)
     020374  012746  013503                                                 MOV     #FMT21,-(SP)
     020400  012746  000003                                                 MOV     #3,-(SP)
     020404  010600                                                        MOV     SP,R0
     020406  104414                                                        TRAP    C$PNTB
     020410  062706  000010                                                ADD     #10,SP
3997 020414                            PRINTB  #FMT22
     020414  012746  013513                                                 MOV     #FMT22,(SP)
     020420  012746  000001                                                 MOV     #1,-(SP)
     020424  010600                                                        MOV     SP,R0
     020425  104414                                                        TRAP    C$PNTB
     020430  062706  000004                                                ADD     #4,SP
3998 020434  013701  002336            MOV     REGNUM,R1
3999 020440  006301                    ASL     R1              ;GET PTR TO USYRT REG ASCII
4000 020442                            PRINTB  #FMT27,#TXTUR,TXTURT(R1)
     020442  016146  020052                                                 MOV     TXTURT(R1),(SP)
     020446  012746  017650                                                 MOV     #TXTUR,-(SP)
     020452  012746  013712                                                 MOV     #FMT27,-(SP)
```

```
        020456  012746  000003                                                  MOV     #3,-(SP)
        020462  010600                                                          MOV     SP,R0
        020464  104414                                                          TRAP    C$PNTB
        020466  062706  000010                                                  ADD     #10,SP
   4001 020472  004737  021262          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
   4002 020476                          PRINTB  #FMT23,GDATA,BDATA,XDATA
        020476  013746  002330                                                  MOV     XDATA,-(SP)
        020502  013746  002326                                                  MOV     BDATA,-(SP)
        020506  013746  002324                                                  MOV     GDATA,-(SP)
        020512  012746  013535                                                  MOV     #FMT23,-(SP)
        020516  012746  000004                                                  MOV     #4,-(SP)
        020522  010600                                                          MOV     SP,R0
        020524  104414                                                          TRAP    C$PNTB
        020526  062706  000012                                                  ADD     #12,SP
   4003 020532  004737  022364          JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
   4004 020536                          ENDMSG
        020536                                                          L10005:
        020536  104423                                                         TRAP    C$MSG
   4005
   4006                         ;-------------------------------------------------------------------------
   4007                         .SBTTL ....ERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT)
   4008                         ;-------------------------------------------------------------------------
   4009
   4010 020540                          BGNMSG  ERR11
        020540                                                          ERR11::
   4011 020540                          PRINTB  #FMT21,#TXT12,MPCSR
        020540  013746  002416                                                  MOV     MPCSR,-(SP)
        020544  012746  016602                                                  MOV     #TXT12,-(SP)
        020550  012746  013503                                                  MOV     #FMT21,-(SP)
        020554  012746  000003                                                  MOV     #3,-(SP)
        020560  010600                                                          MOV     SP,R0
        020562  104414                                                          TRAP    C$PNTB
        020564  062706  000010                                                  ADD     #10,SP
   4012 020570                          PRINTB  #FMT22
        020570  012746  013513                                                  MOV     #FMT22,-(SP)
        020574  012746  000001                                                  MOV     #1,-(SP)
        020600  010600                                                          MOV     SP,R0
        020602  104414                                                          TRAP    C$PNTB
        020604  062706  000004                                                  ADD     #4,SP
   4013 020610  013701  002336          MOV     REGNUM,R1
   4014 020614  006301                  ASL     R1                      ;GET PTR TO VIA REG ASCII
   4015 020616                          PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
        020616  016146  017766                                                  MOV     TXTVRT(R1),-(SP)
        020622  012746  017353                                                  MOV     #TXTVR,-(SP)
        020626  012746  013712                                                  MOV     #FMT27,-(SP)
        020632  012746  000003                                                  MOV     #3,-(SP)
        020636  010600                                                          MOV     SP,R0
        020640  104414                                                          TRAP    C$PNTB
        020642  062706  000010                                                  ADD     #10,SP
   4016 020646  004737  021262          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
   4017 020652                          PRINTB  #FMT23,GDATA,BDATA,XDATA
        020652  013746  002330                                                  MOV     XDATA,-(SP)
        020656  013746  002326                                                  MOV     BDATA,-(SP)
        020662  013746  002324                                                  MOV     GDATA,-(SP)
        020666  012746  013535                                                  MOV     #FMT23,-(SP)
        020672  012746  000004                                                  MOV     #4,(SP)
        020676  010600                                                          MOV     SP,R0
```

....ERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT

```
      020700  104414                                                          TRAP    C$PNTB
      020702  062706  000012                                                  ADD     #12,SP
4018  020706  004737  022032          JSR     PC,ERR11$               ;GET & PRINT VIA REGISTERS
4019  020712                         ENDMSG
      020712                                                        L10006:
      020712  104423                                                          TRAP    C$MSG
4020
4021
4022                          ;-------------------------------------------------------------------------
4023                          .SBTTL ....ERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)
4024                          ;-------------------------------------------------------------------------
4025  020714                  BGNMSG  ERR12
      020714                                                        ERR12::
4026  020714                  PRINTB  #FMT21,#TXT12,MPCSR
      020714  013746  002416                                                  MOV     MPCSR,-(SP)
      020720  012746  016602                                                  MOV     #TXT12,-(SP)
      020724  012746  013503                                                  MOV     #FMT21,-(SP)
      020730  012746  000003                                                  MOV     #3,-(SP)
      020734  010600                                                          MOV     SP,R0
      020736  104414                                                          TRAP    C$PNTB
      020740  062706  000010                                                  ADD     #10,SP
4027  020744                  PRINTB  #FMT22
      020744  012746  013513                                                  MOV     #FMT22,-(SP)
      020750  012746  000001                                                  MOV     #1,-(SP)
      020754  010600                                                          MOV     SP,R0
      020756  104414                                                          TRAP    C$PNTB
      020760  062706  000004                                                  ADD     #4,SP
4028  020764  013701  002336          MOV     REGNUM,R1
4029  020770  006301                  ASL     R1                      ;GET PTR TO USYRT REG ASCII
4030  020772                  PRINTB  #FMT27,#TXTUR,TXTURT(R1)
      020772  016146  020052                                                  MOV     TXTURT(R1),-(SP)
      020776  012746  017650                                                  MOV     #TXTUR,-(SP)
      021002  012746  013712                                                  MOV     #FMT27,-(SP)
      021006  012746  000003                                                  MOV     #3,-(SP)
      021012  010600                                                          MOV     SP,R0
      021014  104414                                                          TRAP    C$PNTB
      021016  062706  000010                                                  ADD     #10,SP
4031  021022  004737  022364          JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
4032  021026                         ENDMSG
      021026                                                        L10007:
      021026  104423                                                          TRAP    C$MSG
4033
4034
4035                          ;-------------------------------------------------------------------------
4036                          .SBTTL ....ERROR HANDLER -- ERR13 -- RAM ADDRESS ERRORS
4037                          ;-------------------------------------------------------------------------
4038  021030                  BGNMSG  ERR13
      021030                                                        ERR13::
4039  021030                  PRINTB  #FMT21,#TXT12,MPCSR
      021030  013746  002416                                                  MOV     MPCSR,-(SP)
      021034  012746  016602                                                  MOV     #TXT12,-(SP)
      021040  012746  013503                                                  MOV     #FMT21,-(SP)
      021044  012746  000003                                                  MOV     #3,(SP)
      021050  010600                                                          MOV     SP,R0
      021052  104414                                                          TRAP    C$PNTB
      021054  062706  000010                                                  ADD     #10,SP
4040  021060                  PRINTB  #FMT40,REGNUM
```

```
....ERROR HANDLER -- ERR13 -- RAM ADDRESS ERRORS

        021060  013746  002336                                                  MOV     REGNUM,-(SP)
        021064  012746  014150                                                  MOV     #FMT40,-(SP)
        021070  012746  000002                                                  MOV     #2,-(SP)
        021074  010600                                                          MOV     SP,R0
        021076  104414                                                          TRAP    C$PNTB
        021100  062706  000006                                                  ADD     #6,SP
4041    021104  004737  021262          JSR     PC,XORGB        ;COMPUTE XOR OF GOOD AND BAD DATA
4042    021110                           PRINTB  #FMT23,GDATA,BDATA,XDATA
        021110  013746  002330                                                  MOV     XDATA,-(SP)
        021114  013746  002326                                                  MOV     BDATA,-(SP)
        021120  013746  002324                                                  MOV     GDATA,-(SP)
        021124  012746  013535                                                  MOV     #FMT23,-(SP)
        021130  012746  000004                                                  MOV     #4,-(SP)
        021134  010600                                                          MOV     SP,R0
        021136  104414                                                          TRAP    C$PNTB
        021140  062706  000012                                                  ADD     #12.SP
4043    021144                           ENDMSG
        021144                                                          L10010:
        021144  104423                                                         TRAP    C$MSG
4044
4045
4046                            ;-------------------------------------------------------------------------
4047                            .SBTTL ....ERROR HANDLER -- ERR14 -- VIA REG ERRORS (VIA PRINTOUT)
4048                            ;-------------------------------------------------------------------------
4049    021146                           BGNMSG  ERR14
        021146                                                          ERR14::
4050    021146                           PRINTB  #FMT21,#TXT12,MPCSR
        021146  013746  002416                                                  MOV     MPCSR,-(SP)
        021152  012746  016602                                                  MOV     #TXT12,-(SP)
        021156  012746  013503                                                  MOV     #FMT21,-(SP)
        021162  012746  000003                                                  MOV     #3,-(SP)
        021166  010600                                                          MOV     SP,R0
        021170  104414                                                          TRAP    C$PNTB
        021172  062706  000010                                                  ADD     #10,SP
4051    021176                           PRINTB  #FMT22
        021176  012746  013513                                                  MOV     #FMT22,-(SP)
        021202  012746  000001                                                  MOV     #1,-(SP)
        021206  010600                                                          MOV     SP,R0
        021210  104414                                                          TRAP    C$PNTB
        021212  062706  000004                                                  ADD     #4,SP
4052    021216  013701  002336          MOV     REGNUM,R1
4053    021222  006301                  ASL     R1                      ;GET PTR TO VIA REG ASCII
4054    021224                           PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
        021224  016146  017766                                                  MOV     TXTVRT(R1),-(SP)
        021230  012746  017353                                                  MOV     #TXTVR,-(SP)
        021234  012746  013712                                                  MOV     #FMT27,-(SP)
        021240  012746  000003                                                  MOV     #3,-(SP)
        021244  010600                                                          MOV     SP,R0
        021246  104414                                                          TRAP    C$PNTB
        021250  062706  000010                                                  ADD     #10,SP
4055    021254  004737  022032          JSR     PC,ERR11$               ;GET & PRINT VIA REGISTERS
4056    021260                           ENDMSG
        021260                                                          L10011:
        021260  104423                                                         TRAP    C$MSG
4057
4058
4059
```

....ERROR HANDLER -- ERR14 -- VIA REG ERRORS (VIA PRINTOUT)

```
                                        .SBTTL ....ERROR HANDLER SUBROUTINES
                                        ;-------------------------------------------------------------------------
                                        ;------------ SUBROUTINES USED ONLY BY ERROR HANDLERS ---------------------
                                        ;-------------------------------------------------------------------------

                                        .SBTTL ........ERROR HANDLER SUBROUTINE -- XORGB

                                        ;       PERFORM EXCLUSIVE OR BETWEEN "GDATA" & "BDATA" PUTTING
                                        ;       THE RESULT IN "XDATA"

021262  010146                  XORGB:  MOV     R1,-(SP)        ;PRESERVE WORKING REGISTER
021264  013701  002324                  MOV     GDATA,R1        ;GET "GOOD" DATA
021270  013737  002326  002330          MOV     BDATA,XDATA     ;AND "BAD" DATA
021276  074137  002330                  XOR     R1,XDATA        ;PERFORM EXCLUSIVE OR
021302  012601                          MOV     (SP)+,R1        ;RESTORE R1
021304  000207                          RTS     PC              ;RETURN

                                        ;-------------------------------------------------------------------------
                                        .SBTTL ........ERROR HANDLER SUBROUTINE -- ERR4$
                                        ;-------------------------------------------------------------------------
                                        ;       IDENTIFY & DUMP THE BYTE SELECT REGISTERS

021306                          ERR4$:  PRINTX  #FMT4,#TXT3,#TXT1
021306  012746  015602                                                  MOV     #TXT1,-(SP)
021312  012746  016004                                                  MOV     #TXT3,-(SP)
021316  012746  012247                                                  MOV     #FMT4,-(SP)
021322  012746  000003                                                  MOV     #3,-(SP)
021326  010600                                                          MOV     SP,R0
021330  104415                                                          TRAP    C$PNTX
021332  062706  000010                                                  ADD     #10,SP
021336                          PRINTX  #FMT4A,BSR0,BSR1,BSR2,BSR3
021336  013746  002210                                                  MOV     BSR3,-(SP)
021342  013746  002206                                                  MOV     BSR2,-(SP)
021346  013746  002204                                                  MOV     BSR1,-(SP)
021352  013746  002202                                                  MOV     BSR0,-(SP)
021356  012746  012305                                                  MOV     #FMT4A,-(SP)
021362  012746  000005                                                  MOV     #5,-(SP)
021366  010600                                                          MOV     SP,R0
021370  104415                                                          TRAP    C$PNTX
021372  062706  000014                                                  ADD     #14,SP
021376                          PRINTX  #FMT4B,#TXT2
021376  012746  015640                                                  MOV     #TXT2,-(SP)
021402  012746  012340                                                  MOV     #FMT4B,-(SP)
021406  012746  000002                                                  MOV     #2,-(SP)
021412  010600                                                          MOV     SP,R0
021414  104415                                                          TRAP    C$PNTX
021416  062706  000006                                                  ADD     #6,SP
021422                          PRINTX  #FMT4C,BSR4,BSR5,BSR6,BSR7
021422  013746  002220                                                  MOV     BSR7,-(SP)
021426  013746  002216                                                  MOV     BSR6,-(SP)
021432  013746  002214                                                  MOV     BSR5,-(SP)
021436  013746  002212                                                  MOV     BSR4,-(SP)
021442  012746  012345                                                  MOV     #FMT4C,-(SP)
021446  012746  000005                                                  MOV     #5,(SP)
021452  010600                                                          MOV     SP,R0
```

```
.....ERROR HANDLER SUBROUTINE -- ERR4$

      021454  104415                                                    TRAP    C$PNTX
      021456  062706  000014                                            ADD     #14,SP
4088  021462                        PRINTX  #FMT4B,#TXT2A
      021462  012746  015702                                            MOV     #TXT2A,-(SP)
      021466  012746  012340                                            MOV     #FMT4B,-(SP)
      021472  012746  000002                                            MOV     #2,-(SP)
      021476  010600                                                    MOV     SP,R0
      021500  104415                                                    TRAP    C$PNTX
      021502  062706  000006                                            ADD     #6,SP
4089  021506                        PRINTX  #FMT4A,BSR10,BSR11,BSR12,BSR13
      021506  013746  002230                                            MOV     BSR13,-(SP)
      021512  013746  002226                                            MOV     BSR12,-(SP)
      021516  013746  002224                                            MOV     BSR11,-(SP)
      021522  013746  002222                                            MOV     BSR10,-(SP)
      021526  012746  012305                                            MOV     #FMT4A,-(SP)
      021532  012746  000005                                            MOV     #5,-(SP)
      021536  010600                                                    MOV     SP,R0
      021540  104415                                                    TRAP    C$PNTX
      021542  062706  000014                                            ADD     #14,SP
4090  021546                        PRINTX  #FMT4B,#TXT2B
      021546  012746  015741                                            MOV     #TXT2B,-(SP)
      021552  012746  012340                                            MOV     #FMT4B -(SP)
      021556  012746  000002                                            MOV     #2,-(SP)
      021562  010600                                                    MOV     SP,R0
      021564  104415                                                    TRAP    C$PNTX
      021566  062706  000006                                            ADD     #6,SP
4091  021572                        PRINTX  #FMT4C,BSR14,BSR15,BSR16,BSR17
      021572  013746  002240                                            MOV     BSR17,-(SP)
      021576  013746  002236                                            MOV     BSR16,-(SP)
      021602  013746  002234                                            MOV     BSR15,-(SP)
      021606  013746  002232                                            MOV     BSR14,-(SP)
      021612  012746  012345                                            MOV     #FMT4C,-(SP)
      021616  012746  000005                                            MOV     #5,-(SP)
      021622  010600                                                    MOV     SP,R0
      021624  104415                                                    TRAP    C$PNTX
      021626  062706  000014                                            ADD     #14,SP
4092  021632  000207                RTS     PC
4093
4094                          ;-------------------------------------------------------------------
4095                          .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR5$
4096                          ;-------------------------------------------------------------------
4097                          ;       COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS
4098  021634                  ERR5$:
4099  021634                        PRINTX  #FMT4,#TXT6,#TXT4
      021634  012746  016034                                            MOV     #TXT4,-(SP)
      021640  012746  016137                                            MOV     #TXT6,-(SP)
      021644  012746  012247                                            MOV     #FMT4,-(SP)
      021650  012746  000003                                            MOV     #3,-(SP)
      021654  010600                                                    MOV     SP,R0
      021656  104415                                                    TRAP    C$PNTX
      021660  062706  000010                                            ADD     #10,SP
4100  021664                        PRINTX  #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS
      021664  013746  002210                                            MOV     WSR6,-(SP)
      021670  013746  002206                                            MOV     WSR4,-(SP)
      021674  013746  002204                                            MOV     WSR2,-(SP)
      021700  013746  002202                                            MOV     WSR0,-(SP)
      021704  012746  012655                                            MOV     #FMT11,-(SP)
```

```
.....ERROR HANDLER SUBROUTINE -- ERR5$

        021710  012746  000005                                                  MOV     #5,-(SP)
        021714  010600                                                          MOV     SP,R0
        021716  104415                                                          TRAP    C$PNTX
        021720  062706  000014                                                  ADD     #14,SP
   4101 021724                          PRINTX  #FMT4B,#TXT4A
        021724  012746  016074                                                  MOV     #TXT4A,-(SP)
        021730  012746  012340                                                  MOV     #FMT4B,-(SP)
        021734  012746  000002                                                  MOV     #2,-(SP)
        021740  010600                                                          MOV     SP,R0
        021742  104415                                                          TRAP    C$PNTX
        021744  062706  000006                                                  ADD     #6,SP
   4102 021750                          PRINTX  #FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS
        021750  013746  002220                                                  MOV     WSR16,-(SP)
        021754  013746  002216                                                  MOV     WSR14,-(SP)
        021760  013746  002214                                                  MOV     WSR12,-(SP)
        021764  013746  002212                                                  MOV     WSR10,-(SP)
        021770  012746  012655                                                  MOV     #FMT11,-(SP)
        021774  012746  000005                                                  MOV     #5,-(SP)
        022000  010600                                                          MOV     SP,R0
        022002  104415                                                          TRAP    C$PNTX
        022004  062706  000014                                                  ADD     #14,SP
   4103 022010                          PRINTB  #ENDEMB
        022010  012746  012116                                                  MOV     #ENDEMB,-(SP)
        022014  012746  000001                                                  MOV     #1,-(SP)
        022020  010600                                                          MOV     SP,R0
        022022  104414                                                          TRAP    C$PNTB
        022024  062706  000004                                                  ADD     #4,SP
   4104 022030  000207                  RTS     PC
   4105
   4106                                 ;-------------------------------------------------------------------------
   4107                                 .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR11$
   4108                                 ;-------------------------------------------------------------------------
   4109                                 ;       COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS
   4110
   4111 022032  004737  004460  ERR11$: JSR     PC,GETVRS               ;GET VIA REGS FOR PRINTOUT
   4112 022036                          PRINTX  #FMT24,#TXT16,#TXT17
        022036  012746  016760                                                  MOV     #TXT17,-(SP)
        022042  012746  016745                                                  MOV     #TXT16,-(SP)
        022046  012746  013614                                                  MOV     #FMT24,-(SP)
        022052  012746  000003                                                  MOV     #3,-(SP)
        022056  010600                                                          MOV     SP,R0
        022060  104415                                                          TRAP    C$PNTX
        022062  062706  000010                                                  ADD     #10,SP
   4113 022066                          PRINTX  #FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6
        022066  013746  002270                                                  MOV     VREGS+6,-(SP)
        022072  013746  002266                                                  MOV     VREGS+4,-(SP)
        022076  013746  002264                                                  MOV     VREGS+2,-(SP)
        022102  013746  002262                                                  MOV     VREGS+0,-(SP)
        022106  012746  013627                                                  MOV     #FMT25,-(SP)
        022112  012746  000005                                                  MOV     #5,-(SP)
        022116  010600                                                          MOV     SP,R0
        022120  104415                                                          TRAP    C$PNTX
        022122  062706  000014                                                  ADD     #14,SP
   4114 022126                          PRINTX  #FMT29  #TXT18
        022126  012746  017015                                                  MOV     #TXT18,-(SP)
        022132  012746  013755                                                  MOV     #FMT29,-(SP)
        022136  012746  000002                                                  MOV     #2,-(SP)
```

```
.....ERROR HANDLER SUBROUTINE -- ERR11$

        022142  010600                                                  MOV     SP,R0
        022144  104415                                                  TRAP    C$PNTX
        022146  062706  000006                                         ADD     #6,SP
   4115 022152                  PRINTX  #FMT26,VREGS+8.,VREGS+10.,VREGS+12.,VREGS+14.
        022152  013746  002300                                         MOV     VREGS+14.,-(SP)
        022156  013746  002276                                         MOV     VREGS+12.,-(SP)
        022162  013746  002274                                         MOV     VREGS+10.,-(SP)
        022166  013746  002272                                         MOV     VREGS+8.,-(SP)
        022172  012746  013657                                         MOV     #FMT26,-(SP)
        022176  012746  000005                                         MOV     #5,-(SP)
        022202  010600                                                  MOV     SP,R0
        022204  104415                                                  TRAP    C$PNTX
        022206  062706  000014                                         ADD     #14,SP
   4116 022212                  PRINTX  #FMT29,#TXT19
        022212  012746  017056                                         MOV     #TXT19,-(SP)
        022216  012746  013755                                         MOV     #FMT29,-(SP)
        022222  012746  000002                                         MOV     #2,-(SP)
        022226  010600                                                  MOV     SP,R0
        022230  104415                                                  TRAP    C$PNTX
        022232  062706  000006                                         ADD     #6,SP
   4117 022236                  PRINTX  #FMT25,VREGS+16.,VREGS+18.,VREGS+20.,VREGS+22.
        022236  013746  002310                                         MOV     VREGS+22.,-(SP)
        022242  013746  002306                                         MOV     VREGS+20.,-(SP)
        022246  013746  002304                                         MOV     VREGS+18.,-(SP)
        022252  013746  002302                                         MOV     VREGS+16.,-(SP)
        022256  012746  013627                                         MOV     #FMT25,-(SP)
        022262  012746  000005                                         MOV     #5,-(SP)
        022266  010600                                                  MOV     SP,R0
        022270  104415                                                  TRAP    C$PNTX
        022272  062706  000014                                         ADD     #14,SP
   4118 022276                  PRINTX  #FMT29,#TXT20
        022276  012746  017112                                         MOV     #TXT20,-(SP)
        022302  012746  013755                                         MOV     #FMT29,-(SP)
        022306  012746  000002                                         MOV     #2,-(SP)
        022312  010600                                                  MOV     SP,R0
        022314  104415                                                  TRAP    C$PNTX
        022316  062706  000006                                         ADD     #6,SP
   4119 022322                  PRINTX  #FMT26,VREGS+24.,VREGS+26.,VREGS+28.,VREGS+30.
        022322  013746  002320                                         MOV     VREGS+30.,-(SP)
        022326  013746  002316                                         MOV     VREGS+28.,-(SP)
        022332  013746  002314                                         MOV     VREGS+26.,-(SP)
        022336  013746  002312                                         MOV     VREGS+24.,-(SP)
        022342  012746  013657                                         MOV     #FMT26,-(SP)
        022346  012746  000005                                         MOV     #5,-(SP)
        022352  010600                                                  MOV     SP,R0
        022354  104415                                                  TRAP    C$PNTX
        022356  062706  000014                                         ADD     #14,SP
   4120 022362  000207                  RTS     PC
   4121
   4122                         ;-------------------------------------------------------
   4123                         .SBTTL .....ERROR HANDLER SUBROUTINE - ERR12$
   4124                         ;-------------------------------------------------------
   4125                         ;       COMMON ERROR ROUTINE TO GET AND PRINTOUT USYRT REGISTERS
   4126
   4127 022364  004737  004360 ERR12$: JSR     PC,GETURS               ;GET USYRT REGS FOR PRINTOUT
   4128 022370                  PRINTX  #FMT24,#TXT13,#TXT14
        022370  012746  016645                                         MOV     #TXT14,-(SP)
```

```
.....ERROR HANDLER SUBROUTINE -- ERR12$

        022374  012746  016630                                          MOV     #TXT13,-(SP)
        022400  012746  013614                                          MOV     #FMT24,-(SP)
        022404  012746  000003                                          MOV     #3,-(SP)
        022410  010600                                                  MOV     SP,R0
        022412  104415                                                  TRAP    C$PNTX
        022414  062706  000010                                          ADD     #10,SP
  4129  022420                          PRINTX  #FMT25,UREGS+0,UREGS+2,UREGS+4,UREGS+6
        022420  013746  002250                                          MOV     UREGS+6,-(SP)
        022424  013746  002246                                          MOV     UREGS+4,-(SP)
        022430  013746  002244                                          MOV     UREGS+2,-(SP)
        022434  013746  002242                                          MOV     UREGS+0,-(SP)
        022440  012746  013627                                          MOV     #FMT25,-(SP)
        022444  012746  000005                                          MOV     #5,-(SP)
        022450  010600                                                  MOV     SP,R0
        022452  104415                                                  TRAP    C$PNTX
        022454  062706  000014                                          ADD     #14,SP
  4130  022460                          PRINTX  #FMT29,#TXT15
        022460  012746  016703                                          MOV     #TXT15,-(SP)
        022464  012746  013755                                          MOV     #FMT29,-(SP)
        022470  012746  000002                                          MOV     #2,-(SP)
        022474  010600                                                  MOV     SP,R0
        022476  104415                                                  TRAP    C$PNTX
        022500  062706  000006                                          ADD     #6,SP
  4131  022504                          PRINTX  #FMT26,UREGS+10,UREGS+12,UREGS+14,UREGS+16
        022504  013746  002260                                          MOV     UREGS+16,-(SP)
        022510  013746  002256                                          MOV     UREGS+14,-(SP)
        022514  013746  002254                                          MOV     UREGS+12,-(SP)
        022520  013746  002252                                          MOV     UREGS+10,-(SP)
        022524  012746  013657                                          MOV     #FMT26,-(SP)
        022530  012746  000005                                          MOV     #5,-(SP)
        022534  010600                                                  MOV     SP,R0
        022536  104415                                                  TRAP    C$PNTX
        022540  062706  000014                                          ADD     #14,SP
  4132  022544  000207                          RTS     PC
  4133
  4134                                  .EVEN
```

LOAD DEVICE PROTECTION TABLE

```
4136                                    .SBTTL  LOAD DEVICE PROTECTION TABLE
4137
4138                                    ;////////////////////////////////////////////////////////////////////////////
4139                                    ;/ THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
4140                                    ;/ PROTECTED FROM TESTING, IF DESIRED.
4141                                    ;////////////////////////////////////////////////////////////////////////////
4142
4143 022546                                    BGNPROT
     022546                                                                                    L$PROT::
4144 022546  177777                            .WORD   -1      ;DON'T CHK CSR ADRS
4145 022550  177777                            .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
4146 022552  177777                            .WORD   -1      ;DON'T CHK DRIVE NO.
4147 022554                                    ENDPROT
```

INITIALIZE SECTION

```
                        .SBTTL  INITIALIZE SECTION

                        ;////////////////////////////////////////////////////////////////////////
                        ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                        ;/ AT THE BEGINNING OF THE TEST SEQUENCE ON THE NEXT UNIT.
                        ;////////////////////////////////////////////////////////////////////////

                                BGNINIT
                                                                                L$INIT::
                                SETVEC  #140,#170000,#340       ;ODT ROM ADDRESS                ;JB REV A-0
022554  012746  000340                                                          MOV     #340,-(SP)
022560  012746  170000                                                          MOV     #170000,-(SP)
022564  012746  000140                                                          MOV     #140,-(SP)
022570  012746  000003                                                          MOV     #3,-(SP)
022574  104437                                                                  TRAP    C$SVEC
022576  062706  000010                                                          ADD     #10,SP

022602  010637  002340                MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
022606  005037  002344                CLR     SUBRPC          ;CLEAR SUBR CALL PC
022612  005037  002400                CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
022616  005037  002376                CLR     ERROR1          ;CLEAR ERROR FLAG
022622  005037  002402                CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
022626  005737  002370                TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
022632  001007                      BNE     6$              ;BR IF NOT
022634  013737  000004  002372      MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
022642  013737  000006  002374      MOV     @#6,SAVE6
022650  000406                      BR      9$

022652  013737  002372  000004  6$: MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
022660  013737  002374  000006      MOV     SAVE6,@#6

022666  012737  000001  002370  9$: MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH

