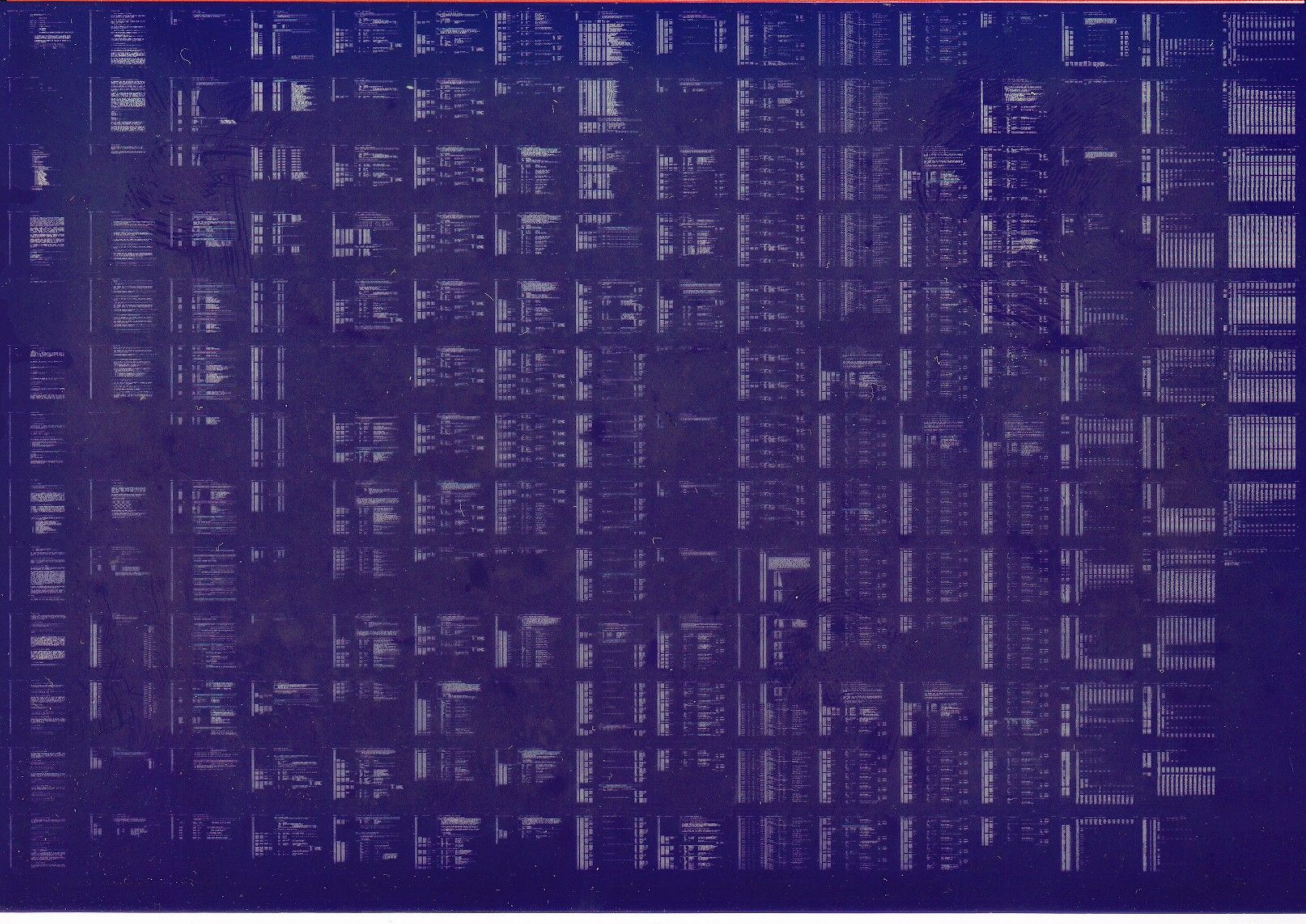
DMV11 DMV11 LINE UNIT DIAG 3 AH-S394A-MC
M8053,M8064 CVDMEAO FICHE 1 OF 1



```
                        .TITLE CVDMEA0 DMV11 LINE UNIT DIAG3
                        .SBTTL PROGRAM DOCUMENT
                        .REM @

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

                        PRODUCT CODE:   AC-S392A-MC

                        PRODUCT NAME:   CVDMEA0 DMV-11 LINE UNIT STATIC DIAGNOSTIC PART #3

                        PRODUCT DATE:   JANUARY 1981

                        MAINTAINER:     DIAGNOSTICS MERRIMACK  CC:38P

                        AUTHORS:        CHRIS BRIENEN
                                        DAVE HOFFMAN
                                        RAY MARSHALL

                        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) 1981 BY DIGITAL EQUIPMENT CORPORATION

                        THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                                DIGITAL         PDP             UNIBUS          MASSBUS
                                DEC             DECUS           DECTAPE
```

41
42
43
44
45
46 HISTORY
47 -------
48
49
50 REV DATE REASON
51 --- ---- ------
52
53 0 14-JAN-81 INITIAL RELEASE
54
55

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:

PDP-11/03 OR PDP-11/23
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

```
159
160                     THIS  PROGRAM  (CVDME)  SHOULD  BE THE LAST OF THE FIVE DMV-\1
161                     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 10 SECONDS PER PASS FOR EACH UNIT (5 SEC FOR PDP-11/23).

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
CVDME-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

6.3.1 START COMMAND

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

```
330                 ISR   INHIBIT STATISTICAL REPORTS
331                 IDU   INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
332                 LOT   LOOP ON TEST
333
334           THE  FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0
335           ARE CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS
336           SWITCH  IS  NOT GIVEN ALL FLAGS ARE CLEARED.  SEE EXAMPLE AT
337           END OF 6.3.1.5.
338
339
340           6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)
341
342           <INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
343           PASSES)  IT  IS  DESIRED  THAT  THE  END  OF PASS MESSAGE BE
344           PRINTED.  THE DEFAULT IS AT THE  END  OF  EVERY  PASS.   SEE
345           EXAMPLE AT END OF 6.3.1.5.
346
347
348           6.3.1.5 EFFECT OF START COMMAND
349
350           THE  EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE
351           PARAMETER DIALOGUE, THE  SOFTWARE  PARAMETER  DIALOGUE,  AND
352           THEN THE DIAGNOSTIC TESTS THEMSELVES.
353
354           THE  HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION
355           "# UNITS?" TO WHICH THE  OPERATOR  REPLIES  WITH  A  DECIMAL
356           NUMBER  N FROM 1 TO 16. THE TERM 'UNIT" REFERS TO THE DEVICE
357           TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
358           THIS  ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL
359           BE BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE  CONTAINING
360           ALL  THE  HARDWARE  INFORMATION  FOR ONE UNIT.  THE OPERATOR
361           MUST SUPPLY N (NUMBER OF UNITS) VALUES  FOR  EACH  QUESTION.
362           HE  MAY  DO  THIS  BY GIVING ONE ANSWER TO EACH QUESTION (IN
363           WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
364           BY  GIVING  N  VALUES, SEPARATED BY COMMAS, TO EACH QUESTION
365           (SERIES WILL BE POSED ONCE).  EACH QUESTION IS  FOLLOWED  BY
366           THE  RESPONSE  RADIX  (D  FOR  DECIMAL,  B FOR BINARY, O FOR
367           OCTAL, L FOR YES/NO) IN PARENTHESES AND  THE  DEFAULT  VALUE
368           AFTER THE PARENTHESES.
369
370           FOLLOWING  THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS
371           TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE  MODE  (QUICK
372           VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.
373
374           WHEN  THE QUESTION '# UNITS?" IS ANSWERED, MEMORY STORAGE IS
375           ALLOCATED FOR THE P-TABLES, AND IF THERE IS  NOT  ENOUGH  TO
376           ACCOMMODATE THEM THE MESSAGE 'TOO MANY UNITS" IS ISSUED.  IN
377           THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE  TO
378           TEST ALL UNITS.
379
380           EXAMPLE:
381
382           STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE
383
384           THIS  COMMAND  WILL CAUSE THREE PASSES TO BE MADE, EACH PASS
385           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.

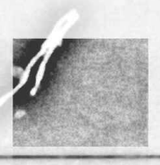
6.3.8 DISPLAY COMMAND

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

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

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

554
555
556 6.3.8.2 EFFECT OF DISPLAY COMMAND
557
558 THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED
559 OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS
560 THAT WERE DROPPED BY THE OPERATOR 'DROP' COMMAND ARE SO
561 DESIGNATED.
562
563
564 6.3.9 FLAGS COMMAND
565
566 **********
567 FLA(GS)
568 **********
569
570
571 6.3.9.1 EFFECT OF FLAGS COMMAND
572
573 THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.
574
575
576 6.3.10 ZFLAGS COMMAND
577
578 **********
579 ZFL(AGS)
580 **********
581
582
583 6.3.10.1 EFFECT OF ZFLAGS COMMAND
584
585 ALL FLAGS ARE CLEARED.
586
587
588 6.3.11 CONTROL CHARACTERS
589
590 A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES
591 A RETURN TO COMMAND MODE.
592
593 A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR
594 DIALOGUES- HARD CORE QUESTIONS (SEE 6.2),HARDWARE DIALOGUE
595 (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE
596 DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.
597
598 A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC
599 CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE
600 REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED,
601 WHICH RESTORES NORMAL TELETYPE OUTPUT.
602
603
604 6.3.12 HARDWARE PARAMETERS
605
606 THE FOLLOWING 3 QUESTIONS WILL BE ASKED ON A START COMMAND.
607 THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE
608 DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN
609 RESPONSE.



1. DEVICE CSR ADDRESS : (0) 160020?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SEL0) RESIDE ON THE LSI-BUS. THE ALLOWABLE RANGE IS 160020-177760 (OCTAL), AND THE DEFAULT VALUE IS 160020.

2. DEVICE VECTOR ADDRESS : (0) 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 &
```

```
788                     ;*  RECEIVE  IN  BCP  MODE  WITH  ODD  VRC ERROR DETECTION THE FOLLOWING
789                     ;*  MESSAGE:
790                     ;*
791                     ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
792                     ;*     357 337 277 177
793                     ;*
794                     ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
795                     ;*  THREE  TIMES.  AFTER  THE  LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
796                     ;*  SENT  BEFORE  DROPING  TXE  &  RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
797                     ;*  UTILIZED.
798                     ;*
799                     ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
800                     ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
801                     ;*****************************************************************************
802
803
804                     ;*****************************************************************************
805                     ;*      TEST 5  <DATA TEST -- BCP XLB EVEN VRC>
806                     ;*
807                     ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
808                     ;*  RECEIVE  IN  BCP  MODE  WITH  EVEN VRC ERROR DETECTION THE FOLLOWING
809                     ;*  MESSAGE:
810                     ;*
811                     ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
812                     ;*     357 337 277 177
813                     ;*
814                     ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
815                     ;*  THREE  TIMES.  AFTER  THE  LAST MESSAGE, SEVERAL SYNC CHARACTERS ARE
816                     ;*  SENT  BEFORE  DROPING  TXE  &  RXE. 7-BIT CHARACTER LENGTHS ARE ALSO
817                     ;*  UTILIZED.
818                     ;*
819                     ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
820                     ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
821                     ;*****************************************************************************
822
823
824                     ;*****************************************************************************
825                     ;*      TEST 6  <DATA TEST -- BOP XLB CRC-CCITT-1>
826                     ;*
827                     ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
828                     ;*  RECEIVE  IN  BOP MODE WITH CRC-CCITT-1 ERROR DETECTION THE FOLLOWING
829                     ;*  SHORT MESSAGE:  125  252  000 377 001
830                     ;*
831                     ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY FLAG CHARACTERS AND REPEATED
832                     ;*  THREE  TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
833                     ;*  LENGTHS ARE ALSO UTILIZED.
834                     ;*
835                     ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
836                     ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
837                     ;*****************************************************************************
838
839
840                     ;*****************************************************************************
841                     ;*      TEST 7  <DATA TEST -- BOP XLB CRC-CCITT-0>
842                     ;*
843                     ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
```

```
844                     ;*  RECEIVE  IN  BOP MODE WITH CRC-CCITT-0 ERROR DETECTION THE FOLLOWING
845                     ;*  SHORT MESSAGE:  125  252  000  377  001
846                     ;*
847                     ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY FLAG CHARACTERS AND REPEATED
848                     ;*  THREE  TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
849                     ;*  LENGTHS ARE ALSO UTILIZED.
850                     ;*
851                     ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
852                     ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
853                     ;*********************************************************************************
854
855
856                     ;*********************************************************************************
857                     ;*      TEST 8  <MODEM CONTROL SIGNAL LOOPBACK TEST>
858                     ;*
859                     ;*  FIRST, THE DMV-11 IS INITIALIZED. THEN, TTL LOOPBACK IS SELECTED,
860                     ;*  AND THE FOLLOWING CHECKS ARE PERFORMED INVOLVING THE MODEM STATUS
861                     ;*  REGISTER :
862                     ;*  - RING, CARRIER, MODEM READY, TEST MODE, CTS ARE CHECKED FOR 1 STATE.
863                     ;*  - RTS IS DE-ASSERTED AND CTS IS CHECKED FOR 0.
864                     ;*  - RTS IS ASSERTED AND CTS IS CHECKED FOR 1.
865                     ;*
866                     ;*  NEXT, IF THE OPTION IS AN M8053 WITH AN H3254 TEST CONNECTOR INSTALLED,
867                     ;*  THE DMV-11 IN INITIALIZED AGAIN, (TTL LOOPBACK IS CLEARED), AND
868                     ;*  THE FOLLOWING CHECKS ARE PERFORMED :
869                     ;*  - RING, CARRIER, MODEM READY, CTS ARE CHECKED FOR 1, TEST MODE IS CHECKED
870                     ;*      FOR 0.
871                     ;*  - RTS IS DE-ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 0.
872                     ;*  - RTS IS ASSERTED, AND CARRIER AND CTS ARE CHECKED FOR 1.
873                     ;*  - DTR IS DE-ASSERTED, AND MODEM READY IS CHECKED FOR 0.
874                     ;*  - DTR IS ASSERTED, AND MODEM READY IS CHECKED FOR 1.
875                     ;*********************************************************************************
876
877
878                     ;*********************************************************************************
879                     ;*      TEST 9  <DDCMP MESSAGE TEST>
880                     ;*
881                     ;*  THIS  TEST  WILL  USE XLB IF IT IS ENABLED -- OTHERWISE TTL LOOPBACK
882                     ;*  WILL  BE  UTILIZED.  THIS  ASSURES  THAT  IT  CAN ALWAYS BE RUN AS A
883                     ;*  GENERAL 'RINGOUT'' OF THE M8053.
884                     ;*
885                     ;*  INITIALIZATION:  BCP  MODE, CRC-16, IDLE = 0, SYNC (S/AR) = 226 OCT.
886                     ;*  (96 HEX.), RXCL & TXCL = 0 (CHAR. LENGTH = 8).
887                     ;*
888                     ;*  THE FOLLOWING SAMPLE DDCMP MESSAGE IS TRANSMITTED & RECEIVED AND ALL
889                     ;*  DATA AND CRC CHARACTERS ARE CHECKED FOR ERRORS:
890                     ;*
891                     ;*          ---------- HEADER ----------   --- DATA (PATTERN K) -----
892                     ;*  SYNC SYNC 201 000 075 003 002 001 CRC CRC 000 377 ... 252 000 CRC CRC
893                     ;*
894                     ;*  THE  ATTEMPT  HERE  IS TO PROVIDE A TEST JUST BELOW THE LEVEL OF THE
895                     ;*  FUNCTIONAL  DIAGNOSTIC.  THE  USYRT  WILL BE RESPONSIBLE FOR ALL CRC
896                     ;*  GENERATION  AND  VERIFICATION BUT THE CRC'S WILL ALSO BE VERIFIED BY
897                     ;*  SOFTWARE.
898                     ;*********************************************************************************
899
```

900

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH DESCRIBES A 'MASTER CLEAR FAILURE'' ERROR, AND PROVIDES THE PC OF THE ERROR CALL AND THE DEVICE REGISTER CONTENTS :

```
  CVDMB DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
  MASTER CLEAR FAILURE

    THE CONTENTS OF ALL BYTE SELECT REG'S ARE:
  BSEL0   BSEL1   BSEL2   BSEL3
   000     000     000     000
     BSEL4   BSEL5   BSEL6   BSEL7
      000     000     121     000
  BSEL10  BSEL11  BSEL12  BSEL13
   000     000     000     000
     BSEL14  BSEL15  BSEL16  BSEL17
      000     000     000     000
```

FOR OTHER ERRORS, THE REPORT MAY BE MORE EXTENSIVE, AND REQUIRE ADDITIONAL DATA TO BE REPORTED.

IF EXTENDED ERROR INFORMATION HAD BEEN INHIBITED USING THE IXE FLAG PRIOR TO RUNNING THE TEST, THE ABOVE ERROR WOULD HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :

```
  CVDMB DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
  MASTER CLEAR FAILURE
```

a

```
CVDMEA.P11    10-DEC-80 09:16          GENERAL EQUATES AND DS INVOCATION & SETUP

                                       .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.
```

```
                        .SBTTL  PROGRAM HEADER
                        ;++
                        ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                        ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                        ;--

002000                          POINTER BGNAU,BGNDU,ERRTBL

                                HEADER .CVDME,A,0,15.,0

002000  103             L$NAME::
002001  126                     .ASCII /C/
002002  104                     .ASCII /V/
002003  115                     .ASCII /D/
002004  105                     .ASCII /M/
002005  000                     .ASCII /E/
002006  000                     .BYTE   0
002007  000                     .BYTE   0
002010                          .BYTE   0
002010  101             L$REV::
002011                          .ASCII  /A/
002011  060             L$DEPO::
002012                          .ASCII  /0/
002012  000000            L$UNIT::
002014                          .WORD   0
002014  000017            L$TIML::
002016                          .WORD   15.
002016  033644            L$HPOP::
002020                          .WORD   L$HARD
002020  000000            L$SPCP::
002022                          .WORD   0
002022  002150            L$HPTP::
002024                          .WORD   L$HW
002024  000000            L$SPTP::
002026                          .WORD   0
002026  034506            L$LADP::
002030                          .WORD   L$LAST
002030  000000            L$STA::
002032                          .WORD   0
002032  000000            L$CO::
002034                          .WORD   0
002034  000000            L$DTYP::
002036                          .WORD   0
002036  000000            L$APT::
002040                          .WORD   0
002040  002124            L$DTP::
002042                          .WORD   L$DISPATCH
002042  000000            L$PRIO::
002044                          .WORD   0
002044  000000            L$ENVI::
002046                          .WORD   0
002046  000000            L$EXP1::
002050                          .WORD   0
002050  003               L$MREV::
002051  003                     .BYTE   C$REVISION
                                .BYTE   C$EDIT
```

```
1028  002052
1029  002052  000000          L$EF::
                                      .WORD   0
1030  002054  000000                  .WORD   0
1031  002056                  L$SPC::
1032  002056  000000                  .WORD   0
1033  002060                  L$DEVP::
1034  002060  003264                  .WORD   L$DVTYP
1035  002062                  L$REPP::
1036  002062  000000                  .WORD   0
1037  002064                  L$EXP4::
1038  002064  000000                  .WORD   0
1039  002066                  L$EXP5::
1040  002066  000000                  .WORD   0
1041  002070                  L$AUT::
1042  002070  023222                  .WORD   L$AU
1043  002072                  L$DUT::
1044  002072  023216                  .WORD   L$DU
1045  002074                  L$LUN::
1046  002074  000000                  .WORD   0
1047  002076                  L$DESP::
1048  002076  003304                  .WORD   L$DESC
1049  002100                  L$LOAD::
1050  002100  104035                  EMT     E$LOAD
1051  002102                  L$ETP::
1052  002102  002172                  .WORD   L$ERRTBL
1053  002104                  L$ICP::
1054  002104  022554                  .WORD   L$INIT
1055  002106                  L$CCP::
1056  002106  023214                  .WORD   L$CLEAN
1057  002110                  L$ACP::
1058  002110  023070                  .WORD   L$AUTO
1059  002112                  L$PRT::
1060  002112  022546                  .WORD   L$PROT
1061  002114                  L$TEST::
1062  002114  000000                  .WORD   0
1063  002116                  L$DLY::
1064  002116  000000                  .WORD   0
1065  002120                  L$HIME::
1066  002120  000000                  .WORD   0
1067
1068
1069                    .EVEN
```

```
1070
1071
1072  002122                          .SBTTL  DISPATCH TABLE
1073
1074                                  SLASH
1075                                  ;;//////////////////////////////////////////////////////////////////////
1076  002122                          ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1077                                  ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1078                                  SLASH
1079  002122                          ;;//////////////////////////////////////////////////////////////////////
1080  002122  000011                          DISPATCH 9.
1081  002124                                                                  .WORD   9
1082  002124  023224                                                 LSDISPATCH::
1083  002126  023524                                                         .WORD   T1
1084  002130  026230                                                         .WORD   T2
1085  002132  027106                                                         .WORD   T3
1086  002134  027650                                                         .WORD   T4
1087  002136  030412                                                         .WORD   T5
1088  002140  031074                                                         .WORD   T6
1089  002142  031556                                                         .WORD   T7
1090  002144  032522                                                         .WORD   T8
1091                                                                        .WORD   T9
```

CVDMEA.P11 10-DEC-80 09:16 DEFAULT HARDWARE P-TABLE

```
                        .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:
```

```
                            .SBTTL  SOFTWARE P-TABLE

                            ;//////////////////////////////////////////////////////////////////////
                            ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                            ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                            ;//////////////////////////////////////////////////////////////////////