                        ;SEE IF PROGRAM JUST STARTED, BR IF YES
                                READEF  #EF.START
022674  012700  000040                                                          MOV     #EF.START,R0
022700  104447                                                                  TRAP    C$REFG
                                BCOMPLETE       STARST
022702  103415                                                                  BCS     STARST

                        ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
                                READEF  #EF.RESTART
022704  012700  000037                                                          MOV     #EF.RESTART,R0
022710  104447                                                                  TRAP    C$REFG
                                BCOMPLETE       STARST
022712  103411                                                                  BCS     STARST

                        ;SEE IF THIS IS A NEW PASS, BR IF YES
                                READEF  #EF.NEW
022714  012700  000035                                                          MOV     #EF.NEW,R0
022720  104447                                                                  TRAP    C$REFG
                                BCOMPLETE       NEWST
022722  103411                                                                  BCS     NEWST

                        ;SEE IF PROGRAM WAS JUST CONTINUED
```

INITIALIZE SECTION

```
4190 022724                            READEF  #EF.CONTINUE
     022724  012700  000036                                                    MOV     #EF.CONTINUE,R0
     022730  104447                                                            TRAP    C$REFG
4191 022732                            BCOMPLETE       ENDIT
     022732  103473                                                            BCS     ENDIT
4192 022734  000414                    BR      GETPRM
4193
4194 022736                    STARST:
4195 022736  005037  002412            CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
4196
4197                           ;CLEAR DEVICE MAP
4198 022742  005037  002404            CLR     DEVMAP
4199 022746                    NEWST:
4200 022746  012737  177777  002334    MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
4201 022754  005237  002412            INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
4202 022760  012737  000001  002406    MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
4203
4204                           ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
4205                           ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
4206 022766                    GETPRM:
4207 022766  005237  002334            INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
4208 022772                            GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
     022772  013700  002334                                                    MOV     LOGDEV,R0
     022776  104442                                                            TRAP    C$GPHRD
     023000  010001                                                            MOV     R0,R1
4209 023002                            BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
     023002  103403                                                            BCS     10$
4210 023004  006337  002406            ASL     DEVPTR          ;SHIFT DEVICE POINTER
4211 023010  000766                    BR      GETPRM          ;SKIP THIS DEVICE
4212 023012  053737  002406  002404  10$:      BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
4213 023020  006337  002406            ASL     DEVPTR          ;SHIFT BIT POINTER
4214
4215 023024  012102                    MOV     (R1)+,R2        ;R2=CSR ADDR VALUE
4216 023026  012703  002416            MOV     #MPCSR,R3       ;R3=POINTER TO CSR ADDR STORAGE AREA
4217
4218 023032  010223              11$:      MOV     R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
4219 023034  005202                    INC     R2              ;BUMP BSEL ADDR
4220 023036  022703  002456            CMP     #BSEL17+2,R3    ;ALL 16 ADDRESSES MOVED ?
4221 023042  001373                    BNE     11$             ;NO: DO ANOTHER ADDRESS
4222                                                           ;YES: CONTINUE
4223
4224 023044  011137  002456            MOV     (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
4225 023050  012137  002460            MOV     (R1)+,MPOVEC
4226 023054  062737  000004  002460    ADD     #4,MPOVEC       ;GET DMV11 OUTPUT INTRPT VECTOR
4227 023062  012137  002462            MOV     (R1)+,MPRIOR    ;GET DMV11 DEVICE PRIORITY
4228 023066  012137  002464            MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
4229 023072  012137  002466            MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
4230 023076  012137  002470            MOV     (R1)+,BRDTYP    ;GET DMV-11 BOARD TYPE
4231 023102  012137  002472            MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
4232 023106  011137  002474            MOV     (R1),BDRATE     ;GET BAUD RATE FOR THIS DEVICE
4233                           ;ISSUE LSI BUS RESET, TO INIT DMV11
4234 023112                            BRESET
     023112  104433                                                            TRAP    C$RESET
4235 023114  005000                    CLR     R0              ;# TIME DELAY TO ALLOW COMPLETION
4236 023116  000...            15$:      NOP                     ;# OF DMV11 MICRODIAGNOSTICS.
4237 023120  077002                    SOB     R0,15$          ;#
4238 023122                    ENDIT:
```

INITIALIZE SECTION

```
4239 023122                            ENDINIT
     023122                                                L10013:
     023122  104411                                              TRAP    C$INIT
```

AUTO DROP UNIT SECTION

```
                                .SBTTL AUTO DROP UNIT SECTION

                                ;//////////////////////////////////////////////////////////////////////////////
                                ;/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
                                ;/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
                                ;//////////////////////////////////////////////////////////////////////////////

                                ;+*****************************************************************************
                                ;
                                ;       THIS ALGORITHM IS THE SAME A CNDMA TEST # 1 EXCEPT THAT TEST
                                ;       WILL JUST REPORT THE FAILURE AND GO ON -- THIS ROUTINE WILL CAUSE THE
                                ;       DEVICE TO BE DROPPED IF A BUS-TIMEOUT OCCURS WHEN ANY OF THE CSR'S
                                ;       ARE ACCESSED WITH EITHER A "TST" OR "TSTB" INSTRUCTION.
                                ;
                                ;-*****************************************************************************

023124                                  BGNAUTO
023124                                                                  L$AUTO::
023124                                  SETVEC  #4,#AD.HIT,#0   ;SETUP INVALID-ADDRESS TRAP VECTOR
023124  012746  000000                                                  MOV     #0,-(SP)
023130  012746  023242                                                  MOV     #AD.HIT,-(SP)
023134  012746  000004                                                  MOV     #4,-(SP)
023140  012746  000003                                                  MOV     #3,-(SP)
023144  104437                                                          TRAP    C$SVEC
023146  062706  000010                                                  ADD     #10,SP
023152  005037  002546                  CLR     TMP0            ;INITIALIZE TRAP FLAG REGISTER
023156  012702  000001                  MOV     #1,R2           ;FLAG BIT
023162  013703  002416                  MOV     BSEL0,R3        ;INIT ADDRESS POINTER

023166  105723          1$:             TSTB    (R3)+           ;ACCESS THE CSR'S BY BYTES.
023170  006302                          ASL     R2
023172  103375                          BCC     1$

023174  013703  002416                  MOV     BSEL0,R3        ;RE-INIT ADDRESS POINTER
023200  012702  000001                  MOV     #1,R2           ;RE-INIT FLAG BIT
023204  005723          2$:             TST     (R3)+           ;ACCESS THE CSR'S BY WORDS.
023206  006302                          ASL     R2
023210  006302                          ASL     R2
023212  103374                          BCC     2$

023214                                  CLRVEC  #4              ;RESTORE THE VECTOR TO DS
023214  012700  000004                                                  MOV     #4,R0
023220  104436                                                          TRAP    C$CVEC
023222  005737  002546                  TST     TMP0            ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP?
023226  001403                          BEQ     AD.OK           ;NO,  EXIT TEST
023230                                  DODU    LOGDEV          ;YES, DROP THIS LOGICAL DEV.
023230  013700  002334                                                  MOV     LOGDEV,R0
023234  104451                                                          TRAP    C$DODU

023236  000240          AD.OK:  NOP                             ;(FOR PATCHING IN A HALT IF NECESSARY)

023240                                  ENDAUTO
023240                                                                  L10014:
023240  104461                                                          TRAP    C$AUTO

023242  050237  002546          AD.HIT: BIS     R2,TMP0         ;FLAG THE HIT IF WE GET IT!
```

AUTO DROP UNIT SECTION

```
4285 023246  000002                    RTI                    ;RETURN
4286
```

CLEANUP CODING SECTION

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;////////////////////////////////////////////////////////////////////////
                                ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;/ AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
                                ;////////////////////////////////////////////////////////////////////////

023250                                  BGNCLN
023250                                                                  L$CLEAN::
023250                                  ENDCLN
023250                                                                  L10015:
023250  104412                                                                  TRAP    C$CLEAN
```

DROP UNIT SECTION

```
                                .SBTTL  DROP UNIT SECTION

                                ;////////////////////////////////////////////////////////////////////////////////
                                ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ;/ TO NO LONGER BE TESTED.
                                ;////////////////////////////////////////////////////////////////////////////////

023252                                  BGNDU
023252                                                                  L$DU::
                                ;ISSUE UNIBUS RESET TO CLEAN UP
023252                                  BRESET
023252  104433                                                                  TRAP    C$RESET
023254                                  ENDDU
023254                                                                  L10016:
023254  104453                                                                  TRAP    C$DU
```

```
                                .SBTTL  ADD UNIT SECTION

                                ;////////////////////////////////////////////////////////////////////////////
                                ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
                                ;/  "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
                                ;////////////////////////////////////////////////////////////////////////////

                                        BGNAU
023256                                                                  L.$AU::
                                        ENDAU
023256                                                                  L10017:
023256  104452                                                                 TRAP    C$AU
```

TEST 1 -- RX DATA FLUSHING TEST

```
4332                                    .SBTTL  TEST 1 -- RX DATA FLUSHING TEST

                                        ;+*********************************************************************
                                        ;*
                                        ;*      TEST 1 -- RX DATA FLUSHING TEST
                                        ;*
                                        ;*
                                        ;*  IN BCP MODE/HALF DUPLEX IT IS DESIRABLE TO HAVE THE ABILITY TO FLUSH
                                        ;*  THE USYRT OF ITS CRC CHARACTERS. THIS FLUSHING IS ACCOMPLISHED BY READING
                                        ;*  TO THE VIA SHIFT REGISTER.
                                        ;*  THIS TEST VERIFIES THAT WHEN THE VIA SR IS READ, 8 PULSES WILL
                                        ;*  BE GENERATED AT THE CB1 PIN (WHICH DIRECTLY FEEDS THE CHARACTER FIFO).
                                        ;*
                                        ;--*********************************************************************
                                        ;
                                        ;       BGNTST
     023260                                                                             T1::
4333
4334 023260  004737  005376                    JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
4335
4336 023264  004537  007234                    JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
4337 023270  063626                            DDCMP!NOCHK!SYNCH!STRIPS ;SET DDCMP,NO CHECK,SYNCH=226
4338 023272  000000                            0                       ;USE 8 BIT CHARS
4339 023274  103003                            BCC     .+8.            ;BR IF NO ERROR
4340 023276                                    ERROR                   ;REPORT STACKED ERROR
     023276  104460                                                                     TRAP    C$ERROR
4341 023300                                    ESCAPE  TST             ;SKIP TO END OF TEST
     023300  104410                                                                     TRAP    C$ESCAPE
     023302  000254                                                                     .WORD   L10020-.
4342
4343 023304  004537  003712                    JSR     R5,WRITEI       ;SET SHIFT REGISTER TO
4344 023310  120013                            VIAACR                  ; "SYSTEM CLOCK RATE" MODE (CB1=CLK)
4345 023312  000210                            210                     ;(BIT 7 PREVIOUSLY SET)
4346 023314  103003                            BCC     .+8.            ;BR IF NO ERROR
4347 023316                                    ERROR                   ; REPORT STACKED ERROR
     023316  104460                                                                     TRAP    C$ERROR
4348 023320                                    ESCAPE  TST             ;SKIP TO END OF TEST
     023320  104410                                                                     TRAP    C$ESCAPE
     023322  000234                                                                     .WORD   L10020-.
4349
4350 023324  004537  010250                    JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
4351 023330  000001                            TSOM                    ;AND KNOCK DOWN TBMT
4352 023332  000007                            7.
4353
4354 023334  004537  010250                    JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
4355 023340  000001                            TSOM                    ;AND KNOCK DOWN TBMT
4356 023342  000010                            8.
4357
4358 023344  004537  010250                    JSR     R5,TXCTRL       ;CLEAR TSOM (GET READY TO SEND DATA)
4359 023350  000000                            000
4360 023352  000000                            0
4361
4362 023354  004537  010136                    JSR     R5,TXCHAR       ;LOAD 125, TX 3RD SYNCH
4363 023360  000125                            125
4364 023362  000010                            8.
4365 023364  103003                            BCC     .+8.            ;BR IF NO ERROR
4366 023366                                    ERROR                   ;REPORT STACKED ERROR
```

TEST 1 -- RX DATA FLUSHING TEST

```
      023366  104460                                                          TRAP    C$ERROR
4367  023370                        ESCAPE  TST             ;SKIP TO END OF TEST
      023370  104410                                                          TRAP    C$ESCAPE
      023372  000164                                                          .WORD   L10020-.
4368                          ;-----------------------------------------------------------------
4369  023374  012702  000004        MOV     #4,R2           ;** TRANSFER 4 CHARACTERS **
4370
4371  023400  004537  003712  1$:   JSR     R5,WRITEI       ;SET RTS & FULL DUPLEX (SO STEPLU WORKS)
4372  023404  120000                VIAORB
4373  023406  000142                TXEN!RXEN!TTLOOP
4374  023410  103003                BCC     .+8.            ;BR IF NO ERROR
4375  023412                        ERROR                   ; REPORT STACKED ERROR
      023412  104460                                                          TRAP    C$ERROR
4376  023414                        ESCAPE  TST             ;SKIP TO END OF TEST
      023414  104410                                                          TRAP    C$ESCAPE
      023416  000140                                                          .WORD   L10020-.
4377
4378  023420  004537  012072        JSR     R5,STEPLU       ;FLIP TSO BIT VALUE(WILL BE SHIFTED INTO
4379  023424  000001                1                       ; FIFO DURING FLUSHING).
4380
4381  023426  004537  003712        JSR     R5,WRITEI       ;CLEAR RTS, SET HDX (SO THAT SR CLOCK WORKS)
4382  023432  120000                VIAORB
4383  023434  000152                TXEN!RXEN!RTSND!TTLOOP  ;#
4384  023436  103003                BCC     .+8.            ;BR IF NO ERROR
4385  023440                        ERROR                   ; REPORT STACKED ERROR
      023440  104460                                                          TRAP    C$ERROR
4386  023442                        ESCAPE  TST             ;SKIP TO END OF TEST
      023442  104410                                                          TRAP    C$ESCAPE
      023444  000112                                                          .WORD   L10020-.
4387
4388  023446  004537  003566        JSR     R5,READI        ;READ VIA SHIFT REGISTER (SHOULD CAUSE
4389  023452  120012                VIASR                   ; 8 CLOCKS FROM CB1 LEAD => FIFO)
4390  023454  000000                000
4391  023456  103003                BCC     .+8.            ;BR IF NO ERROR
4392  023460                        ERROR                   ; REPORT STACKED ERROR
      023460  104460                                                          TRAP    C$ERROR
4393  023462                        ESCAPE  TST             ;SKIP TO END OF TEST
      023462  104410                                                          TRAP    C$ESCAPE
      023464  000072                                                          .WORD   L10020-.
4394
4395  023466  077234                SOB     R2,1$           ;** LOOP UNTIL ALL 4 SENT (VIA CB1) **
4396                          ;-----------------------------------------------------------------
4397  023470  004537  010350        JSR     R5,RXCHAR       ;READ AND CHECK FOR 377
4398  023474  000377                377                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
4399  023476  000000                0                       ;* DIDN'T WORK.
4400  023500  100000                NOCRDA
4401  023502  103003                BCC     .+8.            ;BR IF NO ERROR
4402  023504                        ERROR                   ;REPORT STACKED ERROR
      023504  104460                                                          TRAP    C$ERROR
4403  023506                        ESCAPE  TST             ;SKIP TO END OF TEST
      023506  104410                                                          TRAP    C$ESCAPE
      023510  000046                                                          .WORD   L10020-.
4404
4405  023512  004537  010350        JSR     R5,RXCHAR       ;READ AND CHECK FOR 000
4406  023516  000003                003                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
4407  023520  000000                0                       ;* DIDN'T WORK.
4408  023522  100000                NOCRDA
```