002170                              BGNSW   SFPTBL
002170  000000                                                          .WORD   L10001-L$SW/2
002172                                                          L$SW::
002172                                                          SFPTBL::

002172                              ENDSW
002172                                                          L10001:
```

```
                        .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
                        ;
000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
```

```
1188    000140          PRI03== 140
1189    000100          PRI02== 100
1190    000040          PRI01== 40
1191    000000          PRI00== 0
1192                    ;
1193                    ;OPERATOR FLAG BITS
1194                    ;
1195    000004          EVL==     4
1196    000010          LOT==    10
1197    000020          ADR==    20
1198    000040          IDU==    40
1199    000100          ISR==   100
1200    000200          UAM==   200
1201    000400          BOE==   400
1202    001000          PNT==  1000
1203    002000          PRI==  2000
1204    004000          IXE==  4000
1205    010000          IBE== 10000
1206    020000          IER== 20000
1207    040000          LOE== 40000
1208    100000          HOE== 100000
```

```
                        .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
```

```
CVDMEA.P11    10-DEC-80 09:16        REGISTER DEFINITIONS -- USYRT

1302      002000                     TXAB    = BIT10          ;TRANSMIT ABORT
1303      001000                     TXEOM   = BIT9           ;TRANSMIT END-OF-MESSAGE
1304      000400                     TXSOM   = BIT8           ;TRANSMIT START-OF-MESSAGE
1305
1306                                 ;;*************************************************************************************
1307                                 ;* USYRT 'SYNC/SECONDARY ADDRESS' REGISTER
1308                                 ;;*************************************************************************************
1309
1310      120404                     PCSARL  = 120404         ;ADDRESS OF THIS REG
1311      000226                     SYNCH   = 226            ;STANDARD SYNCH CHARACTER
1312
1313                                 ;;*************************************************************************************
1314                                 ;* USYRT 'MODE CONTROL'
1315                                 ;;*************************************************************************************
1316
1317      120405                     PCSARH  = 120405         ;ADDRESS OF THIS REG
1318
1319                                         ;BIT DEFINITIONS ON BYTE BASIS:
1320
1321      000200                     APA     = BIT7           ;'ALL PARTIES ADDRESS' ENABLE
1322      000100                     PROTO   = BIT6           ;SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
1323      000040                     STRIP   = BIT5           ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1324      000020                     SECAD   = BIT4           ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1325      000010                     IDLE    = BIT3           ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1326      000007                     XYZ     = BIT2!BIT1!BIT0 ;CRC/PARITY SELECTION CONTROL
1327
1328                                         ;BIT DEFINITIONS ON WORD BASIS:
1329
1330      100000                     APAD    = BIT15          ;'ALL PARTIES ADDRESS' ENABLE
1331      040000                     DDCMP   = BIT14          ;CODE FOR DDCMP MODE
1332      020000                     STRIPS  = BIT13          ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1333      010000                     SECADR  = BIT12          ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1334      004000                     IDLES   = BIT11          ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1335      000400                     CRC0S   = BIT8           ;CODE FOR CRC-CCITT-0 SELECTION
1336      001400                     CRC16   = BIT9!BIT8      ;CODE FOR CRC-16 SELECTION
1337      003400                     NOCHK   = BIT10!BIT9!BIT8 ;CODE FOR NO ERROR CHECKING
1338      002400                     EVRC    = BIT10!BIT8     ;CODE FOR VRC EVEN CHECK
1339      002000                     OVRC    = BIT10          ;CODE FOR VRC ODD CHECK
1340
1341                                 ;;*************************************************************************************
1342                                 ;* USYRT 'DATA LENGTH SELECT' REGISTER
1343                                 ;;*************************************************************************************
1344
1345      120407                     PCR     = 120407         ;ADDRESS OF THIS REG
1346
1347                                         ;BIT DEFINITIONS:
1348
1349      000340                     TXDL    = BIT7!BIT6!BIT5 ;TRANSMIT DATA LENGTH SELECTION
1350      000020                     EXADD   = BIT4           ;EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
1351      000010                     EXCON   = BIT3           ;EXTENDED CONTROL FIELD -- NOT USED OR TESTED
1352      000007                     RXDL    = BIT2!BIT1!BIT0 ;RECEIVER DATA LENGTH SELECTION
1353
1354                                 ;;*************************************************************************************
1355                                 ;* USYRT STATUS REGISTER (ADDR. A400)
1356                                 ;;*************************************************************************************
1357      122000                     USTATR = 122000          ;USYRT STATUS REGISTER ADDRESS = A400 (HEX)
```

```
1358
1359                                    ;BIT DEFINITIONS:
1360
1361        000200          RDA     = BIT7          ;RECEIVER DATA AVAILABLE
1362        000100          TBMT    = BIT6          ;TRANSMITTER BUFFER EMPTY
1363        000040          RXACT   = BIT5          ;RECEIVER ACTIVE
1364        000020          RSA     = BIT4          ;RECEIVER STATUS AVAILABLE
1365        000010          TSO     = BIT3          ;TRANSMITTER SERIAL OUTPUT
1366        000004          TXACT   = BIT2          ;TRANSMITTER ACTIVE
1367        000002          TXU     = BIT1          ;TRANSMITTER UNDERRUN
1368        000001          SFR     = BIT0          ;SYNC/FLAG RECEIVED
```

```
                        .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
```

```
1425
1426                            ;       INITIALIZED TO 001 (HEX = 01) -- PORT A IS READ ONLY (EXCEPT FOR
1427                            ;                                       BIT0 WHICH ENABLES USYRT INTERNAL LOOPBACK).
1428
1429
1430
1431                            ;;*************************************************************************************
1432                            ;* TIMER 1 LOW ORDER (LATCH & COUNTER) -- 'T1L-L'' & 'T1C-L'' -- WRITE & READ
1433                            ;;*************************************************************************************
1434
1435    120004                  VIAT1A  = 120004                ;ADDRESS OF THIS REGISTER -- HEX = A0X4
1436
1437                            ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1438                            ;  WHEN READING, LOW ORDER COUNTER IS READ.
1439
1440
1441
1442                            ;;*************************************************************************************
1443                            ;* TIMER 1 HIGH ORDER COUNTER & TRIGGER -- 'T1L-H AND TRIGGER'' & 'T1C-H''
1444                            ;*       -- WRITE & READ
1445                            ;;*************************************************************************************
1446
1447    120005                  VIAT1B  = 120005                ;ADDRESS OF THIS REGISTER -- HEX = A0X5
1448
1449                            ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1450                            ;                 ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1451
1452                            ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1453
1454
1455
1456                            ;;*************************************************************************************
1457                            ;* TIMER 1 LOW ORDER LATCH -- 'T1L-L'' -- READ/WRITE
1458                            ;;*************************************************************************************
1459
1460    120006                  VIAT1C  = 120006                ;ADDRESS OF THIS REGISTER -- HEX = A0X6
1461
1462                            ;  THE LOW ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1463                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1464
1465
1466
1467                            ;;*************************************************************************************
1468                            ;* TIMER 1 HIGH ORDER LATCH -- 'T1L-H'' -- READ/WRITE
1469                            ;;*************************************************************************************
1470
1471    120007                  VIAT1D  = 120007                ;ADDRESS OF THIS REGISTER -- HEX = A0X7
1472
1473                            ;  THE HIGH ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1474                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1475
1476
1477
1478                            ;;*************************************************************************************
1479                            ;* TIMER 2 LOW ORDER (LATCH & COUNTER) -- 'T2L-L'' & 'T2C-L'' -- WRITE & READ
1480                            ;;*************************************************************************************
```

```
1481
1482          120010          VIAT2A  = 120010                ;ADDRESS OF THIS REGISTER -- HEX = A0X8
1483
1484                          ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1485                          ;  WHEN READING, LOW ORDER COUNTER IS READ.
1486
1487
1488
1489                          ;;*****************************************************************************
1490                          ;* TIMER 2 HIGH ORDER COUNTER & TRIGGER -- 'T2L-H AND TRIGGER' & 'T2C-H'
1491                          ;*       -- WRITE & READ
1492                          ;;*****************************************************************************
1493
1494          120011          VIAT2B  = 120011                ;ADDRESS OF THIS REGISTER -- HEX = A0X9
1495
1496                          ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1497                          ;                ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1498
1499                          ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1500
1501                          ;;*****************************************************************************
1502                          ;* SHIFT REGISTER -- 'SR' -- READ/WRITE
1503                          ;;*****************************************************************************
1504
1505          120012          VIASR   = 120012                ;ADDRESS OF THIS REGISTER -- HEX = A0XA
1506
1507                          ;  SHIFTING IS CONTROLLED BY THE SETTING OF VIASRC (ACR2 ---> ACR4) IN VIAACR
1508
1509
1510
1511                          ;;*****************************************************************************
1512                          ;* AUXILIARY CONTROL REGISTER -- 'ACR' -- READ/WRITE
1513                          ;;*****************************************************************************
1514
1515          120013          VIAACR  = 120013                ;ADDRESS OF THIS REGISTER -- HEX = A0XB
1516
1517          000300          T1MODE  = BIT7!BIT6             ;CONTROL THE MODE OF TIMER # 1
1518
1519                                          ;BIT 7:
1520                                          ; 0     PB7 DISABLED -- ONLY T1TO IN VIAIFR REFLECTS TIMEOUT
1521                                          ; 1     PB7 & T1TO REFLECT TIMEOUT
1522
1523                                          ;BIT 6:
1524                                          ; 0     TIMER 1 IN ONE-SHOT MODE
1525                                          ; 1     TIMER 1 IN CONTINUOUS SQUARE WAVE MODE
1526
1527          000040          T2MODE  = BIT5                  ;CONTROLS THE MODE OF TIMER # 1
1528
1529                                          ; 0     PULSE COUNTING MODE
1530                                          ; 1     INTERVAL TIMER MODE
1531
1532          000034          SRMODE  = BIT4!BIT3!BIT2        ;CONTROLS THE MODE OF THE SHIFT REGISTER
1533
1534                                          ; 0     SR DISABLED
1535                                          ; 1     SHIFT IN UNDER CONTROL OF T2, SHFT PULSES GEN'D ON CB1
1536                                          ; 2     SHIFT IN AT SYS. CLOCK RATE, SHFT PULSES GEN'D ON CB1
```

```
1537                                    ; 3     SHIFT IN UNDER CONTROL OF EXTERNAL INPUT PULSES
1538                                    ; 4     SHIFT OUT -- FREE RUNNING -- RATE CONTROLLED BY T2
1539                                    ; 5     SHIFT OUT -- RATE CONTROLLED BY T2 -- PULSES ON CB1
1540                                    ; 6     SHIFT OUT -- SYS. CLOCK RATE -- PULSES ON CB1
1541                                    ; 7     SHIFT OUT -- UNDER CONTROL OF PULSES APPLIED TO CB1
1542
1543        000002      PBLENB  = BIT1              ;PB LATCH CONTROL -- 1 ENABLES LATCH
1544        000001      PALENB  = BIT0              ;PA LATCH CONTROL -- 1 ENABLES LATCH
1545
1546
1547
1548
1549                    ;;*****************************************************************************
1550                    ;* PERIPHERAL CONTROL REGISTER -- 'PCR' -- READ/WRITE
1551                    ;;*****************************************************************************
1552
1553        120014      VIAPCR  = 120014            ;ADDRESS OF THIS REGISTER -- HEX = A0XC
1554
1555        000340      CB2CTL  = BIT7!BIT6!BIT5    ;CB2 MODE SELECT
1556        000020      CB1CTL  = BIT4              ;CB1 MODE SELECT
1557        000016      CA2CTL  = BIT3!BIT2!BIT1    ;CA2 MODE SELECT
1558        000001      CA1CTL  = BIT0              ;CA1 MODE SELECT
1559
1560
1561
1562                    ;;*****************************************************************************
1563                    ;* INTERRUPT FLAG REGISTER -- 'IFR' -- READ ONLY
1564                    ;;*****************************************************************************
1565
1566        120015      VIAIFR  = 120015            ;ADDRESS OF THIS REGISTER -- HEX = A0XD
1567
1568        000200      FLGIRQ  = BIT7              ;SET WHEN A FLAG IN THIS REG. GOES HIGH AND
1569                                                ;ITS CORRESPONDING BIT IN VIAIER IS SET.
1570                                                ;(I.E. VIAIER IS THE ENABLE REGISTER FOR THE
1571                                                ;FOR THE SETTING OF IRQ AND THE ISSUEANCE OF
1572                                                ;AN INTERRUPT TO THE 6502 WHEN IRQ IS SET.)
1573
1574        000100      FLGT1   = BIT6              ;TIMEOUT OF TIMER 1
1575        000040      FLGT2   = BIT5              ;TIMEOUT OF TIMER 2
1576        000020      FLGCB1  = BIT4              ;ACTIVE TRANSITION OF PIN 18 (CB1)
1577        000010      FLGCB2  = BIT3              ;ACTIVE TRANSITION OF PIN 19 (CB2)
1578        000004      FLGSR   = BIT2              ;COMPLETION OF 8 SHIFTS
1579        000002      FLGCA1  = BIT1              ;ACTIVE TRANSITION OF PIN 40 (CA1)
1580        000001      FLGCA2  = BIT0              ;ACTIVE TRANSITION OF PIN 39 (CA2)
1581
1582
1583
1584                    ;;*****************************************************************************
1585                    ;* INTERRUPT ENABLE REGISTER -- 'IER' -- READ/WRITE
1586                    ;;*****************************************************************************
1587
1588        120016      VIAIER  = 120016            ;ADDRESS OF THIS REGISTER -- HEX = A0XE
1589
1590        000200      INTSC   = BIT7              ;CONTROLS THE SETTING OR CLEARING OF BITS IN
1591                                                ;THE REST OF IER.  IF = 0  THE OTHER BITS IN
1592                                                ;THIS REG., IF SET, WILL CLEAR THEIR RESPECTIVE
```

```
1593                                            ;BITS IN THE INT. ENAB. REG..  IF = 1, THE
1594                                            ;RESPECTIVE BITS WILL BE SET.
1595
1596                            ;  WHEN WRITING THIS REG., THE COMMENT ABOVE HOLDS.
1597
1598                            ;  WHEN READING THIS REG., THE CURRENT STATE OF THE INT. ENABLE REG. IS RETURNED.
1599
1600                            ;  THE BIT ASSIGNMENTS ARE THE SAME AS FOR VIAIFR AS DEFINED ABOVE.
1601
1602
1603
1604                            ;;*****************************************************************************
1605                            ;* OUTPUT REGISTER A -- 'ORA'' -- READ ONLY (OR READ/WRITE UNDER CONTROL OF 'DDPA')
1606                            ;;*****************************************************************************
1607
1608         120017             VIAORA  = 120017                ;ADDRESS OF THIS REGISTER -- HEX = A0XF
1609
1610                            ;  THIS ADDRESS ACCESSES THE SAME DATA AS 'VIAMS'' EXCEPT THAT NO 'HANDSHAKING''
1611                            ;  WILL TAKE PLACE (I.E. THERE IS NO CHANGE IN IRQ OR CA2 AS A RESULT OF
1612                            ;  READING ORA THROUGH THIS ADDRESS)
1613
1614                            ;THE BIT ASSIGNMENTS ARE THE SAME AS FOR 'VIAMS'' ABOVE.
1615
1616
1617
```

```
1618                            .SBTTL  REGISTER DEFINITIONS -- MISC
1619
1620                            ;;*****************************************************************************
1621                            ;* SWITCH PACKS
1622                            ;;*****************************************************************************
1623
1624    121000                  SWPBOT  = 121000                ;'BOOT ADDRESS'' SWITCH PACK [A200]
1625    121400                  SWPDDCMP = 121400               ;'DDCMP ADDRESS'' SWITCH PACK [A300]
1626
1627                            ;MISCELLANEOUS EQUATES
1628
1629    100000                  TCCHEK  = BIT15         ;FLAG TO REQUEST H3254,5 CHECK
1630    001000                  RAMADR  = 001000        ;STARTING ADRS OF RAM PAGE 2 (ADRS 0200 HEX)
1631
1632    000002                  EIAV35  = BIT1                  ;SELECT V.35 OR EIA 423/232C
1633    000001                  INTGRL  = BIT0                  ;SELECT INTEGRAL MODEM
1634
1635    040000                  NORXEN  = BIT14         ;KILL RXEN DURING ''INITRN''
1636    001000                  NOLOOP  = BIT9          ;KILL TTLOOP DURING ''INITRN''
1637
1638    000200                  NCTBMT  = BIT7          ;DISABLE INITIAL TBMT=0 CHECK IN TXCHAR
1639
1640    100000                  NOCRDA  = BIT15         ;DISABLE INITIAL RDA=0 CHECK IN RXCHAR
1641    040000                  NFCRDA  = BIT14         ;DISABLE FINAL RDA=1 CHECK IN RXCHAR
1642    020000                  NCRACT  = BIT13         ;DISABLE RXACT=1 CHECK AFTER CLOCKING (RXCHAR)
1643
```

```
                        .SBTTL  GLOBAL DATA SECTION

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

                        ;++*************************************************************************
                        ; CONTROL BLOCK FOR STACKED ERROR MESSAGES
                        ;--*************************************************************************

                                ERRTBL
                                                                        L$ERRTBL::
002172                  ERRTYP::        .WORD   0
002174                  ERRNBR::        .WORD   0
002176                  ERRMSG::        .WORD   0
002200                  ERRBLK::        .WORD   0

                        ;;*************************************************************************
                        ;* STORAGE FOR DEVICE REGISTERS
                        ;;*************************************************************************
002202                  WSR0:                           ;STORAGE FOR DEVICE CSR REGISTERS
002202  000000          BSR0:   .WORD   0
002204                  WSR2:
002204  000000          BSR1:   .WORD   0
002206                  WSR4:
002206  000000          BSR2:   .WORD   0
002210                  WSR6:
002210  000000          BSR3:   .WORD   0
002212                  WSR10:
002212  000000          BSR4:   .WORD   0
002214                  WSR12:
002214  000000          BSR5:   .WORD   0
002216                  WSR14:
002216  000000          BSR6:   .WORD   0
002220                  WSR16:
002220  000000          BSR7:   .WORD   0
002222  000000          BSR10:  .WORD   0
002224  000000          BSR11:  .WORD   0
002226  000000          BSR12:  .WORD   0
002230  000000          BSR13:  .WORD   0
002232  000000          BSR14:  .WORD   0
002234  000000          BSR15:  .WORD   0
002236  000000          BSR16:  .WORD   0
002240  000000          BSR17:  .WORD   0

002242  000010          UREGS:  .BLKW   8.      ;THE FIRST 7 ARE FOR THE USYRT'S ACTUAL
                                                ;REGISTERS.  THE LAST ONE IS FOR THE STATUS
                                                ;REG. (USTATR).
002262  000020          VREGS:  .BLKW   16.     ;STORAGE FOR VIA REGISTERS FOR PRINTOUT
```

```
1694
1695                            ;*********************************************************************************
1696                            ;* MISCELLANEOUS STORAGE
1697                            ;*********************************************************************************
1697  002322  000000            TDATA:   .WORD   0              ;TEST DATA
1698  002324  000000            GDATA:   .WORD   0              ;GOOD DATA
1699  002326  000000            BDATA:   .WORD   0              ;BAD DATA
1700  002330  000000            XDATA:   .WORD   0              ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
1701  002332  000000            SCRACH:  .WORD   0              ;GEN'L PURPOSE SCRATCH WORD
1702  002334  000000            LOGDEV:  .WORD   0              ;LOGICAL DEVICE NUMBER
1703  002336  000000            REGNUM:  .WORD   0              ;CONTAINS A DEVICE REGISTER NUMBER
1704  002340  000000            PSTACK:  .WORD   0              ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
1705  002342  000000            PRIOR:   .WORD   0              ;CPU PRIORITY FOR PRINTOUT
1706  002344  000000            SUBRPC:  .WORD   0              ;PC OF SUBR CALL FOR ERROR REPORTS
1707  002346  000000            INTFLG:  .WORD   0              ;INTERRUPT RECEIVED FLAGS
1708                                                            ;  BIT 0 FOR TX, BIT 1 FOR RCV
1709  002350  000000            ERRFLG:  .WORD   0              ;SUBROUTINE ERROR FLAG
1710  002352  000000            TIMFLG:  .WORD   0              ;EVENT TIME-OUT FLAG
1711  002354  000000            RETADR:  .WORD   0              ;SUBR ERROR RETURN ADDRESS
1712  002356  000000            REDBYT:  .WORD   0              ;LO BYTE CONTAINS BYTE READ FROM LU REG
1713  002360  000000            WRIBYT:  .WORD   0              ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
1714  002362  000000            LOADAT:  .WORD   0              ;CONTAINS TEST DATA LOADED INTO REG
1715  002364  000000            GOODAT:  .WORD   0              ;STORAGE FOR EXPECTED DATA
1716  002366  000000            BADDAT:  .WORD   0              ;STORAGE FOR ACTUAL DATA
1717  002370  000000            FRSTIM:  .WORD   0              ;FLAG=0 IF PROGRAM JUST LOADED
1718  002372  000000            SAVE4:   .WORD   0              ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
1719  002374  000000            SAVE6:   .WORD   0              ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
1720  002376  000000            ERROR1:  .WORD   0              ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
1721  002400  000000            CHPTYP:  .WORD   0              ;USYRT CHIP TYPE, =0 FOR SMC, ELSE =1
1722  002402  000000            SAVLEN:  .WORD   0              ;SAVED TX AND RCV CHAR LENGTHS
1723  002404  000000            DEVMAP:  .WORD   0              ;BIT MAP OF ACTIVE DEVICES
1724  002406  000000            DEVPTR:  .WORD   0              ;DEVICE MAP BIT POINTER
1725  002410  000000            UNIT:    .WORD   0              ;CONTAINS UNIT NO. (1 TO N)
1726  002412  000000            STARES:  .WORD   0              ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
1727  002414  000000            TSTNUM:  .WORD   0              ;NO. OF CURRENT TEST (FOR SOME TESTS)
1728
```