TEST 1 -- RX DATA FLUSHING TEST

```
4409 023524  103003                    BCC     .+8.          ;BR IF NO ERROR
4410 023526                            ERROR                 ;REPORT STACKED ERROR
     023526  104460                                                  TRAP    C$ERROR
4411 023530                            ESCAPE  TST           ;SKIP TO END OF TEST
     023530  104410                                                  TRAP    C$ESCAPE
     023532  000024                                                  .WORD   L10020-.
4412
4413 023534  004537  010350            JSR     R5,RXCHAR     ;READ AND CHECK FOR 377
4414 023540  000360                    360                   ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
4415 023542  000000                    0                     ;* DIDN'T WORK
4416 023544  100000                    NOCRDA
4417 023546  103003                    BCC     .+8.          ;BR IF NO ERROR
4418 023550                            ERROR                 ;REPORT STACKED ERROR
     023550  104460                                                  TRAP    C$ERROR
4419 023552                            ESCAPE  TST           ;SKIP TO END OF TEST
     023552  104410                                                  TRAP    C$ESCAPE
     023554  000002                                                  .WORD   L10020-.
4420 023556                    ENDTST
     023556                                                  L10020:
     023556  104401                                                  TRAP    C$ETST
```

```
4434                                    .SBTTL  TEST 2 -- INTEGRAL MODEM INTERFACE TEST

                                        ;+*****************************************************************************
                                        ;*
                                        ;*      TEST 2 -- INTEGRAL MODEM INTERFACE TEST
                                        ;*
                                        ;*  THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM AND A MESSAGE IS
                                        ;*  TRANSMITTED,  RECEIVED,  AND CHECKED USING A TURNAROUND CONNECTOR ON
                                        ;*  THE  BOARD  OR  AT THE END OF A CABLE. THE FOLLOWING MESSAGE WILL BE
                                        ;*  SENT IN BCP MODE WITH CRC-16 SPECIFIED:
                                        ;*
                                        ;*      SYNC SYNC 000 125 252 377 000 CRC1 CRC2 SYNC
                                        ;*
                                        ;*  IF  THE  P-TABLE  FOR  THE  CURRENT  UNIT INDICATES THAT NO EXTERNAL
                                        ;*  TURNAROUND IS PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
                                        ;*
                                        ;-*****************************************************************************

                                        ;       BGNTST
     023560                                                                                     T2::
4435 023560  012737  000002  002414            MOV     #2.,TSTNUM      ;SET TEST NO. FOR POSSIBLE PRINTOUT
4436 023566  004737  005376                    JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
4437 023572  004537  007532                    JSR     R5,CKLPBK       ;SEE IF THIS INTERFACE CAN BE RUN
4438 023576  000001                            INTGRL
4439 023600  103002                            BCC     2$              ;BR IF YES
4440 023602                                    EXIT    TST             ;WRONG INTERFACE - SKIP TEST
     023602  104432                                                            TRAP    C$EXIT
     023604  001630                                                            .WORD   L10021-.
4441
4442                                    ;LOAD 6502 MICROCODE FOR INTEGRAL MODEM TEST INTO RAM PAGE 2
4443 023606  012701  025436            2$:     MOV     #MCODE,R1       ;GET STARTING ADRS OF DMV MICROCODE
4444 023612  012702  001000                    MOV     #RAMADR,R2      ;GET STARTING ADRS OF RAM PAGE 2
4445 023616  112137  023634            3$:     MOVB    (R1)+,6$        ;SET DATA BYTE TO BE WRITTEN
4446 023622  010237  023632                    MOV     R2,4$           ;SET RAM WRITE ADRS
4447 023626  004537  003712                    JSR     R5,WRITEI       ;WRITE A DATA BYTE INTO RAM
4448 023632  000000                    4$:     .WORD   0
4449 023634  000000                    6$:     .WORD   0
4450 023636  005202                            INC     R2              ;INCR RAM ADRS
4451 023640  020127  026263                    CMP     R1,#ENDCOD      ;SEE IF ALL CODE LOADED YET
4452 023644  103764                            BLO     3$              ;BR IF NOT
4453
4454                                    ;READ AND VERIFY 6502 MICROCODE IN RAM
4455 023646  012701  025436                    MOV     #MCODE,R1       ;GET STARTING ADRS OF DMV MICROCODE TO CHECK
4456 023652  012702  001000                    MOV     #RAMADR,R2      ;GET STARTING ADRS OF RAM PAGE 2
4457 023656  010237  023666            8$:     MOV     R2,10$          ;SET RAM READ ADRS
4458 023662  004537  003566                    JSR     R5,READI        ;READ A RAM BYTE
4459 023666  000000                    10$:    .WORD   0
4460 023670  000000                    12$:    .WORD   0
4461 023672  122137  023670                    CMPB    (R1)+,12$       ;SEE IF BYTE IS CORRECT
4462 023676  001422                            BEQ     16$             ;BR IF CORRECT
4463 023700  010237  002336                    MOV     R2,REGNUM       ;SET RAM ADRS FOR ERROR REPORT
4464 023704  005037  002324                    CLR     GDATA           ;SET EXPECTED RAM DATA
4465 023710  116137  177777  002324            MOVB    -1(R1),GDATA
4466 023716  005037  002326                    CLR     BDATA           ;SET ACTUAL RAM DATA
4467 023722  113737  023670  002326            MOVB    12$,BDATA
4468                                    ;REPORT RAM ERROR LOADING MICROCODE
4469 023730                                    GEDF    EM79,ERR13
```

```
                                                        ;       "DEVICE FATAL" ERROR # 38
      023730  104455                                                                      TRAP    C$ERDF
      023732  000046                                                                      .WORD   38
      023734  015125                                                                      .WORD   EM79
      023736  021030                                                                      .WORD   ERR13
4470  023740                          ESCAPE  TST
      023740  104410                                                                      TRAP    C$ESCAPE
      023742  001472                                                                      .WORD   L10021-.
4471  023744  005202          16$:    INC     R2              ;INCR RAM ADRS
4472  023746  020127  026263        CMP     R1,#ENDCOD      ;SEE IF ALL CODE CHECKED YET
4473  023752  103741                BLO     8$              ;BR IF NOT
4474
4475                          ;SET UP VIA AND USYRT FOR OPERATION
4476  023754  004537  003712        JSR     R5,WRITEI       ;RESET THE USYRT
4477  023760  120000                VIAORB
4478  023762  000031                RTSND!DTR!PRESET
4479  023764  004537  003712        JSR     R5,WRITEI       ;CLEAR USYRT RESET BIT
4480  023770  120000                VIAORB
4481  023772  000030                RTSND!DTR
4482  023774  004537  003712        JSR     R5,WRITEI       ;SET SYNCH CHAR = 226
4483  024000  120404                PCSARL
4484  024002  000226                226
4485  024004  012737  065400  024024    MOV     #DDCMP!STRIPS!IDLES!CRC16,18$   ;SET DDCMP,STRIP,IDLE, CRC16
4486  024012  000337  024024        SWAB    18$             ;GET DATA INTO LO BYTE
4487  024016  004537  003712        JSR     R5,WRITEI       ;PROGRAM THE PCSARH
4488  024022  120405                PCSARH
4489  024024  000000          18$:    .WORD   0
4490  024026  004537  005410        JSR     R5,CKUSTS       ;CHK USYRT STATUS FOR INITIALIZED STATE
4491  024032  000110                110                     ;  TBMT=1, TSO=1
4492  024034  103003                BCC     .+8.            ;IF ERROR, PRINT REPORT
4493  024036                        ERROR
      024036  104460                                                                      TRAP    C$ERROR
4494  024040                          ESCAPE  TST
      024040  104410                                                                      TRAP    C$ESCAPE
      024042  001372                                                                      .WORD   L10021-.
4495  024044  004537  003712        JSR     R5,WRITEI       ;SET TSOM IN USYRT
4496  024050  120403                TDSRH
4497  024052  000001                TSOM
4498  024054  004537  003712        JSR     R5,WRITEI       ;LOAD 237 CHAR FOR INTGRL MODEM SYNCHRONIZATION
4499  024060  120402                TDSRL
4500  024062  000237                237
4501  024064  004537  006014        JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
4502  024070  000000                0
4503  024072  103003                BCC     .+8.            ;IF ERROR, REPORT ERROR
4504  024074                        ERROR
      024074  104460                                                                      TRAP    C$ERROR
4505  024076                          ESCAPE  TST
      024076  104410                                                                      TRAP    C$ESCAPE
      024100  001334                                                                      .WORD   L10021-.
4506
4507                          ;INITIATE 6502 TEST OF INTEGRAL MODEM
4508  024102  005077  156320        CLR     @SEL4           ;CLEAR SEL4
4509  024106  012777  001000  156312    MOV     #RAMADR,@SEL4   ;SET START ADRS OF RAM CODE IN SEL4
4510  024114  112777  000005  156300    MOVB    #EXECUT,@BSEL2  ;ISSUE M-LOOP CMND TO EXECUTE AT PC IN SEL4
4511
4512  024122  012701  001750  ;WAIT SEVERAL MILLI-SEC FOR COMPLETION OF TEST
                                MOV     #1000.,R1       ;INIT WAIT LOOP COUNTER
4513  024126  005301          22$:    DEC     R1              ;DECREMENT COUNTER
```

```
4514 024130  000240                          NOP
4515 024132  001375                          BNE     22$             ;BR IF NOT DONE COUNTING YET
4516 024134  132777  000200  156260          BITB    @MRDY,@BSEL2    ;SEE IF M-LOOP FINISHED PROPERLY
4517 024142  001013                          BNE     24$             ;BR IF YES
4518 024144  004737  004166                  JSR     PC,GETWSR       ;GET CSR'S FOR PRINTOUT
4519 024150  012737  000005  002324          MOV     #EXECUT,GDATA   ;IDENTIFY REQUESTED FUNCTION
4520                                 ;REPORT MRDY TIMEOUT ERROR
4521 024156                                  GEDF    EM4,ERR4
                                                     ;       "DEVICE FATAL" ERROR # 39
     024156  104455                                                          TRAP    C$ERDF
     024160  000047                                                          .WORD   39
     024162  014307                                                          .WORD   EM4
     024164  020120                                                          .WORD   ERR4
4522 024166                                  ESCAPE  TST
     024166  104410                                                          TRAP    C$ESCAPE
     024170  001244                                                          .WORD   L10021-.
4523
4524 024172  105777  156234          24$:    TSTB    @BSEL6          ;SEE IF ANY ERRORS OCCURRED IN TEST
4525 024176  001002                          BNE     26$             ;BR IF ERROR DETECTED
4526 024200  000137  025434                  JMP     90$             ;SUCCESSFUL COMPLETION OF TEST
4527 024204  012737  000007  002336  26$:    MOV     #7,REGNUM       ;SET REG NO. FOR PRINTOUT
4528 024212  005037  002324                  CLR     GDATA           ;CLEAR EXPECTED DATA AREA
4529 024216  005037  002326                  CLR     BDATA           ;CLEAR ACTUAL DATA AREA
4530                                 ;CHK FOR ERROR 1
4531 024222  127727  156204  000001          CMPB    @BSEL6,#1       ;CHK FOR ERROR 1
4532 024230  001006                          BNE     28$             ;BR IF NOT
4533                                 ;REPORT TBMT NOT SET
4534 024232                                  GEDF    EM73,ERR12
                                                     ;       "DEVICE FATAL" ERROR # 40
     024232  104455                                                          TRAP    C$ERDF
     024234  000050                                                          .WORD   40
     024236  014777                                                          .WORD   EM73
     024240  020714                                                          .WORD   ERR12
4535 024242                                  ESCAPE  TST
     024242  104410                                                          TRAP    C$ESCAPE
     024244  001170                                                          .WORD   L10021-.
4536                                 ;CHK FOR ERROR 2
4537 024246  127727  156160  000002  28$:    CMPB    @BSEL6,#2       ;CHK FOR ERROR 2
4538 024254  001006                          BNE     30$             ;BR IF NOT
4539                                 ;REPORT TBMT NOT SET
4540 024256                                  GEDF    EM73,ERR12
                                                     ;       "DEVICE FATAL" ERROR # 41
     024256  104455                                                          TRAP    C$ERDF
     024260  000051                                                          .WORD   41
     024262  014777                                                          .WORD   EM73
     024264  020714                                                          .WORD   ERR12
4541 024266                                  ESCAPE  TST
     024266  104410                                                          TRAP    C$ESCAPE
     024270  001144                                                          .WORD   L10021-.
4542                                 ;CHK FOR ERROR 3
4543 024272  127727  156134  000003  30$:    CMPB    @BSEL6,#3       ;CHK FOR ERROR 3
4544 024300  001006                          BNE     31$             ;BR IF NOT
4545                                 ;REPORT TBMT NOT SET
4546 024302                                  GEDF    EM73,ERR12
                                                     ;       "DEVICE FATAL" ERROR # 42
     024302  104455                                                          TRAP    C$ERDF
     024304  000052                                                          .WORD   42
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
      024306  014777                                                                  .WORD   EM73
      024310  020714                                                                  .WORD   ERR12
4547  024312                              ESCAPE  TST
      024312  104410                                                                  TRAP    C$ESCAPE
      024314  001120                                                                  .WORD   L10021-.
4548                              ;CHK FOR ERROR 4
4549  024316  127727  156110  000004  31$:    CMPB    @#BSEL6,#4      ;CHK FOR ERROR 4
4550  024324  001011                        BNE     32$             ;BR IF NOT
4551                              ;REPORT CARRIER NOT SET
4552  024326  012737  000001  002336        MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
4553  024334                              GEDF    EM80,ERR14
                                                        ;       "DEVICE FATAL" ERROR # 43
      024334  104455                                                                  TRAP    C$ERDF
      024336  000053                                                                  .WORD   43
      024340  015161                                                                  .WORD   EM80
      024342  021146                                                                  .WORD   ERR14
4554  024344                              ESCAPE  TST
      024344  104410                                                                  TRAP    C$ESCAPE
      024346  001066                                                                  .WORD   L10021-.
4555                              ;CHK FOR ERROR 5
4556  024350  127727  156056  000005  32$:    CMPB    @#BSEL6,#5      ;CHK FOR ERROR 5
4557  024356  001006                        BNE     34$             ;BR IF NOT
4558                              ;REPORT TBMT NOT SET
4559  024360                              GEDF    EM73,ERR12
                                                        ;       "DEVICE FATAL" ERROR # 44
      024360  104455                                                                  TRAP    C$ERDF
      024362  000054                                                                  .WORD   44
      024364  014777                                                                  .WORD   EM73
      024366  020714                                                                  .WORD   ERR12
4560  024370                              ESCAPE  TST
      024370  104410                                                                  TRAP    C$ESCAPE
      024372  001042                                                                  .WORD   L10021-.
4561                              ;CHK FOR ERROR 6
4562  024374  127727  156032  000006  34$:    CMPB    @#BSEL6,#6      ;CHK FOR ERROR 6
4563  024402  001006                        BNE     36$             ;BR IF NOT
4564                              ;REPORT TBMT NOT SET
4565  024404                              GEDF    EM73,ERR12
                                                        ;       "DEVICE FATAL" ERROR # 45
      024404  104455                                                                  TRAP    C$ERDF
      024406  000055                                                                  .WORD   45
      024410  014777                                                                  .WORD   EM73
      024412  020714                                                                  .WORD   ERR12
4566  024414                              ESCAPE  TST
      024414  104410                                                                  TRAP    C$ESCAPE
      024416  001016                                                                  .WORD   L10021-.
4567                              ;CHK FOR ERROR 7
4568  024420  127727  156006  000007  36$:    CMPB    @#BSEL6,#7      ;CHK FOR ERROR 7
4569  024426  001006                        BNE     38$             ;BR IF NOT
4570                              ;REPORT TBMT NOT SET
4571  024430                              GEDF    EM73,ERR12
                                                        ;       "DEVICE FATAL" ERROR # 46
      024430  104455                                                                  TRAP    C$ERDF
      024432  000056                                                                  .WORD   46
      024434  014777                                                                  .WORD   EM73
      024436  020714                                                                  .WORD   ERR12
4572  024440                              ESCAPE  TST
      024440  104410                                                                  TRAP    C$ESCAPE
```

```
     024442  000772                                                                  .WORD   L10021-.
4573                                ;CHK FOR ERROR 8
4574 024444  127727  155762  000010 38$:    CMPB    @#SEL6,#8.      ;CHK FOR ERROR 8
4575 024452  001006                        BNE     40$             ;BR IF NOT
4576                                ;REPORT TBMT NOT SET
4577 024454                                GEDF    EM73,ERR12
                                                    ;       "DEVICE FATAL" ERROR # 47
     024454  104455                                                                 TRAP    C$ERDF
     024456  000057                                                                 .WORD   47
     024460  014777                                                                 .WORD   EM73
     024462  020714                                                                 .WORD   ERR12
4578 024464                                ESCAPE  TST
     024464  104410                                                                 TRAP    C$ESCAPE
     024466  000746                                                                 .WORD   L10021-.
4579                                ;CHK FOR ERROR 9
4580 024470  127727  155736  000011 40$:    CMPB    @#SEL6,#9.      ;CHK FOR ERROR 9
4581 024476  001006                        BNE     42$             ;BR IF NOT
4582                                ;REPORT RDA NOT SET
4583 024500                                GEDF    EM75,ERR12
                                                    ;       "DEVICE FATAL" ERROR # 48
     024500  104455                                                                 TRAP    C$ERDF
     024502  000060                                                                 .WORD   48
     024504  015035                                                                 .WORD   EM75
     024506  020714                                                                 .WORD   ERR12
4584 024510                                ESCAPE  TST
     024510  104410                                                                 TRAP    C$ESCAPE
     024512  000722                                                                 .WORD   L10021-.
4585                                ;CHK FOR ERROR 10
4586 024514  127727  155712  000012 42$:    CMPB    @#SEL6,#10.     ;CHK FOR ERROR 10
4587 024522  001017                        BNE     44$             ;BR IF NOT
4588                                ;REPORT RCV'D DATA MISCOMPARE ERROR
4589 024524  012737  000000  002336        MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
4590 024532  112737  000000  002324        MOVB    #000,GDATA      ;SET EXPECTED DATA
4591 024540  117737  155670  002326        MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
4592 024546                                GEDF    EM34,ERR10
                                                    ;       "DEVICE FATAL" ERROR # 49
     024546  104455                                                                 TRAP    C$ERDF
     024550  000061                                                                 .WORD   49
     024552  014502                                                                 .WORD   EM34
     024554  020364                                                                 .WORD   ERR10
4593 024556                                ESCAPE  TST
     024556  104410                                                                 TRAP    C$ESCAPE
     024560  000654                                                                 .WORD   L10021 .
4594                                ;CHK FOR ERROR 11
4595 024562  127727  155644  000013 44$:    CMPB    @#SEL6,#11.     ;CHK FOR ERROR 11
4596 024570  001006                        BNE     46$             ;BR IF NOT
4597                                ;REPORT RDA NOT SET
4598 024572                                GEDF    EM75,ERR12
                                                    ;       "DEVICE FATAL" ERROR # 50
     024572  104455                                                                 TRAP    C$ERDF
     024574  000062                                                                 .WORD   50
     024576  015035                                                                 .WORD   EM75
     024600  020714                                                                 .WORD   ERR12
4599 024602                                ESCAPE  TST
     024602  104410                                                                 TRAP    C$ESCAPE
     024604  000630                                                                 .WORD   L10021-.
4600                                ;CHK FOR ERROR 12
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
4601 024606  127727  155620  000014  46$:    CMPB    @#SEL6,#12.     ;CHK FOR ERROR 12
4602 024614  001017                          BNE     48$             ;BR IF NOT
4603                                 ;REPORT RCV'D DATA MISCOMPARE ERROR
4604 024     127737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
4605 024     127737  000125  002324          MOVB    #125,GDATA      ;SET EXPECTED DATA
4606 024     127737  155576  002326          MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
4607 024                                     GEDF    EM34,ERR10
                                                     ;       "DEVICE FATAL" ERROR # 51
     024640  104455                                                          TRAP    C$ERDF
     024642  000063                                                          .WORD   51
     024644  014502                                                          .WORD   EM34
     024646  020364                                                          .WORD   ERR10
4608 024650                                  ESCAPE  TST
     024650  104410                                                          TRAP    C$ESCAPE
     024652  000562                                                          .WORD   L10021-.
4609                                 ;CHK FOR ERROR 13
4610 024654  127727  155552  000015  48$:    CMPB    @#SEL6,#13.     ;CHK FOR ERROR 13
4611 024662  001006                          BNE     50$             ;BR IF NOT
4612                                 ;REPORT RDA NOT SET
4613 024664                                  GEDF    EM75,ERR12
                                                     ;       "DEVICE FATAL" ERROR # 52
     024664  104455                                                          TRAP    C$ERDF
     024666  000064                                                          .WORD   52
     024670  015035                                                          .WORD   EM75
     024672  020714                                                          .WORD   ERR12
4614 024674                                  ESCAPE  TST
     024674  104410                                                          TRAP    C$ESCAPE
     024676  000536                                                          .WORD   L10021-.
4615                                 ;CHK FOR ERROR 14
4616 024700  127727  155526  000016  50$:    CMPB    @#SEL6,#14.     ;CHK FOR ERROR 14
4617 024706  001017                          BNE     52$             ;BR IF NOT
4618                                 ;REPORT RCV'D DATA MISCOMPARE ERROR
4619 024710  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
4620 024716  112737  000252  002324          MOVB    #252,GDATA      ;SET EXPECTED DATA
4621 024724  117737  155504  002326          MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
4622 024732                                  GEDF    EM34,ERR10
                                                     ;       "DEVICE FATAL" ERROR # 53
     024732  104455                                                          TRAP    C$ERDF
     024734  000065                                                          .WORD   53
     024736  014502                                                          .WORD   EM34
     024740  020364                                                          .WORD   ERR10
4623 024742                                  ESCAPE  TST
     024742  104410                                                          TRAP    C$ESCAPE
     024744  000470                                                          .WORD   L10021-.
4624                                 ;CHK FOR ERROR 15
4625 024746  127727  155460  000017  52$:    CMPB    @#SEL6,#15.     ;CHK FOR ERROR 15
4626 024754  001006                          BNE     54$             ;BR IF NOT
4627                                 ;REPORT RDA NOT SET
4628 024756                                  GEDF    EM75,ERR12
                                                     ;       "DEVICE FATAL" ERROR # 54
     024756  104455                                                          TRAP    C$ERDF
     024760  000066                                                          .WORD   54
     024762  015035                                                          .WORD   EM75
     024764  020714                                                          .WORD   ERR12
4629 024766                                  ESCAPE  TST
     024766  104410                                                          TRAP    C$ESCAPE
     024770  000444                                                          .WORD   L10021-.
```

```
                                        ;CHK FOR ERROR 16
 025772  127727  155434  000020  54$:   CMPB    @#SEL6,#16.     ;CHK FOR ERROR 16
 026000  001017                          BNE     56$             ;BR IF NOT
                                        ;REPORT RCV'D DATA MISCOMPARE ERROR
 026002  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
 026010  112737  000377  002324          MOVB    #377,GDATA      ;SET EXPECTED DATA
 026016  117737  155412  002326          MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
 026024                                  GEDF    EM34,ERR10
                                                 ;       "DEVICE FATAL" ERROR # 55
 025024  104455                                                  TRAP    C$ERDF
 025026  000067                                                  .WORD   55
 025030  014502                                                  .WORD   EM34
 025032  020364                                                  .WORD   ERR10
 026034                                  ESCAPE  TST
 025034  104410                                                  TRAP    C$ESCAPE
 025036  000376                                                  .WORD   L10021-.
                                        ;CHK FOR ERROR 17
 026040  127727  155366  000021  56$:   CMPB    @#SEL6,#17.     ;CHK FOR ERROR 17
 026046  001006                          BNE     58$             ;BR IF NOT
                                        ;REPORT RDA NOT SET
 026050                                  GEDF    EM75,ERR12
                                                 ;       "DEVICE FATAL" ERROR # 56
 025050  104455                                                  TRAP    C$ERDF
 025052  000070                                                  .WORD   56
 025054  015035                                                  .WORD   EM75
 025056  020714                                                  .WORD   ERR12
 026060                                  ESCAPE  TST
 025060  104410                                                  TRAP    C$ESCAPE
 025062  000352                                                  .WORD   L10021-.
                                        ;CHK FOR ERROR 18
 026064  127727  155342  000022  58$:   CMPB    @#SEL6,#18.     ;CHK FOR ERROR 18
 026072  001017                          BNE     60$             ;BR IF NOT
                                        ;REPORT RCV'D DATA MISCOMPARE ERROR
 026074  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
 026102  112737  000000  002324          MOVB    #000,GDATA      ;SET EXPECTED DATA
 026110  117737  155320  002326          MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
 026116                                  GEDF    EM34,ERR10
                                                 ;       "DEVICE FATAL" ERROR # 57
 025116  104455                                                  TRAP    C$ERDF
 025120  000071                                                  .WORD   57
 025122  014502                                                  .WORD   EM34
 025124  020364                                                  .WORD   ERR10
 026126                                  ESCAPE  TST
 025126  104410                                                  TRAP    C$ESCAPE
 025130  000304                                                  .WORD   L10021-.
                                        ;CHK FOR ERROR 19
 026132  127727  155274  000023  60$:   CMPB    @#SEL6,#19.     ;CHK FOR ERROR 19
 026140  001006                          BNE     62$             ;BR IF NOT
                                        ;REPORT RERR NOT SET
 026142                                  GEDF    EM36,ERR10
                                                 ;       "DEVICE FATAL" ERROR # 58
 025142  104455                                                  TRAP    C$ERDF
 025144  000072                                                  .WORD   58
 025146  014551                                                  .WORD   EM36
 025150  020364                                                  .WORD   ERR10
 026152                                  ESCAPE  TST
 025152  104410                                                  TRAP    C$ESCAPE
```

```
      025154  000260                                                                  .WORD   L10021-.
4660                                ;CHK FOR ERROR 20
4661  025156  127727  155250  000024 62$:    CMPB    @#SEL6,#20.     ;CHK FOR ERROR 20
4662  025164  001006                BNE     64$             ;BR IF NOT
4663                                ;REPORT RDA NOT SET
4664  025166                        GEDF    EM75,ERR12
                                                      ;       "DEVICE FATAL" ERROR # 59
      025166  104455                                                                  TRAP    C$ERDF
      025170  000073                                                                  .WORD   59
      025172  015035                                                                  .WORD   EM75
      025174  020714                                                                  .WORD   ERR12
4665  025176                        ESCAPE  TST
      025176  104410                                                                  TRAP    C$ESCAPE
      025200  000234                                                                  .WORD   L10021-.
4666                                ;CHK FOR ERROR 21
4667  025202  127727  155224  000025 64$:    CMPB    @#SEL6,#21.     ;CHK FOR ERROR 21
4668  025210  001017                BNE     66$             ;BR IF NOT
4669                                ;REPORT RCV'D DATA MISCOMPARE ERROR
4670  025212  012737  000000  002336        MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
4671  025220  112737  000160  002324        MOVB    #160,GDATA      ;SET EXPECTED DATA
4672  025226  117737  155202  002326        MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
4673  025234                        GEDF    EM34,ERR10
                                                      ;       "DEVICE FATAL" ERROR # 60
      025234  104455                                                                  TRAP    C$ERDF
      025236  000074                                                                  .WORD   60
      025240  014502                                                                  .WORD   EM34
      025242  020364                                                                  .WORD   ERR10
4674  025244                        ESCAPE  TST
      025244  104410                                                                  TRAP    C$ESCAPE
      025246  000166                                                                  .WORD   L10021-.
4675                                ;CHK FOR ERROR 22
4676  025250  127727  155156  000026 66$:    CMPB    @#SEL6,#22.     ;CHK FOR ERROR 22
4677  025256  001006                BNE     68$             ;BR IF NOT
4678                                ;REPORT RDA NOT SET
4679  025260                        GEDF    EM75,ERR12
                                                      ;       "DEVICE FATAL" ERROR # 61
      025260  104455                                                                  TRAP    C$ERDF
      025262  000075                                                                  .WORD   61
      025264  015035                                                                  .WORD   EM75
      025266  020714                                                                  .WORD   ERR12
4680  025270                        ESCAPE  TST
      025270  104410                                                                  TRAP    C$ESCAPE
      025272  000142                                                                  .WORD   L10021-.
4681                                ;CHK FOR ERROR 23
4682  025274  127727  155132  000027 68$:    CMPB    @#SEL6,#23.     ;CHK FOR ERROR 23
4683  025302  001017                BNE     70$             ;BR IF NOT
4684                                ;REPORT RCV'D DATA MISCOMPARE ERROR
4685  025304  012737  000000  002336        MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
4686  025312  112737  000034  002324        MOVB    #034,GDATA      ;SET EXPECTED DATA
4687  025320  117737  155110  002326        MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
4688  025326                        GEDF    EM34,ERR10
                                                      ;       "DEVICE FATAL" ERROR # 62
      025326  104455                                                                  TRAP    C$ERDF
      025330  000076                                                                  .WORD   62
      025332  014502                                                                  .WORD   EM34
      025334  020364                                                                  .WORD   ERR10
4689  025336                        ESCAPE  TST
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
                025336  104410                                                          TRAP    C$ESCAPE
                025340  000074                                                          .WORD   L10021-.
4690                                    ;CHK FOR ERROR 24
4691            025342  127727  155064  000030  70$:    CMPB    @#SEL6,#24.     ;CHK FOR ERROR 24
4692            025350  001006                  BNE     72$             ;BR IF NOT
4693                                    ;REPORT RDA NOT SET
4694            025352                  GEDF    EM75,ERR12
                                                        ;       "DEVICE FATAL" ERROR # 63
                025352  104455                                                          TRAP    C$ERDF
                025354  000077                                                          .WORD   63
                025356  015035                                                          .WORD   EM75
                025360  020714                                                          .WORD   ERR12
4695            025362                  ESCAPE  TST
                025362  104410                                                          TRAP    C$ESCAPE
                025364  000050                                                          .WORD   L10021-.
4696                                    ;CHK FOR ERROR 25
4697            025366  127727  155040  000031  72$:    CMPB    @#SEL6,#25.     ;CHK FOR ERROR 25
4698            025374  001011                  BNE     74$             ;BR IF NOT
4699                                    ;REPORT CARRIER NOT CLEARED
4700            025376  012737  000001  002336          MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
4701            025404                  GEDF    EM81,ERR14
                                                        ;       "DEVICE FATAL" ERROR # 64
                025404  104455                                                          TRAP    C$ERDF
                025406  000100                                                          .WORD   64
                025410  015201                                                          .WORD   EM81
                025412  021146                                                          .WORD   ERR14
4702            025414                  ESCAPE  TST
                025414  104410                                                          TRAP    C$ESCAPE
                025416  000016                                                          .WORD   L10021-.
4703            025420  004737  004166  74$:    JSR     PC,GETWSR       ;GET CSR'S FOR PRINTOUT
4704                                    ;REPORT INVALID ERROR CODE FROM 6502
4705            025424                  GEDF    EM82,ERR3
                                                        ;       "DEVICE FATAL" ERROR # 65
                025424  104455                                                          TRAP    C$ERDF
                025426  000101                                                          .WORD   65
                025430  015225                                                          .WORD   EM82
                025432  020072                                                          .WORD   ERR3
4706
4707            025434                  90$:
4708            025434                  ENDTST
                025434                                                          L10021:
                025434  104401                                                          TRAP    C$ETST
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
   4710 025436                          MCODE:
   4711                                 ;
   4712                                 ;LINE#  LOC   CODE        LINE
   4713                                 ;
   4714                                 ;0001   0000                    .=$0200                 ;START OF MICROCODE FOR INTEGRAL MOD
EM TEST
   4715                                 ;0002   0200
   4716                                 ;0003   0200
   4717                                 ;0005   0200
   4718                                 ;0006   0200  ;*****************************************************************************
   4719                                 ;0007   0200  ; THIS IS THE 6502 MICROCODE WHICH IS LOADED INTO RAM AND EXECUTED FOR THE
   4720                                 ;0008   0200  ; PURPOSE OF TESTING THE INTEGRAL MODEM ON THE M8064, AT 56K BAUD. AFTER THE
   4721                                 ;0009   0200  ; LSI-11 PROGRAM DOES SOME INITIAL SETUP, IT TRANSFERS CONTROL TO THIS CODE
   4722                                 ;0010   0200  ; IN RAM, AND WAITS FOR COMPLETION OF THE TEST, AS INDICATED BY MRDY SET.
   4723                                 ;0011   0200  ; THIS CODE TRANSMITS, RECEIVES, AND CHECKS THE FOLLOWING CHARACTERS :
   4724                                 ;0012   0200  ; 2 SYNCH CHARACTERS, 5 DATA CHARACTERS 000, 125, 252, 377, 000, 2 CRC-16
   4725                                 ;0013   0200  ; CHARACTERS 160 AND 034, AND 2 TERMINATING SYNCHS. THE MESSAGE IS SENT USING
   4726                                 ;0014   0200  ; CHARACTER (DDCMP) MODE, THE SYNCH CHARACTER USED IS 226, STRIP SYNCH AND
   4727                                 ;0015   0200  ; IDLE MODES ARE SET, AND THE DATA CLOCK IS PROVIDED BY THE INTEGRAL MODEM.
   4728                                 ;0016   0200  ; ALL DATA AND CRC CHARACTERS ARE CHECKED AS THEY ARE RECEIVED, AND THE CRC
   4729                                 ;0017   0200  ; ERROR CHECK BIT IS CHECKED TO BE SET WITH RECEPTION OF THE LAST DATA
   4730                                 ;0018   0200  ; CHARACTER (000).
   4731                                 ;0019   0200  ;*****************************************************************************
   4732                                 ;0020   0200
   4733                                 ;0021   0200            ;EQUATES FOR BIT DEFINITIONS
   4734                                 ;0022   0200          BIT0    =@1
   4735                                 ;0023   0200          BIT1    =@2
   4736                                 ;0024   0200          BIT2    =@4
   4737                                 ;0025   0200          BIT3    =@10
   4738                                 ;0026   0200          BIT4    =@20
   4739                                 ;0027   0200          BIT5    =@40
   4740                                 ;0028   0200          BIT6    =@100
   4741                                 ;0029   0200          BIT7    =@200
   4742                                 ;0030   0200          BIT8    =@400
   4743                                 ;0031   0200          BIT9    =@1000
   4744                                 ;0032   0200          BIT10   =@2000
   4745                                 ;0033   0200          BIT11   =@4000
   4746                                 ;0034   0200          BIT12   =@10000
   4747                                 ;0035   0200          BIT13   =@20000
   4748                                 ;0036   0200          BIT14   =@40000
   4749                                 ;0037   0200          BIT15   =@100000
   4750                                 ;0038   0200
   4751                                 ;0039   0200
   4752                                 ;0040   0200
   4753                                 ;0041   0200            ;ADDRESS EQUATES FOR CSR REGISTERS
   4754                                 ;0042   0200          SEL0    =$10
   4755                                 ;0043   0200          BSEL0   =SEL0
   4756                                 ;0044   0200          BSEL1   =SEL0+1
   4757                                 ;0045   0200          SEL2    =SEL0+2
   4758                                 ;0046   0200          BSEL2   =SEL0+2
   4759                                 ;0047   0200          BSEL3   =SEL0+3
   4760                                 ;0048   0200          SEL4    =SEL0+4
   4761                                 ;0049   0200          BSEL4   =SEL0+4
   4762                                 ;0050   0200          BSEL5   =SEL0+5
   4763                                 ;0051   0200          SEL6    =SEL0+6
   4764                                 ;0052   0200          BSEL6   =SEL0+6
   4765                                 ;0053   0200          BSEL7   =SEL0+7
   4766                                 ;0054   0200
```

```
4767                                    ;0055  0200
4768                                    ;0056  0200
4769                                    ;0057  0200          ;VERSATILE INTERFACE ADAPTER REGISTER EQUATES
4770                                    ;0058  0200  OREGB   = $A000                    ;OUTPUT REGISTER B
4771                                    ;0059  0200  OREGA   = OREGB+1                 ;OUTPUT REGISTER A
4772                                    ;0060  0200  DDRB    = OREGB+2                 ;DATA DIRECTION REGISTER B
4773                                    ;0061  0200  DDRA    = OREGB+3                 ;DATA DIRECTION REGISTER A
4774                                    ;0062  0200  T1LL    = OREGB+6                 ;TIMER 1 LATCH LOW BITS
4775                                    ;0063  0200  T1LH    = OREGB+7                 ;TIMER 1 LATCH HIGH BITS
4776                                    ;0064  0200  ACR     = OREGB+$B                ;AUXILIARY CONTROL REGISTER
4777                                    ;0065  0200  PCR     = OREGB+$C                ;PERIPHERAL CONTROL REGISTER
4778                                    ;0066  0200
4779                                    ;0067  0200
4780                                    ;0068  0200
4781                                    ;0069  0200          ;VIA OUTPUT REGISTER B BIT EQUATES
4782                                    ;0070  0200  NULCLK  = BIT7
4783                                    ;0071  0200  RXEN    = BIT6
4784                                    ;0072  0200  TXEN    = BIT5
4785                                    ;0073  0200  DTR     = BIT4
4786                                    ;0074  0200  RTSND   = BIT3
4787                                    ;0075  0200  HDX     = BIT2
4788                                    ;0076  0200  TTLOOP  = BIT1
4789                                    ;0077  0200  PRESET  = BIT0
4790                                    ;0078  0200
4791                                    ;0079  0200
4792                                    ;0080  0200
4793                                    ;0081  0200          ;VIA OUTPUT REGISTER A BIT EQUATES
4794                                    ;0082  0200  RING    = BIT7
4795                                    ;0083  0200  CARIER  = BIT6
4796                                    ;0084  0200  MDMRDY  = BIT5
4797                                    ;0085  0200  BDRATE  = BIT4
4798                                    ;0086  0200  CTS     = BIT3
4799                                    ;0087  0200  TM      = BIT2
4800                                    ;0088  0200  RCVDAT  = BIT1
4801                                    ;0089  0200  UMAINT  = BIT0
4802                                    ;0090  0200
4803                                    ;0091  0200
4804                                    ;0092  0200          ;USYRT REGISTER ADDRESS EQUATES
4805                                    ;0093  0200  RXDB    = $A100
4806                                    ;0094  0200  RDSR    = RXDB+1
4807                                    ;0095  0200  TXDB    = RXDB+2
4808                                    ;0096  0200  TDSR    = RXDB+3
4809                                    ;0097  0200  SAR     = RXDB+4
4810                                    ;0098  0200  PCSAR   = RXDB+5
4811                                    ;0099  0200  PCTLR   = RXDB+7
4812                                    ;0100  0200
4813                                    ;0101  0200
4814                                    ;0102  0200
4815                                    ;0103  0200          ;USYRT TDSR REGISTER BIT EQUATES
4816                                    ;0104  0200  TEOM    = BIT1
4817                                    ;0105  0200  TSOM    = BIT0
4818                                    ;0106  0200
4819                                    ;0107  0200
4820                                    ;0108  0200
4821                                    ;0109  0200          ;USYRT RDSR BIT EQUATES
4822                                    ;0110  0200  RERR    = BIT7
4823                                    ;0111  0200
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
 4824                                   ;0112  0200
 4825                                   ;0113  0200
 4826                                   ;0114  0200              ;USYRT STATUS REGISTER EQUATES
 4827                                   ;0115  0200              USTATR = $A400
 4828                                   ;0116  0200              RDA    = BIT7
 4829                                   ;0117  0200              TBMT   = BIT6
 4830                                   ;0118  0200              RXACT  = BIT5
 4831                                   ;0119  0200              RSA    = BIT4
 4832                                   ;0120  0200              TSO    = BIT3
 4833                                   ;0121  0200              TXACT  = BIT2
 4834                                   ;0122  0200              TXUERR = BIT1
 4835                                   ;0123  0200              SYNFLG = BIT0
 4836                                   ;0124  0200
 4837                                   ;0125  0200
 4838                                   ;0126  0200
 4839                                   ;0127  0200              ;MISCELLANEOUS EQUATES
 4840                                   ;0128  0200              SYNCH  = @226
 4841                                   ;0129  0200
 4842                                   ;0130  0200
 4843                                   ;0131  0200  A0 00               LDY     #0
 4844 025436   240   000                        .BYTE   240,000
 4845                                   ;0132  0202  84 16               STY     BSEL6          ;CLEAR BSEL6
 4846 025440   204   026                        .BYTE   204,026
 4847                                   ;0133  0204  84 17               STY     BSEL7          ;CLEAR BSEL7
 4848 025442   204   027                        .BYTE   204,027
 4849                                   ;0134  0206              ;TURN ON THE USYRT, CLOCK
 4850                                   ;0135  0206  A2 60               LDX     #TXEN!RXEN     ;ASSERT TXEN,RXEN,RTS,DTR
 4851 025444   242   140                        .BYTE   242,140
 4852                                   ;0136  0208  8E 00 A0            STX     OREGB          ;  AND RELEASE INT MODEM RESET
 4853 025446   216   000   240                  .BYTE   216,000,240
 4854                                   ;0137  020B  A2 00               LDX     #0             ;INIT TBMT TIME-OUT COUNTER
 4855 025451   242   000                        .BYTE   242,000
 4856                                   ;0138  020D  2C 00 A4            BIT     USTATR         ;SEE IF TBMT SET
 4857 025453   054   000   244                  .BYTE   054,000,244
 4858                                   ;0139  0210  70 08               BVS     .+10           ;BR IF TBMT SET
 4859 025456   160   010                        .BYTE   160,010
 4860                                   ;0140  0212  E8                  INX                    ;INCREMENT TIME-OUT COUNTER
 4861 025460   350                              .BYTE   350
 4862                                   ;0141  0213  D0 F8               BNE     .-6            ;BR IF NO TIME-OUT
 4863 025461   320   370                        .BYTE   320,370
 4864                                   ;0142  0215                      ; ***  ERROR 1  ***
 4865                                   ;0143  0215  A0 01               LDY     #1             ;SET CODE FOR TBMT TIME-OUT ERROR
 4866 025463   240   001                        .BYTE   240,001
 4867                                   ;0144  0217  4C 90 03            JMP     A100           ;GO TAKE ERROR EXIT
 4868 025465   114   220   003                  .BYTE   114,220,003
 4869                                   ;0145  021A              ;LOAD FIRST SYNCH CHAR INTO TRANSMITTER
 4870                                   ;0146  021A  A2 96               LDX     #SYNCH         ;LOAD FIRST SYNCH CHAR
 4871 025470   242   226                        .BYTE   242,226
 4872                                   ;0147  021C  8E 02 A1            STX     TXDB
 4873 025472   216   002   241                  .BYTE   216,002,241
 4874                                   ;0148  021F  A2 00               LDX     #0             ;INIT TBMT TIME OUT COUNTER
 4875 025475   242   000                        .BYTE   242,000
 4876                                   ;0149  0221  2C 00 A4            BIT     USTATR         ;SEE IF TBMT SET
 4877 025477   054   000   244                  .BYTE   054,000,244
 4878                                   ;0150  0224  70 08               BVS     .+10           ;BR IF TBMT SET
 4879 025502   160   010                        .BYTE   160,010
 4880                                   ;0151  0226  E8                  INX                    ;INCREMENT TIME-OUT COUNTER
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
4881 025504  350                          .BYTE   350
4882                              ;0152  0227  D0 F8          BNE     *-6             ;BR IF NO TIME-OUT
4883 025505  320  370                     .BYTE   320,370
4884                              ;0153  0229                  ; *** ERROR 2 ***
4885                              ;0154  0229  A0 02          LDY     #2              ;SET CODE FOR TBMT TIME-OUT ERROR
4886 025507  240  002                     .BYTE   240,002
4887                              ;0155  022B  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
4888 025511  114  220  003                .BYTE   114,220,003
4889                              ;0156  022E                  ;LOAD SECOND SYNCH CHAR INTO TRANSMITTER
4890                              ;0157  022E  A2 96          LDX     #SYNCH          ;LOAD SECOND SYNCH CHAR
4891 025514  242  226                     .BYTE   242,226
4892                              ;0158  0230  8E 02 A1       STX     TXDB
4893 025516  216  002  241                .BYTE   216,002,241
4894                              ;0159  0233  A2 00          LDX     #0              ;INIT TBMT TIME-OUT COUNTER
4895 025521  242  000                     .BYTE   242,000
4896                              ;0160  0235  2C 00 A4       BIT     USTATR          ;SEE IF TBMT SET
4897 025523  054  000  244                .BYTE   054,000,244
4898                              ;0161  0238  70 08          BVS     *+10            ;BR IF TBMT SET
4899 025526  160  010                     .BYTE   160,010
4900                              ;0162  023A  E8             INX                     ;INCREMENT TIME-OUT COUNTER
4901 025530  350                          .BYTE   350
4902                              ;0163  023B  D0 F8          BNE     *-6             ;BR IF NO TIME-OUT
4903 025531  320  370                     .BYTE   320,370
4904                              ;0164  023D                  ; *** ERROR 3 ***
4905                              ;0165  023D  A0 03          LDY     #3              ;SET CODE FOR TBMT TIME-OUT ERROR
4906 025533  240  003                     .BYTE   240,003
4907                              ;0166  023F  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
4908 025535  114  220  003                .BYTE   114,220,003
4909                              ;0167  0242                  ;CHECK FOR CARRIER SET
4910                              ;0168  0242  2C 01 A0       BIT     OREGA           ;SEE IF CARRIER SET YET
4911 025540  054  001  240                .BYTE   054,001,240
4912                              ;0169  0245  70 05          BVS     *+7             ;BR IF CARRIER SET
4913 025543  160  005                     .BYTE   160,005
4914                              ;0170  0247                  ; *** ERROR 4 ***
4915                              ;0171  0247  A0 04          LDY     #4              ;SET CODE FOR CARRIER NOT SET ERROR
4916 025545  240  004                     .BYTE   240,004
4917                              ;0172  0249  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
4918 025547  114  220  003                .BYTE   114,220,003
4919                              ;0173  024C                  ;LOAD TRANSMITTER WITH 000 CHAR
4920                              ;0174  024C  A2 00          LDX     #000            ;CLEAR TSOM
4921 025552  242  000                     .BYTE   242,000
4922                              ;0175  024E  8E 03 A1       STX     TDSR
4923 025554  216  003  241                .BYTE   216,003,241
4924                              ;0176  0251  8E 02 A1       STX     TXDB            ;LOAD 000 CHAR
4925 025557  216  002  241                .BYTE   216,002,241
4926                              ;0177  0254  A2 00          LDX     #0              ;INIT TBMT TIME-OUT COUNTER
4927 025562  242  000                     .BYTE   242,000
4928                              ;0178  0256  2C 00 A4       BIT     USTATR          ;SEE IF TBMT SET
4929 025564  054  000  244                .BYTE   054,000,244
4930                              ;0179  0259  70 08          BVS     *+10            ;BR IF TBMT SET
4931 025567  160  010                     .BYTE   160,010
4932                              ;0180  025B  E8             INX                     ;INCREMENT TIME-OUT COUNTER
4933 025571  350                          .BYTE   350
4934                              ;0181  025C  D0 F8          BNE     *-6             ;BR IF NO TIME-OUT
4935 025572  320  370                     .BYTE   320,370
4936                              ;0182  025E                  ; *** ERROR 5 ***
4937                              ;0183  025E  A0 05          LDY     #5              ;SET CODE FOR TBMT TIME-OUT ERROR
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
4938 025574      240      005                    .BYTE   240,005
4939                                    ;0184  0260  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
4940 025576      114      220      003           .BYTE   114,220,003
4941                                    ;0185  0263                    ;LOAD TRANSMITTER WITH 125 CHAR
4942                                    ;0186  0263  A2 55             LDX     #@125           ;LOAD 125 CHAR
4943 025601      242      125                    .BYTE   242,125
4944                                    ;0187  0265  8E 02 A1          STX     TXDB
4945 025603      216      002      241           .BYTE   216,002,241
4946                                    ;0188  0268  A2 00             LDX     #0              ;INIT TBMT TIME-OUT COUNTER
4947 025606      242      000                    .BYTE   242,000
4948                                    ;0189  026A  2C 00 A4          BIT     USTATR          ;SEE IF TBMT SET
4949 025610      054      000      244           .BYTE   054,000,244
4950                                    ;0190  026D  70 08             BVS     .+10            ;BR IF TBMT SET
4951 025613      160      010                    .BYTE   160,010
4952                                    ;0191  026F  E8                INX                     ;INCREMENT TIME-OUT COUNTER
4953 025615      350                             .BYTE   350
4954                                    ;0192  0270  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
4955 025616      320      370                    .BYTE   320,370
4956                                    ;0193  0272                    ; *** ERROR 6 ***
4957                                    ;0194  0272  A0 06             LDY     #6              ;SET CODE FOR TBMT TIME-OUT ERROR
4958 025620      240      006                    .BYTE   240,006
4959                                    ;0195  0274  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
4960 025622      114      220      003           .BYTE   114,220,003
4961                                    ;0196  0277                    ;LOAD TRANSMITTER WITH 252 CHAR
4962                                    ;0197  0277  A2 AA             LDX     #@252           ;LOAD 252 CHAR
4963 025625      242      252                    .BYTE   242,252
4964                                    ;0198  0279  8E 02 A1          STX     TXDB
4965 025627      216      002      241           .BYTE   216,002,241
4966                                    ;0199  027C  A2 00             LDX     #0              ;INIT TBMT TIME-OUT COUNTER
4967 025632      242      000                    .BYTE   242,000
4968                                    ;0200  027E  2C 00 A4          BIT     USTATR          ;SEE IF TBMT SET
4969 025634      054      000      244           .BYTE   054,000,244
4970                                    ;0201  0281  70 08             BVS     .+10            ;BR IF TBMT SET
4971 025637      160      010                    .BYTE   160,010
4972                                    ;0202  0283  E8                INX                     ;INCREMENT TIME-OUT COUNTER
4973 025641      350                             .BYTE   350
4974                                    ;0203  0284  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
4975 025642      320      370                    .BYTE   320,370
4976                                    ;0204  0286                    ; *** ERROR 7 ***
4977                                    ;0205  0286  A0 07             LDY     #7              ;SET CODE FOR TBMT TIME-OUT ERROR
4978 025644      240      007                    .BYTE   240,007
4979                                    ;0206  0288  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
4980 025646      114      220      003           .BYTE   114,220,003
4981                                    ;0207  028B                    ;LOAD TRANSMITTER WITH 377 CHAR AND END OF MESSAGE
4982                                    ;0208  028B  A2 FF             LDX     #@377           ;LOAD 377 CHAR
4983 025651      242      377                    .BYTE   242,377
4984                                    ;0209  028D  8E 02 A1          STX     TXDB
4985 025653      216      002      241           .BYTE   216,002,241
4986                                    ;0210  0290  A2 00             LDX     #0              ;INIT TBMT TIME-OUT COUNTER
4987 025656      242      000                    .BYTE   242,000
4988                                    ;0211  0292  2C 00 A4          BIT     USTATR          ;SEE IF TBMT SET
4989 025660      054      000      244           .BYTE   054,000,244
4990                                    ;0212  0295  70 08             BVS     .+10            ;BR IF TBMT SET
4991 025663      160      010                    .BYTE   160,010
4992                                    ;0213  0297  E8                INX                     ;INCREMENT TIME-OUT COUNTER
4993 025665      350                             .BYTE   350
4994                                    ;0214  0298  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
```