```
1729
1730   002416
1731   002416
1732   002416  160020          ;;********** CURRENT DEVICE PARAMETERS  **************
1733   002416                  BSEL0:
1734   002416                  SEL0:
1735   002416  160020  MPCSR:  .WORD   160020          ;POINTER TO DMV11 CSR'S
1736   002420  160021  BSEL1:  .WORD   160021          ;POINTER TO BSEL1
1737   002422                  BSEL2:
1738   002422  160022  SEL2:   .WORD   160022          ;POINTER TO SEL2
1739   002424  160023  BSEL3:  .WORD   160023          ;POINTER TO BSEL3
1740   002426                  BSEL4:
1741   002426  160024  SEL4:   .WORD   160024          ;POINTER TO SEL4
1742   002430  160025  BSEL5:  .WORD   160025          ;POINTER TO BSEL5
1743   002432                  BSEL6:
1744   002432  160026  SEL6:   .WORD   160026          ;POINTER TO SEL6
1745   002434  160027  BSEL7:  .WORD   160027          ;POINTER TO BSEL7
1746   002436                  BSEL10:
1747   002436  160030  SEL10:  .WORD   160030          ;POINTER TO SEL10
1748   002440  160031  BSEL11: .WORD   160031          ;POINTER TO BSEL11
1749   002442                  BSEL12:
1750   002442  160032  SEL12:  .WORD   160032          ;POINTER TO SEL12
1751   002444  160033  BSEL13: .WORD   160033          ;POINTER TO BSEL13
1752   002446                  BSEL14:
1753   002446  160034  SEL14:  .WORD   160034          ;POINTER TO SEL14
1754   002450  160035  BSEL15: .WORD   160035          ;POINTER TO BSEL15
1755   002452                  BSEL16:
1756   002452  160036  SEL16:  .WORD   160036          ;POINTER TO SEL16
1757   002454  160037  BSEL17: .WORD   160037          ;POINTER TO BSEL17
1758
1759   002456  000300          MPIVEC: .WORD      300   ;DMV11 INPUT INTERRUPT VECTOR
1760   002460  000304          MPOVEC: .WORD      304   ;DMV11 OUTPUT INTERRUPT VECTOR
1761   002462  000240          MPRIOR: .WORD      240   ;DMV11 DEVICE PRIORITY
1762   002464  000000          LUSWI1: .WORD        0   ;LINE UNIT SWITCH PACK #1
1763   002466  000000          LUSWI2: .WORD        0   ;LINE UNIT SWITCH PACK #2
1764   002470  000000          BRDTYP: .WORD        0   ;0=M8064, 1=M8053/V.35,2=M8053/EIA
1765   002472  000000          TSTCON: .WORD        0   ;TEST CONNECTOR INDICATOR
       002474  000001          BDRATE: .WORD        1   ;BAUD RATE = 56 K
                                                        ;          0 = 19.2 K
                                                        ;          1 = 56 K
```

```
1766
1767  002476  120400          ;TABLE OF USYRT REGISTER ADDRESSES
1768  002500  120401  USYREG: .WORD   120400          ;ADDRESS OF RDSRL
1769  002502  120402          .WORD   120401          ;ADDRESS OF RDSRH
1770  002504  120403          .WORD   120402          ;ADDRESS OF TDSRL
1771  002506  120404          .WORD   120403          ;ADDRESS OF TDSRH
1772  002510  120405          .WORD   120404          ;ADDRESS OF PCSARL
1773  002512  120407          .WORD   120405          ;ADDRESS OF PCSARH
1774  002514  122000          .WORD   120407          ;ADDRESS OF PCR
1775                          .WORD   122000          ;ADDRESS OF USYRT STATUS REG
1776
1777  002516  000010  ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
                      REDDAT: .BLKB   8.
1778
1779                  ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
1780  002526  000000  REG0:   .WORD   0
1781  002530  000000  REG1:   .WORD   0
1782  002532  000000  REG2:   .WORD   0
1783  002534  000000  REG3:   .WORD   0
1784  002536  000000  REG4:   .WORD   0
1785  002540  000000  REG5:   .WORD   0
1786  002542  000000  REG6:   .WORD   0
1787  002544  000000  REG7:   .WORD   0
1788
1789                  ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
1790  002546  000000  TMP0:   .WORD   0
1791  002550  000000  TMP1:   .WORD   0
1792  002552  000000  TMP2:   .WORD   0
1793  002554  000000  TMP3:   .WORD   0
1794  002556  000000  TMP4:   .WORD   0
1795  002560  000000  TMP5:   .WORD   0
1796  002562  000000  TMP6:   .WORD   0
1797  002564  000000  TMP7:   .WORD   0
1798
1799                  ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
1800  002566  UPBITS:
1801  002566      377           .BYTE   377             ;MASK FOR RDBR
1802  002567      000           .BYTE   000             ;MASK FOR RDSR
1803  002570      000           .BYTE   000             ;MASK FOR TDBR
1804  002571      360           .BYTE   360             ;MASK FOR TDSR
1805  002572      000           .BYTE   000             ;MASK FOR SSAR
1806  002573      000           .BYTE   000             ;MASK FOR PCSAR
1807  002574      347           .BYTE   347             ;MASK FOR PCR
1808
1809  002575      200  TDSRNRW: .BYTE   200             ;TDSR NON-R/W BITS
```

```
1810
1811
1812  002576
1813  002576  377          .SBTTL  DATA TEST PATTERNS
1814  002577  377          ;***** DATA PATTERN E *****
1815  002600  377          PATE:
1816  002601  377                  .BYTE   377
1817  002602  377                  .BYTE   377
1818  002603  377                  .BYTE   377
1819  002604  377                  .BYTE   377
1820  002605  366                  .BYTE   377
1821                               .BYTE   377
1822                               .BYTE   377
1823  002606                       .BYTE   366
1824  002606  000
1825  002607  000          ;***** DATA PATTERN F *****
1826  002610  000          PATF:
1827  002611  000                  .BYTE   000
1828  002612  000                  .BYTE   000
1829  002613  000                  .BYTE   000
1830  002614  000                  .BYTE   000
1831  002615  110                  .BYTE   000
1832                               .BYTE   000
1833                               .BYTE   000
1834  002616                       .BYTE   110
1835  002616  000
1836  002617  001          ;***** DATA PATTERN G *****
1837  002620  003          PATG:
1838  002621  004                  .BYTE   000
1839  002622  005                  .BYTE   001
1840  002623  007                  .BYTE   003
1841  002624  100                  .BYTE   004
1842  002625  101                  .BYTE   005
1843  002626  103                  .BYTE   007
1844  002627  104                  .BYTE   100
1845  002630  105                  .BYTE   101
1846  002631  107                  .BYTE   103
1847  002632  000                  .BYTE   104
1848  002633  017                  .BYTE   105
1849  002634  027                  .BYTE   107
1850  002635  041                  .BYTE   000
1851  002636  200                  .BYTE   017
1852  002637  277                  .BYTE   027
1853  002640  103                  .BYTE   041
1854  002641  144                  .BYTE   200
1855  002642  115                  .BYTE   277
1856  002643  157                  .BYTE   103
1857  002644  000                  .BYTE   144
1858                               .BYTE   115
1859                               .BYTE   157
1860  002645                       .BYTE   000
1861  002645  125
1862  002646  252          ;***** DATA PATTERN X *****
1863  002647  000          PATX:
1864  002650  377                  .BYTE   125
1865  002651  001                  .BYTE   252
                                   .BYTE   000
                                   .BYTE   377
                                   .BYTE   001
```

```
1866  002652  002                    .BYTE   002
1867  002653  004                    .BYTE   004
1868  002654  010                    .BYTE   010
1869  002655  020                    .BYTE   020
1870  002656  040                    .BYTE   040
1871  002657  100                    .BYTE   100
1872  002660  200                    .BYTE   200
1873  002661  376                    .BYTE   376
1874  002662  375                    .BYTE   375
1875  002663  373                    .BYTE   373
1876  002664  367                    .BYTE   367
1877  002665  357                    .BYTE   357
1878  002666  337                    .BYTE   337
1879  002667  277                    .BYTE   277
1880  002670  177                    .BYTE   177
1881
1882  002671  125                    .BYTE   125
1883  002672  252                    .BYTE   252
1884  002673  000                    .BYTE   000
1885  002674  377                    .BYTE   377
1886  002675  001                    .BYTE   001
1887  002676  002                    .BYTE   002
1888  002677  004                    .BYTE   004
1889  002700  010                    .BYTE   010
1890  002701  020                    .BYTE   020
1891  002702  040                    .BYTE   040
1892  002703  100                    .BYTE   100
1893  002704  200                    .BYTE   200
1894  002705  376                    .BYTE   376
1895  002706  375                    .BYTE   375
1896  002707  373                    .BYTE   373
1897  002710  367                    .BYTE   367
1898  002711  357                    .BYTE   357
1899  002712  337                    .BYTE   337
1900  002713  277                    .BYTE   277
1901  002714  177                    .BYTE   177
1902
1903  002715  125                    .BYTE   125
1904  002716  252                    .BYTE   252
1905  002717  000                    .BYTE   000
1906  002720  377                    .BYTE   377
1907  002721  001                    .BYTE   001
1908  002722  002                    .BYTE   002
1909  002723  004                    .BYTE   004
1910  002724  010                    .BYTE   010
1911  002725  020                    .BYTE   020
1912  002726  040                    .BYTE   040
1913  002727  100                    .BYTE   100
1914  002730  200                    .BYTE   200
1915  002731  376                    .BYTE   376
1916  002732  375                    .BYTE   375
1917  002733  373                    .BYTE   373
1918  002734  367                    .BYTE   367
1919  002735  357                    .BYTE   357
1920  002736  337                    .BYTE   337
1921  002737  277                    .BYTE   277
```

CVDMEA.P11 10-DEC-80 09:16

```
1922  002740      177
1923
1924
1925  002741
1926  002741      000
1927  002742      041
1928  002743      102
1929  002744      143
1930  002745      204
1931  002746      245
1932  002747      306
1933  002750      347
1934  002751      000
1935  002752      001
1936  002753      002
1937  002754      004
1938  002755      040
1939  002756      100
1940  002757      200
1941  002760      000
1942  002761      346
1943  002762      345
1944  002763      343
1945  002764      307
1946  002765      247
1947  002766      147
1948  002767      347
1949  002770      242
1950  002771      105
1951  002772      347
1952  002773      010
1953  002774      020
1954  002775      367
1955  002776      357
1956  002777      030
1957  003000      027
1958  003001      377
1959
1960
1961  003002
1962  003002      000
1963  003003      000
1964  003004      001
1965  003005      002
1966  003006      004
1967  003007      020
1968  003010      040
1969  003011      010
1970
1971
1972  003012
1973  003012      000
1974  003013      377
1975  003014      376
1976  003015      375
1977  003016      373
```

DATA TEST PATTERNS

```
EPATX:  .BYTE   177

;***** DATA PATTERN I *****
PATI:
        .BYTE   000
        .BYTE   041
        .BYTE   102
        .BYTE   143
        .BYTE   204
        .BYTE   245
        .BYTE   306
        .BYTE   347
        .BYTE   000
        .BYTE   001
        .BYTE   002
        .BYTE   004
        .BYTE   040
        .BYTE   100
        .BYTE   200
        .BYTE   000
        .BYTE   346
        .BYTE   345
        .BYTE   343
        .BYTE   307
        .BYTE   247
        .BYTE   147
        .BYTE   347
        .BYTE   242
        .BYTE   105
        .BYTE   347
        .BYTE   010
        .BYTE   020
        .BYTE   367
        .BYTE   357
        .BYTE   030
        .BYTE   027
        .BYTE   377

;***** DATA PATTERN J *****
PATJ:
        .BYTE   000
        .BYTE   000
        .BYTE   001
        .BYTE   002
        .BYTE   004
        .BYTE   020
        .BYTE   040
        .BYTE   010

;***** DATA PATTERN K *****
PATK:
        .BYTE   000
        .BYTE   377
        .BYTE   376
        .BYTE   375
        .BYTE   373
```

```
1978  003017  376                .BYTE   376
1979  003020  177                .BYTE   177
1980  003021  377                .BYTE   377
1981  003022  000                .BYTE   000
1982  003023  001                .BYTE   001
1983  003024  002                .BYTE   002
1984  003025  004                .BYTE   004
1985  003026  010                .BYTE   010
1986  003027  200                .BYTE   200
1987  003030  125                .BYTE   125
1988  003031  252                .BYTE   252
1989  003032  000                .BYTE   000
1990
1991                             ;***** DATA PATTERN L *****
1992  003033                     PATL:
1993  003033  000                .BYTE   000
1994  003034  017                .BYTE   017
1995  003035  016                .BYTE   016
1996  003036  015                .BYTE   015
1997  003037  013                .BYTE   013
1998  003040  016                .BYTE   016
1999  003041  017                .BYTE   017
2000  003042  017                .BYTE   017
2001  003043  000                .BYTE   000
2002  003044  001                .BYTE   001
2003  003045  002                .BYTE   002
2004  003046  004                .BYTE   004
2005  003047  010                .BYTE   010
2006  003050  000                .BYTE   000
2007  003051  005                .BYTE   005
2008  003052  012                .BYTE   012
2009  003053  000                .BYTE   000
```

CVDMEA.P11 10-DEC-80 09:16 DATA TEST PATTERNS

```
2010
2011                                    ;***** DATA PATTERN Q *****
2012  003054        000                 PATQ:   .BYTE   000
2013  003055        002                         .BYTE   002
2014  003056        014                         .BYTE   014
2015  003057        060                         .BYTE   060
2016  003060        001                         .BYTE   001
2017  003061        007                         .BYTE   007
2018  003062        037                         .BYTE   037
2019  003063        177                         .BYTE   177
2020
2021
2022  003064                            ENDPAT:
2023                                    .EVEN
```

```
2024
2025
2026
2027                                    ;*** RECEIVED DATA BUFFER (64. WORDS) ***
2028  003064  000100                    RCVBUF: .BLKW   64.
2029
2030
2031
2032
```

```
                                        .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  034115  032460  020063                                                                  .ASCIZ  /M8053 OR M8064/
003272  051117  046440  030070
003300  032066     000
        003304                                                                                  .EVEN

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

        000012                          .RADIX  10.
003304                                          DESCRIPT        <DMV-11 LINE UNIT TESTS - PART 3 OF 3>
003304                                                                                  L$DESC::
003304  046504  026526  030461                                                                  .ASCIZ  /DMV-11 LINE UNI
003312  046040  047111  020105
003320  047125  052111  052040
003326  051505  051524  026440
003334  050040  051101  020124
003342  020063  043117  031440
003350     000
        003352                                                                                  .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,@BSEL1   ;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,@BSEL2    ;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,GDATA    ;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,@SEL2   ;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,@SEL2     ;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    @SEL6,@(R5)+    ;PUT DATA WHERE CALLER WANTS IT
003564  000205                                  RTS     R5              ;RETURN
```

```
                                        .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
```

```
2208                                    .SBTTL ....M-LOOP -- WRITE
2209                                    ;++*************************************************************************
2210                                    ; WRITE - WRITE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
2211                                    ;
2212                                    ;  CALLING SEQUENCE:
2213                                    ;
2214                                    ;       JSR     R5,WRITE
2215                                    ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
2216                                    ;       .WORD   <ADDRESS OF DATA BYTE>
2217                                    ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2218                                    ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
2219                                    ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
2220                                    ;
2221                                    ; N$:   <RESUMPTION OF NORMAL PROCESSING>
2222                                    ;
2223                                    ;--*************************************************************************
2224
2225  003700  012577  176522   WRITE:  MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
2226  003704  113577  176522           MOVB    @(R5)+,@SEL6    ;MAKE DATA AVAILABLE TO M-LOOP
2227  003710  000404                   BR      MLWRI           ;THE REST OF THIS ROUTINE IS THE SAME AS 'WRITEI'
2228
2229
2230
2231
```

```
                                            .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
                                            ;       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>
                                            ;
                                            ;
                                            ;--*************************************************************************

003712                                      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
```

```
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294  004024  117737  176366  002202  GETBSR: MOVB    @#SEL0,@#SR0      ;PUT THE CURRENT CSR VALUES INTO THE PRINT-OUT
2295  004032  117737  176362  002204          MOVB    @#SEL1,@#SR1      ;TABLE
2296  004040  117737  176356  002206          MOVB    @#SEL2,@#SR2
2297  004046  117737  176352  002210          MOVB    @#SEL3,@#SR3
2298  004054  117737  176346  002212          MOVB    @#SEL4,@#SR4
2299  004062  117737  176342  002214          MOVB    @#SEL5,@#SR5
2300  004070  117737  176336  002216          MOVB    @#SEL6,@#SR6
2301  004076  117737  176332  002220          MOVB    @#SEL7,@#SR7
2302  004104  117737  176326  002222          MOVB    @#SEL10,@#SR10
2303  004112  117737  176322  002224          MOVB    @#SEL11,@#SR11
2304  004120  117737  176316  002226          MOVB    @#SEL12,@#SR12
2305  004126  117737  176312  002230          MOVB    @#SEL13,@#SR13
2306  004134  117737  176306  002232          MOVB    @#SEL14,@#SR14
2307  004142  117737  176302  002234          MOVB    @#SEL15,@#SR15
2308  004150  117737  176276  002236          MOVB    @#SEL16,@#SR16
2309  004156  117737  176272  002240          MOVB    @#SEL17,@#SR17
2310  004164  000207                          RTS     PC                ;RETURN TO CALLER
2311
2312                                  .SBTTL ....GETWSR -- GET WORD SELECT REGISTERS
2313                                  ; 'WORD'' VERSION OF ABOVE SUBROUTINE
2314
2315  004166  017737  176224  002202  GETWSR: MOV     @#SEL0,@#WSR0     ;MOVE THE 4 WORD REGISTERS TO THE OTHERWISE
2316  004174  017737  176222  002204          MOV     @#SEL2,@#WSR2     ;BYTE TABLE
2317  004202  017737  176220  002206          MOV     @#SEL4,@#WSR4
2318  004210  017737  176216  002210          MOV     @#SEL6,@#WSR6
2319  004216  017737  176214  002212          MOV     @#SEL10,@#WSR10
2320  004224  017737  176212  002214          MOV     @#SEL12,@#WSR12
2321  004232  017737  176210  002216          MOV     @#SEL14,@#WSR14
2322  004240  017737  176206  002220          MOV     @#SEL16,@#WSR16
2323  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
```

CVDMEA.P11 10-DEC-80 09:16 ....STALL -- DELAY FOR 10.5 MICRO-SEC'S (ON LSI-11) SEQ 58

```
2380
```

```
2381                                    .SBTTL
2382
2383                                    ;*********************************************************************************
2384                                    ;* GETURS - LOAD INTO THE 8 WORD STORAGE AREA (UREGS) THE CONTENTS OF THE
2385                                    ;*       VARIOUS USYRT REGISTERS
2386                                    ;*
2387                                    ;*          CALLING SEQUENCE:
2388                                    ;*
2389                                    ;*********************************************************************************
2390  004360  012737  002242  004422   GETURS: MOV     #UREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
2391  004366  012737  120400  004420           MOV     #USYRT,4$       ;INIT POINTER TO REGISTER ADDRESSES
2392
2393  004374  005037  002260                   CLR     UREGS+14.       ;CLEAR STORAGE WORD
2394  004400  004537  003454                   JSR     R5,READ         ;READ THE USYRT STATUS REGISTER
2395  004404  122000                           .WORD   USTATR          ;STATUS REGISTER'S ADDRESS WITHIN DMV-11
2396  004406  002260                           .WORD   UREGS+14.       ;ADDRESS ALLOCATED TO THAT REG. W/IN 'UREGS'
2397
2398  004410  005077  000006           3$:     CLR     @5$             ;CLEAR STORAGE WORD
2399  004414  004537  003454                   JSR     R5,READ         ;READ A LINE UNIT REG
2400  004420  000000           4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
2401  004422  000000           5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
2402
2403  004424  005237  004420           6$:     INC     4$              ;INCREMENT REG NO.
2404  004430  023727  004420  120406           CMP     4$,#USYRT+6     ;THIS IS NOT A VALID REGISTER ADDRESS
2405  004436  001772                           BEQ     6$              ;SO IT MUST BE BYPASSED
2406
2407  004440  062737  000002  004422           ADD     #2,5$           ;ADVANCE ADDRESS OF STORAGE AREA POINTER
2408  004446  023727  004420  120410           CMP     4$,#USYRT+10    ;SEE IF ALL REGS READ YET
2409  004454  001355                           BNE     3$              ;BR IF NOT
2410
2411  004456  000207                           RTS     PC              ;RETURN
2412
2413
2414
2415                                    ;*********************************************************************************
2416                                    ;* GETVRS: - LOAD INTO THE 16 WORD STORAGE AREA (VREGS) THE CONTENTS OF THE
2417                                    ;*       VARIOUS VIA REGISTERS.
2418                                    ;*
2419                                    ;*       CALLING SEQUENCE :
2420                                    ;*********************************************************************************
2421  004460  012737  002262  004506   GETVRS: MOV     #VREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
2422  004466  012737  120000  004504           MOV     #VIA,4$         ;INIT POINTER TO REGISTER ADDRESSES
2423  004474  005077  000006           3$:     CLR     @5$             ;CLEAR STORAGE WORD
2424  004500  004537  003454                   JSR     R5,READ         ;READ A VIA REG
2425  004504  000000           4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
2426  004506  000000           5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
2427  004510  005237  004504           6$:     INC     4$              ;INCREMENT REG NO.
2428  004514  062737  000002  004506           ADD     #2,5$           ;INCREMENT STORAGE ADRS
2429  004522  023727  004504  120020           CMP     4$,#VIA+16.     ;SEE IF ALL VIA REGS READ YET
2430  004530  001361                           BNE     3$              ;BR IF NOT
2431  004532  000207                           RTS     PC              ;RETURN
```

```
                                .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    077,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
```

```
2488  004636  010137  004650              MOV     R1,4$           ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
2489
2490  004642  004537  003712              JSR     R5,WRITEI       ;WRITE TO
2491  004646  120013                      VIAACR                  ;THE VIA'S ACR
2492  004650  000000              4$:     .WORD   0               ;THE NEW REGISTER SETTING
2493
2494  004652  004537  003712              JSR     R5,WRITEI       ;WRITE TO
2495  004656  120006                      VIAT1C                  ;LOW ORDER LATCH REGISTER (T1L-L)
2496  004660  000000              7$:     .WORD   0               ;THE VALUE PASSED
2497
2498  004662  113737  004661  004676      MOVB    7$+1,8$         ;SETUP FOR AND
2499  004670  004537  003712              JSR     R5,WRITEI       ;WRITE TO
2500  004674  120007                      VIAT1D                  ;HIGH ORDER LATCH REGESTER (T1L-H)
2501  004676  000000              8$:     .WORD   0               ;THE VALUE PASSED
2502
2503  004700  004537  003712              JSR     R5,WRITEI       ;WRITE TO
2504  004704  120004                      VIAT1A                  ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2505  004706  000000              10$:    .WORD   0               ;THE VALUE PASSED
2506
2507  004710  113737  004707  004724      MOVB    10$+1,11$       ;SETUP FOR AND
2508  004716  004537  003712              JSR     R5,WRITEI       ;WRITE TO
2509  004722  120005                      VIAT1B                  ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2510  004724  000000              11$:    .WORD   0               ;THE VALUE PASSED
2511
2512                                      ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2513
2514  004726  012601                      MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2515  004730  005205                      INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2516                                                              ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2517
2518  004732  000205                      RTS     R5              ;NOW, RETURN
2519
2520
```

```
                                .SBTTL ....INITT2 -- INITIALIZE TIMER #2
                                ;*************************************************************************
                                ;* INITT2 - INITIALIZE TIMER # 2
                                ;*
                                ;*        CALLING SEQUENCE:
                                ;*
                                ;*              JSR     R5,INITT2
                                ;*              .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          INITT2: 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
```

```
2577  005040  004537  003712            JSR     R5,WRITEI       ;WRITE TO
2578  005044  120013                    VIAACR                  ;THE VIA'S ACR
2579  005046  000000            4$:     .WORD   0               ;THE NEW REGISTER SETTING
2580
2581  005050  004537  003712            JSR     R5,WRITEI       ;WRITE TO
2582  005054  120010                    VIAT2A                  ;LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
2583  005056  000000            10$:    .WORD   0               ;THE VALUE PASSED
2584
2585  005060  113737  005057  005074    MOVB    10$+1,11$       ;SETUP FOR AND
2586  005066  004537  003712            JSR     R5,WRITEI       ;WRITE TO
2587  005072  120011                    VIAT2B                  ;HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
2588  005074  000000            11$:    .WORD   0               ;THE VALUE PASSED
2589
2590                            ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2591
2592  005076  012601                    MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2593  005100  005205                    INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2594                                                            ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2595
2596  005102  000205                    RTS     R5              ;THEN RETURN
2597
```

```
                        .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$:     ADD     #2,R1           ;INCR USYRT REG ADRS PTR
005252  020127  000020          CMP     R1,#16.         ;SEE IF ALL REGS READ YET
005256  002727                  BLT     6$              ;BR 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
```

```
                                ;*************************************************************************
                                ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
                                ;*************************************************************************
005270  010146          WAIT50: MOV     R1,-(SP)        ;SAVE R1
005272  012701  000005          MOV     #5.,R1          ;INIT COUNTER
005276  077101          3$:     SOB     R1,3$           ;DELAY HERE FOR 23.8 MICRO-SEC'S
005300  012601          MOV     (SP)+,R1        ;RESTORE R1
005302  000207          RTS     PC              ;RETURN

                        ;       OVERHEAD (JSR, MOV, MOV, MOV, & RTS) ADD UP TO 25.25 MICRO-SEC'S

                        ;       THEREFORE, ACTUAL TOTAL DELAY IS 49.35 MICRO-SECONDS



                        .SBTTL ....SETVIA -- SET UP VIA REGISTERS
                        ;*************************************************************************
                        ;* SETVIA - SET UP THE VIA REGISTERS
                        ;*
                        ;*      THIS SUBROUTINE PROGRAMS THE VIA REGISTERS FOR NORMAL OPERATION, BY
                        ;*      LOADING THE DDRB, DDRA, ORB, ACR, PCR, IER.
                        ;*
                        ;*      CALLING SEQUENCE :
                        ;*         JSR  PC,SETVIA
                        ;*************************************************************************
005304                  SETVIA:
005304  004537  003712          JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
005310  120002          VIADPB
005312  000377          377
005314  004537  003712          JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
005320  120003          VIADPA                  ; (BIT0 IS ONLY OUTPUT BIT)
005322  000001          001
005324  004537  003712          JSR     R5,WRITEI       ;DISABLE USYRT INTERNAL LOOPBACK
005330  120017          VIAORA
005332  000000          000
005334  004537  003712          JSR     R5,WRITEI       ;INIT PORT B
005340  120000          VIAORB
005342  000030          DTR!RTSND
005344  004537  003712          JSR     R5,WRITEI       ;SET ACR FOR :  T1 SQUARE WAVE OUTPUT MODE,
005350  120013          VIAACR                  ;               T2 ONE-SHOT OUTPUT MODE,
005352  000350          350                     ;               SR AT SYS CLOCK RATE ON CB1
005354  004537  003712          JSR     R5,WRITEI       ;SET PCR FOR :  CB1 NEG TRANS INPUT MODE,
005360  120014          VIAPCR                  ;               CA2 NEG TRANS INPUT MODE,
005362  000022          022                     ;               CA1 NEG TRANS INPUT MODE
005364  004537  003712          JSR     R5,WRITEI       ;DISABLE ALL MICRO-INTRPTS
005370  120016          VIAIER
005372  000177          177
005374  000207          RTS     PC              ;RETURN
```

```
                        .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$,BDATA
                        ;STACK 'USYRT STATUS INCORRECT'' ERROR
005456                  GTDF    EM68,ERR10
                        ;       QUEUE 'DEVICE FATAL'' ERROR # 7
005456  012737  000001  002172          MOV     #T.EDF,ERRTYP
005464  012737  000007  002174          MOV     #7,ERRNBR
005472  012737  014650  002176          MOV     #EM68,ERRMSG
005500  012737  020364  002200          MOV     #ERR10,ERRBLK
005506  000261                  SEC                     ;SET C BIT FOR ERROR
005510  005205          2$:     INC     R5              ;INCREMENT R5 PAST ARGUMENT
005512  000205                  RTS     R5              ;RETURN
```

```
2751                                    .SBTTL ....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)
2752                                    ;*************************************************************************
2753                                    ;* CKTACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TXACT IN THE USYRT
2754                                    ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
2755                                    ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
2756                                    ;*
2757                                    ;*      CALLING SEQUENCE :
2758                                    ;*      JSR     R5,CKTACT
2759                                    ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TXACT>
2760                                    ;*************************************************************************
2761  005514                            CKTACT:
2762  005514  012737  000007  002336            MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
2763  005522  004537  003566                    JSR     R5,READI        ;READ USYRT STATUS
2764  005526  122000                            USTATR
2765  005530  000000                    1$:     .WORD   0
2766  005532  032725  000001                    BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TXACT
2767  005536  001422                            BEQ     2$              ;BR IF EXPECTED TXACT = 0
2768  005540  132737  000004  005530            BITB    #TXACT,1$       ;SEE IF TXACT = 1
2769  005546  001040                            BNE     3$              ;BR IF TXACT = 1
2770                                    ;STACK 'TXACT NOT SET'' MSG
2771  005550                                    GTDF    EM69,ERR12
2772                                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 8
2773  005550  012737  000001  002172                                                      MOV     #T.EDF,ERRTYP
2774  005556  012737  000010  002174                                                      MOV     #8,ERRNBR
2775  005564  012737  014677  002176                                                      MOV     #EM69,ERRMSG
2776  005572  012737  020714  002200                                                      MOV     #ERR12,ERRBLK
2777  005600  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2778  005602  000423                            BR      4$              ;TAKE ERROR EXIT
2779  005604  132737  000004  005530    2$:     BITB    #TXACT,1$       ;SEE IF TXACT = 0
2780  005612  001416                            BEQ     3$              ;BR IF TXACT = 0
2781                                    ;STACK 'TXACT NOT CLEARED'' MSG
2782  005614                                    GTDF    EM70,ERR12
2783                                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 9
2784  005614  012737  000001  002172                                                      MOV     #T.EDF,ERRTYP
2785  005622  012737  000011  002174                                                      MOV     #9,ERRNBR
2786  005630  012737  014715  002176                                                      MOV     #EM70,ERRMSG
2787  005636  012737  020714  002200                                                      MOV     #ERR12,ERRBLK
2788  005644  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2789  005646  000401                            BR      4$              ;TAKE ERROR EXIT
2790  005650  000241                    3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
2791  005652  000205                    4$:     RTS     R5              ;RETURN
2792
2793
2794
2795
```

```
                                .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
```

```
                                .SBTTL ....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY
                                ;*****************************************************************************
                                ;* CKTBMT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TBMT 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,CKTBMT
                                ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TBMT>
                                ;*****************************************************************************
006014                          CKTBMT:
006014  012737  000007  002336          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006022  004537  003566                  JSR     R5,READI        ;READ USYRT STATUS
006026  122000                          USTATR
006030  000000                  1$:     .WORD   0
006032  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TBMT
006036  001422                          BEQ     2$              ;BR IF EXPECTED TBMT = 0
006040  132737  000100  006030          BITB    #TBMT,1$        ;SEE IF TBMT = 1
006046  001040                          BNE     3$              ;BR IF TBMT = 1
                                ;STACK 'TBMT NOT SET'' MSG
006050                                  GTDF    EM73,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 12
006050  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006056  012737  000014  002174                                                  MOV     #12,ERRNBR
006064  012737  014777  002176                                                  MOV     #EM73,ERRMSG
006072  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006100  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006102  000423                          BR      4$              ;TAKE ERROR EXIT
006104  132737  000100  006030  2$:     BITB    #TBMT,1$        ;SEE IF TBMT = 0
006112  001416                          BEQ     3$              ;BR IF TBMT = 0
                                ;STACK 'TBMT NOT CLEARED'' MSG
006114                                  GTDF    EM74,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 13
006114  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
006122  012737  000015  002174                                                  MOV     #13,ERRNBR
006130  012737  015014  002176                                                  MOV     #EM74,ERRMSG
006136  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
006144  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006146  000401                          BR      4$              ;TAKE ERROR EXIT
006150  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006152  000205                  4$:     RTS     R5              ;RETURN
```

```
                                .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
```

```
3074  007032  001416                    BEQ     5$              ;BR IF REOM = 0
3075                              ;STACK 'REOM NOT CLEARED'' MSG
3076  007034                            GTDF    EM30,ERR12
3077                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 23
3078  007034  012737  000001  002172                                    MOV     #T.EDF,ERRTYP
3079  007042  012737  000027  002174                                    MOV     #23,ERRNBR
3080  007050  012737  014444  002176                                    MOV     #EM30,ERRMSG
3081  007056  012737  020714  002200                                    MOV     #ERR12,ERRBLK
3082  007064  000261                    SEC                     ;SET C BIT TO FLAG ERROR
3083  007066  000401                    BR      6$              ;TAKE ERROR EXIT
3084  007070  000241              5$:   CLC                     ;CLEAR C BIT FOR NO ERRORS
3085  007072  000205              6$:   RTS     R5              ;RETURN
3086
3087
```

```
                                .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
```

```
                                        VIAORB
007414  000142          13$:    TXEN!RXEN!TTLOOP        ;* THIS VALUE MIGHT BE MODIFIED ABOVE

007416  004537  003712          JSR     R5,WRITEI       ;SET TSOM IN USYRT
007422  120403                  TDSRH
007424  000001                  TSOM
007426  004537  003712          JSR     R5,WRITEI       ;LOAD SYNCH CHAR INTO TX BUF
007432  120402                  TDSRL
007434  000226                  SYNCH
007436  004537  006014          JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
007442  000000                  0
007444  103427                  BCS     7$              ;IF ERROR, EXIT SUBROUTINE
007446  005001                  CLR     R1              ;INIT CYCLE COUNTER
007450  004537  012072  4$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
007454  000001                  1
007456  004537  003566          JSR     R5,READI        ;READ USYRT STATUS REG
007462  122000                  USTATR
007464  000000          5$:     .WORD   0
007466  132737  000100  007464  BITB    #TBMT,5$        ;SEE IF TBMT IS SET YET
007474  001010                  BNE     6$              ;BR IF YES
007476  005201                  INC     R1              ;INCR CYCLE COUNTER
007500  020127  000003          CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
007504  002761                  BLT     4$              ;BR IF LESS THAN 3 CYCLES
007506  004537  006014          JSR     R5,CKTBMT       ;GO STACK 'TBMT NOT SET'' MSG
007512  000001                  1
007514  103403                  BCS     7$              ;IF ERROR, EXIT SUBROUTINE
007516  004537  005514  6$:     JSR     R5,CKTACT       ;CHK FOR TXACT = 1
007522  000001                  1
007524  012601          7$:     MOV     (SP)+,R1        ;RESTORE R1
007526  000205                  RTS     R5              ;RETURN (IF C = 1, WE HAD AN ERROR)

007530  000142          20$:    TXEN!RXEN!TTLOOP        ;DEFAULT VALUE FOR VIAORB: ENABLE
                                                        ;TX AND RX ON USYRT, ASSERT RTS, DTR
```

```
                                .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
```

```
3277  007670  005737  002472          18$:    TST     TSTCON          ;SEE IF H3254, H3255 USED
3278  007674  001002                          BNE     23$             ;BR IF NOT
3279  007676  000137  010072          20$:    JMP     46$             ;GO RUN THE TEST
3280  007702  023727  002472  000003  23$:    CMP     TSTCON,#3       ;SEE IF OPERATOR SPEC'D V.35
3281  007710  001006                          BNE     28$             ;BR IF NO
3282  007712  023727  002470  000001          CMP     BRDTYP,#1       ;TSTCOM MATCH BRDTYP?
3283  007720  001766                          BEQ     20$             ;YES: RUN TEST
3284  007722  000137  010006                  JMP     40$             ;WRONG INTERFACE, GO SKIP TEST
3285
3286  007726  023727  002472  000002  28$:    CMP     TSTCON,#2       ;SEE IF OPERATOR SPEC'D EIA
3287  007734  001760                          BEQ     20$             ;BR IF YES, TO RUN EIA
3288  007736  000137  010006                  JMP     40$             ;WRONG INTERFACE, GO SKIP TEST
3289
3290                                  ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
3291
3292  007742  005737  002470          32$:    TST     BRDTYP          ;SEE IF INTEGRAL MODEM SELECTED
3293  007746  001343                          BNE     16$             ;BR IF NOT (TEST FOR V35/EIA)
3294  007750  000137  007652                  JMP     12$             ;SEE IF INTEGRAL MODEM CAN BE RUN
3295
3296                                  ;PRINT 'FOR BAUD RATE SPECIFIED,"
3297  007754                          38$:
3298  007754  023727  002412  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3299  007762  001063                          BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3300  007764                                  PRINTF  #FMT30
3301  007764  012746  013762                                                  MOV     #FMT30,-(SP)
3302  007770  012746  000001                                                  MOV     #1,-(SP)
3303  007774  010600                                                          MOV     SP,R0
3304  007776  104417                                                          TRAP    C$PNTF
3305  010000  062706  000004                                                  ADD     #4,SP
3306  010004  000434                          BR      48$             ;GO TO PRINT 'TEST NOT RUN"
3307                                  ;PRINT "IMPROPER CONNECTOR TYPE SPECIFIED"
3308  010006                          40$:
3309  010006  023727  002412  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3310  010014  001046                          BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3311  010016                                  PRINTF  #FMT31          ; *********
3312  010016  012746  014017                                                  MOV     #FMT31,-(SP)
3313  010022  012746  000001                                                  MOV     #1,-(SP)
3314  010026  010600                                                          MOV     SP,R0
3315  010030  104417                                                          TRAP    C$PNTF
3316  010032  062706  000004                                                  ADD     #4,SP
3317  010036  000417                          BR      48$             ;GO TO PRINT 'TEST NOT RUN"
3318
3319                                  ;PRINT 'FOR OPTION SPECIFIED,"
3320  010040  023727  002412  000001  42$:    CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3321  010046  001031                          BNE     50$             ;BR IF NOT TO SKIP PRINTING
3322  010050                                  PRINTF  #FMT32
3323  010050  012746  014065                                                  MOV     #FMT32,-(SP)
3324  010054  012746  000001                                                  MOV     #1,-(SP)
3325  010060  010600                                                          MOV     SP,R0
3326  010062  104417                                                          TRAP    C$PNTF
3327  010064  062706  000004                                                  ADD     #4,SP
3328  010070  000402                          BR      48$             ;GO TO PRINT 'TEST NOT RUN"
3329
3330                                  ;*** BRANCH HERE TO RUN TEST ***
3331  010072  000241          46$:    CLC                             ;CLEAR C BIT TO RUN TEST
3332  010074  000417                  BR      52$                     ;EXIT
```

```
3333
3334                                    ;*** BRANCH HERE TO SKIP TEST ***
3335                                    ;PRINT 'TEST XX NOT RUN'
3336  010076                            48$:
3337  010076  023727  002412  000001            CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3338  010104  001012                            BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3339  010106                                    PRINTF  #FMT19,TSTNUM
3340  010106  013746  002414                                                    MOV     TSTNUM,-(SP)
3341  010112  012746  013452                                                    MOV     #FMT19,-(SP)
3342  010116  012746  000002                                                    MOV     #2,-(SP)
3343  010122  010600                                                            MOV     SP,R0
3344  010124  104417                                                            TRAP    C$PNTF
3345  010126  062706  000006                                                    ADD     #6,SP
3346  010132  000261                    50$:    SEC                     ;SET C BIT TO SKIP TEST
3347  010134  000205                    52$:    RTS     R5              ;RETURN
3348
3349
3350
```

```
                        .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)
```

CVDMEA.P11 10-DEC-80 09:16 ....TXCTRL -- CONTROL MESSAGE TRANSMISSION (TDSRH)

```
3405                                    .SBTTL ....TXCTRL -- CONTROL MESSAGE TRANSMISSION (TDSRH)
3406                                    ;*****************************************************************************
3407                                    ;* TXCTRL - THIS SUBROUTINE ALLOWS CONTROL OF MESSAGE TRANSMISSION BY LOADING
3408                                    ;*       THE USYRT TDSRH WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
3409                                    ;*       FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
3410                                    ;*       PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
3411                                    ;*       MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
3412                                    ;*       IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
3413                                    ;*       IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
3414                                    ;*       DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
3415                                    ;*
3416                                    ;*       CALLING SEQUENCE :
3417                                    ;*       JSR     R5,TXCTRL
3418                                    ;*       .WORD   <DATA FOR TDSRH IN LO BYTE>
3419                                    ;*       .WORD   <NUMBER OF CYCLES TO CLOCK>
3420                                    ;*****************************************************************************
3421  010250                            TXCTRL:
3422  010250  010146                           MOV     R1,-(SP)        ;SAVE R1
3423  010252  010246                           MOV     R2,-(SP)        ;SAVE R2
3424  010254  012537  010266                   MOV     (R5)+,2$        ;GET DATA FOR TDSRH
3425  010260  004537  003712                   JSR     R5,WRITEI       ;LOAD DATA INTO TDSRH
3426  010264  120403                           TDSRH
3427  010266  000000                   2$:     .WORD   0
3428  010270  005001                           CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
3429  010272  012502                           MOV     (R5)+,R2        ;GET DESIRED NO. OF CYCLES
3430  010274  001422                           BEQ     6$              ;BR IF NO CLOCKING DONE
3431  010276  004537  005514           3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
3432  010302  000001                           1
3433  010304  103416                           BCS     6$              ;BR TO EXIT IF ERROR
3434  010306  020102                           CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3435  010310  001411                           BEQ     5$              ;BR IF YES
3436  010312  004537  006014                   JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
3437  010316  000000                           0
3438  010320  103410                           BCS     6$              ;BR TO EXIT IF ERROR
3439  010322  004537  012072                   JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3440  010326  000001                           1
3441  010330  005201                           INC     R1              ;INCR CYCLE COUNT
3442  010332  000761                           BR      3$              ;KEEP CLOCKING
3443  010334  004537  006014           5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
3444  010340  000001                           1
3445  010342  012602                   6$:     MOV     (SP)+,R2        ;RESTORE R2
3446  010344  012601                           MOV     (SP)+,R1        ;RESTORE R1
3447  010346  000205                           RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
3448
3449
```