TEST 2 -- INTEGRAL MODEM INTERFACE TEST

```
 4995 025666  320  370                    .BYTE   320,370
 4996                          ;0215  029A                    ; *** ERROR 8 ***
 4997                          ;0216  029A  A0 08             LDY     #8              ;SET CODE FOR TBMT TIME-OUT ERROR
 4998 025670  240  010                    .BYTE   240,010
 4999                          ;0217  029C  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
 5000 025672  114  220  003               .BYTE   114,220,003
 5001                          ;0218  029F                    ;LOAD TRANSMITTER WITH 000 CHAR
 5002                          ;0219  029F  A2 00             LDX     #000            ;LOAD 000 CHAR
 5003 025675  242  000                    .BYTE   242,000
 5004                          ;0220  02A1  8E 02 A1          STX     TXDB
 5005 025677  216  002  241               .BYTE   216,002,241
 5006                          ;0221  02A4  A2 00             LDX     #0              ;INIT RDA TIME-OUT COUNTER
 5007 025702  242  000                    .BYTE   242,000
 5008                          ;0222  02A6  2C 00 A4          BIT     USTATR          ;SEE IF RDA SET
 5009 025704  054  000  244               .BYTE   054,000,244
 5010                          ;0223  02A9  30 08             BMI     .+10            ;BR IF RDA SET
 5011 025707  060  010                    .BYTE   060,010
 5012                          ;0224  02AB  E8                INX                     ;INCREMENT TIME-OUT COUNTER
 5013 025711  350                         .BYTE   350
 5014                          ;0225  02AC  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
 5015 025712  320  370                    .BYTE   320,370
 5016                          ;0226  02AE                    ; *** ERROR 9 ***
 5017                          ;0227  02AE  A0 09             LDY     #9              ;SET CODE FOR RDA TIME-OUT ERROR
 5018 025714  240  011                    .BYTE   240,011
 5019                          ;0228  02B0  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
 5020 025716  114  220  003               .BYTE   114,220,003
 5021                          ;0229  02B3                    ;READ AND CHECK 000 CHAR
 5022                          ;0230  02B3  AD 00 A1          LDA     RXDB            ;READ RECEIVER BUFFER
 5023 025721  255  000  241               .BYTE   255,000,241
 5024                          ;0231  02B6  C9 00             CMP     #0000           ;CHK FOR 000
 5025 025724  311  000                    .BYTE   311,000
 5026                          ;0232  02B8  F0 05             BEQ     .+7             ;BR IF 000
 5027 025726  360  005                    .BYTE   360,005
 5028                          ;0233  02BA                    ; *** ERROR 10 ***
 5029                          ;0234  02BA  A0 0A             LDY     #10             ;SET CODE FOR DATA MISCOMPARE ERROR
 5030 025730  240  012                    .BYTE   240,012
 5031                          ;0235  02BC  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
 5032 025732  114  220  003               .BYTE   114,220,003
 5033                          ;0236  02BF  A2 00             LDX     #0              ;INIT RDA TIME-OUT COUNTER
 5034 025735  242  000                    .BYTE   242,000
 5035                          ;0237  02C1  2C 00 A4          BIT     USTATR          ;SEE IF RDA SET
 5036 025737  054  000  244               .BYTE   054,000,244
 5037                          ;0238  02C4  30 08             BMI     .+10            ;BR IF RDA SET
 5038 025742  060  010                    .BYTE   060,010
 5039                          ;0239  02C6  E8                INX                     ;INCREMENT TIME-OUT COUNTER
 5040 025744  350                         .BYTE   350
 5041                          ;0240  02C7  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
 5042 025745  320  370                    .BYTE   320,370
 5043                          ;0241  02C9                    ; *** ERROR 11 ***
 5044                          ;0242  02C9  A0 0B             LDY     #11             ;SET CODE FOR RDA TIME-OUT ERROR FOR
 5045 025747  240  013                    .BYTE   240,013
 5046                          ;0243  02CB  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
 5047 025751  114  220  003               .BYTE   114,220,003
 5048                          ;0244  02CE  A2 02             LDX     #TEOM           ;SET TEOM TO TERMINATE MSG
 5049 025754  242  002                    .BYTE   242,002
 5050                          ;0245  02D0  8E 03 A1          STX     TDSR
 5051 025756  216  003  241               .BYTE   216,003,241
```

```
5052                                    ;0246  02D3                    ;READ AND CHECK 125 CHAR
5053                                    ;0247  02D3  AD 00 A1           LDA     RXDB            ;READ RECEIVER BUFFER
5054 025761    255     000     241              .BYTE   255,000,241
5055                                    ;0248  02D6  C9 55              CMP     #@125           ;CHK FOR 125
5056 025764    311     125                      .BYTE   311,125
5057                                    ;0249  02D8  F0 05              BEQ     *+7             ;BR IF 125
5058 025766    360     005                      .BYTE   360,005
5059                                    ;0250  02DA                     ; *** ERROR 12 ***
5060                                    ;0251  02DA  A0 0C              LDY     #12             ;SET CODE FOR DATA MISCOMPARE ERROR
5061 025770    240     014                      .BYTE   240,014
5062                                    ;0252  02DC  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5063 025772    114     220     003              .BYTE   114,220,003
5064                                    ;0253  02DF  A2 00              LDX     #0              ;INIT RDA TIME-OUT COUNTER
5065 025775    242     000                      .BYTE   242,000
5066                                    ;0254  02E1  2C 00 A4           BIT     USTATR          ;SEE IF RDA SET
5067 025777    054     000     244              .BYTE   054,000,244
5068                                    ;0255  02E4  30 08              BMI     *+10            ;BR IF RDA SET
5069 026002    060     010                      .BYTE   060,010
5070                                    ;0256  02E6  E8                 INX                     ;INCREMENT TIME-OUT COUNTER
5071 026004    350                              .BYTE   350
5072                                    ;0257  02E7  D0 F8              BNE     *-6             ;BR IF NO TIME-OUT
5073 026005    320     370                      .BYTE   320,370
5074                                    ;0258  02E9                     ; *** ERROR 13 ***
5075                                    ;0259  02E9  A0 0D              LDY     #13             ;SET CODE FOR RDA TIME-OUT ERROR
5076 026007    240     015                      .BYTE   240,015
5077                                    ;0260  02EB  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5078 026011    114     220     003              .BYTE   114,220,003
5079                                    ;0261  02EE                     ;READ AND CHECK 252 CHAR
5080                                    ;0262  02EE  AD 00 A1           LDA     RXDB            ;READ RECEIVER BUFFER
5081 026014    255     000     241              .BYTE   255,000,241
5082                                    ;0263  02F1  C9 AA              CMP     #@252           ;CHK FOR 252
5083 026017    311     252                      .BYTE   311,252
5084                                    ;0264  02F3  F0 05              BEQ     *+7             ;BR IF 252
5085 026021    360     005                      .BYTE   360,005
5086                                    ;0265  02F5                     ; *** ERROR 14 ***
5087                                    ;0266  02F5  A0 0E              LDY     #14             ;SET CODE FOR DATA MISCOMPARE ERROR
5088 026023    240     016                      .BYTE   240,016
5089                                    ;0267  02F7  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5090 026025    114     220     003              .BYTE   114,220,003
5091                                    ;0268  02FA  A2 00              LDX     #0              ;INIT RDA TIME-OUT COUNTER
5092 026030    242     000                      .BYTE   242,000
5093                                    ;0269  02FC  2C 00 A4           BIT     USTATR          ;SEE IF RDA SET
5094 026032    054     000     244              .BYTE   054,000,244
5095                                    ;0270  02FF  30 08              BMI     *+10            ;BR IF RDA SET
5096 026035    060     010                      .BYTE   060,010
5097                                    ;0271  0301  E8                 INX                     ;INCREMENT TIME-OUT COUNTER
5098 026037    350                              .BYTE   350
5099                                    ;0272  0302  D0 F8              BNE     *-6             ;BR IF NO TIME-OUT
5100 026040    320     370                      .BYTE   320,370
5101                                    ;0273  0304                     ; *** ERROR 15 ***
5102                                    ;0274  0304  A0 0F              LDY     #15             ;SET CODE FOR RDA TIME-OUT ERROR FOR
5103 026042    240     017                      .BYTE   240,017
5104                                    ;0275  0306  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5105 026044    114     220     003              .BYTE   114,220,003
5106                                    ;0276  0309                     ;READ AND CHECK 377 CHAR
5107                                    ;0277  0309  AD 00 A1           LDA     RXDB            ;READ RECEIVER BUFFER
5108 026047    255     000     241              .BYTE   255,000,241
```

```
5109                                    ;0278  030C  C9 FF              CMP     #@377           ;CHK FOR 377
5110 026052     311     377                     .BYTE   311,377
5111                                    ;0279  030E  F0 05              BEQ     *+7             ;BR IF 377
5112 026054     360     005                     .BYTE   360,005
5113                                    ;0280  0310                     ; *** ERROR 16 ***
5114                                    ;0281  0310  A0 10              LDY     #16             ;SET CODE FOR DATA MISCOMPARE ERROR
5115 026056     240     020                     .BYTE   240,020
5116                                    ;0282  0312  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5117 026060     114     220     003             .BYTE   114,220,003
5118                                    ;0283  0315  A2 00              LDX     #0              ;INIT RDA TIME-OUT COUNTER
5119 026063     242     000                     .BYTE   242,000
5120                                    ;0284  0317  2C 00 A4           BIT     USTATR          ;SEE IF RDA SET
5121 026065     054     000     244             .BYTE   054,000,244
5122                                    ;0285  031A  30 08              BMI     *+10            ;BR IF RDA SET
5123 026070     060     010                     .BYTE   060,010
5124                                    ;0286  031C  E8                 INX                     ;INCREMENT TIME-OUT COUNTER
5125 026072     350                             .BYTE   350
5126                                    ;0287  031D  D0 F8              BNE     *-6             ;BR IF NO TIME-OUT
5127 026073     320     370                     .BYTE   320,370
5128                                    ;0288  031F                     ; *** ERROR 17 ***
5129                                    ;0289  031F  A0 11              LDY     #17             ;SET CODE FOR RDA TIME-OUT ERROR
5130 026075     240     021                     .BYTE   240,021
5131                                    ;0290  0321  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5132 026077     114     220     003             .BYTE   114,220,003
5133                                    ;0291  0324                     ;READ AND CHECK 000 CHAR
5134                                    ;0292  0324  AD 00 A1           LDA     RXDB            ;READ RECEIVER BUFFER
5135 026102     255     000     241             .BYTE   255,000,241
5136                                    ;0293  0327  C9 00              CMP     #@000           ;CHK FOR 000
5137 026105     311     000                     .BYTE   311,000
5138                                    ;0294  0329  F0 05              BEQ     *+7             ;BR IF 000
5139 026107     360     005                     .BYTE   360,005
5140                                    ;0295  032B                     ; *** ERROR 18 ***
5141                                    ;0296  032B  A0 12              LDY     #18             ;SET CODE FOR DATA MISCOMPARE ERROR
5142 026111     240     022                     .BYTE   240,022
5143                                    ;0297  032D  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5144 026113     114     220     003             .BYTE   114,220,003
5145                                    ;0298  0330  AE 01 A1           LDX     RDSR            ;CHECK FOR RERR BIT SET
5146 026116     256     001     241             .BYTE   256,001,241
5147                                    ;0299  0333  30 05              BMI     *+7             ;BR IF RERR BIT SET (NO CRC ERROR)
5148 026121     060     005                     .BYTE   060,005
5149                                    ;0300  0335                     ; *** ERROR 19 ***
5150                                    ;0301  0335  A0 13              LDY     #19
5151 026123     240     023                     .BYTE   240,023
5152                                    ;0302  0337  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5153 026125     114     220     003             .BYTE   114,220,003
5154                                    ;0303  033A  A2 00              LDX     #0              ;INIT RDA TIME-OUT COUNTER
5155 026130     242     000                     .BYTE   242,000
5156                                    ;0304  033C  2C 00 A4           BIT     USTATR          ;SEE IF RDA SET
5157 026132     054     000     244             .BYTE   054,000,244
5158                                    ;0305  033F  30 08              BMI     *+10            ;BR IF RDA SET
5159 026135     060     010                     .BYTE   060,010
5160                                    ;0306  0341  E8                 INX                     ;INCREMENT TIME-OUT COUNTER
5161 026137     350                             .BYTE   350
5162                                    ;0307  0342  D0 F8              BNE     *-6             ;BR IF NO TIME-OUT
5163 026140     320     370                     .BYTE   320,370
5164                                    ;0308  0344                     ; *** ERROR 20 ***
5165                                    ;0309  0344  A0 14              LDY     #20             ;SET CODE FOR RDA TIME-OUT ERROR
```

```
5166 026142  240  024                    .BYTE   240,024
5167                          ;0310  0346  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5168 026144  114  220  003               .BYTE   114,220,003
5169                          ;0311  0349                    ;READ AND CHECK FIRST CRC CHAR (160)
5170                          ;0312  0349  AD 00 A1          LDA     RXDB            ;READ RECEIVER BUFFER
5171 026147  255  000  241               .BYTE   255,000,241
5172                          ;0313  034C  C9 70             CMP     #160            ;CHK FOR 160
5173 026152  311  160                    .BYTE   311,160
5174                          ;0314  034E  F0 05             BEQ     .+7             ;BR IF 160
5175 026154  360  005                    .BYTE   360,005
5176                          ;0315  0350                    ; ***  ERROR 21  ***
5177                          ;0316  0350  A0 15             LDY     #21             ;SET CODE FOR DATA MISCOMPARE ERROR
5178 026156  240  025                    .BYTE   240,025
5179                          ;0317  0352  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5180 026160  114  220  003               .BYTE   114,220,003
5181                          ;0318  0355  A2 00             LDX     #0              ;INIT RDA TIME-OUT COUNTER
5182 026163  242  000                    .BYTE   242,000
5183                          ;0319  0357  2C 00 A4          BIT     USTATR          ;SEE IF RDA SET
5184 026165  054  000  244               .BYTE   054,000,244
5185                          ;0320  035A  30 08             BMI     .+10            ;BR IF RDA SET
5186 026170  060  010                    .BYTE   060,010
5187                          ;0321  035C  E8                INX                     ;INCREMENT TIME OUT COUNTER
5188 026172  350                         .BYTE   350
5189                          ;0322  035D  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
5190 026173  320  370                    .BYTE   320,370
5191                          ;0323  035F                    ; ***  ERROR 22  ***
5192                          ;0324  035F  A0 16             LDY     #22             ;SET CODE FOR RDA TIME-OUT ERROR
5193 026175  240  026                    .BYTE   240,026
5194                          ;0325  0361  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5195 026177  114  220  003               .BYTE   114,220,003
5196                          ;0326  0364                    ;READ AND CHECK 2ND CRC CHAR (034)
5197                          ;0327  0364  AD 00 A1          LDA     RXDB            ;READ RECEIVER BUFFER
5198 026202  255  000  241               .BYTE   255,000,241
5199                          ;0328  0367  C9 1C             CMP     #034            ;CHK FOR 034
5200 026205  311  034                    .BYTE   311,034
5201                          ;0329  0369  F0 05             BEQ     .+7             ;BR IF 034
5202 026207  360  005                    .BYTE   360,005
5203                          ;0330  036B                    ; ***  ERROR 23  ***
5204                          ;0331  036B  A0 17             LDY     #23             ;SET CODE FOR DATA MISCOMPARE ERROR
5205 026211  240  027                    .BYTE   240,027
5206                          ;0332  036D  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5207 026213  114  220  003               .BYTE   114,220,003
5208                          ;0333  0370  A2 00             LDX     #0              ;INIT RDA TIME OUT COUNTER
5209 026216  242  000                    .BYTE   242,000
5210                          ;0334  0372  2C 00 A4          BIT     USTATR          ;SEE IF RDA SET
5211 026220  054  000  244               .BYTE   054,000,244
5212                          ;0335  0375  30 08             BMI     .+10            ;BR IF RDA SET
5213 026223  060  010                    .BYTE   060,010
5214                          ;0336  0377  E8                INX                     ;INCREMENT TIME-OUT COUNTER
5215 026225  350                         .BYTE   350
5216                          ;0337  0378  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
5217 026226  320  370                    .BYTE   320,370
5218                          ;0338  037A                    ; ***  ERROR 24  ***
5219                          ;0339  037A  A0 18             LDY     #24             ;SET CODE FOR RDA TIME-OUT ERROR
5220 026230  240  030                    .BYTE   240,030
5221                          ;0340  037C  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5222 026232  114  220  003               .BYTE   114,220,003
```

```
5223                                    ;0341  037F                    ;DROP RTS, CHECK FOR CARRIER TO DROP
5224                                    ;0342  037F  A2 68               LDX     #TXEN!RXEN!RTSND  ;DE-ASSERT RTS
5225 026235      242     150                    .BYTE   242,150
5226                                    ;0343  0381  8E 00 A0           STX     OREGB
5227 026237      216     000     240            .BYTE   216,000,240
5228                                    ;0344  0384  A2 00              LDX     #0                ;INIT CARRIER DROP TIME-OUT COUNTER
5229 026242      242     000                    .BYTE   242,000
5230                                    ;0345  0386  2C 01 A0           BIT     OREGA             ;SEE IF CARRIER CLEARED
5231 026244      054     001     240            .BYTE   054,001,240
5232                                    ;0346  0389  50 05              BVC     .+7               ;BR IF CARRIER CLEARED
5233 026247      120     005                    .BYTE   120,005
5234                                    ;0347  038B  E8                 INX                       ;INCREMENT TIME-OUT COUNTER
5235 026251      350                            .BYTE   350
5236                                    ;0348  038C  D0 F8              BNE     .-6               ;BR IF NO TIME-OUT
5237 026252      320     370                    .BYTE   320,370
5238                                    ;0349  038E                     ; *** ERROR 25 ***
5239                                    ;0350  038E  A0 19              LDY     #25               ;SET CODE FOR CARRIER DROP TIME-OUT
5240 026254      240     031                    .BYTE   240,031
5241                                    ;0351  0390                     ;COME HERE FOR EXIT
5242                                    ;0352  0390  84 16          A100 STY     BSEL6             ;PUT ERROR NO. (IF ANY) INTO BSEL6
5243 026256      204     026                    .BYTE   204,026
5244                                    ;0353  0392  85 17              STA     BSEL7             ;PUT BAD DATA (IF ANY) INTO BSEL7
5245 026260      205     027                    .BYTE   205,027
5246                                    ;0354  0394  60                 RTS                       ;RETURN CONTROL TO LSI-11 PROGRAM
5247 026262      140                            .BYTE   140
5248                                    ;0355  0395
5249                                    ;0356  0395
5250                                    ;
5251                                    ;
5252                                    ;ERRORS = 0000
5253                                    ;
5254 026263                             ENDCOD:
5255                                    .EVEN
```

TEST 3 -- DATA TEST -- BCP XLB CRC-16

```
5273                                    .SBTTL  TEST 3 -- DATA TEST -- BCP XLB CRC-16

                                        ;+*****************************************************************************
                                        ;*
                                        ;*      TEST 3 -- DATA TEST -- BCP XLB CRC-16
                                        ;*
                                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                                        ;*  RECEIVE  IN  BCP  MODE  WITH  CRC-16  ERROR  DETECTION THE FOLLOWING
                                        ;*  MESSAGE:
                                        ;*
                                        ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
                                        ;*     357 337 277 177
                                        ;*
                                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
                                        ;*  THREE  TIMES WITH CRC'S FOLLOWING EACH ONE. THE LAST TRANSMISSION OF
                                        ;*  THE  CRC  WILL BE FOLLOWED BY SEVERAL SYNC CHARACTERS BEFORE DROPPING
                                        ;*  TXE & RXE. 8-BIT CHARACTER LENGTHS ARE ALSO UTILIZED.
                                        ;*
                                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                                        ;*
                                        ;-*****************************************************************************
                                        ;
                                        ;       BGNTST
     026264                                                                                     T3::
5274 026264  012737  000003  002526            MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
5275 026272  004737  005376                    JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
5276 026276  042737  001000  026354            BIC     #NOLOOP,1$      ;
5277 026304  012737  000011  027126            MOV     #9.,3$          ;*INIT ENTRAN COUNT/TTLOOP STATUS
5278 026312  023727  002470  000000            CMP     BRDTYP,#0       ;IS THIS AN M8064?
5279 026320  001412                           BEQ     2$              ; YES: USE TTLOOP (NOT XLB).
5280 026322  023727  002472  000004            CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
5281 026330  001406                           BEQ     2$              ;BR IF NO
5282 026332  052737  001000  026354            BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
5283 026340  052737  100000  027126            BIS     #BIT15,3$       ;      AND SET MSB OF ENTRAN STATUS (NOLOOP)
5284                                    ;-----------------------------------------------------------------------------
5285 026346  004537  007234            2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
5286 026352  065626                           DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
5287 026354  000000            1$:     0                               ;USE 8 BIT CHARS
5288 026356  103003                           BCC     .+8.            ;BR IF NO ERROR
5289 026360                                    ERROR                   ;REPORT STACKED ERROR
     026360  104460                                                                     TRAP    C$ERROR
5290 026362                                    ESCAPE  TST             ;SKIP TO END OF TEST
     026362  104410                                                                     TRAP    C$ESCAPE
     026364  000554                                                                     .WORD   L10022-.
5291
5292 026366  004537  010136                    JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
5293 026372  000226                           SYNCH
5294 026374  000007                           7.
5295 026376  103003                           BCC     .+8.            ;BR IF NO ERROR
5296 026400                                    ERROR                   ;REPORT STACKED ERROR
     026400  104460                                                                     TRAP    C$ERROR
5297 026402                                    ESCAPE  TST             ;SKIP TO END OF TEST
     026402  104410                                                                     TRAP    C$ESCAPE
     026404  000534                                                                     .WORD   L10022-.
5298
5299 026406  004537  010136                    JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
```

```
5300 026412  000226                    SYNCH
5301 026414  000010                    8.
5302 026416  103003                    BCC     .+8.            ;BR IF NO ERROR
5303 026420                            ERROR                   ;REPORT STACKED ERROR
     026420  104460                                                    TRAP    C$ERROR
5304 026422                            ESCAPE  TST             ;SKIP TO END OF TEST
     026422  104410                                                    TRAP    C$ESCAPE
     026424  000514                                                    .WORD   L10022-.
5305
5306 026426  004537  010250            JSR     R5,TXCTRL       ;CLEAR TSOM
5307 026432  000000                    000
5308 026434  000000                    0
5309
5310 026436  004537  010136            JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
5311 026442  000125                    125
5312 026444  000010                    8.
5313 026446  103003                    BCC     .+8.            ;BR IF NO ERROR
5314 026450                            ERROR                   ;REPORT STACKED ERROR
     026450  104460                                                    TRAP    C$ERROR
5315 026452                            ESCAPE  TST             ;SKIP TO END OF TEST
     026452  104410                                                    TRAP    C$ESCAPE
     026454  000464                                                    .WORD   L10022-.
5316
5317 026456  004537  010136            JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
5318 026462  000252                    252
5319 026464  000010                    8.
5320 026466  103003                    BCC     .+8.            ;BR IF NO ERROR
5321 026470                            ERROR                   ;REPORT STACKED ERROR
     026470  104460                                                    TRAP    C$ERROR
5322 026472                            ESCAPE  TST             ;SKIP TO END OF TEST
     026472  104410                                                    TRAP    C$ESCAPE
     026474  000444                                                    .WORD   L10022-.
5323
5324 026476  004537  010136            JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 252(DATA2)
5325 026502  000000                    000
5326 026504  000010                    8.
5327 026506  103003                    BCC     .+8.            ;BR IF NO ERROR
5328 026510                            ERROR                   ;REPORT STACKED ERROR
     026510  104460                                                    TRAP    C$ERROR
5329 026512                            ESCAPE  TST             ;SKIP TO END OF TEST
     026512  104410                                                    TRAP    C$ESCAPE
     026514  000424                                                    .WORD   L10022-.
5330
5331 026516  004537  010136            JSR     R5,TXCHAR       ;LOAD 377(DATA4), TX 000(DATA3)
5332 026522  000377                    377
5333 026524  000010                    8.
5334 026526  103003                    BCC     .+8.            ;BR IF NO ERROR
5335 026530                            ERROR                   ;REPORT STACKED ERROR
     026530  104460                                                    TRAP    C$ERROR
5336 026532                            ESCAPE  TST             ;SKIP TO END OF TEST
     026532  104410                                                    TRAP    C$ESCAPE
     026534  000404                                                    .WORD   L10022-.
5337
5338 026536  004537  010136            JSR     R5,TXCHAR       ;LOAD 001(DATA5)
5339 026542  000001                    001
5340 026544  000000                    0
5341 026546  103003                    BCC     .+8.            ;BR IF NO ERROR
```

TEST 3 -- DATA TEST -- BCP XLB CRC-16

```
5342 026550                            ERROR                   ;REPORT STACKED ERROR
     026550  104460                                                            TRAP    C$ERROR
5343 026552                            ESCAPE  TST             ;SKIP TO END OF TEST
     026552  104410                                                            TRAP    C$ESCAPE
     026554  000364                                                            .WORD   L10022-.
5344
5345 026556  004537  011624            JSR     R5,RCV1ST       ;CLOCK AND RCV 125
5346 026562  000000                    0
5347 026564  103003                    BCC     .+8.            ;BR IF NO ERROR
5348 026566                            ERROR                   ;REPORT STACKED ERROR
     026566  104460                                                            TRAP    C$ERROR
5349 026570                            ESCAPE  TST             ;SKIP TO END OF TEST
     026570  104410                                                            TRAP    C$ESCAPE
     026572  000346                                                            .WORD   L10022-.
5350
5351 026574  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
5352 026600  000125                    125
5353 026602  000000                    0
5354 026604  000010                    8.
5355 026606  103003                    BCC     .+8.            ;BR IF NO ERROR
5356 026610                            ERROR                   ;REPORT STACKED ERROR
     026610  104460                                                            TRAP    C$ERROR
5357 026612                            ESCAPE  TST             ;SKIP TO END OF TEST
     026612  104410                                                            TRAP    C$ESCAPE
     026614  000324                                                            .WORD   L10022-.
5358                            ;-------------------------------------------------------------
5359                            ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATX"
5360                            ;-------------------------------------------------------------
5361 026616  012702  002646            MOV     #PATX+1,R2      ;SET UP TABLE POINTER
5362 026622  112237  026660    5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
5363 026626  116237  000003  026640    MOVB    3(R2),10$       ;SET UP TRANSMIT CHARACTER
5364
5365 026634  004537  010136            JSR     R5,TXCHAR       ;LOAD A CHARACTER
5366 026640  000000            10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
5367 026642  000000                    0
5368 026644  103003                    BCC     .+8.            ;BR IF NO ERROR
5369 026646                            ERROR                   ;REPORT STACKED ERROR
     026646  104460                                                            TRAP    C$ERROR
5370 026650                            ESCAPE  TST             ;SKIP TO END OF TEST
     026650  104410                                                            TRAP    C$ESCAPE
     026652  000266                                                            .WORD   L10022-.
5371
5372 026654  004537  010350            JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
5373 026660  000000            20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
5374 026662  000000                    0
5375 026664  000010                    8.
5376 026666  103003                    BCC     .+8.            ;BR IF NO ERROR
5377 026670                            ERROR                   ;REPORT STACKED ERROR
     026670  104460                                                            TRAP    C$ERROR
5378 026672                            ESCAPE  TST             ;SKIP TO END OF TEST
     026672  104410                                                            TRAP    C$ESCAPE
     026674  000244                                                            .WORD   L10022-.
5379
5380 026676  022702  002665            CMP     #PATX+16.,R2    ;CHECK FOR 20TH CHARACTER OF TABLE
5381 026702  001347                    BNE     5$              ;BR IF NOT DONE
5382                            ;-------------------------------------------------------------
5383 026704  004537  010250            JSR     R5,TXCTRL       ;LOAD 1ST TEOM
```

TEST 3 -- DATA TEST -- BCP XLB CRC-16

```
 5384 026710  000002                    TEOM
 5385 026712  000000                    0
 5386 026714  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 357(DATA17), RCV 337(DATA18)
 5387 026720  000357                    357
 5388 026722  000000                    0
 5389 026724  000010                    8.
 5390 026726  103003                    BCC     .+8.            ;BR IF NO ERROR
 5391 026730                            ERROR                   ;REPORT STACKED ERROR
      026730  104460                                                    TRAP    C$ERROR
 5392 026732                            ESCAPE  TST             ;SKIP TO END OF TEST
      026732  104410                                                    TRAP    C$ESCAPE
      026734  000204                                                    .WORD   L10022-.
 5393
 5394 026736  004537  010250            JSR     R5,TXCTRL       ;LOAD 2ND TEOM
 5395 026742  000002                    TEOM
 5396 026744  000000                    0
 5397 026746  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 337(DATA18), RCV 277(DATA19)
 5398 026752  000337                    337
 5399 026754  000000                    0
 5400 026756  000010                    8.
 5401 026760  103003                    BCC     .+8.            ;BR IF NO ERROR
 5402 026762                            ERROR                   ;REPORT STACKED ERROR
      026762  104460                                                    TRAP    C$ERROR
 5403 026764                            ESCAPE  TST             ;SKIP TO END OF TEST
      026764  104410                                                    TRAP    C$ESCAPE
      026766  000152                                                    .WORD   L10022-.
 5404
 5405 026770  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 277(DATA19), RCV 177(DATA20)
 5406 026774  000277                    277
 5407 026776  000000                    0
 5408 027000  000010                    8.
 5409 027002  103003                    BCC     .+8.            ;BR IF NO ERROR
 5410 027004                            ERROR                   ;REPORT STACKED ERROR
      027004  104460                                                    TRAP    C$ERROR
 5411 027006                            ESCAPE  TST             ;SKIP TO END OF TEST
      027006  104410                                                    TRAP    C$ESCAPE
      027010  000130                                                    .WORD   L10022-.
 5412
 5413 027012  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 177(DATA20), RCV FIRST CRC BYTE
 5414 027016  100177                    RXERR!177
 5415 027020  000001                    RERCHK
 5416 027022  000010                    8.
 5417 027024  103003                    BCC     .+8.            ;BR IF NO ERROR
 5418 027026                            ERROR                   ;REPORT STACKED ERROR
      027026  104460                                                    TRAP    C$ERROR
 5419 027030                            ESCAPE  TST             ;SKIP TO END OF TEST
      027030  104410                                                    TRAP    C$ESCAPE
      027032  000106                                                    .WORD   L10022-.
 5420
 5421 027034  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
 5422 027040  000156                    156
 5423 027042  000000                    0
 5424 027044  000010                    8.
 5425 027046  103003                    BCC     .+8.            ;BR IF NO ERROR
 5426 027050                            ERROR                   ;REPORT STACKED ERROR
      027050  104460                                                    TRAP    C$ERROR
 5427 027052                            ESCAPE  TST             ;SKIP TO END OF TEST
```

```
        027052  104410                                                          TRAP    C$ESCAPE
        027054  000064                                                          .WORD   L10022-.
5428
5429    027056  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
5430    027062  000236                  236
5431    027064  000000                  0
5432    027066  000010                  8.
5433    027070  103003                  BCC     .+8.            ;BR IF NO ERROR
5434    027072                          ERROR                   ;REPORT STACKED ERROR
        027072  104460                                                          TRAP    C$ERROR
5435    027074                          ESCAPE  TST             ;SKIP TO END OF TEST
        027074  104410                                                          TRAP    C$ESCAPE
        027076  000042                                                          .WORD   L10022-.
5436                            ;-------------------------------------------------------------
5437    027100  005337  002526          DEC     REG0            ;DECREMENT COUNT
5438    027104  001406                  BEQ     40$             ;BR IF TRIPLE LOOP IS COMPLETED
5439
5440    027106  004537  010250          JSR     R5,TXCTRL       ;CLEAR TEOM, SET TSOM
5441    027112  000001                  TSOM
5442    027114  000001                  1
5443    027116  000137  026346          JMP     2$              ;AND RUN TX/RX AGAIN
5444
5445    027122  004537  011772  40$:    JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
5446    027126  000011          3$:     9.
5447    027130  103003                  BCC     .+8.            ;BR IF NO ERROR
5448    027132                          ERROR                   ;REPORT STACKED ERROR
        027132  104460                                                          TRAP    C$ERROR
5449    027134                          ESCAPE  TST             ;SKIP TO END OF TEST
        027134  104410                                                          TRAP    C$ESCAPE
        027136  000002                                                          .WORD   L10022-.
5450    027140          ENDTST
        027140                                                  L10022:
        027140  104401                                                          TRAP    C$ETST
```

```
   5468                                 .SBTTL  TEST 4 -- DATA TEST -- BCP XLB ODD VRC

                                        ;+*****************************************************************************
                                        ;*
                                        ;*      TEST 4 -- DATA TEST -- BCP XLB ODD VRC
                                        ;*
                                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                                        ;*  RECEIVE  IN  BCP  MODE  WITH  ODD  VRC ERROR DETECTION THE FOLLOWING
                                        ;*  MESSAGE:
                                        ;*
                                        ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
                                        ;*     357 337 277 177
                                        ;*
                                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
                                        ;*  THREE  TIMES.  AFTER  THE  LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
                                        ;*  SENT  BEFORE  DROPING  TXE  &  RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
                                        ;*  UTILIZED.
                                        ;*
                                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                                        ;*
                                        ;-*****************************************************************************
                                        ;
                                        ;       BGNTST
     027142                                                                                     T4::
   5469 027142  004737  005376          JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
   5470 027146  042737  001000  027224  BIC     #NOLOOP,1$      ;
   5471 027154  012737  000011  027670  MOV     #9.,3$          ;INIT ENTRAN COUNT/TTLOOP STATUS
   5472 027162  023727  002470  000000  CMP     BRDTYP,#0       ;IS THIS AN M8064?
   5473 027170  001412                  BEQ     2$              ; YES: USE TTLOOP (NOT XLB).
   5474 027172  023727  002472  000004  CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
   5475 027200  001406                  BEQ     2$              ;BR IF NO
   5476 027202  052737  001000  027224  BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
   5477 027210  052737  100000  027670  BIS     #BIT15,3$       ;      AND SPECIFY NOLOOP IN ENTRAN
   5478                                 ;-----------------------------------------------------------------------
   5479
   5480 027216  004537  007234          2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
   5481 027222  062226                          DDCMP!STRIPS!OVRC!SYNCH ;SET DDCMP,STRIP SYNC,ODD VRC,SYNCH=226
   5482 027224  000347                  1$:     TXDL!RXDL               ;USE 7 BIT CHARS FOR RX & TX
   5483 027226  103003                          BCC     .+8.            ;BR IF NO ERROR
   5484 027230                                  ERROR                   ;REPORT STACKED ERROR
        027230  104460                                                                          TRAP    C$ERROR
   5485 027232                                  ESCAPE  TST             ;SKIP TO END OF TEST
        027232  104410                                                                          TRAP    C$ESCAPE
        027234  000446                                                                          .WORD   L10023-.
   5486
   5487 027236  004537  010136          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
   5488 027242  000226                  SYNCH
   5489 027244  000007                  7.
   5490 027246  103003                  BCC     .+8.            ;BR IF NO ERROR
   5491 027250                          ERROR                   ;REPORT STACKED ERROR
        027250  104460                                                                          TRAP    C$ERROR
   5492 027252                          ESCAPE  TST             ;SKIP TO END OF TEST
        027252  104410                                                                          TRAP    C$ESCAPE
        027254  000426                                                                          .WORD   L10023-.
   5493
   5494 027256  004537  010136          JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
```

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

```
5495 027262  000226                    SYNCH
5496 027264  000010                    8.
5497 027266  103003                    BCC     .+8.            ;BR IF NO ERROR
5498 027270                            ERROR                   ;REPORT STACKED ERROR
     027270  104460                                                    TRAP    C$ERROR
5499 027272                            ESCAPE  TST             ;SKIP TO END OF TEST
     027272  104410                                                    TRAP    C$ESCAPE
     027274  000406                                                    .WORD   L10023-.
5500
5501 027276  004537  010250            JSR     R5,TXCTRL       ;CLEAR TSOM
5502 027302  000000                    000
5503 027304  000000                    0
5504
5505 027306  004537  010136            JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
5506 027312  000125                    125
5507 027314  000010                    8.
5508 027316  103003                    BCC     .+8.            ;BR IF NO ERROR
5509 027320                            ERROR                   ;REPORT STACKED ERROR
     027320  104460                                                    TRAP    C$ERROR
5510 027322                            ESCAPE  TST             ;SKIP TO END OF TEST
     027322  104410                                                    TRAP    C$ESCAPE
     027324  000356                                                    .WORD   L10023-.
5511
5512 027326  004537  010136            JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
5513 027332  000252                    252
5514 027334  000010                    8.
5515 027336  103003                    BCC     .+8.            ;BR IF NO ERROR
5516 027340                            ERROR                   ;REPORT STACKED ERROR
     027340  104460                                                    TRAP    C$ERROR
5517 027342                            ESCAPE  TST             ;SKIP TO END OF TEST
     027342  104410                                                    TRAP    C$ESCAPE
     027344  000336                                                    .WORD   L10023-.
5518
5519 027346  004537  010136            JSR     R5,TXCHAR       ;LOAD 000(DATA3)
5520 027352  000000                    000
5521 027354  000000                    0
5522 027356  103003                    BCC     .+8.            ;BR IF NO ERROR
5523 027360                            ERROR                   ;REPORT STACKED ERROR
     027360  104460                                                    TRAP    C$ERROR
5524 027362                            ESCAPE  TST             ;SKIP TO END OF TEST
     027362  104410                                                    TRAP    C$ESCAPE
     027364  000316                                                    .WORD   L10023-.
5525
5526 027366  004537  011624            JSR     R5,RCV1ST       ;CLOCK AND RCV 125(DATA1)
5527 027372  000000                    0
5528 027374  103003                    BCC     .+8.            ;BR IF NO ERROR
5529 027376                            ERROR                   ;REPORT STACKED ERROR
     027376  104460                                                    TRAP    C$ERROR
5530 027400                            ESCAPE  TST             ;SKIP TO END OF TEST
     027400  104410                                                    TRAP    C$ESCAPE
     027402  000300                                                    .WORD   L10023-.
5531
5532 027404  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
5533 027410  000125                    125
5534 027412  000001                    RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5535 027414  000010                    8.
5536 027416  103003                    BCC     .+8.            ;BR IF NO ERROR
```

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

```
5537 027420                                ERROR                   ;REPORT STACKED ERROR
     027420  104460                                                                TRAP    C$ERROR
5538 027422                                ESCAPE  TST             ;SKIP TO END OF TEST
     027422  104410                                                                TRAP    C$ESCAPE
     027424  000256                                                                .WORD   L10023-.
5539                               ;-----------------------------------------------------------------
5540                               ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATX"
5541                               ;-----------------------------------------------------------------
5542 027426  012702  002646                MOV     #PATX+1,R2      ;SET UP TABLE POINTER
5543 027432  112237  027470          5$:   MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
5544 027436  116237  000001  027450        MOVB    1(R2),10$       ;SET UP TRANSMIT CHARACTER
5545
5546 027444  004537  010136                JSR     R5,TXCHAR       ;LOAD A CHARACTER
5547 027450  000000                  10$:   000                     ;** HOLE FOR NEXT TX CHARACTER
5548 027452  000000                        0
5549 027454  103003                        BCC     .+8.            ;BR IF NO ERROR
5550 027456                                ERROR                   ;REPORT STACKED ERROR
     027456  104460                                                                TRAP    C$ERROR
5551 027460                                ESCAPE  TST             ;SKIP TO END OF TEST
     027460  104410                                                                TRAP    C$ESCAPE
     027462  000220                                                                .WORD   L10023-.
5552
5553 027464  004537  010350                JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
5554 027470  000000                  20$:   000                     ;** HOLE FOR EXPECTED CHARACTER
5555 027472  000001                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5556 027474  000010                        8.
5557 027476  103003                        BCC     .+8.            ;BR IF NO ERROR
5558 027500                                ERROR                   ;REPORT STACKED ERROR
     027500  104460                                                                TRAP    C$ERROR
5559 027502                                ESCAPE  TST             ;SKIP TO END OF TEST
     027502  104410                                                                TRAP    C$ESCAPE
     027504  000176                                                                .WORD   L10023-.
5560
5561 027506  022702  002737                CMP     #EPATX-1,R2
5562 027512  001347                        BNE     5$
5563                               ;-----------------------------------------------------------------
5564 027514  004537  010250                JSR     R5,TXCTRL       ;LOAD TSOM
5565 027520  000001                        TSOM
5566 027522  000000                        0
5567 027524  103003                        BCC     .+8.            ;BR IF NO ERROR
5568 027526                                ERROR                   ;REPORT STACKED ERROR
     027526  104460                                                                TRAP    C$ERROR
5569 027530                                ESCAPE  TST             ;SKIP TO END OF TEST
     027530  104410                                                                TRAP    C$ESCAPE
     027532  000150                                                                .WORD   L10023-.
5570
5571 027534  004537  010350                JSR     R5,RXCHAR       ;READ & CHK 277, RCV 177
5572 027540  000277                        277
5573 027542  000001                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5574 027544  000010                        8.
5575 027546  103003                        BCC     .+8.            ;BR IF NO ERROR
5576 027550                                ERROR                   ;REPORT STACKED ERROR
     027550  104460                                                                TRAP    C$ERROR
5577 027552                                ESCAPE  TST             ;SKIP TO END OF TEST
     027552  104410                                                                TRAP    C$ESCAPE
     027554  000126                                                                .WORD   L10023-.
5578
```

TEST 4 -- DATA TEST -- BCP XLB ODD VRC

```
5579 027556  004537  010250          JSR     R5,TXCTRL       ;LOAD 2ND TSOM
5580 027562  000001                  TSOM
5581 027564  000000                  0
5582 027566  103003                  BCC     .+8.            ;BR IF NO ERROR
5583 027570                          ERROR                   ;REPORT STACKED ERROR
     027570  104460                                                  TRAP    C$ERROR
5584 027572                          ESCAPE  TST             ;SKIP TO END OF TEST
     027572  104410                                                  TRAP    C$ESCAPE
     027574  000106                                                  .WORD   L10023-.
5585
5586 027576  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 177, RCV FIRST SYNC
5587 027602  000177                  177
5588 027604  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5589 027606  000010                  8.
5590 027610  103003                  BCC     .+8.            ;BR IF NO ERROR
5591 027612                          ERROR                   ;REPORT STACKED ERROR
     027612  104460                                                  TRAP    C$ERROR
5592 027614                          ESCAPE  TST             ;SKIP TO END OF TEST
     027614  104410                                                  TRAP    C$ESCAPE
     027616  000064                                                  .WORD   L10023-.
5593
5594 027620  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 1ST SYNC, RCV SECOND SYNC
5595 027624  000226                  SYNCH
5596 027626  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5597 027630  000010                  8.
5598 027632  103003                  BCC     .+8.            ;BR IF NO ERROR
5599 027634                          ERROR                   ;REPORT STACKED ERROR
     027634  104460                                                  TRAP    C$ERROR
5600 027636                          ESCAPE  TST             ;SKIP TO END OF TEST
     027636  104410                                                  TRAP    C$ESCAPE
     027640  000042                                                  .WORD   L10023-.
5601
5602 027642  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND SYNC, RCV NEXT ONE
5603 027646  000226                  SYNCH
5604 027650  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5605 027652  000010                  8.
5606 027654  103003                  BCC     .+8.            ;BR IF NO ERROR
5607 027656                          ERROR                   ;REPORT STACKED ERROR
     027656  104460                                                  TRAP    C$ERROR
5608 027660                          ESCAPE  TST             ;SKIP TO END OF TEST
     027660  104410                                                  TRAP    C$ESCAPE
     027662  000020                                                  .WORD   L10023-.
5609
5610 027664  004537  011772          JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
5611 027670  000011          3$:     9.
5612 027672  103003                  BCC     .+8.            ;BR IF NO ERROR
5613 027674                          ERROR                   ;REPORT STACKED ERROR
     027674  104460                                                  TRAP    C$ERROR
5614 027676                          ESCAPE  TST             ;SKIP TO END OF TEST
     027676  104410                                                  TRAP    C$ESCAPE
     027700  000002                                                  .WORD   L10023-.
5615 027702          ENDTST
     027702                                                  L10023:
     027702  104401                                                  TRAP    C$ETST
```

```
5633                                    .SBTTL  TEST 5 -- DATA TEST -- BCP XLB EVEN VRC

                                        ;+*********************************************************************************
                                        ;*
                                        ;*      TEST 5 -- DATA TEST -- BCP XLB EVEN VRC
                                        ;*
                                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                                        ;*  RECEIVE  IN  BCP  MODE  WITH  EVEN VRC ERROR DETECTION THE FOLLOWING
                                        ;*  MESSAGE:
                                        ;*
                                        ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
                                        ;*     357 337 277 177
                                        ;*
                                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
                                        ;*  THREE  TIMES.  AFTER  THE  LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
                                        ;*  SENT  BEFORE  DROPING  TXE  &  RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
                                        ;*  UTILIZED.
                                        ;*
                                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                                        ;*
                                        ;-*********************************************************************************
                                        ;
                                        ;       BGNTST
     027704                                                                                             T5::
5634 027704  004737  005376             JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
5635 027710  042737  001000  027766     BIC     #NOLOOP,1$      ;
5636 027716  012737  000011  030432     MOV     #9.,3$          ;INIT ENTRAN COUNT/STATUS
5637 027724  023727  002470  000000     CMP     BRDTYP,#0       ;IS BOARD TYPE M8064?
5638 027732  001412                     BEQ     2$              ; YES: SPECIFY TTLOOP (NOT XLB)
5639 027734  023727  002472  000004     CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
5640 027742  001406                     BEQ     2$              ;BR IF NO
5641 027744  052737  001000  027766     BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
5642 027752  053737  100000  030432     BIS     BIT15,3$        ;      AND SPECIFY NOLOOP IN ENTRAN
5643                                    ;-----------------------------------------------------------------------------
5644 027760  004537  007234         2$: JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
5645 027764  062626                     DDCMP!STRIPS!EVRC!SYNCH  ;SET DDCMP,STRIP SYNCH,EVEN VRC,SYNCH=226
5646 027766  000347                 1$: TXDL!RXDL               ;USE 7 BIT CHARS FOR TX & RX
5647 027770  103003                     BCC     .+8.            ;BR IF NO ERROR
5648 027772                             ERROR                   ;REPORT STACKED ERROR
     027772  104460                                                             TRAP    C$ERROR
5649 027774                             ESCAPE  TST             ;SKIP TO END OF TEST
     027774  104410                                             TRAP    C$ESCAPE
     027776  000446                                             .WORD   L10024-.
5650
5651 030000  004537  010136             JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
5652 030004  000226                     SYNCH
5653 030006  000007                     7.
5654 030010  103003                     BCC     .+8.            ;BR IF NO ERROR
5655 030012                             ERROR                   ;REPORT STACKED ERROR
     030012  104460                                                             TRAP    C$ERROR
5656 030014                             ESCAPE  TST             ;SKIP TO END OF TEST
     030014  104410                                             TRAP    C$ESCAPE
     030016  000426                                             .WORD   L10024 .
5657
5658 030020  004537  010136             JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
5659 030024  000226                     SYNCH
```