```
                        .SBTTL ....RXCHAR -- RECEIVE A CHARACTER
                        ;*************************************************************************
                        ;* RXCHAR - THIS SUBROUTINE READS THE USYRT RDSR AND CHECKS THE CONTENTS
                        ;*       AGAINST THE DATA PASSED IN THE WORD FOLLOWING THE CALL.
                        ;*       IF BIT0 = 0 IN THE SECOND WORD FOLLOWING THE CALL, THE RERR BIT IS
                        ;*       NOT CHECKED AGAINST THE EXPECTED VALUE. THEN, IT CLOCKS
                        ;*       THE LINE UNIT FOR THE NO. OF CYCLES PASSED IN THE THIRD WORD
                        ;*       FOLLOWING THE CALL. THE PROGRAM CONTINUALLY MONITORS RDA AND RXACT.
                        ;*       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,RXCHAR
                        ;*       .WORD   <EXPECTED RDSRL IN LO BYTE, RDSRH IN HI BYTE>
                        ;*       .WORD   <=0 FOR NO RERR CHK, =1 FOR RERR CHK>
                        ;*       .WORD   <NUMBER OF CYCLES TO CLOCK (IN LO BYTE)>
                        ;*               <SPECIAL DISABLE SWITCHES: NOCRDA,NFCRDA,NCRACT(IN HI BYTE)>
                        ;*************************************************************************
010350                  RXCHAR:
010350  010146                  MOV     R1,-(SP)        ;SAVE R1
010352  010246                  MOV     R2,-(SP)        ;SAVE R2
010354  004537  003566          JSR     R5,READI        ;READ RDSRH
010360  120401                  RDSRH
010362  000000          2$:     .WORD   0
010364  004537  003566          JSR     R5,READI        ;READ RDSRL
010370  120400                  RDSRL
010372  000000          1$:     .WORD   0
010374  111501                  MOVB    (R5),R1         ;GET EXPECTED RDSRL
010376  042701  177400          BIC     #177400,R1      ;MASK OFF UNUSED BITS
010402  023727  002402  000347  CMP     SAVLEN,#TXDL!RXDL ;SEE IF 7-BIT CHARS BEING USED
010410  001005                  BNE     3$              ;BR IF NOT 7-BIT CHARS
010412  142737  000200  010372  BICB    #BIT7,1$        ;CLEAR 8TH BIT FOR COMPARE
010420  142701  000200          BICB    #BIT7,R1
010424  123701  010372  3$:     CMPB    1$,R1           ;COMPARE RCV'D CHAR TO EXPECTED
010430  001462                  BEQ     6$              ;BR IF MATCH
010432  004537  003566          JSR     R5,READI        ;READ USYRT STATUS REG
010436  122000                  USTATR
010440  000000          4$:     .WORD   0
010442  132737  000002  010440  BITB    #TXU,4$         ;SEE IF TX UNDERRUN OCCURRED
010450  001421                  BEQ     5$              ;BR IF NOT
010452  012737  000007  002336  MOV     #7,REGNUM       ;SET USYRT REG NO. FOR STATUS REG
                        ;STACK 'TX UNDERRUN'' ERROR
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
010510  000137  011610          JMP     20$             ;TAKE ERROR EXIT
010514  005037  002336  5$:     CLR     REGNUM          ;SET USYRT REG NO. FOR RDSRL
010520  005037  002324          CLR     GDATA           ;SET EXPECTED DATA
010524  110137  002324          MOVB    R1,GDATA
010530  005037  002326          CLR     BDATA           ;SET ACTUAL DATA
010534  113737  010372  002326  MOVB    1$,BDATA
                        ;STACK 'RCV'D DATA MISCOMPARE'' ERROR
```

```
3506  010542                                    GTDF    EM34,ERR10
3507
3508  010542  012737  000001  002172                                ;       QUEUE 'DEVICE FATAL'' ERROR # 27
3509  010550  012737  000033  002174                                        MOV     #T.EDF,ERRTYP
3510  010556  012737  014502  002176                                        MOV     #27,ERRNBR
3511  010564  012737  020364  002200                                        MOV     #EM34,ERRMSG
3512  010572  000137  011610                                              MOV     #ERR10,ERRBLK
                                            JMP     20$             ;TAKE ERROR EXIT
3513  010576  116501  000001          6$:   MOVB    1(R5),R1        ;GET RDSRH
3514  010602  042701  177400                BIC     #177400,R1      ;MASK OFF UNUSED BITS
3515  010606  123701  010362                CMPB    2$,R1           ;COMPARE RCV'D STATUS TO EXPECTED
3516  010612  001016                        BNE     7$              ;BR IF MISMATCH
3517  010614  000137  011474                JMP     17$             ;CONTINUE
3518  010620  012737  000001  002336        MOV     #1,REGNUM       ;SET USYRT REG NO. FOR RDSRH
3519  010626  005037  002324                CLR     GDATA           ;SET EXPECTED DATA
3520  010632  110137  002324                MOVB    R1,GDATA
3521  010636  005037  002326                CLR     BDATA           ;SET ACTUAL DATA
3522  010642  113737  010362  002326        MOVB    2$,BDATA
3523  010650  012737  000001  002336  7$:   MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
3524  010656  032765  000001  000002        BIT     #RERCHK,2(R5)   ;SEE IF RCV ERROR BIT SHOULD BE IGNORED
3525  010664  001447                        BEQ     9$              ;BR IF YES
3526                                  ;CHECK RERR BIT
3527  010666  132701  000200                BITB    #RERR,R1        ;SEE IF EXPECTED BIT = 1
3528  010672  001022                        BNE     8$              ;BR IF YES
3529  010674  132737  000200  010362        BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 0
3530  010702  001440                        BEQ     9$              ;BR IF YES
3531                                  ;STACK 'RERR NOT CLEARED'' MSG
3532  010704                                GTDF    EM35,ERR12
3533
3534  010704  012737  000001  002172                                ;       QUEUE 'DEVICE FATAL'' ERROR # 28
3535  010712  012737  000034  002174                                        MOV     #T.EDF,ERRTYP
3536  010720  012737  014530  002176                                        MOV     #28,ERRNBR
3537  010726  012737  020714  002200                                        MOV     #EM35,ERRMSG
3538  010734  000137  011610                                              MOV     #ERR12,ERRBLK
                                            JMP     20$             ;TAKE ERROR EXIT
3539  010740  132737  000200  010362  8$:   BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 1
3540  010746  001016                        BNE     9$              ;BR IF YES
3541                                  ;STACK 'RERR NOT SET'' MSG
3542  010750                                GTDF    EM36,ERR12
3543
3544  010750  012737  000001  002172                                ;       QUEUE 'DEVICE FATAL'' ERROR # 29
3545  010756  012737  000035  002174                                        MOV     #T.EDF,ERRTYP
3546  010764  012737  014551  002176                                        MOV     #29,ERRNBR
3547  010772  012737  020714  002200                                        MOV     #EM36,ERRMSG
3548  011000  000137  011610                                              MOV     #ERR12,ERRBLK
                                            JMP     20$             ;TAKE ERROR EXIT
3549                                  ;CHECK ROR BIT
3550  011004  132701  000010          9$:   BITB    #ROR,R1         ;SEE IF EXPECTED BIT = 1
3551  011010  001022                        BNE     10$             ;BR IF YES
3552  011012  132737  000010  010362        BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 0
3553  011020  001440                        BEQ     11$             ;BR IF YES
3554                                  ;STACK 'ROR NOT CLEARED'' MSG
3555  011022                                GTDF    EM16,ERR12
3556
3557  011022  012737  000001  002172                                ;       QUEUE 'DEVICE FATAL'' ERROR # 30
3558  011030  012737  000036  002174                                        MOV     #T.EDF,ERRTYP
3559  011036  012737  014340  002176                                        MOV     #30,ERRNBR
3560  011044  012737  020714  002200                                        MOV     #EM16,ERRMSG
3561  011052  000137  011610                                              MOV     #ERR12,ERRBLK
                                            JMP     20$             ;TAKE ERROR EXIT
```

```
3562  011056  132737  000010  010362  10$:    BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 1
3563  011064  001016                          BNE     11$             ;BR IF YES
3564                                  ;STACK 'ROR NOT SET'' MSG
3565  011066                                  GTDF    EM14,ERR12
3566                                                                  ;       QUEUE 'DEVICE FATAL'' ERROR # 31
3567  011066  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
3568  011074  012737  000037  002174                                                  MOV     #31,ERRNBR
3569  011102  012737  014324  002176                                                  MOV     #EM14,ERRMSG
3570  011110  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3571  011116  000137  011610                          JMP     20$             ;TAKE ERROR EXIT
3572                                  ;CHECK RABGA BIT
3573  011122  132701  000004          11$:    BITB    #RABGA,R1       ;SEE IF EXPECTED BIT = 1
3574  011126  001022                          BNE     12$             ;BR IF YES
3575  011130  132737  000004  010362          BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 0
3576  011136  001440                          BEQ     13$             ;BR IF YES
3577                                  ;STACK 'RABGA NOT CLEARED'' MSG
3578  011140                                  GTDF    EM39,ERR12
3579                                                                  ;       QUEUE 'DEVICE FATAL'' ERROR # 32
3580  011140  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
3581  011146  012737  000040  002174                                                  MOV     #32,ERRNBR
3582  011154  012737  014566  002176                                                  MOV     #EM39,ERRMSG
3583  011162  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3584  011170  000137  011610                          JMP     20$             ;TAKE ERROR EXIT
3585  011174  132737  000004  010362  12$:    BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 1
3586  011202  001016                          BNE     13$             ;BR IF YES
3587                                  ;STACK 'RABGA NOT SET'' MSG
3588  011204                                  GTDF    EM40,ERR12
3589                                                                  ;       QUEUE 'DEVICE FATAL'' ERROR # 33
3590  011204  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
3591  011212  012737  000041  002174                                                  MOV     #33,ERRNBR
3592  011220  012737  014610  002176                                                  MOV     #EM40,ERRMSG
3593  011226  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3594  011234  000137  011610                          JMP     20$             ;TAKE ERROR EXIT
3595                                  ;CHECK REOM BIT
3596  011240  132701  000002          13$:    BITB    #REOM,R1        ;SEE IF EXPECTED BIT = 1
3597  011244  001022                          BNE     14$             ;BR IF YES
3598  011246  132737  000002  010362          BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 0
3599  011254  001440                          BEQ     15$             ;BR IF YES
3600                                  ;STACK 'REOM NOT CLEARED'' MSG
3601  011256                                  GTDF    EM30,ERR12
3602                                                                  ;       QUEUE 'DEVICE FATAL'' ERROR # 34
3603  011256  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
3604  011264  012737  000042  002174                                                  MOV     #34,ERRNBR
3605  011272  012737  014444  002176                                                  MOV     #EM30,ERRMSG
3606  011300  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3607  011306  000137  011610                          JMP     20$             ;TAKE ERROR EXIT
3608  011312  132737  000002  010362  14$:    BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 1
3609  011320  001016                          BNE     15$             ;BR IF YES
3610                                  ;STACK 'REOM NOT SET'' MSG
3611  011322                                  GTDF    EM31,ERR12
3612                                                                  ;       QUEUE 'DEVICE FATAL'' ERROR # 35
3613  011322  012737  000001  002172                                                  MOV     #T.EDF,ERRTYP
3614  011330  012737  000043  002174                                                  MOV     #35,ERRNBR
3615  011336  012737  014465  002176                                                  MOV     #EM31,ERRMSG
3616  011344  012737  020714  002200                                                  MOV     #ERR12,ERRBLK
3617  011352  000137  011610                          JMP     20$             ;TAKE ERROR EXIT
```

```
3618
3619  011356  132701  000001          ;CHECK RSOM BIT
3620  011362  001022          15$:    BITB    #RSOM,R1        ;SEE IF EXPECTED BIT = 1
3621  011364  132737  000001  010362  BNE     16$             ;BR IF YES
3622  011372  001440                  BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 0
3623                                  BEQ     17$             ;BR IF YES
3624  011374                  ;STACK 'RSOM NOT CLEARED'' MSG
3625                                  GTDF    EM28,ERR12
3626  011374  012737  000001  002172                  ;       QUEUE 'DEVICE FATAL'' ERROR # 36
3627  011402  012737  000044  002174                          MOV     #T.EDF,ERRTYP
3628  011410  012737  014406  002176                          MOV     #36,ERRNBR
3629  011416  012737  020714  002200                          MOV     #EM28,ERRMSG
3630  011424  000137  011610                                  MOV     #ERR12,ERRBLK
3631  011430  132737  000001  010362  16$:    JMP     20$             ;TAKE ERROR EXIT
3632  011436  001016                  BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 1
3633                                  BNE     17$             ;BR IF YES
3634  011440                  ;STACK 'RSOM NOT SET'' MSG
3635                                  GTDF    EM29,ERR12
3636  011440  012737  000001  002172                  ;       QUEUE 'DEVICE FATAL'' ERROR # 37
3637  011446  012737  000045  002174                          MOV     #T.EDF,ERRTYP
3638  011454  012737  014427  002176                          MOV     #37,ERRNBR
3639  011462  012737  020714  002200                          MOV     #EM29,ERRMSG
3640  011470  000137  011610                                  MOV     #ERR12,ERRBLK
3641                                  JMP     20$             ;TAKE ERROR EXIT
3642  011474  116502  000004  17$:    MOVB    4(R5),R2        ;GET DESIRED NO. OF CYCLES
3643  011500  005001                  CLR     R1              ;INIT CYCLE COUNT
3644
3645  011502  136527  000005  000040  18$:    BITB    5(R5),#BIT5     ;* IS RXACT CHECK TO BE DISABLED ?
3646  011510  001004                  BNE     31$             ;* BR IF YES
3647  011512  004537  005654          JSR     R5,CKRACT       ;CHK FOR RACT = 1
3648  011516  000001                  1
3649  011520  103433                  BCS     20$             ;BR TO EXIT IF ERROR
3650
3651  011522  020102          31$:    CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3652  011524  001415                  BEQ     19$             ;BR IF YES
3653
3654  011526  136527  000005  000200          BITB    5(R5),#BIT7     ;* SEE IF INITIAL RDA CHECK DESIRED
3655  011534  001004                  BNE     22$             ;* BR IF NO
3656  011536  004537  006154          JSR     R5,CKRDA        ;CHK FOR RDA = 0
3657  011542  000000                  0
3658  011544  103421                  BCS     20$             ;BR TO EXIT IF ERROR
3659
3660  011546  004537  012072  22$:    JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3661  011552  000001                  1
3662  011554  005201                  INC     R1              ;INCR CYCLE COUNT
3663  011556  000751                  BR      18$             ;CONTINUE CLOCKING
3664
3665  011560  136527  000005  000100  19$:    BITB    5(R5),#BIT6     ;* IS FINAL RDA CHECK TO BE SKIPPED ?
3666  011566  001004                  BNE     30$             ;* BR IF YES
3667  011570  004537  006154          JSR     R5,CKRDA        ;CHK RDA = 1
3668  011574  000001                  1
3669  011576  103404                  BCS     20$             ;BR IF ERROR
3670
3671  011600  062705  000006  30$:    ADD     #6,R5           ;FIX UP RETURN ADRS
3672  011604  000241                  CLC                     ;SET C = 0 FOR NO ERROR
3673  011606  000403                  BR      21$             ;TAKE ERROR-FREE EXIT
```

CVDMEA.P11 10-DEC-80 09:16 ....RXCHAR -- RECEIVE A CHARACTER

```
3674  011610  062705  000006  20$:    ADD     #6,R5           ;FIX UP RETURN ADDRESS
3675  011614  000261                  SEC                     ;SET C BIT FOR ERROR
3676  011616  012602          21$:    MOV     (SP)+,R2        ;RESTORE R2
3677  011620  012601                  MOV     (SP)+,R1        ;RESTORE R1
3678  011622  000205                  RTS     R5              ;RETURN
3679
```

```
                                .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)
```

CVDMEA.P11 10-DEC-80 09:16 ....RCV1ST -- RECEIVE FIRST CHARACTER OF MESSAGE SEQ 89

3736

```
                                .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
```

```
CVDMEA.P11    10-DEC-80 09:16          ....STEPLU -- CLOCK THE USYRT N TIMES

 3776                                  .SBTTL ....STEPLU -- CLOCK THE USYRT N TIMES
 3777                                  ;*************************************************************************
 3778                                  ;* STEPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NUMBER OF CYCLES
 3779                                  ;*       PASSED IN THE WORD FOLLOWING THE CALL. THE VIA ACR MUST BE PREVIOUSLY
 3780                                  ;*       SET UP FOR T1 ONE-SHOT MODE, AND THE T1 LATCHES MUST BE PREVIOUSLY SET
 3781                                  ;*       TO CONTROL THE WIDTH OF THE CLOCK PULSE. ALL THAT THIS SUBROUTINE
 3782                                  ;*       DOES IS TO LOAD 000 INTO THE HI BYTE OF THE T1 COUNTER, FOR THE
 3783                                  ;*       DESIRED NUMBER OF TIMES.
 3784                                  ;*
 3785                                  ;*       CALLING SEQUENCE :
 3786                                  ;*       JSR     R5,STEPLU
 3787                                  ;*       .WORD   <NUMBER OF CYCLES TO CLOCK>
 3788                                  ;*************************************************************************
 3789  012072                          STEPLU:
 3790  012072  010146                         MOV     R1,-(SP)        ;SAVE R1
 3791  012074  012501                         MOV     (R5)+,R1        ;INIT CYCLE COUNTER
 3792  012076  004537  003712         1$:     JSR     R5,WRITEI       ;LOAD T1C-H, START COUNTER, CLOCK 1 CYCLE
 3793  012102  120005                         VIAT1B
 3794  012104  000000                         000
 3795  012106  005301                         DEC     R1              ;DECR CYCLE COUNTER
 3796  012110  001372                         BNE     1$              ;BR IF ALL CYCLES NOT DONE YET
 3797  012112  012601                         MOV     (SP)+,R1        ;RESTORE R1
 3798  012114  000205                         RTS     R5              ;RETURN
 3799
 3800
 3801
 3802
 3803
 3804
```

```
                        .SBTTL  GLOBAL ERROR REPORT SECTION

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

                        .NLIST  BEX
012116 047045 047045 000 ENDEMB: .ASCIZ  /%N%N/
012123    045 000116    NEWLIN: .ASCIZ  /%N/            ;USED TO TERMINATE ERROR MESSAGES

012126 047045 040445 040506 FMT2:   .ASCIZ  /%N%AFAILING REG = %T%ASEL%01/
012163    045 022516 020101 FMT3:   .ASCIZ  /%N%A  EXPECTED:  %03%A   ACTUAL:  %03%A   XOR:  %03/
012247    045 022516 052101 FMT4:   .ASCIZ  /%N%ATHE CONTENTS OF ALL%T%N%T/
012305    045 022516 030523 FMT4A:  .ASCIZ  /%N%S1%03%S5%03%S5%03%S5%03/
012340 047045 052045    000 FMT4B:  .ASCIZ  /%N%T/
012345    045 022516 032523 FMT4C:  .ASCIZ  /%N%S5%03%S5%03%S5%03%S5%03/
012400 047045 040445 020040 FMT5:   .ASCIZ  /%N%A  WHEN %03%A LOADED INTO BSEL1/
012443    045 022516 020101 FMT5A:  .ASCIZ  /%N%A  ATTEMPTING 'M-LOOP'' FUNCTION CODE %02%A (%T%A)/
012530 047045 040445 042115 FMT7:   .ASCIZ  /%N%AMDIAG #%03%A FAILED/

012560 047045 040445 020040 FMT10:  .ASCIZ  /%N%A  EXPECTED:%08%A  ACTUAL:%08%A  XOR:%08/
012634 040445 020040 051514 FMT10A: .ASCIZ  /%A  LSI ADDR:%08/
012655    045 022516 034117 FMT11:  .ASCIZ  /%N%08%08%08%08/
012674 047045 047045 052045 FMT12:  .ASCIZ  /%N%N%T/
012703    045 022516 022524 FMT13:  .ASCIZ  /%N%T%03%S2%03%S2%03%S2%03%S2%03%S2%03/
012751    045 031123 047445 FMT14:  .ASCIZ  /%S2%03%S2%03/
012766 040445 020040 042504 FMT15:  .ASCIZ  /%A  DETECTED IN %T%T%A --/
013020 040445 020040 042504 FMT15A: .ASCIZ  /%A  DETECTED @ TEST PATTERN ELEMENT # %D2/
013072 047045 052045 047445 FMT16:  .ASCIZ  /%N%T%03%S4%03%S%03/
013115    045 022516 022524 FMT16A: .ASCIZ  /%N%T%03%S%03%S%03%S4%03%S%03%S%03/
013157    045 020101 020040 FMT17:  .ASCIZ  /%A    VALUE SENT TO NPR CONTROL REGISTER:  %03/
013236 047045 040445 020040 FMT17A: .ASCIZ  /%N%A    VALUE READ FROM CONTROL REGISTER:    %03/
013317    045 022516 020101 FMT17B: .ASCIZ  /%N%A    LSI-11 MEMORY ADDRESS ACCESSED:%08/
013372 047045 040445 020040 FMT17C: .ASCIZ  /%N%A  INFORMATION ON THE FIRST OF %D5%A ERRORS:/