TEST 5 -- DATA TEST -- BCP XLB EVEN VRC

```
5660 030026  000010                    8.
5661 030030  103003                    BCC     .+8.            ;BR IF NO ERROR
5662 030032                            ERROR                   ;REPORT STACKED ERROR
     030032  104460                                                          TRAP    C$ERROR
5663 030034                            ESCAPE  TST             ;SKIP TO END OF TEST
     030034  104410                                                          TRAP    C$ESCAPE
     030036  000406                                                          .WORD   L10024-.
5664
5665 030040  004537  010250            JSR     R5,TXCTRL       ;CLEAR TSOM
5666 030044  000000                    000
5667 030046  000000                    0
5668
5669 030050  004537  010136            JSR     R5,TXCHAR       ;LOAD 125(DATA1),   3RD SYNCH
5670 030054  000125                    125
5671 030056  000010                    8.
5672 030060  103003                    BCC     .+8.            ;BR IF NO ERROR
5673 030062                            ERROR                   ;REPORT STACKED ERROR
     030062  104460                                                          TRAP    C$ERROR
5674 030064                            ESCAPE  TST             ;SKIP TO END OF TEST
     030064  104410                                                          TRAP    C$ESCAPE
     030066  000356                                                          .WORD   L10024-.
5675
5676 030070  004537  010136            JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
5677 030074  000252                    252
5678 030076  000010                    8.
5679 030100  103003                    BCC     .+8.            ;BR IF NO ERROR
5680 030102                            ERROR                   ;REPORT STACKED ERROR
     030102  104460                                                          TRAP    C$ERROR
5681 030104                            ESCAPE  TST             ;SKIP TO END OF TEST
     030104  104410                                                          TRAP    C$ESCAPE
     030106  000336                                                          .WORD   L10024-.
5682
5683 030110  004537  010136            JSR     R5,TXCHAR       ;LOAD 000(DATA3)
5684 030114  000000                    000
5685 030116  000000                    0
5686 030120  103003                    BCC     .+8.            ;BR IF NO ERROR
5687 030122                            ERROR                   ;REPORT STACKED ERROR
     030122  104460                                                          TRAP    C$ERROR
5688 030124                            ESCAPE  TST             ;SKIP TO END OF TEST
     030124  104410                                                          TRAP    C$ESCAPE
     030126  000316                                                          .WORD   L10024-.
5689
5690 030130  004537  011624            JSR     R5,RCV1ST       ;CLOCK AND RCV 125(DATA1)
5691 030134  000000                    0
5692 030136  103003                    BCC     .+8.            ;BR IF NO ERROR
5693 030140                            ERROR                   ;REPORT STACKED ERROR
     030140  104460                                                          TRAP    C$ERROR
5694 030142                            ESCAPE  TST             ;SKIP TO END OF TEST
     030142  104410                                                          TRAP    C$ESCAPE
     030144  000300                                                          .WORD   L10024-.
5695
5696 030146  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 125(DATA1),  CV 252(DATA2)
5697 030152  000125                    125
5698 030154  000001                    RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5699 030156  000010                    8.
5700 030160  103003                    BCC     .+8.            ;BR IF NO ERROR
5701 030162                            ERROR                   ;REPORT STACKED ERROR
```

```
        030162  104460                                                          TRAP    C$ERROR
5702    030164                          ESCAPE  TST             ;SKIP TO END OF TEST
        030164  104410                                                          TRAP    C$ESCAPE
        030166  000256                                                          .WORD   L10024-.
5703                            ;-----------------------------------------------------------
5704                            ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATX"
5705                            ;-----------------------------------------------------------
5706    030170  012702  002646                MOV     #PATX+1,R2      ;SET UP TABLE POINTER
5707    030174  112237  030232        5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
5708    030200  116237  000001  030212        MOVB    1(R2),10$       ;SET UP TRANSMIT CHARACTER
5709
5710    030206  004537  010136                JSR     R5,TXCHAR       ;LOAD A CHARACTER
5711    030212  000000                10$:     000                     ;** HOLE FOR NEXT TX CHARACTER
5712    030214  000000                        0
5713    030216  103003                        BCC     .+8.            ;BR IF NO ERROR
5714    030220                          ERROR                           ;REPORT STACKED ERROR
        030220  104460                                                          TRAP    C$ERROR
5715    030222                          ESCAPE  TST             ;SKIP T  END OF TEST
        030222  104410                                                          TRAP    C$ESCAPE
        030224  000220                                                          .WORD   L10024-.
5716
5717    030226  004537  010350                JSR     R5,RXCHAR       ;CLK/RECEIVE CHECK PREVIOUS CHARACTER
5718    030232  000000                20$:     000                     ;** HOLE FOR EXPECTED CHARACTER
5719    030234  000001                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5720    030236  000010                        8.
5721    030240  103003                        BCC     .+8.            ;BR IF NO ERROR
5722    030242                          ERROR                           ;REPORT STACKED ERROR
        030242  104460                                                          TRAP    C$ERROR
5723    030244                          ESCAPE  TST             ;SKIP TO END OF TEST
        030244  104410                                                          TRAP    C$ESCAPE
        030246  000176                                                          .WORD   L10024-.
5724
5725    030250  022702  002737                CMP     #EPATX-1,R2
5726    030254  001347                        BNE     5$
5727                            ;-----------------------------------------------------------
5728    030256  004537  010250                JSR     R5,TXCTRL       ;LOAD 1ST TSOM
5729    030262  000001                        TSOM
5730    030264  000000                        0
5731    030266  103003                        BCC     .+8.            ;BR IF NO ERROR
5732    030270                          ERROR                           ;REPORT STACKED ERROR
        030270  104460                                                          TRAP    C$ERROR
5733    030272                          ESCAPE  TST             ;SKIP TO END OF TEST
        030272  104410                                                          TRAP    C$ESCAPE
        030274  000150                                                          .WORD   L10024-.
5734
5735    030276  004537  010350                JSR     R5,RXCHAR       ;READ & CHK 277, RCV 177
5736    030302  000277                        277
5737    030304  000001                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5738    030306  000010                        8.
5739    030310  103003                        BCC     .+8.            ;BR IF NO ERROR
5740    030312                          ERROR                           ;REPORT STACKED ERROR
        030312  104460                                                          TRAP    C$ERROR
5741    030314                          ESCAPE  TST             ;SKIP TO END OF TEST
        030314  104410                                                          TRAP    C$ESCAPE
        030316  000126                                                          .WORD   L10024-.
5742
5743    030320  004537  010250                JSR     R5,TXCTRL       ;LOAD 2ND TSOM
```

TEST 5 - DATA TEST - BCP XLB EVEN VRC

```
5744 030324  000001                    TSOM
5745 030326  000000                    0
5746 030330  103003                    BCC     .+8.            ;BR IF NO ERROR
5747 030332                            ERROR                   ;REPORT STACKED ERROR
     030332  104460                                                    TRAP    C$ERROR
5748 030334                            ESCAPE  TST             ;SKIP TO END OF TEST
     030334  104410                                                    TRAP    C$ESCAPE
     030336  000106                                                    .WORD   L10024-.
5749
5750 030340  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 177, RCV FIRST SYNC
5751 030344  000177                    177
5752 030346  000001                    RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5753 030350  000010                    8.
5754 030352  103003                    BCC     .+8.            ;BR IF NO ERROR
5755 030354                            ERROR                   ;REPORT STACKED ERROR
     030354  104460                                                    TRAP    C$ERROR
5756 030356                            ESCAPE  TST             ;SKIP TO END OF TEST
     030356  104410                                                    TRAP    C$ESCAPE
     030360  000064                                                    .WORD   L10024-.
5757
5758 030362  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 1ST SYNC, RCV SECOND SYNC
5759 030366  000226                    SYNCH
5760 030370  000001                    RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5761 030372  000010                    8.
5762 030374  103003                    BCC     .+8.            ;BR IF NO ERROR
5763 030376                            ERROR                   ;REPORT STACKED ERROR
     030376  104460                                                    TRAP    C$ERROR
5764 030400                            ESCAPE  TST             ;SKIP TO END OF TEST
     030400  104410                                                    TRAP    C$ESCAPE
     030402  000042                                                    .WORD   L10024-.
5765
5766 030404  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 2ND SYNC, RCV NEXT ONE
5767 030410  000226                    SYNCH
5768 030412  000001                    RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
5769 030414  000010                    8.
5770 030416  103003                    BCC     .+8.            ;BR IF NO ERROR
5771 030420                            ERROR                   ;REPORT STACKED ERROR
     030420  104460                                                    TRAP    C$ERROR
5772 030422                            ESCAPE  TST             ;SKIP TO END OF TEST
     030422  104410                                                    TRAP    C$ESCAPE
     030424  000020                                                    .WORD   L10024-.
5773
5774 030426  004537  011772            JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
5775 030432  000011            3$:     9.                      ;
5776 030434  103003                    BCC     .+8.            ;BR IF NO ERROR
5777 030436                            ERROR                   ;REPORT STACKED ERROR
     030436  104460                                                    TRAP    C$ERROR
5778 030440                            ESCAPE  TST             ;SKIP TO END OF TEST
     030440  104410                                                    TRAP    C$ESCAPE
     030442  000002                                                    .WORD   L10024-.
5779 030444                    ENDTST
     030444                                                  L10024:
     030444  104401                                                    TRAP    C$ETST
```

```
5793                                    .SBTTL  TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

                                        ;;*****************************************************************************
                                        ;*
                                        ;*      TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1
                                        ;*
                                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                                        ;*  RECEIVE  IN  BOP MODE WITH CRC-CCITT-1 ERROR DETECTION THE FOLLOWING
                                        ;*  SHORT MESSAGE:  125  252  000 377 001
                                        ;*
                                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY FLAG CHARACTERS AND REPEATED
                                        ;*  THREE  TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
                                        ;*  LENGTHS ARE ALSO UTILIZED.
                                        ;*
                                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                                        ;*
                                        ;--*****************************************************************************
                                        ;
                                        ;       BGNTST
     030446                                                                                     T6::
5794 030446  012737  000003  002526            MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
5795 030454  004737  005376                    JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
5796 030460  042737  001000  030522            BIC     #NOLOOP,1$      ;
5797 030466  023727  002470  000000            CMP     BRDTYP,#0       ;IS BOARD TYPE = M8064 ?
5798 030474  001407                           BEQ     2$              ; YES: SPECIFY TTLOOP (NO XLB)
5799 030476  023727  002472  000004            CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
5800 030504  001403                           BEQ     2$              ;BR IF NO
5801 030506  052737  001000  030522            BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
5802                                    ;-----------------------------------------------------------------------------
5803 030514  004537  007234            2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
5804 030520  000000                           0                       ;SET BOP MODE,CRC-CCITT=>1'S CHECK
5805 030522  000000            1$:     0                               ;USE 8 BIT CHARS
5806 030524  103003                           BCC     .+8.            ;BR IF NO ERROR
5807 030526                                    ERROR                   ;REPORT STACKED ERROR
     030526  104460                                                                     TRAP    C$ERROR
5808 030530                                    ESCAPE  TST             ;SKIP TO END OF TEST
     030530  104410                                                                     TRAP    C$ESCAPE
     030532  000374                                                                     .WORD   L10025-.
5809
5810 030534  004537  010250                    JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
5811 030540  000001                           TSOM
5812 030542  000007                           7.
5813 030544  004537  010250                    JSR     R5,TXCTRL       ;LOAD 3RD FLAG, TX 2ND FLAG
5814 030550  000001                           TSOM
5815 030552  000010                           8.
5816 030554  004537  010250                    JSR     R5,TXCTRL       ;LOAD 4TH FLAG, TX 3RD FLAG
5817 030560  000001                           TSOM
5818 030562  000010                           8.
5819 030564  004537  010250                    JSR     R5,TXCTRL       ;CLEAR TSOM
5820 030570  000000                           000
5821 030572  000000                           0
5822 030574  004537  010136                    JSR     R5,TXCHAR       ;LOAD DATA1(125), TX 4TH FLAG
5823 030600  000125                           125
5824 030602  000010                           8.
5825 030604  103003                           BCC     .+8.            ;BR IF NO ERROR
5826 030606                                    ERROR                   ;REPORT STACKED ERROR
```

TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

```
                030606  104460                                                          TRAP    C$ERROR
5827            030610                          ESCAPE  TST             ;SKIP TO END OF TEST
                030610  104410                                                          TRAP    C$ESCAPE
                030612  000314                                                          .WORD   L10025-.
5828
5829            030614  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA2(252), TX DATA1(125)
5830            030620  000252                  252
5831            030622  000010                  8.
5832            030624  103003                  BCC     .+8.            ;BR IF NO ERROR
5833            030626                          ERROR                   ;REPORT STACKED ERROR
                030626  104460                                                          TRAP    C$ERROR
5834            030630                          ESCAPE  TST             ;SKIP TO END OF TEST
                030630  104410                                                          TRAP    C$ESCAPE
                030632  000274                                                          .WORD   L10025-.
5835
5836            030634  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA3(000), TX DATA2(252)
5837            030640  000000                  000
5838            030642  000010                  8.
5839            030644  103003                  BCC     .+8.            ;BR IF NO ERROR
5840            030646                          ERROR                   ;REPORT STACKED ERROR
                030646  104460                                                          TRAP    C$ERROR
5841            030650                          ESCAPE  TST             ;SKIP TO END OF TEST
                030650  104410                                                          TRAP    C$ESCAPE
                030652  000254                                                          .WORD   L10025-.
5842
5843            030654  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA4(377), TX DATA3(000)
5844            030660  000377                  377
5845            030662  000010                  8.
5846            030664  103003                  BCC     .+8.            ;BR IF NO ERROR
5847            030666                          ERROR                   ;REPORT STACKED ERROR
                030666  104460                                                          TRAP    C$ERROR
5848            030670                          ESCAPE  TST             ;SKIP TO END OF TEST
                030670  104410                                                          TRAP    C$ESCAPE
                030672  000234                                                          .WORD   L10025-.
5849
5850            030674  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA5(001), TX DATA4(377)
5851            030700  000001                  001
5852            030702  000011                  9.
5853            030704  103003                  BCC     .+8.            ;BR IF NO ERROR
5854            030706                          ERROR                   ;REPORT STACKED ERROR
                030706  104460                                                          TRAP    C$ERROR
5855            030710                          ESCAPE  TST             ;SKIP TO END OF TEST
                030710  104410                                                          TRAP    C$ESCAPE
                030712  000214                                                          .WORD   L10025-.
5856
5857            030714  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
5858            030720  000002                  TEOM
5859            030722  000000                  0
5860
5861            030724  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV DATA1(125)
5862            030730  000000                  0
5863            030732  103003                  BCC     .+8.            ;BR IF NO ERROR
5864            030734                          ERROR                   ;REPORT STACKED ERROR
                030734  104460                                                          TRAP    C$ERROR
5865            030736                          ESCAPE  TST             ;SKIP TO END OF TEST
                030736  104410                                                          TRAP    C$ESCAPE
                030740  000166                                                          .WORD   L10025-.
```

TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

```
5866
5867 030742  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA1(125), RCV DATA2(252)
5868 030746  000525                  RXSOM!125               ; & CHECK RSOM=1
5869 030750  000000                  0
5870 030752  000010                  8.
5871 030754  103003                  BCC     .+8.            ;BR IF NO ERROR
5872 030756                          ERROR                   ;REPORT STACKED ERROR
     030756  104460                                                  TRAP    C$ERROR
5873 030760                          ESCAPE  TST             ;SKIP TO END OF TEST
     030760  104410                                                  TRAP    C$ESCAPE
     030762  000144                                                  .WORD   L10025-.
5874
5875 030764  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
5876 030770  000002                  TEOM
5877 030772  000000                  0
5878
5879 030774  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA2(252), RCV DATA3(000)
5880 031000  000252                  252
5881 031002  000000                  0
5882 031004  000010                  8.
5883 031006  103003                  BCC     .+8.            ;BR IF NO ERROR
5884 031010                          ERROR                   ;REPORT STACKED ERROR
     031010  104460                                                  TRAP    C$ERROR
5885 031012                          ESCAPE  TST             ;SKIP TO END OF TEST
     031012  104410                                                  TRAP    C$ESCAPE
     031014  000112                                                  .WORD   L10025-.
5886
5887 031016  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA3(000), RCV DATA4(377)
5888 031022  000000                  000
5889 031024  000000                  0
5890 031026  000010                  8.
5891 031030  103003                  BCC     .+8.            ;BR IF NO ERROR
5892 031032                          ERROR                   ;REPORT STACKED ERROR
     031032  104460                                                  TRAP    C$ERROR
5893 031034                          ESCAPE  TST             ;SKIP TO END OF TEST
     031034  104410                                                  TRAP    C$ESCAPE
     031036  000070                                                  .WORD   L10025-.
5894
5895 031040  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA4(377), RCV DATA5(001)
5896 031044  000377                  377
5897 031046  000000                  0
5898 031050  020010                  NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
5899 031052  103003                  BCC     .+8.            ;BR IF NO ERROR
5900 031054                          ERROR                   ;REPORT STACKED ERROR
     031054  104460                                                  TRAP    C$ERROR
5901 031056                          ESCAPE  TST             ;SKIP TO END OF TEST
     031056  104410                                                  TRAP    C$ESCAPE
     031060  000046                                                  .WORD   L10025-.
5902
5903 031062  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA5(001), RCV FIRST FLAG
5904 031066  001001                  RXEOM!001               ; & CHECK REOM
5905 031070  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
5906 031072  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
5907 031074  103003                  BCC     .+8.            ;BR IF NO ERROR
5908 031076                          ERROR                   ;REPORT STACKED ERROR
     031076  104460                                                  TRAP    C$ERROR
5909 031100                          ESCAPE  TST             ;SKIP TO END OF TEST
```

TEST 6 -- DATA TEST -- BOP XLB CRC-CCITT-1

```
        031100  104410                                                          TRAP    C$ESCAPE
        031102  000024                                                          .WORD   L10025-.
   5910                         ;-------------------- ----------------------------------------------------
   5911 031104  005337  002526          DEC     REG0            ;DECREMENT COUNT
   5912 031110  001406                  BEQ     40$             ;BR IF TRIPLE LOOP IS COMPLETED
   5913
   5914 031112  004537  010250          JSR     R5,TXCTRL       ;CLEAR TEOM, SET TSOM
   5915 031116  000001                  TSOM
   5916 031120  000001                  1
   5917 031122  000137  030514          JMP     2$              ;AND RUN TX/RX AGAIN
   5918 031126                  40$:
   5919 031126                  ENDTST
        031126                                                          L10025:
        031126  104401                                                          TRAP    C$ETST
```

```
 5933                                   .SBTTL  TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

                                        ;+*****************************************************************************
                                        ;*
                                        ;*      TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0
                                        ;*
                                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
                                        ;*  RECEIVE  IN  BOP MODE WITH CRC-CCITT-0 ERROR DETECTION THE FOLLOWING
                                        ;*  SHORT MESSAGE:  125  252  000  377  001
                                        ;*
                                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY FLAG CHARACTERS AND REPEATED
                                        ;*  THREE  TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
                                        ;*  LENGTHS ARE ALSO UTILIZED.
                                        ;*
                                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                                        ;*
                                        ;-*****************************************************************************
                                        ;
                                        ;       BGNTST
      031130                                                                                    T7::
 5934 031130  012737  000003  002526           MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
 5935 031136  004737  005376                   JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
 5936 031142  042737  001000  031204           BIC     #NOLOOP,1$      ;
 5937 031150  023727  002470  000000           CMP     BRDTYP,#0       ;BOARD TYPE = M8064 ?
 5938 031156  001407                          BEQ     2$              ; YES: SPECIFY TTLOOP (NOT XLB)
 5939 031160  023727  002472  000004           CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
 5940 031166  001403                          BEQ     2$              ;BR IF NO
 5941 031170  052737  001000  031204           BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
 5942                                   ;-----------------------------------------------------------------------------
 5943 031176  004537  007234           2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
 5944 031202  000400                          CRCOS                   ;SET BOP MODE,CRC-CCITT=>0'S CHECK
 5945 031204  000000                   1$:     0                       ;USE 8 BIT CHARS
 5946 031206  103003                          BCC     .+8.            ;BR IF NO ERROR
 5947 031210                                   ERROR                   ;REPORT STACKED ERROR
      031210  104460                                                                    TRAP    C$ERROR
 5948 031212                                   ESCAPE  TST             ;SKIP TO END OF TEST
      031212  104410                                                                    TRAP    C$ESCAPE
      031214  000374                                                                    .WORD   L10026-.
 5949
 5950 031216  004537  010250                   JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
 5951 031222  000001                          TSOM
 5952 031224  000007                          7.
 5953 031226  004537  010250                   JSR     R5,TXCTRL       ;LOAD 3RD FLAG, TX 2ND FLAG
 5954 031232  000001                          TSOM
 5955 031234  000010                          8.
 5956 031236  004537  010250                   JSR     R5,TXCTRL       ;LOAD 4TH FLAG, TX 3RD FLAG
 5957 031242  000001                          TSOM
 5958 031244  000010                          8.
 5959 031246  004537  010250                   JSR     R5,TXCTRL       ;CLEAR TSOM
 5960 031252  000000                          000
 5961 031254  000000                          0
 5962 031256  004537  010136                   JSR     R5,TXCHAR       ;LOAD DATA1(125), TX 4TH FLAG
 5963 031262  000125                          125
 5964 031264  000010                          8.
 5965 031266  103003                          BCC     .+8.            ;BR IF NO ERROR
 5966 031270                                   ERROR                   ;REPORT STACKED ERROR
```

TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

```
      031270  104460                                                    TRAP    C$ERROR
 5967 031272                          ESCAPE  TST             ;SKIP TO END OF TEST
      031272  104410                                                    TRAP    C$ESCAPE
      031274  000314                                                    .WORD   L10026-.
 5968
 5969 031276  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA2(252), TX DATA1(125)
 5970 031302  000252                  252
 5971 031304  000010                  8.
 5972 031306  103003                  BCC     .+8.            ;BR IF NO ERROR
 5973 031310                          ERROR                   ;REPORT STACKED ERROR
      031310  104460                                                    TRAP    C$ERROR
 5974 031312                          ESCAPE  TST             ;SKIP TO END OF TEST
      031312  104410                                                    TRAP    C$ESCAPE
      031314  000274                                                    .WORD   L10026-.
 5975
 5976 031316  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA3(000), TX DATA2(252)
 5977 031322  000000                  000
 5978 031324  000010                  8.
 5979 031326  103003                  BCC     .+8.            ;BR IF NO ERROR
 5980 031330                          ERROR                   ;REPORT STACKED ERROR
      031330  104460                                                    TRAP    C$ERROR
 5981 031332                          ESCAPE  TST             ;SKIP TO END OF TEST
      031332  104410                                                    TRAP    C$ESCAPE
      031334  000254                                                    .WORD   L10026-.
 5982
 5983 031336  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA4(377), TX DATA3(000)
 5984 031342  000377                  377
 5985 031344  000010                  8.
 5986 031346  103003                  BCC     .+8.            ;BR IF NO ERROR
 5987 031350                          ERROR                   ;REPORT STACKED ERROR
      031350  104460                                                    TRAP    C$ERROR
 5988 031352                          ESCAPE  TST             ;SKIP TO END OF TEST
      031352  104410                                                    TRAP    C$ESCAPE
      031354  000234                                                    .WORD   L10026-.
 5989
 5990 031356  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA5(001), TX DATA4(377)
 5991 031362  000001                  001
 5992 031364  000011                  9.
 5993 031366  103003                  BCC     .+8.            ;BR IF NO ERROR
 5994 031370                          ERROR                   ;REPORT STACKED ERROR
      031370  104460                                                    TRAP    C$ERROR
 5995 031372                          ESCAPE  TST             ;SKIP TO END OF TEST
      031372  104410                                                    TRAP    C$ESCAPE
      031374  000214                                                    .WORD   L10026-.
 5996
 5997 031376  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
 5998 031402  000002                  TEOM
 5999 031404  000000                  0
 6000 031406  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV DATA1(125)
 6001 031412  000000                  0
 6002 031414  103003                  BCC     .+8.            ;BR IF NO ERROR
 6003 031416                          ERROR                   ;REPORT STACKED ERROR
      031416  104460                                                    TRAP    C$ERROR
 6004 031420                          ESCAPE  TST             ;SKIP TO END OF TEST
      031420  104410                                                    TRAP    C$ESCAPE
      031422  000166                                                    .WORD   L10026-.
 6005
```

TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

```
6006 031424  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA1(125), RCV DATA2(252)
6007 031430  000525                  RXSOM!125               ; & CHECK RSOM=1
6008 031432  000000                  0
6009 031434  000010                  8.
6010 031436  103003                  BCC     .+8.            ;BR IF NO ERROR
6011 031440                          ERROR                   ;REPORT STACKED ERROR
     031440  104460                                                  TRAP    C$ERROR
6012 031442                          ESCAPE  TST             ;SKIP TO END OF TEST
     031442  104410                                                  TRAP    C$ESCAPE
     031444  000144                                                  .WORD   L10026-.
6013
6014 031446  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
6015 031452  000002                  TEOM
6016 031454  000000                  0
6017 031456  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA2(252), RCV DATA3(000)
6018 031462  000252                  252
6019 031464  000000                  0
6020 031466  000010                  8.
6021 031470  103003                  BCC     .+8.            ;BR IF NO ERROR
6022 031472                          ERROR                   ;REPORT STACKED ERROR
     031472  104460                                                  TRAP    C$ERROR
6023 031474                          ESCAPE  TST             ;SKIP TO END OF TEST
     031474  104410                                                  TRAP    C$ESCAPE
     031476  000112                                                  .WORD   L10026-.
6024
6025 031500  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA3(000), RCV DATA4(377)
6026 031504  000000                  000
6027 031506  000000                  0
6028 031510  000010                  8.
6029 031512  103003                  BCC     .+8.            ;BR IF NO ERROR
6030 031514                          ERROR                   ;REPORT STACKED ERROR
     031514  104460                                                  TRAP    C$ERROR
6031 031516                          ESCAPE  TST             ;SKIP TO END OF TEST
     031516  104410                                                  TRAP    C$ESCAPE
     031520  000070                                                  .WORD   L10026-.
6032
6033 031522  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA4(377), RCV DATA5(001)
6034 031526  000377                  377
6035 031530  000000                  0
6036 031532  020010                  NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
6037 031534  103003                  BCC     .+8.            ;BR IF NO ERROR
6038 031536                          ERROR                   ;REPORT STACKED ERROR
     031536  104460                                                  TRAP    C$ERROR
6039 031540                          ESCAPE  TST             ;SKIP TO END OF TEST
     031540  104410                                                  TRAP    C$ESCAPE
     031542  000046                                                  .WORD   L10026-.
6040
6041 031544  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA5(001), RCV FIRST FLAG
6042 031550  001001                  RXEOM!001               ; & CHECK REOM
6043 031552  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
6044 031554  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
6045 031556  103003                  BCC     .+8.            ;BR IF NO ERROR
6046 031560                          ERROR                   ;REPORT STACKED ERROR
     031560  104460                                                  TRAP    C$ERROR
6047 031562                          ESCAPE  TST             ;SKIP TO END OF TEST
     031562  104410                                                  TRAP    C$ESCAPE
     031564  000024                                                  .WORD   L10026-.
```

```
6048                                    ;-------------------------------------------------------------------
6049 031566  005337  002526                     DEC     REG0            ;DECREMENT COUNT
6050 031572  001406                             BEQ     40$             ;BR IF TRIPLE LOOP IS COMPLETED
6051
6052 031574  004537  010250                     JSR     R5,TXCTRL       ;CLEAR TEOM, SET TSOM
6053 031600  000001                             TSOM
6054 031602  000001                             1
6055 031604  000137  031176                     JMP     2$              ;AND RUN TX/RX AGAIN
6056 031610                             40$:
6057 031610                             ENDTST
     031610                                                                     L10026:
     031610  104401                                                                       TRAP    C$ETST
```

```
6077                                    .SBTTL  TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

                                        ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                        ;*
                                        ;*      TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST
                                        ;*
                                        ;*  FIRST, THE DMV-11 IS INITIALIZED, THEN, TTL LOOPBACK IS SELECTED,
                                        ;*  AND THE FOLLOWING CHECKS ARE PERFORMED INVOLVING THE MODEM STATUS
                                        ;*  REGISTER :
                                        ;*  - RING, CARRIER, MODEM READY, TEST MODE, CTS ARE CHECKED FOR 1 STATE.
                                        ;*  - RTS IS DE-ASSERTED AND CTS IS CHECKED FOR 0.
                                        ;*  - RTS IS ASSERTED AND CTS IS CHECKED FOR 1.
                                        ;*
                                        ;*  NEXT, IF THE OPTION IS AN M8053 WITH AN H3254 TEST CONNECTOR INSTALLED,
                                        ;*  THE DMV-11 IN INITIALIZED AGAIN, (TTL LOOPBACK IS CLEARED), AND
                                        ;*  THE FOLLOWING CHECKS ARE PERFORMED :
                                        ;*  - RING (IF EIA), CARRIER, MODEM READY, CTS ARE CHECKED FOR 1, TEST
                                        ;*    MODE IS CHECKED FOR 0.
                                        ;*  - RTS IS DE-ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 0.
                                        ;*  - RTS IS ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 1.
                                        ;*  - DTR IS DE-ASSERTED, AND MODEM READY IS CHECKED FOR 0.
                                        ;*  - DTR IS ASSERTED, AND MODEM READY IS CHECKED FOR 1.
                                        ;*
                                        ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                        ;
                                                BGNTST
     031612                                                                                     T8::
6078 031612  012737  000010  002414             MOV     #8.,TSTNUM      ;SET TEST NO. FOR POSSIBLE PRINTOUT
6079 031620  004537  007532                     JSR     R5,CKLPBK       ;CHK FOR H3254/5 INSTALLED
6080 031624  100000                             TCCHEK
6081 031626  103002                             BCC     2$              ;BR IF YES, TO RUN TEST
6082 031630                                     EXIT    TST             ;NO TEST CONNECTOR, SKIP TEST
     031630  104432                                                                       TRAP    C$EXIT
     031632  000722                                                                       .WORD   L10027-.
6083 031634                             2$:
6084 031634                                     BGNSUB
     031634                                                             T8.1:
     031634  104402                                                                       TRAP    C$BSUB
6085                                    ;INIT DMV, SET TTL LOOPBACK, CHK MODEM STATUS
6086 031636  004737  003352                     JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR TO INIT DMV11
6087 031642  004537  003712                     JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
6088 031646  120002                             VIADPB
6089 031650  000377                             377
6090 031652  004537  003712                     JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
6091 031656  120003                             VIADPA
6092 031660  000000                             000
6093 031662  004537  003712                     JSR     R5,WRITEI       ;SET TTL LOOPBACK
6094 031666  120000                             VIAORB
6095 031670  000002                             TTLOOP
6096 031672  012737  000001  002336             MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
6097                                    ;CHK FOR RING, CARRIER, MODEM RDY, CTS, TEST MODE, = 1
6098 031700  004537  003566                     JSR     R5,READ         ;READ MODEM STATUS
6099 031704  120017                             VIAORA
6100 031706  000000                     4$:     .WORD   0
6101 031710  142737  000023  031706             BICB    #SPEED!RCVDAT!UMAINT,4$ ;CLEAR UNNEEDED BITS
6102 031716  123727  031706  000354             CMPB    4$,#RING!CARIER!MDMRDY!CTS!TM ;CHK FOR BITS SET
6103 031724  001414                             BEQ     8$              ;BR IF ALL BITS SET
```

TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

```
6104 031726  012737  000354  002324          MOV     #RING!CARIER!MDMRDY!CTS!TM,GDATA ;SET EXPECTED DATA
6105 031734  013737  031706  002326          MOV     4$,BDATA          ;SET ACTUAL DATA
6106                                 ;REPORT "MODEM STATUS INCORRECT"
6107 031742                          GEDF    EM83,ERR11
                                                     ;       "DEVICE FATAL" ERROR # 66
     031742  104455                                                    TRAP    C$ERDF
     031744  000102                                                    .WORD   66
     031746  015262                                                    .WORD   EM83
     031750  020540                                                    .WORD   ERR11
6108 031752                          ESCAPE  SUB
     031752  104410                                                    TRAP    C$ESCAPE
     031754  000106                                                    .WORD   L10030-.
6109                                 ;DE-ASSERT RTS, CHK FOR CTS = 0
6110 031756  004537  003712          8$:     JSR     R5,WRITEI         ;DE-ASSERT RTS
6111 031762  120000                          VIAORB
6112 031764  000012                          RTSND!TTLOOP
6113 031766  004537  003566                  JSR     R5,READI          ;READ MODEM STATUS
6114 031772  120017                          VIAORA
6115 031774  000000                  10$:    .WORD   0
6116 031776  132737  000010  031774        BITB    #CTS,10$          ;CHK FOR CTS = 0
6117 032004  001406                          BEQ     12$               ;BR IF YES
6118                                 ;REPORT CTS NOT CLEARED
6119 032006                          GEDF    EM84,ERR14
                                                     ;       "DEVICE FATAL" ERROR # 67
     032006  104455                                                    TRAP    C$ERDF
     032010  000103                                                    .WORD   67
     032012  015311                                                    .WORD   EM84
     032014  021146                                                    .WORD   ERR14
6120 032016                          ESCAPE  SUB
     032016  104410                                                    TRAP    C$ESCAPE
     032020  000042                                                    .WORD   L10030-.
6121                                 ;ASSERT RTS, CHK FOR CTS = 1
6122 032022  004537  003712          12$:    JSR     R5,WRITEI         ;ASSERT RTS
6123 032026  120000                          VIAORB
6124 032030  000002                          TTLOOP
6125 032032  004537  003566                  JSR     R5,READI          ;READ MODEM STATUS
6126 032036  120017                          VIAORA
6127 032040  000000                  14$:    .WORD   0
6128 032042  132737  000010  032040        BITB    #CTS,14$          ;CHK FOR CTS = 1
6129 032050  001004                          BNE     15$               ;BR IF YES
6130                                 ;REPORT CTS NOT SET
6131 032052                          GEDF    EM85,ERR14
                                                     ;       "DEVICE FATAL" ERROR # 68
     032052  104455                                                    TRAP    C$ERDF
     032054  000104                                                    .WORD   68
     032056  015326                                                    .WORD   EM85
     032060  021146                                                    .WORD   ERR14
6132 032062                          15$:
6133 032062                          ENDSUB
     032062                                                          L10030:
     032062  104403                                                    TRAP    C$ESUB
6134                                 ;SEE IF BOARD IS M8053 WITH H3254 INSTALLED
6135 032064  005737  002470                  TST     BRDTYP            ;SEE IF M8053
6136 032070  001002                          BNE     17$               ;BR IF YES
6137 032072  000137  032554          16$:    JMP     A1                ;SKIP THIS SECTION OF CODE
6138 032076  023727  002472  000000  17$:    CMP     TSTCON,#0         ;SEE IF H3254 INSTALLED
6139 032104  001372                          BNE     16$               ;BR IF NOT, TO SKIP CODE
```

```
6140 032106                                    BGNSUB
     032106                                                                T8.2:
     032106  104402                                                        TRAP    C$BSUB
6141                                ;INIT DMV, (TTL LOOPBACK IS CLEARED), CHK MODEM STATUS
6142 032110  004737  003352                    JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR TO INIT DMV11
6143 032114  004537  003712                    JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
6144 032120  120002                            VIADPB
6145 032122  000377                            377
6146 032124  004537  003712                    JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
6147 032130  120003                            VIADPA
6148 032132  000000                            000
6149 032134  004537  003712                    JSR     R5,WRITEI       ;DISABLE TTLOOP
6150 032140  120000                            VIAORB
6151 032142  000000                            000
6152 032144  012737  000001  002336            MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
6153                                ;-----------------------------------------------------------------
6154                                ;CHK FOR RING (IF EIA), CARRIER, MODEM RDY, CTS = 1, TEST MODE, = 0
6155                                ;-----------------------------------------------------------------
6156 032152  004537  003566                    JSR     R5,READI        ;READ MODEM STATUS
6157 032156  120017                            VIAORA
6158 032160  000000                18$:        .WORD   0
6159
6160 032162  023727  002470  000001            CMP     BRDTYP,#1       ;IS V.35 THE SELECTED I/F ?
6161 032170  001013                            BNE     21$             ; NO: BR TO DO CHECK WITH RING
6162                                                                    ;YES: REMOVE RING BEFORE CHECKING
6163 032172  142737  000223  032160            BICB    #RING!SPEED!RCVDAT!UMAINT,18$ ;CLEAR UNNEEDED BITS
6164 032200  123727  032160  000150            CMPB    18$,#CARIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
6165 032206  001427                            BEQ     20$             ;BR IF STATUS CORRECT
6166 032210  012737  000150  002324            MOV     #CARIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
6167 032216  000412                            BR      19$
6168                                                                    ; DO CHECK WITH RING....
6169 032220  142737  000023  032160  21$:       BICB    #SPEED!RCVDAT!UMAINT,18$ ;CLEAR UNNEEDED BITS
6170 032226  123727  032160  000350            CMPB    18$,#RING!CARIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
6171 032234  001414                            BEQ     20$             ;BR IF STATUS CORRECT
6172 032236  012737  000350  002324            MOV     #RING!CARIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
6173
6174                                ;REPORT "MODEM STATUS INCORRECT"
6175 032244  013737  032160  002326  19$:       MOV     18$,BDATA       ;SET ACTUAL DATA
6176 032252                                    GEDF    EM83,ERR11
                                                                        ;       "DEVICE FATAL" ERROR # 69
     032252  104455                                                            TRAP    C$ERDF
     032254  000105                                                            .WORD   69
     032256  015262                                                            .WORD   EM83
     032260  020540                                                            .WORD   ERR11
6177 032262                                    ESCAPE  SUB
     032262  104410                                                            TRAP    C$ESCAPE
     032264  000266                                                            .WORD   L10031-.
6178                                ;DE-ASSERT RTS, CHK FOR CTS,CARRIER = 0
6179 032266  004537  003712            20$:    JSR     R5,WRITEI       ;DE-ASSERT RTS
6180 032272  120000                            VIAORB
6181 032274  000010                            RTSND
6182 032276  004537  003566                    JSR     R5,READI        ;READ MODEM STATUS
6183 032302  120017                            VIAORA
6184 032304  000000                22$:        .WORD   0
6185 032306  132737  000010  032304            BITB    #CTS,22$        ;CHK FOR CTS = 0
6186 032314  001406                            BEQ     24$             ;BR IF YES
6187                                ;REPORT CTS NOT CLEARED
```

TEST 8 - MODEM CONTROL SIGNAL LOOPBACK TEST

```
6188 032316                                    GEDF    EM84,ERR14
                                                       ;       "DEVICE FATAL" ERROR # 70
     032316  104455                                                                    TRAP    C$ERDF
     032320  000106                                                                    .WORD   70
     032322  015311                                                                    .WORD   EM84
     032324  021146                                                                    .WORD   ERR14
6189 032326                                    ESCAPE  SUB
     032326  104410                                                                    TRAP    C$ESCAPE
     032330  000222                                                                    .WORD   L10031-.
6190 032332  132737  000100  032304  24$:     BITB    #CARIER,22$     ;CHK FOR CARRIER = 0
6191 032340  001406                          BEQ     26$             ;BR IF YES
6192                                 ;REPORT CARRIER NOT CLEARED
6193 032342                                    GEDF    EM86,ERR14
                                                       ;       "DEVICE FATAL" ERROR # 71
     032342  104455                                                                    TRAP    C$ERDF
     032344  000107                                                                    .WORD   71
     032346  015342                                                                    .WORD   EM86
     032350  021146                                                                    .WORD   ERR14
6194 032352                                    ESCAPE  SUB
     032352  104410                                                                    TRAP    C$ESCAPE
     032354  000176                                                                    .WORD   L10031-.
6195                                 ;ASSERT RTS, CHK FOR CTS,CARRIER = 1
6196 032356  004537  003712          26$:     JSR     R5,WRITEI       ;ASSERT RTS
6197 032362  120000                          VIAORB
6198 032364  000000                          000
6199 032366  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
6200 032372  120017                          VIAORA
6201 032374  000000                  28$:    .WORD   0
6202 032376  132737  000010  032374          BITB    #CTS,28$        ;CHK FOR CTS = 1
6203 032404  001006                          BNE     30$             ;BR IF YES
6204                                 ;REPORT CTS NOT SET
6205 032406                                    GEDF    EM85,ERR14
                                                       ;       "DEVICE FATAL" ERROR # 72
     032406  104455                                                                    TRAP    C$ERDF
     032410  000110                                                                    .WORD   72
     032412  015326                                                                    .WORD   EM85
     032414  021146                                                                    .WORD   ERR14
6206 032416                                    ESCAPE  SUB
     032416  104410                                                                    TRAP    C$ESCAPE
     032420  000132                                                                    .WORD   L10031-.
6207 032422  132737  000100  032374  30$:     BITB    #CARIER,28$     ;CHK FOR CARRIER = 1
6208 032430  001006                          BNE     32$             ;BR IF YES
6209                                 ;REPORT CARRIER NOT SET
6210 032432                                    GEDF    EM87,ERR14
                                                       ;       "DEVICE FATAL" ERROR # 73
     032432  104455                                                                    TRAP    C$ERDF
     032434  000111                                                                    .WORD   73
     032436  015363                                                                    .WORD   EM87
     032440  021146                                                                    .WORD   ERR14
6211 032442                                    ESCAPE  SUB
     032442  104410                                                                    TRAP    C$ESCAPE
     032444  000106                                                                    .WORD   L10031-.
6212                                 ;DE-ASSERT DTR, CHK FOR MODEM READY = 0
6213 032446  004537  003712          32$:     JSR     R5,WRITEI       ;DE-ASSERT DTR
6214 032452  120000                          VIAORB
6215 032454  000020                          DTR
6216 032456  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
```

TEST 8 - MODEM CONTROL SIGNAL LOOPBACK TEST

```
6217 032462  120017                        VIAORA
6218 032464  000000                34$:    .WORD   0
6219 032466  132737  000040  032464        BITB    #MDMRDY,34$     ;CHK FOR MODEM READY = 0
6220 032474  001406                        BEQ     36$             ;BR IF YES
6221                               ;REPORT MODEM READY NOT CLEARED
6222 032476                                GEDF    EM88,ERR14
                                                   ;       "DEVICE FATAL" ERROR # 74
     032476  104455                                                        TRAP    C$ERDF
     032500  000112                                                        .WORD   74
     032502  015403                                                        .WORD   EM88
     032504  021146                                                        .WORD   ERR14
6223 032506                                ESCAPE  SUB
     032506  104410                                                        TRAP    C$ESCAPE
     032510  000042                                                        .WORD   L10031-.
6224                               ;ASSERT DTR, CHK FOR MODEM READY = 1
6225 032512  004537  003712        36$:    JSR     R5,WRITEI       ;ASSERT DTR
6226 032516  120000                        VIAORB
6227 032520  000000                        000
6228 032522  004537  003566                JSR     R5,READI        ;READ MODEM STATUS
6229 032526  120017                        VIAORA
6230 032530  000000                38$:    .WORD   0
6231 032532  132737  000040  032530        BITB    #MDMRDY,38$     ;CHK FOR MODEM READY = 1
6232 032540  001004                        BNE     40$             ;BR IF YES
6233                               ;REPORT MODEM READY NOT SET
6234 032542                                GEDF    EM89,ERR14
                                                   ;       "DEVICE FATAL" ERROR # 75
     032542  104455                                                        TRAP    C$ERDF
     032544  000113                                                        .WORD   75
     032546  015426                                                        .WORD   EM89
     032550  021146                                                        .WORD   ERR14
6235 032552                        40$:
6236 032552                                ENDSUB
     032552                                                    L10031:
     032552  104403                                                        TRAP    C$ESUB
6237 032554                        A1:
6238 032554                        ENDTST
     032554                                                    L10027:
     032554  104401                                                        TRAP    C$ETST
```

```
                                                 .SBTTL  TEST 9 -- DDCMP MESSAGE TEST

                                                 ;*******************************************************************
                                                 ;*
                                                 ;*      TEST 9 -- DDCMP MESSAGE TEST
                                                 ;*
                                                 ;*   THIS  TEST  WILL  USE XLB IF IT IS ENABLED -- OTHERWISE TTL LOOPBACK
                                                 ;*   WILL  BE  UTILIZED.  THIS  ASSURES  THAT  IT  CAN ALWAYS BE RUN AS A
                                                 ;*   GENERAL "RINGOUT" OF THE M8053.
                                                 ;*
                                                 ;*   INITIALIZATION:  BCP  MODE, CRC-16, IDLE = 0, SYNC (S/AR) = 226 OCT.
                                                 ;*   (96 HEX.), RXCL & TXCL = 0 (CHAR. LENGTH = 8).
                                                 ;*
                                                 ;*   THE FOLLOWING SAMPLE DDCMP MESSAGE IS TRANSMITTED & RECEIVED AND ALL
                                                 ;*   DATA AND CRC CHARACTERS ARE CHECKED FOR ERRORS:
                                                 ;*
                                                 ;*               ----------- HEADER -----------    --- DATA (PATTERN K) -----
                                                 ;*   SYNC SYNC 201 000 075 003 002 001 CRC CRC 000 377 ... 252 000 CRC CRC
                                                 ;*
                                                 ;*   THE  ATTEMPT  HERE  IS TO PROVIDE A TEST JUST BELOW THE LEVEL OF THE
                                                 ;*   FUNCTIONAL  DIAGNOSTIC.  THE  USYRT  WILL BE RESPONSIBLE FOR ALL CRC
                                                 ;*   GENERATION  AND  VERIFICATION BUT THE CRC'S WILL ALSO BE VERIFIED BY
                                                 ;*   SOFTWARE.
                                                 ;*
                                                 ;-*******************************************************************
                                                 ;
                                                 ;       BGNTST
     032556                                                                                        T9::
     032556  012737  000003  002526                      MOV     #3,REG0          ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
     032564  004737  005376                              JSR     PC,INIDMV        ;INIT DMV-11, ENTER MAINT LOOP
     032570  042737  001000  032632                      BIC     #NOLOOP,1$       ;
     032576  023727  002470  000000                      CMP     BRDTYP,#0        ;BOARD TYPE = M8064 ?
     032604  001407                                      BEQ     2$               ; YES: SPECIFY TTLOOP (NOT XLB)
     032606  023727  002472  000004                      CMP     TSTCON,#4        ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
     032614  001403                                      BEQ     2$               ;BR IF NO
     032616  052737  001000  032632                      BIS     #NOLOOP,1$       ; YES: SPECIFY NO TTLOOP (INITRN)
                                                 ;-------------------------------------------------------------------
     032624  004537  007234                      2$:     JSR     R5,INITRN        ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
     032630  065626                                      DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
     032632  000000                              1$:     0                        ;USE 8 BIT CHARS
     032634  103003                                      BCC     .+8.             ;BR IF NO ERROR
     032636                                              ERROR                    ;REPORT STACKED ERROR
     032636  104460                                                                                TRAP    C$ERROR
     032640                                              ESCAPE  TST              ;SKIP TO END OF TEST
     032640  104410                                                                                TRAP    C$ESCAPE
     032642  001032                                                                                .WORD   L10032-.

     032644  004537  010250                              JSR     R5,TXCTRL        ;SET TSOM, TX 1ST SYNCH
     032650  000001                                      TSOM
     032652  000007                                      7.
     032654  004537  010250                              JSR     R5,TXCTRL        ;TX 2ND SYNCH
     032660  000001                                      TSOM
     032662  000010                                      8.
     032664  004537  010250                              JSR     R5,TXCTRL        ;CLEAR TSOM
     032670  000000                                      000
     032672  000000                                      0
     032674  004537  010136                              JSR     R5,TXCHAR        ;LOAD 201(HEADR1), TX 3RD SYNCH
```

TEST 9 -- DDCMP MESSAGE TEST

```
 6286 032700  000201                    201
 6287 032702  000010                    8.
 6288 032704  103003                    BCC     .+8.            ;BR IF NO ERROR
 6289 032706                            ERROR                   ;REPORT STACKED ERROR
      032706  104460                                                    TRAP    C$ERROR
 6290 032710                            ESCAPE  TST             ;SKIP TO END OF TEST
      032710  104410                                                    TRAP    C$ESCAPE
      032712  000762                                                    .WORD   L10032-.
 6291
 6292 032714  004537  010136            JSR     R5,TXCHAR       ;LOAD 000(HEADR2), TX HEADR1
 6293 032720  000000                    000
 6294 032722  000010                    8.
 6295 032724  103003                    BCC     .+8.            ;BR IF NO ERROR
 6296 032726                            ERROR                   ;REPORT STACKED ERROR
      032726  104460                                                    TRAP    C$ERROR
 6297 032730                            ESCAPE  TST             ;SKIP TO END OF TEST
      032730  104410                                                    TRAP    C$ESCAPE
      032732  000742                                                    .WORD   L10032-.
 6298
 6299 032734  004537  010136            JSR     R5,TXCHAR       ;LOAD 075(HEADR3), TX HEADR2
 6300 032740  000075                    075
 6301 032742  000010                    8.
 6302 032744  103003                    BCC     .+8.            ;BR IF NO ERROR
 6303 032746                            ERROR                   ;REPORT STACKED ERROR
      032746  104460                                                    TRAP    C$ERROR
 6304 032750                            ESCAPE  TST             ;SKIP TO END OF TEST
      032750  104410                                                    TRAP    C$ESCAPE
      032752  000722                                                    .WORD   L10032-.
 6305
 6306 032754  004537  010136            JSR     R5,TXCHAR       ;LOAD 003(HEADR4), TX HEADR3
 6307 032760  000003                    003
 6308 032762  000010                    8.
 6309 032764  103003                    BCC     .+8.            ;BR IF NO ERROR
 6310 032766                            ERROR                   ;REPORT STACKED ERROR
      032766  104460                                                    TRAP    C$ERROR
 6311 032770                            ESCAPE  TST             ;SKIP TO END OF TEST
      032770  104410                                                    TRAP    C$ESCAPE
      032772  000702                                                    .WORD   L10032-.
 6312
 6313 032774  004537  010136            JSR     R5,TXCHAR       ;LOAD 002(HEADR5)
 6314 033000  000002                    002
 6315 033002  000000                    0
 6316 033004  103003                    BCC     .+8.            ;BR IF NO ERROR
 6317 033006                            ERROR                   ;REPORT STACKED ERROR
      033006  104460                                                    TRAP    C$ERROR
 6318 033010                            ESCAPE  TST             ;SKIP TO END OF TEST
      033010  104410                                                    TRAP    C$ESCAPE
      033012  000662                                                    .WORD   L10032-.
 6319
 6320 033014  004537  011624            JSR     R5,RCV1ST       ;CLOCK AND RCV 201
 6321 033020  000000                    0
 6322 033022  103003                    BCC     .+8.            ;BR IF NO ERROR
 6323 033024                            ERROR                   ;REPORT STACKED ERROR
      033024  104460                                                    TRAP    C$ERROR
 6324 033026                            ESCAPE  TST             ;SKIP TO END OF TEST
      033026  104410                                                    TRAP    C$ESCAPE
      033030  000644                                                    .WORD   L10032-.
```

TEST 9 -- DDCMP MESSAGE TEST

```
6325
6326 033032  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 201(HEADR1), RCV HEADR2
6327 033036  000201                  201
6328 033040  000000                  0
6329 033042  000010                  8.
6330 033044  103003                  BCC     .+8.            ;BR IF NO ERROR
6331 033046                          ERROR                   ;REPORT STACKED ERROR
     033046  104460                                                  TRAP    C$ERROR
6332 033050                          ESCAPE  TST             ;SKIP TO END OF TEST
     033050  104410                                                  TRAP    C$ESCAPE
     033052  000622                                                  .WORD   L10032-.
6333
6334 033054  004537  010136          JSR     R5,TXCHAR       ;LOAD 001(HEADR6)
6335 033060  000001                  001
6336 033062  000000                  0
6337 033064  103003                  BCC     .+8.            ;BR IF NO ERROR
6338 033066                          ERROR                   ;REPORT STACKED ERROR
     033066  104460                                                  TRAP    C$ERROR
6339 033070                          ESCAPE  TST             ;SKIP TO END OF TEST
     033070  104410                                                  TRAP    C$ESCAPE
     033072  000602                                                  .WORD   L10032-.
6340
6341 033074  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 000(HEADR2), RCV HEADR3
6342 033100  000000                  000
6343 033102  000000                  0
6344 033104  000010                  8.
6345 033106  103003                  BCC     .+8.            ;BR IF NO ERROR
6346 033110                          ERROR                   ;REPORT STACKED ERROR
     033110  104460                                                  TRAP    C$ERROR
6347 033112                          ESCAPE  TST             ;SKIP TO END OF TEST
     033112  104410                                                  TRAP    C$ESCAPE
     033114  000560                                                  .WORD   L10032-.
6348
6349 033116  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
6350 033122  000002                  TEOM                    ;(STARTS CRC 16 CHARACTER)
6351 033124  000000                  0
6352 033126  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 075(HEADR3), RCV HEADR4
6353 033132  000075                  075
6354 033134  000000                  0
6355 033136  000010                  8.
6356 033140  103003                  BCC     .+8.            ;BR IF NO ERROR
6357 033142                          ERROR                   ;REPORT STACKED ERROR
     033142  104460                                                  TRAP    C$ERROR
6358 033144                          ESCAPE  TST             ;SKIP TO END OF TEST
     033144  104410                                                  TRAP    C$ESCAPE
     033146  000526                                                  .WORD   L10032-.
6359
6360 033150  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 003(HEADR4), RCV HEADR5
6361 033154  000003                  003
6362 033156  000000                  0
6363 033160  000010                  8.
6364 033162  103003                  BCC     .+8.            ;BR IF NO ERROR
6365 033164                          ERROR                   ;REPORT STACKED ERROR
     033164  104460                                                  TRAP    C$ERROR
6366 033166                          ESCAPE  TST             ;SKIP TO END OF TEST
     033166  104410                                                  TRAP    C$ESCAPE
     033170  000504                                                  .WORD   L10032-.
```

TEST 9 -- DDCMP MESSAGE TEST

```
6367
6368 033172  004537  010250          JSR     R5,TXCTRL       ;CLEAR TEOM
6369 033176  000000                  000
6370 033200  000000                  0
6371                        ;-----------------------------------------------------------------
6372 033202  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(DATA1)
6373 033206  000000                  000
6374 033210  000000                  0
6375 033212  103003                  BCC     .+8.            ;BR IF NO ERROR
6376 033214                          ERROR                   ;REPORT STACKED ERROR
     033214  104460                                                  TRAP    C$ERROR
6377 033216                          ESCAPE  TST             ;SKIP TO END OF TEST
     033216  104410                                                  TRAP    C$ESCAPE
     033220  000454                                                  .WORD   L10032-.
6378
6379 033222  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 002(HEADR5), RCV HEADR6
6380 033226  000002                  002
6381 033230  000000                  0
6382 033232  000010                  8.
6383 033234  103003                  BCC     .+8.            ;BR IF NO ERROR
6384 033236                          ERROR                   ;REPORT STACKED ERROR
     033236  104460                                                  TRAP    C$ERROR
6385 033240                          ESCAPE  TST             ;SKIP TO END OF TEST
     033240  104410                                                  TRAP    C$ESCAPE
     033242  000432                                                  .WORD   L10032-.
6386
6387 033244  004537  010136          JSR     R5,TXCHAR       ;LOAD 377(DATA2)
6388 033250  000377                  377
6389 033252  000000                  0
6390 033254  103003                  BCC     .+8.            ;BR IF NO ERROR
6391 033256                          ERROR                   ;REPORT STACKED ERROR
     033256  104460                                                  TRAP    C$ERROR
6392 033260                          ESCAPE  TST             ;SKIP TO END OF TEST
     033260  104410                                                  TRAP    C$ESCAPE
     033262  000412                                                  .WORD   L10032-.
6393
6394 033264  004537  010350          JSR     R5,RXCHAR       ;READ/CHK 001(HEADR6), "RCV" FIRST CRC BYTE
6395 033270  100001                  RXERR!001
6396 033272  000001                  RERCHK
6397 033274  000010                  8.
6398 033276  103003                  BCC     .+8.            ;BR IF NO ERROR
6399 033300                          ERROR                   ;REPORT STACKED ERROR
     033300  104460                                                  TRAP    C$ERROR
6400 033302                          ESCAPE  TST             ;SKIP TO END OF TEST
     033302  104410                                                  TRAP    C$ESCAPE
     033304  000370                                                  .WORD   L10032-.
6401
6402 033306  004537  010136          JSR     R5,TXCHAR       ;LOAD 376(DATA3)
6403 033312  000376                  376
6404 033314  000000                  0
6405 033316  103003                  BCC     .+8.            ;BR IF NO ERROR
6406 033320                          ERROR                   ;REPORT STACKED ERROR
     033320  104460                                                  TRAP    C$ERROR
6407 033322                          ESCAPE  TST             ;SKIP TO END OF TEST
     033322  104410                                                  TRAP    C$ESCAPE
     033324  000350                                                  .WORD   L10032-.
6408
```

```
 6409 033326  004537  010350                JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
 6410 033332  000043                        043
 6411 033334  000000                        0
 6412 033336  000010                        8.
 6413 033340  103003                        BCC     .+8.            ;BR IF NO ERROR
 6414 033342                                ERROR                   ;REPORT STACKED ERROR
      033342  104460                                                                TRAP    C$ERROR
 6415 033344                                ESCAPE  TST             ;SKIP TO END OF TEST
      033344  104410                                                                TRAP    C$ESCAPE
      033346  000326                                                                .WORD   L10032-.
 6416
 6417 033350  004537  010136                JSR     R5,TXCHAR       ;LOAD 375(DATA4)
 6418 033354  000375                        375
 6419 033356  000000                        0
 6420 033360  103003                        BCC     .+8.            ;BR IF NO ERROR
 6421 033362                                ERROR                   ;REPORT STACKED ERROR
      033362  104460                                                                TRAP    C$ERROR
 6422 033364                                ESCAPE  TST             ;SKIP TO END OF TEST
      033364  104410                                                                TRAP    C$ESCAPE
      033366  000306                                                                .WORD   L10032-.
 6423
 6424 033370  004537  010350                JSR     R5,RXCHAR       ;READ & CHK SECOND CRC BYTE; RCV DATA1
 6425 033374  000035                        035
 6426 033376  000000                        0
 6427 033400  000010                        8.
 6428 033402  103003                        BCC     .+8.            ;BR IF NO ERROR
 6429 033404                                ERROR                   ;REPORT STACKED ERROR
      033404  104460                                                                TRAP    C$ERROR
 6430 033406                                ESCAPE  TST             ;SKIP TO END OF TEST
      033406  104410                                                                TRAP    C$ESCAPE
      033410  000264                                                                .WORD   L10032-.
 6431
 6432                                       ;-------------------------------------------------------------
 6433                                       ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATK"
 6434                                       ;-------------------------------------------------------------
 6435 033412  012702  003012                        MOV     #PATK,R2        ;SET UP TABLE POINTER
 6436 033416  112237  033454                5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED RX CHARACTER
 6437 033422  116237  000003  033434                MOVB    3(R2),10$       ;SET UP TRANSMIT CHARACTER
 6438
 6439 033430  004537  010136                        JSR     R5,TXCHAR       ;LOAD A CHARACTER
 6440 033434  000000                        10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
 6441 033436  000000                                0
 6442 033440  103003                                BCC     .+8.            ;BR IF NO ERROR
 6443 033442                                        ERROR                   ;REPORT STACKED ERROR
      033442  104460                                                                TRAP    C$ERROR
 6444 033444                                        ESCAPE  TST             ;SKIP TO END OF TEST
      033444  104410                                                                TRAP    C$ESCAPE
      033446  000226                                                                .WORD   L10032-.
 6445
 6446 033450  004537  010350                        JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
 6447 033454  000000                        20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
 6448 033456  000000                                0
 6449 033460  000010                                8.
 6450 033462  103003                                BCC     .+8.            ;BR IF NO ERROR
 6451 033464                                        ERROR                   ;REPORT STACKED ERROR
      033464  104460                                                                TRAP    C$ERROR
 6452 033466                                        ESCAPE  TST             ;SKIP TO END OF TEST
```

TEST 9 -- DDCMP MESSAGE TEST

```
        033466  104410                                                          TRAP    C$ESCAPE
        033470  000204                                                          .WORD   L10032-.
   6453
   6454 033472  022702  003027              CMP     #PATK+13.,R2    ;CHECK FOR 14TH CHARACTER OF TABLE
   6455 033476  001347                      BNE     5$              ;BR IF NOT DONE
   6456                          ;-----------------------------------------------------------------
   6457 033500  004537  010250              JSR     R5,TXCTRL       ;SET TEOM
   6458 033504  000002                      TEOM
   6459 033506  000000                      0
   6460 033510  004537  010350              JSR     R5,RXCHAR       ;READ/CHK 200(DATA13), RCV 125(DATA14)
   6461 033514  000200                      200
   6462 033516  000000                      0
   6463 033520  000010                      8.
   6464 033522  103003                      BCC     .+8.            ;BR IF NO ERROR
   6465 033524                              ERROR                   ;REPORT STACKED ERROR
        033524  104460                                                          TRAP    C$ERROR
   6466 033526                              ESCAPE  TST             ;SKIP TO END OF TEST
        033526  104410                                                          TRAP    C$ESCAPE
        033530  000144                                                          .WORD   L10032-.
   6467
   6468 033532  004537  010250              JSR     R5,TXCTRL       ;SET TEOM
   6469 033536  000002                      TEOM
   6470 033540  000000                      0
   6471 033542  004537  010350              JSR     R5,RXCHAR       ;READ/CHK 125(DATA14), RCV 252(DATA15)
   6472 033546  000125                      125
   6473 033550  000000                      0
   6474 033552  000010                      8.
   6475 033554  103003                      BCC     .+8.            ;BR IF NO ERROR
   6476 033556                              ERROR                   ;REPORT STACKED ERROR
        033556  104460                                                          TRAP    C$ERROR
   6477 033560                              ESCAPE  TST             ;SKIP TO END OF TEST
        033560  104410                                                          TRAP    C$ESCAPE
        033562  000112                                                          .WORD   L10032-.
   6478
   6479 033564  004537  010350              JSR     R5,RXCHAR       ;READ/CHK 252(DATA15), RCV 000(DATA16)
   6480 033570  000252                      252
   6481 033572  000000                      0
   6482 033574  000010                      8.
   6483 033576  103003                      BCC     .+8.            ;BR IF NO ERROR
   6484 033600                              ERROR                   ;REPORT STACKED ERROR
        033600  104460                                                          TRAP    C$ERROR
   6485 033602                              ESCAPE  TST             ;SKIP TO END OF TEST
        033602  104410                                                          TRAP    C$ESCAPE
        033604  000070                                                          .WORD   L10032-.
   6486
   6487 033606  004537  010350              JSR     R5,RXCHAR       ;READ/CHK 000(DATA16), RCV FIRST CRC BYTE
   6488 033612  100000                      RXERR!000
   6489 033614  000001                      RERCHK
   6490 033616  000010                      8.
   6491 033620  103003                      BCC     .+8.            ;BR IF NO ERROR
   6492 033622                              ERROR                   ;REPORT STACKED ERROR
        033622  104460                                                          TRAP    C$ERROR
   6493 033624                              ESCAPE  TST             ;SKIP TO END OF TEST
        033624  104410                                                          TRAP    C$ESCAPE
        033626  000046                                                          .WORD   L10032-.
   6494
   6495 033630  004537  010350              JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
```

TEST 9 -- DDCMP MESSAGE TEST

```
 6496 033634  000231                        231
 6497 033636  000000                        0
 6498 033640  000010                        8.
 6499 033642  103003                        BCC     .+8.            ;BR IF NO ERROR
 6500 033644                                ERROR                   ;REPORT STACKED ERROR
      033644  104460                                                        TRAP    C$ERROR
 6501 033646                                ESCAPE  TST             ;SKIP TO END OF TEST
      033646  104410                                                        TRAP    C$ESCAPE
      033650  000024                                                        .WORD   L10032-.
 6502
 6503 033652  004537  010350                JSR     R5,RXCHAR       ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
 6504 033656  000176                        176
 6505 033660  000000                        0
 6506 033662  000010                        8.
 6507 033664  103003                        BCC     .+8.            ;BR IF NO ERROR
 6508 033666                                ERROR                   ;REPORT STACKED ERROR
      033666  104460                                                        TRAP    C$ERROR
 6509 033670                                ESCAPE  TST             ;SKIP TO END OF TEST
      033670  104410                                                        TRAP    C$ESCAPE
      033672  000002                                                        .WORD   L10032-.
 6510 033674                        ENDTST
      033674                                                    L10032:
      033674  104401                                                        TRAP    C$ETST
```

```
6512                                    .SBTTL  HARDWARE PARAMETER CODING SECTION
6513
6514
6515
6516                                    ;////////////////////////////////////////////////////////////////////////
6517                                    ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
6518                                    ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
6519                                    ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
6520                                    ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
6521                                    ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
6522                                    ;/ WITH THE OPERATOR.
6523                                    ;////////////////////////////////////////////////////////////////////////
6524
6525 033676                                 BGNHRD
      033676  000027                                                    .WORD L10033-L$HARD/2
      033700                                                        L$HARD::
6526
6527 033700                                 GPRMA   ADDRES,0,0,160020,177776,YES
      033700  000031                                                    .WORD   T$CODE
      033702  033756                                                    .WORD   ADDRES
      033704  160020                                                    .WORD   T$LOLIM
      033706  177776                                                    .WORD   T$HILIM
6528 033710                                 GPRMA   VECTOR,2,0,0,674,YES
      033710  001031                                                    .WORD   T$CODE
      033712  034004                                                    .WORD   VECTOR
      033714  000000                                                    .WORD   T$LOLIM
      033716  000674                                                    .WORD   T$HILIM
6529 033720                                 GPRMD   PRIRTY,4,0,7000,4,7,YES
      033720  002032                                                    .WORD   T$CODE
      033722  034035                                                    .WORD   PRIRTY
      033724  007000                                                    .WORD   7000
      033726  000004                                                    .WORD   T$LOLIM
      033730  000007                                                    .WORD   T$HILIM
6530 033732                                 GPRMD   BDTY.M,12,0,7,0,2,YES
      033732  005032                                                    .WORD   T$CODE
      033734  034066                                                    .WORD   BDTY.M
      033736  000007                                                    .WORD   7
      033740  000000                                                    .WORD   T$LOLIM
      033742  000002                                                    .WORD   T$HILIM
6531 033744                                 GPRMD   TCON.M,14,0,7,0,4,YES
      033744  006032                                                    .WORD   T$CODE
      033746  034151                                                    .WORD   TCON.M
      033750  000007                                                    .WORD   7
      033752  000000                                                    .WORD   T$LOLIM
      033754  000004                                                    .WORD   T$HILIM
6532
6533 033756                                 ENDHRD
                                                                        .EVEN
      033756                                                        L10033:
6534
6535                                            .NLIST  BEX
6536 033756     104     105     126    ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
6537 034004     104     105     126    VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
6538 034035     104     105     126    PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
6539 034066     102     117     101    BDTY.M: .ASCIZ  /BOARD TYPE (0=M8064, 1=M8053 V.35, 2=M8053 EIA) : /
6540 034151     124     125     122    TCON.M: .ASCII  /TURNAROUND CONNECTOR TYPE -/<15><12>
6541 034206     050     060     075            .ASCII  /(0=M3254&M3255, 1=INTEGRAL MODEM CABLE, 2=EIA CABLE,/<15><12>
```

HARDWARE PARAMETER CODING SECTION

```
6542 034274      040      063      075            .ASCIZ  / 3=V.35 CABLE, 4=NONE) : /
6543                                        .LIST   BEX
6544                                        .EVEN
```

SOFTWARE PARAMETER CODING SECTION

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;////////////////////////////////////////////////////////////////////////
                                ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ;/ WITH THE OPERATOR.
                                ;////////////////////////////////////////////////////////////////////////

034326                                  BGNSFT
034326  000000                                                  .WORD L10034-L$SOFT/2
034330                                                  L$SOFT::

034330                                  ENDSFT
                                                                .EVEN
034330                                                  L10034:
```

****** PATCH AREA FOR DEBUG ******

```
                                .SBTTL ****** PATCH AREA FOR DEBUG ******
                                PATCH:
                                        .=.+200
                                        NOP
                                        NOP
                                        NOP
                                ;*********************************
                                .SBTTL  "ENDMOD" STATEMENT
                                        ENDMOD
                                .SBTTL  "LASTAD" STATEMENT & END OF PROGRAM
                                        LASTAD
                                                                        .EVEN
                                                                        .WORD   0
                                                                        .WORD   0
                                L$LAST::
                                .END
```

SYMBOL TABLE

```
ABC     = 000160          BSR0    002202          C$ESEG= 000005          EM34    014502          EXADD = 000020
ADDRES  033756            BSR1    002204          C$ESUB= 000003          EM35    014530          EXCON = 000010
ADR     = 000020 G        BSR10   002222          C$ETST= 000001          EM36    014551          EXECUT= 000005
AD.HIT  023242            BSR11   002224          C$EXIT= 000032          EM39    014566          E$END = 002100
AD.OK   023236            BSR12   002226          C$GETB= 000026          EM4     014307          E$LOAD= 000035
APA     = 000200          BSR13   002230          C$GETW= 000027          EM40    014610          FLGCA1= 000002
APAD    = 100000          BSR14   002232          C$GMAN= 000043          EM54    014626          FLGCA2= 000001
ASSEMB= 000010            BSR15   002234          C$GPHR= 000042          EM68    014650          FLGCB1= 000020
A1      032554            BSR16   002236          C$GPLO= 000030          EM69    014677          FLGCB2= 000010
BADDAT  002366            BSR17   002240          C$GPRI= 000040          EM70    014715          FLGIRQ= 000200
BDATA   002326            BSR2    002206          C$INIT= 000011          EM71    014737          FLGSR = 000004
BDRATE  002474            BSR3    002210          C$INLP= 000020          EM72    014755          FLGT1 = 000100
BDTY.M  034066            BSR4    002212          C$MANI= 000050          EM73    014777          FLGT2 = 000040
BIT0    = 000001 G        BSR5    002214          C$MEM = 000031          EM74    015014          FMT10   012560
BIT00   = 000001 G        BSR6    002216          C$MSG = 000023          EM75    015035          FMT10A  012634
BIT01   = 000002 G        BSR7    002220          C$OPEN= 000034          EM76    015051          FMT11   012655
BIT02   = 000004 G        CARIER= 000100          C$PNTB= 000014          EM77    015071          FMT12   012674
BIT03   = 000010 G        CA1CTL= 000001          C$PNTF= 000017          EM78    015105          FMT13   012703
BIT04   = 000020 G        CA2CTL= 000016          C$PNTS= 000016          EM79    015125          FMT14   012751
BIT05   = 000040 G        CB1CTL= 000020          C$PNTX= 000015          EM80    015161          FMT15   012766
BIT06   = 000100 G        CB2CTL= 000340          C$QIO = 000377          EM81    015201          FMT15A  013020
BIT07   = 000200 G        CHKTSO  007074          C$RDBU= 000007          EM82    015225          FMT16   013072
BIT08   = 000400 G        CHPTYP  002400          C$REFG= 000047          EM83    015262          FMT16A  013115
BIT09   = 001000 G        CKLPBK  007532          C$RESE= 000033          EM84    015311          FMT17   013157
BIT1    = 000002 G        CKRACT  005654          C$REVI= 000003          EM85    015326          FMT17A  013236
BIT10   = 002000 G        CKRDA   006154          C$RFLA= 000021          EM86    015342          FMT17B  013317
BIT11   = 004000 G        CKROR   006454          C$RPT = 000025          EM87    015363          FMT17C  013372
BIT12   = 010000 G        CKRSA   006314          C$SEFG= 000046          EM88    015405          FMT19   013452
BIT13   = 020000 G        CKSEOM  006614          C$SPRI= 000041          EM89    015426          FMT2    012126
BIT14   = 040000 G        CKTACT  005514          C$SVEC= 000037          EM90    015450          FMT21   013503
BIT15   = 100000 G        CKTBMT  006014          C$TPRI= 000013          EM91    015501          FMT22   013513
BIT2    = 000004 G        CKUSTS  005410          DDCMP = 040000          ENDCOD  026263          FMT23   013535
BIT3    = 000010 G        CRCOS = 000400          DEVMAP  002404          ENDEMB  012116          FMT24   013614
BIT4    = 000020 G        CRC16 = 001400          DEVPTR  002406          ENDIT   025122          FMT25   013627
BIT5    = 000040 G        CTS     = 000010        DFPTBL  002150 G        ENDPAT  003064          FMT26   013657
BIT6    = 000100 G        C$AU    = 000052        DIAGMC= 000000          ENTRAN  011774          FMT27   013712
BIT7    = 000200 G        C$AUTO= 000061          DOTBMT= 000007          EPATX   002740          FMT28   013721
BIT8    = 000400 G        C$BRK = 000022          DTR     = 000020        ERRBLK  002200 G        FMT29   013755
BIT9    = 001000 G        C$BSEG= 000004          DTRL    = 000000        ERRFLG  002550          FMT3    012165
BOE     = 000400 G        C$BSUB= 000002          D.BUG = 000000          ERRMSG  002176 G        FMT30   013762
BRDTYP  002470            C$CEFG= 000045          EF.CON= 000036 G        ERRNBR  002174 G        FMT31   014017
BSEL0   002416            C$CLCK= 000062          EF.NEW= 000035 G        ERROR1  002376          FMT32   014065
BSEL1   002420            C$CLEA= 000012          EF.PWR= 000034 G        ERRTYP  002172 G        FMT33   014117
BSEL10  002436            C$CLOS= 000035          EF.RES= 000037 G        ERR10   020364 G        FMT4    012247
BSEL11  002440            C$CLP1= 000006          EF.STA= 000040 G        ERR11   020540 G        FMT4A   012305
BSEL12  002442            C$CVEC= 000036          EIAV35= 000002          ERR11$  022032          FMT4B   012340
BSEL13  002444            C$DCLN= 000044          EM100   015536          ERR12   020714 G        FMT4C   012345
BSEL14  002446            C$DODU= 000051          EM101   015556          ERR12$  022364          FMT40   014150
BSEL15  002450            C$DRPT= 000024          EM14    014324          ERR13   021030 G        FMT5    012400
BSEL16  002452            C$DU    = 000053        EM16    014340          ERR14   021146 G        FMT5A   012447
BSEL17  002454            C$EDIT= 000003          EM2     014214          ERR5    020072 G        FMT7    012530
BSEL2   002422            C$ERDF= 000055          EM25    014560          ERR4    020120 G        FRSTIM  002320
BSEL3   002424            C$ERHR= 000056          EM28    014406          ERR4$   021506          F$AU    = 000015
BSEL4   002426            C$ERRO= 000060          EM29    014427          ERR5$   021654          F$AUTO= 000020
BSEL5   002430            C$ERSF= 000054          EM3     014265          ERR7A   020244 G        F$BGN   = 000040
BSEL6   002432            C$ERSO= 000057          EM30    014444          EVT     = 000004 G      F$CLEA= 000001
BSEL7   002434            C$ESCA= 000010          EM31    014465          EVRC    = 002400        F$DU    = 000016
```

SYMBOL TABLE

```
F$END  = 000041        INTGRL= 000001        L$HIME  002120 G      MLWRI   003722      PSTACK  002340
F$HARD= 000004        INTSC = 000200        L$HPCP  002016 G      MPCSR   002416      RABGA = 000004
F$HW   = 000013        ISR   = 000100 G      L$HPTP  002022 G      MPIVEC  002456      RAMADR= 001000
F$INIT= 000006        IXE   = 004000 G      L$HW    002150 G      MPOVEC  002460      RCVBUF  003064
F$JMP  = 000050        I$AU  = 000041        L$ICP   002104 G      MPRIOR  002462      RCVDAT= 000002
F$MOD  = 000000        I$AUTO= 000041        L$INIT  022554 G      MRDY   = 000200      RCV1ST  011624
F$MSG  = 000011        I$CLN = 000041        L$LADP  002026 G      MREQ   = 000001      RDA    = 000200
F$PROT= 000021        I$DU  = 000041        L$LAST  034542 G      MSTCLR  003352      RDSRH  = 120401
F$PWR  = 000017        I$HRD = 000041        L$LOAD  002100 G      NCRACT= 020000      RDSRL  = 120400
F$RPT  = 000012        I$INIT= 000041        L$LUN   002074 G      NCTBMT= 000200      READ    003454
F$SEG  = 000003        I$MOD = 000041        L$MREV  002050 G      NEWLIN  012123      READI   003566
F$SOFT= 000005        I$MSG = 000041        L$NAME  002000 G      NEWST   022746      REDBYT  002356
F$SRV  = 000010        I$PROT= 000040        L$PRIO  002042 G      NFCRDA= 040000      REDDAT  002516
F$SUB  = 000002        I$PTAB= 000041        L$PROT  022546 G      NOCHK  = 003400      REDLOC= 000001
F$SW   = 000014        I$PWR = 000041        L$PRT   002112 G      NOCRDA= 100000      REDPAG= 000005
F$TEST= 000001        I$RPT = 000041        L$REPP  002062 G      NOLOOP= 001000      REGNUM  002336
GDATA   002324        I$SEG = 000041        L$REV   002010 G      NORXEN= 040000      REG0    002526
GETBSR  004024        I$SETU= 000041        L$SOFT  034330 G      NULCLK= 000200      REG1    002530
GETPRM  022766        I$SFT = 000041        L$SPC   002056 G      OVRC   = 002000      REG2    002532
GETURS  004360        I$SRV = 000041        L$SPCP  002020 G      O$APTS= 000000      REG3    002534
GETVRS  004400        I$SUB = 000041        L$SPTP  002024 G      O$AU   = 000001      REG4    002536
GETWSR  004160        I$TST = 000041        L$STA   002030 G      O$BGNR= 000000      REG5    002540
GOODAT  002364        J$JMP = 000167        L$SW    002172 G      O$BGNS= 000000      REG6    002542
G$CNTO= 000200        LOADAT  002362        L$TEST  002114 G      O$DU   = 000001      REG7    002544
G$DELM= 000372        LOE   = 040000 G      L$TIML  002014 G      O$ERRT= 000001      REGM   = 000002
G$DISP= 000003        LOGDEV  002334        L$UNIT  002012 G      O$GNSW= 000000      REPCHK= 000001
G$EXCP= 000400        LOT   = 000010 G      L10000  002170        O$POIN= 000001      RERT   = 000200
G$HILI= 000002        LUSWI1  002464        L10001  002172        O$SETU= 000000      RETADR  002354
G$LOLI= 000001        LUSWI2  002466        L10002  020116        PALENB= 000001      RING   = 000200
G$NO   = 000000        LU1MOD  002000 G      L10003  020242        PATCH   034330      ROR    = 000020
G$OFFS= 000400        L$ACP   002110 G      L10004  020362        PATE    002576      RSA    = 000320
G$OFSI= 000376        L$APT   002036 G      L10005  020536        PATF    002606      RSOM   = 000001
G$PRMA= 000001        L$AU    023256 G      L10006  020712        PATG    002616      RSTCHK  005104
G$PRMD= 000002        L$AUT   002070 G      L10007  021026        PATI    002741      RTSND  = 000010
G$PRML= 000000        L$AUTO  025124 G      L10010  021144        PATJ    003002      RUN    = 000200
G$RADA= 000140        L$CCP   002106 G      L10011  021260        PATK    003012      RXABGA= 002000
G$RADB= 000000        L$CLEA  023250 G      L10013  023122        PATL    003033      RXACT  = 000040
G$RADD= 000040        L$CO    002032 G      L10014  023240        PATQ    003054      RXCHAR  010350
G$RADL= 000120        L$DEPO  002011 G      L10015  023250        PATX    002645      RXDL   = 000007
G$RADO= 000020        L$DESC  003304 G      L10016  023254        PBLENB= 000002      RXEN   = 000100
G$XFER= 000004        L$DESP  002076 G      L10017  023256        PCR    = 120407      RXEOM  = 001000
G$YES  = 000010        L$DEVP  002060 G      L10020  023556        PCSARH= 120405      RXERR  = 100000
HDX    = 000004        L$DISP  002124 G      L10021  025434        PCSARL= 120404      RXOR   = 004000
HELP   = 000000        L$DLY   002116 G      L10022  027140        PNT    = 001000 G    RXSOM  = 000400
HOE    = 100000 G      L$DTP   002040 G      L10023  027702        PRESET= 000001      SAVE4   002572
IBE    = 010000 G      L$DTYP  002034 G      L10024  030444        PRI    = 002000 G    SAVE6   002574
IDLE   = 000010        L$DU    023252 G      L10025  031126        PRIOR   002342      SAVLEN  002402
IDLES  = 004000        L$DUT   002072 G      L10026  031610        PRIRTY  034035      SCRACH  002332
IDU    = 000040 G      L$DVTY  003264 G      L10027  032554        PRI00  = 000000 G    SECAD  = 000020
IEI    = 000001        L$EF    002052 G      L10030  032062        PRI01  = 000040 G    SECADR= 010000
IEO    = 000020        L$ENVI  002044 G      L10031  032552        PRI02  = 000100 G    SEL0    002416
IER    = 020000 G      L$ERRT  002172 G      L10032  033674        PRI03  = 000140 G    SEL10   002436
INIDMV  005576        L$ETP   002102 G      L10033  033756        PRI04  = 000200 G    SEL12   002442
INITRN  007234        L$EXP1  002046 G      L10034  034330        PRI05  = 000240 G    SEL14   002446
INITT1  004534        L$EXP4  002064 G      MCIR   = 000100        PRI06  = 000300 G    SEL16   002452
INITT2  004734        L$EXP5  002066 G      MCODE   025436        PRI07  = 000340 G    SEL2    002422
INTFLG  002346        L$HARD  033700 G      MDMRDY= 000040        PRUTO  = 000100        SEL4    002426
```

SYMBOL TABLE

```
SEL6     002432          TTLOOP= 000002          TXTVRT  017766          T$NEST= 177777          UREGS   002242
SETVIA   005304          TXAB  = 002000          TXTVR0  017371          T$NS0 = 000000          USTATR= 122000
SFPTBL   002172 G        TXACT = 000004          TXTVR1  017375          T$NS1 = 000005          USYREG  002476
SFR    = 000001          TXCHAR  010136          TXTVR2  017401          T$NS2 = 000002          USYRT = 120400
SPEED  = 000020          TXCTRL  010250          TXTVR3  017406          T$PTNU= 000000          VECTOR  034004
SRMODE= 000034           TXDL  = 000340          TXTVR4  017413          T$SAVL= 177777          VIA   = 120000
STALL    004356          TXEN  = 000040          TXTVR5  017420          T$SEGL= 177777          VIAACR= 120013
STARES   002412          TXEOM = 001000          TXTVR6  017425          T$SUBN= 000000          VIADPA= 120003
STARST   022736          TXERR = 100000          TXTVR7  017432          T$TAGL= 177777          VIADPB= 120002
STEPLU   012072          TXGA  = 004000          TXTVR8  017437          T$TAGN= 010035          VIAIER= 120016
STRIP  = 000040          TXSOM = 000400          TXTVR9  017444          T$TEMP= 000000          VIAIFR= 120015
STRIPS= 020000           TXTMLT  017744          TXT1    015602          T$TEST= 000011          VIAMS = 120001
STRTML= 000301           TXTML0  017154          TXT10   016402          T$TSTM= 177777          VIAORA= 120017
STUREG   004250          TXTML1  017160          TXT11   016422          T$TSTS= 000001          VIAORB= 120000
SUBRPC   002344          TXTML2  017174          TXT11A  016474          T$$AU = 010017          VIAPCR= 120014
SVCGBL= 000000           TXTML3  017211          TXT11B  016532          T$$AUT= 010014          VIASR = 120012
SVCINS= 000001           TXTML4  017233          TXT12   016602          T$$CLE= 010015          VIAT1A= 120004
SVCSUB= 000001           TXTML5  017254          TXT13   016630          T$$DU = 010016          VIAT1B= 120005
SVCTAG= 000001           TXTML6  017304          TXT14   016645          T$$HAR= 010033          VIAT1C= 120006
SVCTST= 000001           TXTML7  017316          TXT15   016703          T$$HW = 010000          VIAT1D= 120007
SWPBOT= 121000           TXTNP   017500          TXT16   016745          T$$INI= 010013          VIAT2A= 120010
SWPDDC= 121400           TXTNPT  020030          TXT17   016760          T$$MSG= 010011          VIAT2B= 120011
SYNCH  = 000226          TXTNP0  017505          TXT18   017015          T$$PRO= 010012          VREGS   002262
S$LSYM= 010000           TXTNP1  017515          TXT19   017056          T$$SOF= 010034          WAITS0  005270
TAB    = 000004          TXTNP2  017525          TXT2    015640          T$$SUB= 010031          WRIBYT  002360
TBMT   = 000100          TXTNP3  017535          TXT2A   015702          T$$SW = 010001          WRILOC= 000002
TCCHEK= 100000           TXTNP4  017552          TXT2B   015741          T$$TES= 010032          WRIPAG= 000004
TCON.M   034151          TXTNP5  017567          TXT20   017112          T.EDF = 000001          WRITE   003700
TDATA    002322          TXTNP6  017604          TXT3    016004          T.EHRD= 000002          WRITFI  003712
TDSRH  = 120403          TXTNP7  017620          TXT4    016034          T.ESF = 000000          WSR0    002202
TDSRL  = 120402          TXTNP8  017634          TXT4A   016074          T.ESFT= 000003          WSR10   002212
TDSRNR   002575          TXTNUL  017152          TXT5    016135          T1      023260 G        WSR12   002214
TEOM   = 000002          TXTUR   017650          TXT6    016137          T1MODE= 000300          WSR14   002216
TERR   = 000200          TXTURT  020052          TXT7    016162          T2      023560 G        WSR16   002220
TGA    = 000010          TXTUR0  017663          TXT7A   016252          T2MODE= 000040          WSR2    002204
TIMFLG   002352          TXTUR1  017671          TXT8    016342          T3      026264 G        WSR4    002206
TM     = 000004          TXTUR2  017677          TXT9    016362          T4      027142 G        WSR6    002210
TMP0     002546          TXTUR3  017705          TXU   = 000002          T5      027704 G        XDATA   002330
TMP1     002550          TXTUR4  017713          T$ARGC= 000005          T6      030446 G        XORGB   021262
TMP2     002552          TXTUR5  017722          T$CODE= 006032          T7      031130 G        XYZ    = 000007
TMP3     002554          TXTUR6  017731          T$ERRN= 000115          T8      031612 G        X$ALWA= 000000
TMP4     002556          TXTUR7  017735          T$EXCP= 000000          T8.1    031634          X$FALS= 000040
TMP5     002560          TXTVR   017353          T$FLAG= 000040          T8.2    032106          X$OFFS= 000400
TMP6     002562          TXTVRA  017451          T$GMAN= 000000          T9      032556 G        X$TRUE= 000020
TMP7     002564          TXTVRB  017454          T$HILI= 000004          UAM    = 000200 G       $E     = 000113
TSC    = 000010          TXTVRC  017460          T$LAST= 000001          UMAINT= 000001          $LSTIN= 000001
TSOM   = 000001          TXTVRD  017464          T$LOLI= 000000          UNIT    002410          $LSTTA= 000001
TSTCON   002472          TXTVRE  017470          T$LSYM= 010000          UPBITS  002566          $T     = 000011
TSTNUM   002414          TXTVRF  017474          T$LTNO= 000011

. ABS.   034542      000
         000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  31536 WORDS  ( 124 PAGES)
DYNAMIC MEMORY:  19748 WORDS  ( 75 PAGES)
ELAPSED TIME:  00:09:36
CNDMEA,BIC,CNDMEA,SEQ/CR/-SP=SVC34,MLB/ML,CNDMEA,P11
```

CNDMEA0 DMV11 LINE U....B1 ....C1 ....D1 ....E1 ....F1 ....G1 ....H1 ....I1 ....J1 ....K1 ....L1 ....M1 ....N1

....B2 ....C2 ....D2 ....E2 ....F2 ....G2 ....H2 ....I2 ....J2 ....K2 ....L2 ....M2 ....N2

....B3 ....C3 ....D3 ....E3 ....F3 ....G3 ....H3 ....I3 ....J3 ....K3 ....L3 ....M3 ....N3

....B4 ....C4 ....D4 ....E4 ....F4 ....G4 ....H4 ....I4 ....J4 ....K4 ....L4 ....M4 ....N4

....B5 ....C5 ....D5 ....E5 ....F5 ....G5 ....H5 ....I5 ....J5 ....K5 ....L5 ....M5 ....N5

....B6 ....C6 ....D6 ....E6 ....F6 ....G6 ....H6 ....I6 ....J6 ....K6 ....L6 ....M6 ....N6

....B7 ....C7 ....D7 ....E7 ....F7 ....G7 ....H7 ....I7 ....J7 ....K7 ....L7 ....M7 ....N7

....B8 ....C8 ....D8 ....E8 ....F8 ....G8 ....H8 ....I8 ....J8 ....K8 ....L8 ....M8 ....N8

....B9 ....C9 ....D9 ....E9 ....F9 ....G9 ....H9 ....I9 ....J9 ....K9 ....L9 ....M9 ....N9

....B10 ....C10 ....D10 ....E10 ....F10 ....G10 ....H10 ....I10 ....J10 ....K10 ....L10 ....M10 ....N10

....B11 ....C11 ....D11 ....E11 ....F11 ....G11 ....H11 ....I11 ....J11 ....K11 ....L11 ....M11 ....N11

....B12 ....C12 ....D12 ....E12 ....F12 ....G12 ....H12 ....I12 ....J12 ....K12 ....L12 ....M12 ....N12

....B13 ....C13 ....D13 ....E13 ....F13 ....G13 ....H13 ....I13 ....J13 ....K13 ....L13 ....M13 ....N13

....B14 ....C14 ....D14