013452 047045 040445 042524 FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
013503    045 022524 033117 FMT21:  .ASCIZ  /%T%06%N/
013513    045 022516 043101 FMT22:  .ASCIZ  /%N%AFAILING REG: /
013535    045 042501 050130 FMT23:  .ASCIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%S5%AXOR: %03%N/
013614 047045 052045 047045 FMT24:  .ASCIZ  /%N%T%N%T%N/
013627    045 031517 051445 FMT25:  .ASCIZ  /%03%S5%03%S5%03%S5%03%N/
013657    045 032123 047445 FMT26:  .ASCIZ  /%S4%03%S5%03%S5%03%S5%03%N/
013712 052045 052045 047045 FMT27:  .ASCIZ  /%T%T%N/
013721    045 042501 052130 FMT28:  .ASCIZ  /%AEXTENDED REG AX%01%A-%T%N/
013755    045 022524 000116 FMT29:  .ASCIZ  /%T%N/
013762 047045 040445 047506 FMT30:  .ASCIZ  /%N%AFOR BAUD RATE SPECIFIED./
014017    045 022516 044501 FMT31:  .ASCIZ  /%N%AIMPROPER CONNECTOR TYPE SPECIFIED/
014065    045 022516 043101 FMT32:  .ASCIZ  /%N%AFOR OPTION SPECIFIED./
014117    045 022516 052101 FMT39:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
014150 047045 040445 040506 FMT40:  .ASCIZ  /%N%AFAILING RAM ADRS: %06%A (OCT)%N/

014214 051525 051131 020124 EM2:    .ASCIZ  /USYRT NOT INITIALIZED BY PROGRAM RESET/
014263    115 041511 047522 EM3:    .ASCIZ  /MICRO-DIAG. FAILURE/
014307    115 042122 020131 EM4:    .ASCIZ  /MRDY TIMEOUT/
014324 047522 020122 047516 EM14:   .ASCIZ  /ROR NOT SET/
```

```
014340  047522  020122  047516  EM16:   .ASCIZ  /ROR NOT CLEARED/
014360  042522  042101  053457  EM25:   .ASCIZ  'READ/WRITE DATA ERROR'
014406  051522  046517  047040  EM28:   .ASCIZ  /RSOM NOT CLEARED/
014427     122  047523  020115  EM29:   .ASCIZ  /RSOM NOT SET/
014444  042522  046517  047040  EM30:   .ASCIZ  /REOM NOT CLEARED/
014465     122  047505  020115  EM31:   .ASCIZ  /REOM NOT SET/
014502  041522  023526  020104  EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
014530  042522  051122  047040  EM35:   .ASCIZ  /RERR NOT CLEARED/
014551     122  051105  020122  EM36:   .ASCIZ  /RERR NOT SET/
014566  040522  043502  020101  EM39:   .ASCIZ  /RABGA NOT CLEARED/
014610  040522  043502  020101  EM40:   .ASCIZ  /RABGA NOT SET/
014626  054124  052440  042116  EM54:   .ASCIZ  /TX UNDERRUN ERROR/
014650  051525  051131  020124  EM68:   .ASCIZ  /USYRT STATUS INCORRECT/
014677     124  040530  052103  EM69:   .ASCIZ  /TXACT NOT SET/
014715     124  040530  052103  EM70:   .ASCIZ  /TXACT NOT CLEARED/
014737     122  040530  052103  EM71:   .ASCIZ  /RXACT NOT SET/
014755     122  040530  052103  EM72:   .ASCIZ  /RXACT NOT CLEARED/
014777     124  046502  020124  EM73:   .ASCIZ  /TBMT NOT SET/
015014  041124  052115  047040  EM74:   .ASCIZ  /TBMT NOT CLEARED/
015035     122  040504  047040  EM75:   .ASCIZ  /RDA NOT SET/
015051     122  040504  047040  EM76:   .ASCIZ  /RDA NOT CLEARED/
015071     122  040523  047040  EM77:   .ASCIZ  /RSA NOT SET/
015105     122  040523  047040  EM78:   .ASCIZ  /RSA NOT CLEARED/
015125     122  046501  042440  EM79:   .ASCIZ  /RAM ERROR LOADING MICROCODE/
015161     103  051101  044522  EM80:   .ASCIZ  /CARRIER NOT SET/
015201     103  051101  044522  EM81:   .ASCIZ  /CARRIER NOT CLEARED/
015225     111  053116  046101  EM82:   .ASCIZ  /INVALID ERROR CODE FROM 6502/
015262  047515  042504  020115  EM83:   .ASCIZ  /MODEM STATUS INCORRECT/
015311     103  051524  047040  EM84:   .ASCIZ  /CTS NOT CLRD/
015326  052103  020123  047516  EM85:   .ASCIZ  /CTS NOT SET/
015342  040503  051122  042511  EM86:   .ASCIZ  /CARRIER NOT CLRD/
015363     103  051101  044522  EM87:   .ASCIZ  /CARRIER NOT SET/
015403     115  042117  046505  EM88:   .ASCIZ  /MODEM RDY NOT CLRD/
015426  047515  042504  020115  EM89:   .ASCIZ  /MODEM RDY NOT SET/
015450  042522  042503  053111  EM90:   .ASCIZ  /RECEIVER OVERRUN NOT SET/
015501     122  041505  044505  EM91:   .ASCIZ  /RECEIVER OVERRUN NOT CLEARED/
015536  051524  020117  044502  EM100:  .ASCIZ  /TSO BIT NOT SET/
015556  051524  020117  044502  EM101:  .ASCIZ  /TSO BIT NOT CLEARED/

                                .SBTTL ....TEXT STRINGS FOR ERROR HANDLERS -- 'TXT___'
                                ;-----------------------------------------------------------------
                                ;---------- TEXT USED BY ERROR HANDLERS --------------------------
                                ;-----------------------------------------------------------------

015602  051502  046105  020060  TXT1:   .ASCIZ  /BSEL0   BSEL1   BSEL2   BSEL3/
015640  020040  020040  051502  TXT2:   .ASCIZ  /    BSEL4   BSEL5   BSEL6   BSEL7/
015702  051502  046105  030061  TXT2A:  .ASCIZ  /BSEL10  BSEL11  BSEL12  BSEL13/
015741     040  020040  041040  TXT2B:  .ASCIZ  /    BSEL14  BSEL15  BSEL16  BSEL17/
016004  041040  052131  020105  TXT3:   .ASCIZ  / BYTE SELECT REG'S ARE:/
016034  020040  051440  046105  TXT4:   .ASCIZ  /   SEL0    SEL2    SEL4    SEL6/
016074  020040  051440  046105  TXT4A:  .ASCIZ  /   SEL10   SEL12   SEL14   SEL16/
016135     102     000          TXT5:   .ASCIZ  /B/
016137     040  042523  042514  TXT6:   .ASCIZ  / SELECT REG'S ARE:/
016162  051040  043505  051511  TXT7:   .ASCIZ  / REGISTERS     ORB  ORA  DDRB DDRA T1CL T1CH T1LL T1LH /
016252  020040  020040  020040  TXT7A:  .ASCIZ  /                T2CL T2CH SR   ACR  PCR  IFR  IER  ORA  /
```

CVDMEA.P11 10-DEC-80 09:16 ....TEXT STRINGS FOR ERROR HANDLERS -- 'TXT___'

```
016342  042440  050130  041505  TXT8:    .ASCIZ  / EXPECTED:     /
016362  040440  052103  040525  TXT9:    .ASCIZ  / ACTUAL:       /
016402  054040  051117  020072  TXT10:   .ASCIZ  / XOR:          /
016422  020040  020116  050040  TXT11:   .ASCIZ  /  N  P  R     R  E  G  I  S  T  E  R  S:/
016474  020040  020040  020040  TXT11A:  .ASCIZ  /                CONTROL  D A T A/
016532  020040  020040  020040  TXT11B:  .ASCIZ  /                   OUT ADDR.      IN ADDR./
016602  042504  044526  042503  TXT12:   .ASCIZ  /DEVICE CSR ADDRESS : /
016630  051525  051131  020124  TXT13:   .ASCIZ  /USYRT REGS :/
016645     122  051504  046122  TXT14:   .ASCIZ  /RDSRL   RDSRH   TDSRL   TDSRH/
016703     040  020040  050040  TXT15:   .ASCIZ  /    PCSARL  PCSARH  PCR     USTAT/
016745     126  040511  051040  TXT16:   .ASCIZ  /VIA REGS :/
016760  051117  020102  020040  TXT17:   .ASCIZ  /ORB     ORA     DDRB    DDRA/
017015     040  020040  052040  TXT18:   .ASCIZ  /    T1CL    T1CH    T1LL    T1LH/
017056  031124  046103  020040  TXT19:   .ASCIZ  /T2CL    T2CH    SR      ACR/
017112  020040  020040  041520  TXT20:   .ASCIZ  /    PCR     IFR     IER     ORA/

017152     021     000          TXTNUL:  .BYTE   21,0                    ;CTL-Q -- THIS (WE HOPE) IS HARMLESS

017154  047516  000120          TXTML0:  .ASCIZ  /NOP/
017160  042522  042101  030440  TXTML1:  .ASCIZ  /READ 1 BYTE/
017174  051127  052111  020105  TXTML2:  .ASCIZ  /WRITE 1 BYTE/
017211     116  051120  047455  TXTML3:  .ASCIZ  /NPR-OUT 256 BYTES/
017233     116  051120  044455  TXTML4:  .ASCIZ  /NPR-IN 256 BYTES/
017254  042523  020124  044515  TXTML5:  .ASCIZ  /SET MICROPROCESSOR'S PC/
017304  047125  042504  044506  TXTML6:  .ASCIZ  /UNDEFINED/
017316  046101  047514  020127  TXTML7:  .ASCIZ  /ALLOW U-PROCESSOR INTERRUPTS/

017353     126  040511  051040  TXTVR:   .ASCIZ  /VIA REGISTER /
017371     117  041122     000  TXTVR0:  .ASCIZ  /ORB/
017375     117  040522     000  TXTVR1:  .ASCIZ  /ORA/
017401     104  051104  000102  TXTVR2:  .ASCIZ  /DDRB/
017406  042104  040522     000  TXTVR3:  .ASCIZ  /DDRA/
017413     124  041461  000114  TXTVR4:  .ASCIZ  /T1CL/
017420  030524  044103     000  TXTVR5:  .ASCIZ  /T1CH/
017425     124  046061  000114  TXTVR6:  .ASCIZ  /T1LL/
017432  030524  044114     000  TXTVR7:  .ASCIZ  /T1LH/
017437     124  041462  000114  TXTVR8:  .ASCIZ  /T2CL/
017444  031124  044103     000  TXTVR9:  .ASCIZ  /T2CH/
017451     123  000122          TXTVRA:  .ASCIZ  /SR/
017454  041501  000122          TXTVRB:  .ASCIZ  /ACR/
017460  041520  000122          TXTVRC:  .ASCIZ  /PCR/
017464  043111  000122          TXTVRD:  .ASCIZ  /IFR/
017470  042511  000122          TXTVRE:  .ASCIZ  /IER/
017474  051117  000101          TXTVRF:  .ASCIZ  /ORA/

017500  050116  020122     000  TXTNP:   .ASCIZ  /NPR /
017505     103  047117  051124  TXTNP0:  .ASCIZ  /CONTROL/
017515     104  052101  020101  TXTNP1:  .ASCIZ  /DATA HI/
017525     104  052101  020101  TXTNP2:  .ASCIZ  /DATA LO/
017535     101  042104  027122  TXTNP3:  .ASCIZ  /ADDR. OUT EX/
017552  042101  051104  020056  TXTNP4:  .ASCIZ  /ADDR. OUT HI/
017567     101  042104  027122  TXTNP5:  .ASCIZ  /ADDR. OUT LO/
017604  042101  051104  020056  TXTNP6:  .ASCIZ  /ADDR. IN EX/
017620  042101  051104  020056  TXTNP7:  .ASCIZ  /ADDR. IN HI/
017634  042101  051104  020056  TXTNP8:  .ASCIZ  /ADDR. IN LO/
```

```
        017650  051525  051131  020124  TXTUR:  .ASCIZ  /USYRT REG /
        017663     122  051504  046122  TXTUR0: .ASCIZ  /RDSRL/
        017671     122  051504  044122  TXTUR1: .ASCIZ  /RDSRH/
        017677     124  051504  046122  TXTUR2: .ASCIZ  /TDSRL/
        017705     124  051504  044122  TXTUR3: .ASCIZ  /TDSRH/
        017713     120  051503  051101  TXTUR4: .ASCIZ  /PCSARL/
        017722  041520  040523  044122  TXTUR5: .ASCIZ  /PCSARH/
        017731     120  051103     000  TXTUR6: .ASCIZ  /PCR/
        017735     125  052123  052101  TXTUR7: .ASCIZ  /USTAT/
                                        .LIST   BEX
  3813          017744                  .EVEN
  3814
  3815
  3816                                  .SBTTL ....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- 'TXT__T''
  3817                                  ;-----------------------------------------------------------------
  3818                                  ;---------- TEXT ADDRESS TABLES USED BY ERROR HANDLERS ----------
  3819                                  ;-----------------------------------------------------------------
  3820
  3821  017744  017154  017160  017174  TXTMLT: .WORD   TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7
  3822  017752  017211  017233  017254
  3823  017760  017304  017316
  3824
  3825  017764  017353                          .WORD   TXTVR
  3826  017766  017371  017375  017401  TXTVRT: .WORD   TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7
  3827  017774  017406  017413  017420
  3828  020002  017425  017432
  3829  020006  017437  017444  017451          .WORD   TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF
  3830  020014  017454  017460  017464
  3831  020022  017470  017474
  3832
  3833  020026  017500                          .WORD   TXTNP
  3834  020030  017505  017515  017525  TXTNPT: .WORD   TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8
  3835  020036  017535  017552  017567
  3836  020044  017604  017620  017634
  3837  020052  017663  017671  017677  TXTURT: .WORD   TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7
  3838  020060  017705  017713  017722
  3839  020066  017731  017735
  3840
  3841
  3842
```

```
                        ;-----------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR3 -- DUMP THE BYTE SELECT REGISTERS
                        ;-----------------------------------------------------------------------
                                BGNMSG  ERR3
020072                                                                  ERR3::
020072
020072  004737  021306           JSR     PC,ERR4$
                                PRINTB  #ENDEMB
020076                                                                          MOV     #ENDEMB,-(SP)
020076  012746  012116                                                          MOV     #1,-(SP)
020102  012746  000001                                                          MOV     SP,R0
020106  010600                                                                  TRAP    C$PNTB
020110  104414                                                                  ADD     #4,SP
020112  062706  000004
                                ENDMSG
020116                                                                  L10002:
020116  104423                                                                  TRAP    C$MSG
                        ;-----------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR4 -- M-LOOP TIMEOUT ERROR HANDLING
                        ;-----------------------------------------------------------------------
                                BGNMSG  ERR4
020120                                                                  ERR4::
020120
020120  105037  002325           CLRB    GDATA+1                 ;MAKE SURE BIT 8 DOESN'T PRINT!
020124  010146                   MOV     R1,-(SP)                ;SAVE THE WORKING REGISTER
020126  013701  002324           MOV     GDATA,R1                ;SAVE THIS FOR LATER
020132  022701  000017           CMP     #17,R1                  ;WAS THIS AN M-LOOP REQUEST?
020136  002012                   BGE     5$                      ;YES, THEN REPORT THE FUNCTION CODE
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$PNTX
020156  062706  000006                                                          ADD     #6,SP
020162  000424                   BR      20$
020164  001001           5$:     BNE     6$                      ;IF IT WAS A 17, THIS IS A 'NOP' AND
020166  005001                   CLR     R1                      ;  THE TEXT POINTER MUST SO REFLECT.
020170  022701  000007   6$:     CMP     #7,R1                   ;IS FUNCTION CODE > 7?
020174  002002                   BGE     7$                      ;NO,  THEN WE CAN HANDLE IT
020176  012701  000006           MOV     #6,R1                   ;YES, THEN IT'S UNDEFINED -- SAY SO
020202  006301           7$:     ASL     R1                      ;CONVERT TO A WORD OFFSET
020204                           PRINTX  #FMT5A,GDATA,TXTMLT(R1) ;REPORT THE FAILING FUNCTION
020204  016146  017744                                                          MOV     TXTMLT(R1),-(SP)
020210  013746  002324                                                          MOV     GDATA,-(SP)
020214  012746  012443                                                          MOV     #FMT5A,-(SP)
020220  012746  000003                                                          MOV     #3,-(SP)
020224  010600                                                                  MOV     SP,R0
020226  104415                                                                  TRAP    C$PNTX
020230  062706  000010                                                          ADD     #10,SP
020234  012601           20$:    MOV     (SP)+,R1                ;RESTORE THE WORKING REGISTER
020236  004737  021634           JSR     PC,ERR5$                ;DUMP THE SELECT REGISTERS
                                ENDMSG
020242                                                                  L10003:
020242  104423                                                                  TRAP    C$MSG
                        ;-----------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS
```

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

3899                    ;-------------------------------------------------------------------------
3900    020244                  BGNMSG  ERR7A
3901    020244                                                          ERR7A::
3902    020244  113701  002336  MOVB    REGNUM,R1
3903    020250  006301          ASL     R1              ;AS PASSED, THIS WAS A BYTE OFFSET
3904    020252                  PRINTB  #FMT15,#TXTUR,TXTURT(R1)
3905    020252  016146  020052                                          MOV     TXTURT(R1),-(SP)
3906    020256  012746  017650                                          MOV     #TXTUR,-(SP)
3907    020262  012746  012766                                          MOV     #FMT15,-(SP)
3908    020266  012746  000003                                          MOV     #3,-(SP)
3909    020272  010600                                                  MOV     SP,R0
3910    020274  104414                                                  TRAP    C$PNTB
3911    020276  062706  000010                                          ADD     #10,SP
3912    020302  004737  021262          JSR     PC,XORGB
3913    020306                  PRINTB  #FMT3,GDATA,BDATA,XDATA
3914    020306  013746  002330                                          MOV     XDATA,-(SP)
3915    020312  013746  002326                                          MOV     BDATA,-(SP)
3916    020316  013746  002324                                          MOV     GDATA,-(SP)
3917    020322  012746  012163                                          MOV     #FMT3,-(SP)
3918    020326  012746  000004                                          MOV     #4,-(SP)
3919    020332  010600                                                  MOV     SP,R0
3920    020334  104414                                                  TRAP    C$PNTB
3921    020336  062706  000012                                          ADD     #12,SP
3922    020342                  PRINTB  #ENDEMB
3923    020342  012746  012116                                          MOV     #ENDEMB,-(SP)
3924    020346  012746  000001                                          MOV     #1,-(SP)
3925    020352  010600                                                  MOV     SP,R0
3926    020354  104414                                                  TRAP    C$PNTB
3927    020356  062706  000004                                          ADD     #4,SP
3928    020362                  ENDMSG
3929    020362                                                  L10004:
3930    020362  104423                                                  TRAP    C$MSG
3931                    ;-------------------------------------------------------------------------
3932                    .SBTTL ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)
3933                    ;-------------------------------------------------------------------------
3934
3935    020364                  BGNMSG  ERR10
3936    020364                                                          ERR10::
3937    020364                  PRINTB  #FMT21,#TXT12,MPCSR
3938    020364  013746  002416                                          MOV     MPCSR,-(SP)
3939    020370  012746  016602                                          MOV     #TXT12,-(SP)
3940    020374  012746  013503                                          MOV     #FMT21,-(SP)
3941    020400  012746  000003                                          MOV     #3,-(SP)
3942    020404  010600                                                  MOV     SP,R0
3943    020406  104414                                                  TRAP    C$PNTB
3944    020410  062706  000010                                          ADD     #10,SP
3945    020414                  PRINTB  #FMT22
3946    020414  012746  013513                                          MOV     #FMT22,-(SP)
3947    020420  012746  000001                                          MOV     #1,-(SP)
3948    020424  010600                                                  MOV     SP,R0
3949    020426  104414                                                  TRAP    C$PNTB
3950    020430  062706  000004                                          ADD     #4,SP
3951    020434  013701  002336  MOV     REGNUM,R1
3952    020440  006301          ASL     R1              ;GET PTR TO USYRT REG ASCII
3953    020442                  PRINTB  #FMT27,#TXTUR,TXTURT(R1)
3954    020442  016146  020052                                          MOV     TXTURT(R1),-(SP)
```

```
3955  020446  012746  017650                                        MOV     #TXTUR,-(SP)
3956  020452  012746  013712                                        MOV     #FMT27,-(SP)
3957  020456  012746  000003                                        MOV     #3,-(SP)
3958  020462  010600                                               MOV     SP,R0
3959  020464  104414                                               TRAP    C$PNTB
3960  020466  062706  000010                                       ADD     #10,SP
3961  020472  004737  021262          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
3962  020476                        PRINTB  #FMT23,GDATA,BDATA,XDATA
3963  020476  013746  002330                                        MOV     XDATA,-(SP)
3964  020502  013746  002326                                        MOV     BDATA,-(SP)
3965  020506  013746  002324                                        MOV     GDATA,-(SP)
3966  020512  012746  013535                                        MOV     #FMT23,-(SP)
3967  020516  012746  000004                                        MOV     #4,-(SP)
3968  020522  010600                                               MOV     SP,R0
3969  020524  104414                                               TRAP    C$PNTB
3970  020526  062706  000012                                       ADD     #12,SP
3971  020532  004737  022364          JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
3972  020536                        ENDMSG
3973  020536                                                      L10005:
3974  020536  104423                                               TRAP    C$MSG
3975
3976                                ;-------------------------------------------------------------------------------
3977                                .SBTTL ....ERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT)
3978                                ;-------------------------------------------------------------------------------
3979
3980  020540                        BGNMSG  ERR11
3981  020540                                                      ERR11::
3982  020540                        PRINTB  #FMT21,#TXT12,MPCSR
3983  020540  013746  002416                                        MOV     MPCSR,-(SP)
3984  020544  012746  016602                                        MOV     #TXT12,-(SP)
3985  020550  012746  013503                                        MOV     #FMT21,-(SP)
3986  020554  012746  000003                                        MOV     #3,-(SP)
3987  020560  010600                                               MOV     SP,R0
3988  020562  104414                                               TRAP    C$PNTB
3989  020564  062706  000010                                       ADD     #10,SP
3990  020570                        PRINTB  #FMT22
3991  020570  012746  013513                                        MOV     #FMT22,-(SP)
3992  020574  012746  000001                                        MOV     #1,-(SP)
3993  020600  010600                                               MOV     SP,R0
3994  020602  104414                                               TRAP    C$PNTB
3995  020604  062706  000004                                       ADD     #4,SP
3996  020610  013701  002336          MOV     REGNUM,R1
3997  020614  006301                  ASL     R1                      ;GET PTR TO VIA REG ASCII
3998  020616                        PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
3999  020616  016146  017766                                        MOV     TXTVRT(R1),-(SP)
4000  020622  012746  017353                                        MOV     #TXTVR,-(SP)
4001  020626  012746  013712                                        MOV     #FMT27,-(SP)
4002  020632  012746  000003                                        MOV     #3,-(SP)
4003  020636  010600                                               MOV     SP,R0
4004  020640  104414                                               TRAP    C$PNTB
4005  020642  062706  000010                                       ADD     #10,SP
4006  020646  004737  021262          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
4007  020652                        PRINTB  #FMT23,GDATA,BDATA,XDATA
4008  020652  013746  002330                                        MOV     XDATA,-(SP)
4009  020656  013746  002326                                        MOV     BDATA,-(SP)
4010  020662  013746  002324                                        MOV     GDATA,-(SP)
```

```
4011  020666  012746  013535                                          MOV     #FMT23,-(SP)
4012  020672  012746  000004                                          MOV     #4,-(SP)
4013  020676  010600                                                 MOV     SP,R0
4014  020700  104414                                                 TRAP    C$PNTB
4015  020702  062706  000012                                         ADD     #12,SP
4016  020706  004737  022032          JSR     PC,ERR11$              ;GET & PRINT VIA REGISTERS
4017  020712                          ENDMSG
4018  020712                                                      L10006:
4019  020712  104423                                                TRAP    C$MSG
4020
4021
4022                                  ;-------------------------------------------------------------------------------
4023                                  .SBTTL ....ERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)
4024                                  ;-------------------------------------------------------------------------------
4025  020714                          BGNMSG  ERR12
4026  020714                                                      ERR12::
4027  020714                          PRINTB  #FMT21,#TXT12,MPCSR
4028  020714  013746  002416                                          MOV     MPCSR,-(SP)
4029  020720  012746  016602                                          MOV     #TXT12,-(SP)
4030  020724  012746  013503                                          MOV     #FMT21,-(SP)
4031  020730  012746  000003                                          MOV     #3,-(SP)
4032  020734  010600                                                 MOV     SP,R0
4033  020736  104414                                                 TRAP    C$PNTB
4034  020740  062706  000010                                         ADD     #10,SP
4035  020744                          PRINTB  #FMT22
4036  020744  012746  013513                                          MOV     #FMT22,-(SP)
4037  020750  012746  000001                                          MOV     #1,-(SP)
4038  020754  010600                                                 MOV     SP,R0
4039  020756  104414                                                 TRAP    C$PNTB
4040  020760  062706  000004                                         ADD     #4,SP
4041  020764  013701  002336          MOV     REGNUM,R1
4042  020770  006301                  ASL     R1                     ;GET PTR TO USYRT REG ASCII
4043  020772                          PRINTB  #FMT27,#TXTUR,TXTURT(R1)
4044  020772  016146  020052                                          MOV     TXTURT(R1),-(SP)
4045  020776  012746  017650                                          MOV     #TXTUR,-(SP)
4046  021002  012746  013712                                          MOV     #FMT27,-(SP)
4047  021006  012746  000003                                          MOV     #3,-(SP)
4048  021012  010600                                                 MOV     SP,R0
4049  021014  104414                                                 TRAP    C$PNTB
4050  021016  062706  000010                                         ADD     #10,SP
4051  021022  004737  022364          JSR     PC,ERR12$              ;GET & PRINT USYRT REGISTERS
4052  021026                          ENDMSG
4053  021026                                                      L10007:
4054  021026  104423                                                TRAP    C$MSG
4055
4056
4057                                  ;-------------------------------------------------------------------------------
4058                                  .SBTTL ....ERROR HANDLER -- ERR13 -- RAM ADDRESS ERRORS
4059                                  ;-------------------------------------------------------------------------------
4060  021030                          BGNMSG  ERR13
4061  021030                                                      ERR13::
4062  021030                          PRINTB  #FMT21,#TXT12,MPCSR
4063  021030  013746  002416                                          MOV     MPCSR,-(SP)
4064  021034  012746  016602                                          MOV     #TXT12,-(SP)
4065  021040  012746  013503                                          MOV     #FMT21,-(SP)
4066  021044  012746  000003                                          MOV     #3,-(SP)
```

```
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #10,SP
                PRINTB  #FMT40,REGNUM
                                                                        MOV     REGNUM,-(SP)
                                                                        MOV     #FMT40,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #6,SP
                JSR     PC,XORGB        ;COMPUTE XOR OF GOOD AND BAD DATA
                PRINTB  #FMT23,GDATA,BDATA,XDATA
                                                                        MOV     XDATA,-(SP)
                                                                        MOV     BDATA,-(SP)
                                                                        MOV     GDATA,-(SP)
                                                                        MOV     #FMT23,-(SP)
                                                                        MOV     #4,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #12,SP
                ENDMSG
                                                                L10010:
                                                                        TRAP    C$MSG


;-------------------------------------------------------------------------------
.SBTTL ....ERROR HANDLER -- ERR14 -- VIA REG ERRORS (VIA PRINTOUT)
;-------------------------------------------------------------------------------
                BGNMSG  ERR14
                                                                ERR14::
                PRINTB  #FMT21,#TXT12,MPCSR
                                                                        MOV     MPCSR,-(SP)
                                                                        MOV     #TXT12,-(SP)
                                                                        MOV     #FMT21,-(SP)
                                                                        MOV     #3,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #10,SP
                PRINTB  #FMT22
                                                                        MOV     #FMT22,-(SP)
                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #4,SP
                MOV     REGNUM,R1
                ASL     R1                      ;GET PTR TO VIA REG ASCII
                PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
                                                                        MOV     TXTVRT(R1),-(SP)
                                                                        MOV     #TXTVR,-(SP)
                                                                        MOV     #FMT27,-(SP)
                                                                        MOV     #3,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #10,SP
                JSR     PC,ERR11$               ;GET & PRINT VIA REGISTERS
                ENDMSG
```

```
021260                                                                              L10011:
021260  104423                                                                              TRAP    C$MSG

                                .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)
```

```
4179  021426  013746  002216                                    MOV     BSR6,-(SP)
4180  021432  013746  002214                                    MOV     BSR5,-(SP)
4181  021436  013746  002212                                    MOV     BSR4,-(SP)
4182  021442  012746  012345                                    MOV     #FMT4C,-(SP)
4183  021446  012746  000005                                    MOV     #5,-(SP)
4184  021452  010600                                            MOV     SP,R0
4185  021454  104415                                            TRAP    C$PNTX
4186  021456  062706  000014                                    ADD     #14,SP
4187  021462                    PRINTX  #FMT4B,#TXT2A
4188  021462  012746  015702                                    MOV     #TXT2A,-(SP)
4189  021466  012746  012340                                    MOV     #FMT4B,-(SP)
4190  021472  012746  000002                                    MOV     #2,-(SP)
4191  021476  010600                                            MOV     SP,R0
4192  021500  104415                                            TRAP    C$PNTX
4193  021502  062706  000006                                    ADD     #6,SP
4194  021506                    PRINTX  #FMT4A,BSR10,BSR11,BSR12,BSR13
4195  021506  013746  002230                                    MOV     BSR13,-(SP)
4196  021512  013746  002226                                    MOV     BSR12,-(SP)
4197  021516  013746  002224                                    MOV     BSR11,-(SP)
4198  021522  013746  002222                                    MOV     BSR10,-(SP)
4199  021526  012746  012305                                    MOV     #FMT4A,-(SP)
4200  021532  012746  000005                                    MOV     #5,-(SP)
4201  021536  010600                                            MOV     SP,R0
4202  021540  104415                                            TRAP    C$PNTX
4203  021542  062706  000014                                    ADD     #14,SP
4204  021546                    PRINTX  #FMT4B,#TXT2B
4205  021546  012746  015741                                    MOV     #TXT2B,-(SP)
4206  021552  012746  012340                                    MOV     #FMT4B,-(SP)
4207  021556  012746  000002                                    MOV     #2,-(SP)
4208  021562  010600                                            MOV     SP,R0
4209  021564  104415                                            TRAP    C$PNTX
4210  021566  062706  000006                                    ADD     #6,SP
4211  021572                    PRINTX  #FMT4C,BSR14,BSR15,BSR16,BSR17
4212  021572  013746  002240                                    MOV     BSR17,-(SP)
4213  021576  013746  002236                                    MOV     BSR16,-(SP)
4214  021602  013746  002234                                    MOV     BSR15,-(SP)
4215  021606  013746  002232                                    MOV     BSR14,-(SP)
4216  021612  012746  012345                                    MOV     #FMT4C,-(SP)
4217  021616  012746  000005                                    MOV     #5,-(SP)
4218  021622  010600                                            MOV     SP,R0
4219  021624  104415                                            TRAP    C$PNTX
4220  021626  062706  000014                                    ADD     #14,SP
4221  021632  000207            RTS     PC
4222
4223                      ;-------------------------------------------------------------------
4224                      .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR5$
4225                      ;-------------------------------------------------------------------
4226                      ;
                          ;       COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS
4227  021634              ERR5$:
4228  021634                    PRINTX  #FMT4,#TXT6,#TXT4
4229  021634  012746  016034                                    MOV     #TXT4,-(SP)
4230  021640  012746  016137                                    MOV     #TXT6,-(SP)
4231  021644  012746  012247                                    MOV     #FMT4,-(SP)
4232  021650  012746  000003                                    MOV     #3,-(SP)
4233  021654  010600                                            MOV     SP,R0
4234  021656  104415                                            TRAP    C$PNTX
```

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

4235  021660  062706  000010                                        ADD     #10,SP
4236  021664
                        PRINTX  #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS
4237  021664  013746  002210                                        MOV     WSR6,-(SP)
4238  021670  013746  002206                                        MOV     WSR4,-(SP)
4239  021674  013746  002204                                        MOV     WSR2,-(SP)
4240  021700  013746  002202                                        MOV     WSR0,-(SP)
4241  021704  012746  012655                                        MOV     #FMT11,-(SP)
4242  021710  012746  000005                                        MOV     #5,-(SP)
4243  021714  010600                                                MOV     SP,R0
4244  021716  104415                                                TRAP    C$PNTX
4245  021720  062706  000014                                        ADD     #14,SP
4246  021724
                        PRINTX  #FMT4B,#TXT4A
4247  021724  012746  016074                                        MOV     #TXT4A,-(SP)
4248  021730  012746  012340                                        MOV     #FMT4B,-(SP)
4249  021734  012746  000002                                        MOV     #2,-(SP)
4250  021740  010600                                                MOV     SP,R0
4251  021742  104415                                                TRAP    C$PNTX
4252  021744  062706  000006                                        ADD     #6,SP
4253  021750
                        PRINTX  #FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS
4254  021750  013746  002220                                        MOV     WSR16,-(SP)
4255  021754  013746  002216                                        MOV     WSR14,-(SP)
4256  021760  013746  002214                                        MOV     WSR12,-(SP)
4257  021764  013746  002212                                        MOV     WSR10,-(SP)
4258  021770  012746  012655                                        MOV     #FMT11,-(SP)
4259  021774  012746  000005                                        MOV     #5,-(SP)
4260  022000  010600                                                MOV     SP,R0
4261  022002  104415                                                TRAP    C$PNTX
4262  022004  062706  000014                                        ADD     #14,SP
4263  022010
                        PRINTB  #ENDEMB
4264  022010  012746  012116                                        MOV     #ENDEMB,-(SP)
4265  022014  012746  000001                                        MOV     #1,-(SP)
4266  022020  010600                                                MOV     SP,R0
4267  022022  104414                                                TRAP    C$PNTB
4268  022024  062706  000004                                        ADD     #4,SP
4269  022030  000207                RTS     PC
4270
4271                    ;-------------------------------------------------------------
4272                    .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR11$
4273                    ;-------------------------------------------------------------
4274                    ;       COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS
4275
4276  022032  004737  004460  ERR11$: JSR     PC,GETVRS               ;GET VIA REGS FOR PRINTOUT
4277  022036                        PRINTX  #FMT24,#TXT16,#TXT17
4278  022036  012746  016760                                        MOV     #TXT17,-(SP)
4279  022042  012746  016745                                        MOV     #TXT16,-(SP)
4280  022046  012746  013614                                        MOV     #FMT24,-(SP)
4281  022052  012746  000003                                        MOV     #3,-(SP)
4282  022056  010600                                                MOV     SP,R0
4283  022060  104415                                                TRAP    C$PNTX
4284  022062  062706  000010                                        ADD     #10,SP
4285  022066
                        PRINTX  #FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6
4286  022066  013746  002270                                        MOV     VREGS+6,-(SP)
4287  022072  013746  002266                                        MOV     VREGS+4,-(SP)
4288  022076  013746  002264                                        MOV     VREGS+2,-(SP)
4289  022102  013746  002262                                        MOV     VREGS+0,-(SP)
4290  022106  012746  013627                                        MOV     #FMT25,-(SP)
```

```
4291  022112  012746  000005                                    MOV     #5,-(SP)
4292  022116  010600                                            MOV     SP,R0
4293  022120  104415                                            TRAP    C$PNTX
4294  022122  062706  000014                                    ADD     #14,SP
4295  022126                  PRINTX  #FMT29  #TXT18
4296  022126  012746  017015                                    MOV     #TXT18,-(SP)
4297  022132  012746  013755                                    MOV     #FMT29,-(SP)
4298  022136  012746  000002                                    MOV     #2,-(SP)
4299  022142  010600                                            MOV     SP,R0
4300  022144  104415                                            TRAP    C$PNTX
4301  022146  062706  000006                                    ADD     #6,SP
4302  022152                  PRINTX  #FMT26,VREGS+8.,VREGS+10.,VREGS+12.,VREGS+14.
4303  022152  013746  002300                                    MOV     VREGS+14.,-(SP)
4304  022156  013746  002276                                    MOV     VREGS+12.,-(SP)
4305  022162  013746  002274                                    MOV     VREGS+10.,-(SP)
4306  022166  013746  002272                                    MOV     VREGS+8.,-(SP)
4307  022172  012746  013657                                    MOV     #FMT26,-(SP)
4308  022176  012746  000005                                    MOV     #5,-(SP)
4309  022202  010600                                            MOV     SP,R0
4310  022204  104415                                            TRAP    C$PNTX
4311  022206  062706  000014                                    ADD     #14,SP
4312  022212                  PRINTX  #FMT29,#TXT19
4313  022212  012746  017056                                    MOV     #TXT19,-(SP)
4314  022216  012746  013755                                    MOV     #FMT29,-(SP)
4315  022222  012746  000002                                    MOV     #2,-(SP)
4316  022226  010600                                            MOV     SP,R0
4317  022230  104415                                            TRAP    C$PNTX
4318  022232  062706  000006                                    ADD     #6,SP
4319  022236                  PRINTX  #FMT25,VREGS+16.,VREGS+18.,VREGS+20.,VREGS+22.
4320  022236  013746  002310                                    MOV     VREGS+22.,-(SP)
4321  022242  013746  002306                                    MOV     VREGS+20.,-(SP)
4322  022246  013746  002304                                    MOV     VREGS+18.,-(SP)
4323  022252  013746  002302                                    MOV     VREGS+16.,-(SP)
4324  022256  012746  013627                                    MOV     #FMT25,-(SP)
4325  022262  012746  000005                                    MOV     #5,-(SP)
4326  022266  010600                                            MOV     SP,R0
4327  022270  104415                                            TRAP    C$PNTX
4328  022272  062706  000014                                    ADD     #14,SP
4329  022276                  PRINTX  #FMT29,#TXT20
4330  022276  012746  017112                                    MOV     #TXT20,-(SP)
4331  022302  012746  013755                                    MOV     #FMT29,-(SP)
4332  022306  012746  000002                                    MOV     #2,-(SP)
4333  022312  010600                                            MOV     SP,R0
4334  022314  104415                                            TRAP    C$PNTX
4335  022316  062706  000006                                    ADD     #6,SP
4336  022322                  PRINTX  #FMT25,VREGS+24.,VREGS+26.,VREGS+28.,VREGS+30.
4337  022322  013746  002320                                    MOV     VREGS+30.,-(SP)
4338  022326  013746  002316                                    MOV     VREGS+28.,-(SP)
4339  022332  013746  002314                                    MOV     VREGS+26.,-(SP)
4340  022336  013746  002312                                    MOV     VREGS+24.,-(SP)
4341  022342  012746  013657                                    MOV     #FMT26,-(SP)
4342  022346  012746  000005                                    MOV     #5,-(SP)
4343  022352  010600                                            MOV     SP,R0
4344  022354  104415                                            TRAP    C$PNTX
4345  022356  062706  000014                                    ADD     #14,SP
4346  022362  000207          RTS     PC
```

```
                                ;-------------------------------------------------------------
                                .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR12$
                                ;-------------------------------------------------------------
                                ;       COMMON ERROR ROUTINE TO GET AND PRINTOUT USYRT REGISTERS

022364  004737  004360    ERR12$: JSR     PC,GETURS               ;GET USYRT REGS FOR PRINTOUT
022370                          PRINTX  #FMT24,#TXT13,#TXT14
022370  012746  016645                                            MOV     #TXT14,-(SP)
022374  012746  016630                                            MOV     #TXT13,-(SP)
022400  012746  013614                                            MOV     #FMT24,-(SP)
022404  012746  000003                                            MOV     #3,-(SP)
022410  010600                                                    MOV     SP,R0
022412  104415                                                    TRAP    C$PNTX
022414  062706  000010                                            ADD     #10,SP
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$PNTX
022454  062706  000014                                            ADD     #14,SP
022460                          PRINTX  #FMT29,#TXT15
022460  012746  016703                                            MOV     #TXT15,-(SP)
022464  012746  013755                                            MOV     #FMT29,-(SP)
022470  012746  000002                                            MOV     #2,-(SP)
022474  010600                                                    MOV     SP,R0
022476  104415                                                    TRAP    C$PNTX
022500  062706  000006                                            ADD     #6,SP
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
022544  000207                          RTS     PC

                                .EVEN
```

```
                                .SBTTL  LOAD DEVICE PROTECTION TABLE

                                ;////////////////////////////////////////////////////////////////////////////
                                ;/ THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
                                ;/ PROTECTED FROM TESTING, IF DESIRED.
                                ;////////////////////////////////////////////////////////////////////////////

                                        BGNPROT
                                                                                L$PROT::
                                        .WORD   -1      ;DON'T CHK CSR ADRS
                                        .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
                                        .WORD   -1      ;DON'T CHK DRIVE NO.
                                        ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

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

                                        BGNINIT
                                                                                LSINIT::
022554  010637  002340                  MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
022560  005037  002344                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
022564  005037  002400                  CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
022570  005037  002376                  CLR     ERROR1          ;CLEAR ERROR FLAG
022574  005037  002402                  CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
022600  005737  002370                  TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
022604  001007                          BNE     6$              ;BR IF NOT
022606  013737  000004  002372          MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
022614  013737  000006  002374          MOV     @#6,SAVE6
022622  000406                          BR      9$

022624  013737  002372  000004  6$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
022632  013737  002374  000006          MOV     SAVE6,@#6

022640  012737  000001  002370  9$:     MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH

                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
022646                                  READEF  #EF.START
022646  012700  000040                                          MOV     #EF.START,R0
022652  104447                                                  TRAP    C$REFG
022654                                  BCOMPLETE       STARST
022654  103415                                                  BCS     STARST

                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
022656                                  READEF  #EF.RESTART
022656  012700  000037                                          MOV     #EF.RESTART,R0
022662  104447                                                  TRAP    C$REFG
022664                                  BCOMPLETE       STARST
022664  103411                                                  BCS     STARST

                                ;SEE IF THIS IS A NEW PASS, BR IF YES
022666                                  READEF  #EF.NEW
022666  012700  000035                                          MOV     #EF.NEW,R0
022672  104447                                                  TRAP    C$REFG
022674                                  BCOMPLETE       NEWST
022674  103411                                                  BCS     NEWST

                                ;SEE IF PROGRAM WAS JUST CONTINUED
022676                                  READEF  #EF.CONTINUE
022676  012700  000036                                          MOV     #EF.CONTINUE,R0
022702  104447                                                  TRAP    C$REFG
022704                                  BCOMPLETE       ENDIT
022704  103470                                                  BCS     ENDIT
022706  000414                          BR      GETPRM

022710                          STARST:
```

```
4461  022710  005037  002412                  CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
4462
4463                                          ;CLEAR DEVICE MAP
4464  022714  005037  002404                  CLR     DEVMAP
4465  022720                                NEWST:
4466  022720  012737  177777  002334          MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
4467  022726  005237  002412                  INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
4468  022732  012737  000001  002406          MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
4469
4470                                          ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
4471                                          ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
4472  022740                                GETPRM:
4473  022740  005237  002334                  INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
4474  022744                                  GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
4475  022744  013700  002334                                                          MOV     LOGDEV,R0
4476  022750  104442                                                                  TRAP    C$GPHRD
4477  022752  010001                                                                  MOV     R0,R1
4478  022754                                  BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
4479  022754  103403                                                                  BCS     10$
4480  022756  006337  002406                  ASL     DEVPTR          ;SHIFT DEVICE POINTER
4481  022762  000766                          BR      GETPRM          ;SKIP THIS DEVICE
4482  022764  053737  002406  002404  10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
4483  022772  006337  002406                  ASL     DEVPTR          ;SHIFT BIT POINTER
4484
4485  022776  012102                          MOV     (R1)+,R2        ;R2=CSR ADDR VALUE
4486  023000  012703  002416                  MOV     #MPCSR,R3       ;R3=POINTER TO CSR ADDR STORAGE AREA
4487
4488  023004  010223                  11$:    MOV     R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
4489  023006  005202                          INC     R2              ;BUMP BSEL ADDR
4490  023010  022703  002456                  CMP     #BSEL17+2,R3    ;ALL 16 ADDRESSES MOVED ?
4491  023014  001373                          BNE     11$             ;NO: DO ANOTHER ADDRESS
4492                                                                  ;YES: CONTINUE
4493
4494  023016  011137  002456                  MOV     (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
4495  023022  012137  002460                  MOV     (R1)+,MPOVEC
4496  023026  062737  000004  002460          ADD     #4,MPOVEC       ;GET DMV11 OUTPUT INTRPT VECTOR
4497  023034  012137  002462                  MOV     (R1)+,MPRIOR    ;GET DMV11 DEVICE PRIORITY
4498  023040  012137  002464                  MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
4499  023044  012137  002466                  MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
4500  023050  012137  002470                  MOV     (R1)+,BRDTYP    ;GET DMV-11 BOARD TYPE
4501  023054  012137  002472                  MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
4502  023060  011137  002474                  MOV     (R1),BDRATE     ;GET BAUD RATE FOR THIS DEVICE
4503                                        ;ISSUE LSI BUS RESET, TO INIT DMV11
4504  023064                                  BRESET
4505  023064  104433                                                                  TRAP    C$RESET
4506  023066                                ENDIT:
4507  023066                                  ENDINIT
4508  023066                                                                  L10013:
4509  023066  104411                                                                  TRAP    C$INIT
```

```
                        .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 CVDMA 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.
                        ;
                        ;--------------------------------------------------------------------------

023070                          BGNAUTO
023070                                                          L$AUTO::

023070                          SETVEC  #4,#AD.HIT,#0   ;SETUP INVALID-ADDRESS TRAP VECTOR
023070  012746  000000                                          MOV     #0,-(SP)
023074  012746  023206                                          MOV     #AD.HIT,-(SP)
023100  012746  000004                                          MOV     #4,-(SP)
023104  012746  000003                                          MOV     #3,-(SP)
023110  104437                                                  TRAP    C$SVEC
023112  062706  000010                                          ADD     #10,SP
023116  005037  002546              CLR     TMP0            ;INITIALIZE TRAP FLAG REGISTER
023122  012702  000001              MOV     #1,R2           ;FLAG BIT
023126  013703  002416              MOV     BSEL0,R3        ;INIT ADDRESS POINTER

023132  105723          1$:         TSTB    (R3)+           ;ACCESS THE CSR'S BY BYTES.
023134  006302                      ASL     R2
023136  103375                      BCC     1$

023140  013703  002416              MOV     BSEL0,R3        ;RE-INIT ADDRESS POINTER
023144  012702  000001              MOV     #1,R2           ;RE-INIT FLAG BIT
023150  005723          2$:         TST     (R3)+           ;ACCESS THE CSR'S BY WORDS.
023152  006302                      ASL     R2
023154  006302                      ASL     R2
023156  103374                      BCC     2$

023160                          CLRVEC  #4              ;RESTORE THE VECTOR TO DS
023160  012700  000004                                          MOV     #4,R0
023164  104436                                                  TRAP    C$CVEC
023166  005737  002546              TST     TMP0            ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP?
023172  001403                      BEQ     AD.OK           ;NO,  EXIT TEST
023174                              DODU    LOGDEV          ;YES, DROP THIS LOGICAL DEV.
023174  013700  002334                                          MOV     LOGDEV,R0
023200  104451                                                  TRAP    C$DODU

023202  000240          AD.OK:      NOP                     ;(FOR PATCHING IN A HALT IF NECESSARY)

023204                          ENDAUTO
023204                                                          L10014:
023204  104461                                                  TRAP    C$AUTO
```

```
 4566  023206  050237  002546  AD.HIT: BIS     R2,TMP0         ;FLAG THE HIT IF WE GET IT!
 4567  023212  000002                  RTI                     ;RETURN
 4568
```

```
                                .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.
                                ;////////////////////////////////////////////////////////////////////////

023214                                  BGNCLN
023214                                                          L$CLEAN::

                                        ENDCLN

023214                                                          L10015:
023214  104412                                                        TRAP    C$CLEAN
```

```
4583
4584
4585
4586
4587
4588
4589
4590  023216
4591  023216
4592
4593  023216
4594  023216  104433
4595  023220
4596  023220
4597  023220  104453
```

```
.SBTTL  DROP UNIT SECTION

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

        BGNDU

L$DU::

;ISSUE UNIBUS RESET TO CLEAN UP
        BRESET

                TRAP    C$RESET

        ENDDU

L10016:
                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

                                                                        L$AU::

                                        ENDAU

                                                                        L10017:
023222  104452                                                                TRAP    C$AU
```

```
                              .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
023224                                                                      T1::

023224  004737  005376                JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

023230  004537  007234                JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
023234  063626                        DDCMP!NOCHK!SYNCH!STRIPS ;SET DDCMP,NO CHECK,SYNCH=226
023236  000000                        0                       ;USE 8 BIT CHARS
023240  103003                        BCC     .+8.            ;BR IF NO ERROR
023242                                ERROR                   ;REPORT STACKED ERROR
023242  104460                                                                  TRAP    C$ERROR
023244                                ESCAPE  TST             ;SKIP TO END OF TEST
023244  104410                                                                  TRAP    C$ESCAPE
023246  000254                                                                  .WORD   L10020-.

023250  004537  003712                JSR     R5,WRITEI       ;SET SHIFT REGISTER TO
023254  120013                        VIAACR                  ; 'SYSTEM CLOCK RATE'' MODE (CB1=CLK)
023256  000210                        210                     ;(BIT 7 PREVIOUSLY SET)
023260  103003                        BCC     .+8.            ;BR IF NO ERROR
023262                                ERROR                   ; REPORT STACKED ERROR
023262  104460                                                                  TRAP    C$ERROR
023264                                ESCAPE  TST             ;SKIP TO END OF TEST
023264  104410                                                                  TRAP    C$ESCAPE
023266  000234                                                                  .WORD   L10020-.

023270  004537  010250                JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
023274  000001                        TSOM                    ;AND KNOCK DOWN TBMT
023276  000007                        7.

023300  004537  010250                JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
023304  000001                        TSOM                    ;AND KNOCK DOWN TBMT
023306  000010                        8.

023310  004537  010250                JSR     R5,TXCTRL       ;CLEAR TSOM (GET READY TO SEND DATA)
023314  000000                        000
023316  000000                        0

023320  004537  010136                JSR     R5,TXCHAR       ;LOAD 125, TX 3RD SYNCH
023324  000125                        125
023326  000010                        8.
023330  103003                        BCC     .+8.            ;BR IF NO ERROR
```

```
4667  023332
4668  023332  104460                ERROR                   ;REPORT STACKED ERROR
                                                                      TRAP    C$ERROR
4669  023334
4670  023334  104410                ESCAPE  TST             ;SKIP TO END OF TEST
                                                                      TRAP    C$ESCAPE
4671  023336  000164                                                  .WORD   L10020-.
4672
                        ;-----------------------------------------------------------
4673  023340  012702  000004        MOV     #4,R2           ;** TRANSFER 4 CHARACTERS **
4674
4675  023344  004537  003712  1$:   JSR     R5,WRITEI       ;SET RTS & FULL DUPLEX (SO STEPLU WORKS)
4676  023350  120000                VIAORB
4677  023352  000142                TXEN!RXEN!TTLOOP
4678  023354  103003                BCC     .+8.            ;BR IF NO ERROR
4679  023356                        ERROR                   ; REPORT STACKED ERROR
4680  023356  104460                                                TRAP    C$ERROR
4681  023360                        ESCAPE  TST             ;SKIP TO END OF TEST
4682  023360  104410                                                TRAP    C$ESCAPE
4683  023362  000140                                                .WORD   L10020-.
4684
4685  023364  004537  012072        JSR     R5,STEPLU       ;FLIP TSO BIT VALUE(WILL BE SHIFTED INTO
4686  023370  000001                1                       ; FIFO DURING FLUSHING).
4687
4688  023372  004537  003712        JSR     R5,WRITEI       ;CLEAR RTS, SET HDX (SO THAT SR CLOCK WORKS)
4689  023376  120000                VIAORB
4690  023400  000156                TXEN!RXEN!RTSND!HDX!TTLOOP
4691  023402  103003                BCC     .+8.            ;BR IF NO ERROR
4692  023404                        ERROR                   ; REPORT STACKED ERROR
4693  023404  104460                                                TRAP    C$ERROR
4694  023406                        ESCAPE  TST             ;SKIP TO END OF TEST
4695  023406  104410                                                TRAP    C$ESCAPE
4696  023410  000112                                                .WORD   L10020-.
4697
4698  023412  004537  003566        JSR     R5,READI        ;READ VIA SHIFT REGISTER (SHOULD CAUSE
4699  023416  120012                VIASR                   ; 8 CLOCKS FROM CB1 LEAD => FIFO)
4700  023420  000000                000
4701  023422  103003                BCC     .+8.            ;BR IF NO ERROR
4702  023424                        ERROR                   ; REPORT STACKED ERROR
4703  023424  104460                                                TRAP    C$ERROR
4704  023426                        ESCAPE  TST             ;SKIP TO END OF TEST
4705  023426  104410                                                TRAP    C$ESCAPE
4706  023430  000072                                                .WORD   L10020-.
4707
4708  023432  077234                SOB     R2,1$           ;** LOOP UNTIL ALL 4 SENT (VIA CB1) **
4709                  ;-----------------------------------------------------------
4710  023434  004537  010350        JSR     R5,RXCHAR       ;READ AND CHECK FOR 377
4711  023440  000377                377                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
4712  023442  000000                0                       ;* DIDN'T WORK.
4713  023444  100000                NOCRDA
4714  023446  103003                BCC     .+8.            ;BR IF NO ERROR
4715  023450                        ERROR                   ;REPORT STACKED ERROR
4716  023450  104460                                                TRAP    C$ERROR
4717  023452                        ESCAPE  TST             ;SKIP TO END OF TEST
4718  023452  104410                                                TRAP    C$ESCAPE
4719  023454  000046                                                .WORD   L10020-.
4720
4721  023456  004537  010350        JSR     R5,RXCHAR       ;READ AND CHECK FOR 000
4722  023462  000003                003                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
```

```
4723  023464  000000                  0                       ;* DIDN'T WORK.
4724  023466  100000                  NOCRDA
4725  023470  103003                  BCC     .+8.            ;BR IF NO ERROR
4726  023472                          ERROR                   ;REPORT STACKED ERROR
4727  023472  104460                                                  TRAP    C$ERROR
4728  023474                          ESCAPE  TST             ;SKIP TO END OF TEST
4729  023474  104410                                                  TRAP    C$ESCAPE
4730  023476  000024                                                  .WORD   L10020-.
4731
4732  023500  004537  010350          JSR     R5,RXCHAR       ;READ AND CHECK FOR 377
4733  023504  000360                  360                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
4734  023506  000000                  0                       ;* DIDN'T WORK
4735  023510  100000                  NOCRDA
4736  023512  103003                  BCC     .+8.            ;BR IF NO ERROR
4737  023514                          ERROR                   ;REPORT STACKED ERROR
4738  023514  104460                                                  TRAP    C$ERROR
4739  023516                          ESCAPE  TST             ;SKIP TO END OF TEST
4740  023516  104410                                                  TRAP    C$ESCAPE
4741  023520  000002                                                  .WORD   L10020-.
4742  023522                  ENDTST
4743  023522                                                  L10020:
4744  023522  104401                                                  TRAP    C$ETST
```

```
4745                            .SBTTL  TEST 2 -- INTEGRAL MODEM INTERFACE TEST
4746
4747                            ;++*************************************************************************
4748                            ;*
4749                            ;*      TEST 2 -- INTEGRAL MODEM INTERFACE TEST
4750                            ;*
4751                            ;*  THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM AND A MESSAGE IS
4752                            ;*  TRANSMITTED,  RECEIVED,  AND CHECKED USING A TURNAROUND CONNECTOR ON
4753                            ;*  THE  BOARD  OR  AT THE END OF A CABLE. THE FOLLOWING MESSAGE WILL BE
4754                            ;*  SENT IN BCP MODE WITH CRC-16 SPECIFIED:
4755                            ;*
4756                            ;*      SYNC SYNC 000 125 252 377 000 CRC1 CRC2 SYNC
4757                            ;*
4758                            ;*  IF  THE  P-TABLE  FOR  THE  CURRENT  UNIT INDICATES THAT NO EXTERNAL
4759                            ;*  TURNAROUND IS PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
4760                            ;*
4761                            ;--*************************************************************************
4762
4763                            ;       BGNTST
4764  023524                                                                        T2::
4765  023524  012737  000002  002414        MOV     #2.,TSTNUM      ;SET TEST NO. FOR POSSIBLE PRINTOUT
4766  023532  004737  005376                 JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
4767  023536  004537  007532                 JSR     R5,CKLPBK       ;SEE IF THIS INTERFACE CAN BE RUN
4768  023542  000001                         INTGRL
4769  023544  103002                         BCC     2$              ;BR IF YES
4770  023546                                 EXIT    TST             ;WRONG INTERFACE - SKIP TEST
4771  023546  104432                                                                TRAP    C$EXIT
4772  023550  001630                                                                .WORD   L10021-.
4773
4774                            ;LOAD 6502 MICROCODE FOR INTEGRAL MODEM TEST INTO RAM PAGE 2
4775  023552  012701  025402    2$:     MOV     #MCODE,R1       ;GET STARTING ADRS OF DMV MICROCODE
4776  023556  012702  001000            MOV     #RAMADR,R2      ;GET STARTING ADRS OF RAM PAGE 2
4777  023562  112137  023600    3$:     MOVB    (R1)+,6$        ;SET DATA BYTE TO BE WRITTEN
4778  023566  010237  023576            MOV     R2,4$           ;SET RAM WRITE ADRS
4779  023572  004537  003712            JSR     R5,WRITEI       ;WRITE A DATA BYTE INTO RAM
4780  023576  000000    4$:     .WORD   0
4781  023600  000000    6$:     .WORD   0
4782  023602  005202            INC     R2              ;INCR RAM ADRS
4783  023604  020127  026227            CMP     R1,#ENDCOD      ;SEE IF ALL CODE LOADED YET
4784  023610  103764                    BLO     3$              ;BR IF NOT
4785
4786                            ;READ AND VERIFY 6502 MICROCODE IN RAM
4787  023612  012701  025402            MOV     #MCODE,R1       ;GET STARTING ADRS OF DMV MICROCODE TO CHECK
4788  023616  012702  001000            MOV     #RAMADR,R2      ;GET STARTING ADRS OF RAM PAGE 2
4789  023622  010237  023632    8$:     MOV     R2,10$          ;SET RAM READ ADRS
4790  023626  004537  003566            JSR     R5,READI        ;READ A RAM BYTE
4791  023632  000000    10$:    .WORD   0
4792  023634  000000    12$:    .WORD   0
4793  023636  122137  023634            CMPB    (R1)+,12$       ;SEE IF BYTE IS CORRECT
4794  023642  001422                    BEQ     16$             ;BR IF CORRECT
4795  023644  010237  002336            MOV     R2,REGNUM       ;SET RAM ADRS FOR ERROR REPORT
4796  023650  005037  002324            CLR     GDATA           ;SET EXPECTED RAM DATA
4797  023654  116137  177777  002324    MOVB    -1(R1),GDATA
4798  023662  005037  002326            CLR     BDATA           ;SET ACTUAL RAM DATA
4799  023666  113737  023634  002326    MOVB    12$,BDATA
4800                            ;REPORT RAM ERROR LOADING MICROCODE
```

```
4801  023674                                        GEDF    EM79,ERR13
4802                                                ;       'DEVICE FATAL'' ERROR # 38
4803  023674  104455                                                          TRAP    C$ERDF
4804  023676  000046                                                          .WORD   38
4805  023700  015125                                                          .WORD   EM79
4806  023702  021030                                                          .WORD   ERR13
4807  023704                                        ESCAPE  TST
4808  023704  104410                                                          TRAP    C$ESCAPE
4809  023706  001472                                                          .WORD   L10021-.
4810  023710  005202                          16$:  INC     R2                ;INCR RAM ADRS
4811  023712  020127  026227                      CMP     R1,#ENDCOD        ;SEE IF ALL CODE CHECKED YET
4812  023716  103741                              BLO     8$                ;BR IF NOT
4813
4814                                        ;SET UP VIA AND USYRT FOR OPERATION
4815  023720  004537  003712                      JSR     R5,WRITEI         ;RESET THE USYRT
4816  023724  120000                              VIAORB
4817  023726  000031                              RTSND!DTR!PRESET
4818  023730  004537  003712                      JSR     R5,WRITEI         ;CLEAR USYRT RESET BIT
4819  023734  120000                              VIAORB
4820  023736  000030                              RTSND!DTR
4821  023740  004537  003712                      JSR     R5,WRITEI         ;SET SYNCH CHAR = 226
4822  023744  120404                              PCSARL
4823  023746  000226                              226
4824  023750  012737  065400  023770              MOV     #DDCMP!STRIPS!IDLES!CRC16,18$   ;SET DDCMP,STRIP,IDLE, CRC16
4825  023756  000337  023770                      SWAB    18$               ;GET DATA INTO LO BYTE
4826  023762  004537  003712                      JSR     R5,WRITEI         ;PROGRAM THE PCSARH
4827  023766  120405                              PCSARH
4828  023770  000000                        18$:  .WORD   0
4829  023772  004537  005410                      JSR     R5,CKUSTS         ;CHK USYRT STATUS FOR INITIALIZED STATE
4830  023776  000110                              110                       ;  TBMT=1, TSO=1
4831  024000  103003                              BCC     .+8.              ;IF ERROR, PRINT REPORT
4832  024002                                        ERROR
4833  024002  104460                                                                 TRAP    C$ERROR
4834  024004                                        ESCAPE  TST
4835  024004  104410                                                                 TRAP    C$ESCAPE
4836  024006  001372                                                                 .WORD   L10021-.
4837  024010  004537  003712                      JSR     R5,WRITEI         ;SET TSOM IN USYRT
4838  024014  120403                              TDSRH
4839  024016  000001                              TSOM
4840  024020  004537  003712                      JSR     R5,WRITEI         ;LOAD 237 CHAR FOR INTGRL MODEM SYNCHRONIZATION
4841  024024  120402                              TDSRL
4842  024026  000237                              237
4843  024030  004537  006014                      JSR     R5,CKTBMT         ;CHK FOR TBMT = 0
4844  024034  000000                              0
4845  024036  103003                              BCC     .+8.              ;IF ERROR, REPORT ERROR
4846  024040                                        ERROR
4847  024040  104460                                                                 TRAP    C$ERROR
4848  024042                                        ESCAPE  TST
4849  024042  104410                                                                 TRAP    C$ESCAPE
4850  024044  001334                                                                 .WORD   L10021-.
4851
4852                                        ;INITIATE 6502 TEST OF INTEGRAL MODEM
4853  024046  005077  156354                      CLR     @SEL4             ;CLEAR SEL4
4854  024052  012777  001000  156346              MOV     #RAMADR,@SEL4     ;SET START ADRS OF RAM CODE IN SEL4
4855  024060  112777  000005  156334              MOVB    #EXECUT,@BSEL2    ;ISSUE M-LOOP CMND TO EXECUTE AT PC IN SEL4
4856                                        ;WAIT SEVERAL MILLI-SEC FOR COMPLETION OF TEST
```

```
 4857  024066  012701  001750            MOV     #1000.,R1       ;INIT WAIT LOOP COUNTER
 4858  024072  005301            22$:    DEC     R1              ;DECREMENT COUNTER
 4859  024074  000240                    NOP
 4860  024076  001375                    BNE     22$             ;BR IF NOT DONE COUNTING YET
 4861  024100  132777  000200  156314    BITB    #MRDY,@BSEL2    ;SEE IF M-LOOP FINISHED PROPERLY
 4862  024106  001013                    BNE     24$             ;BR IF YES
 4863  024110  004737  004166            JSR     PC,GETWSR       ;GET CSR'S FOR PRINTOUT
 4864  024114  012737  000005  002324    MOV     #EXECUT,GDATA   ;IDENTIFY REQUESTED FUNCTION
 4865                              ;REPORT MRDY TIMEOUT ERROR
 4866  024122                            GEDF    EM4,ERR4
 4867                                                    ;       'DEVICE FATAL'' ERROR # 39
 4868  024122  104455                                                    TRAP    C$ERDF
 4869  024124  000047                                                    .WORD   39
 4870  024126  014307                                                    .WORD   EM4
 4871  024130  020120                                                    .WORD   ERR4
 4872  024132                            ESCAPE  TST
 4873  024132  104410                                                    TRAP    C$ESCAPE
 4874  024134  001244                                                    .WORD   L10021-.
 4875
 4876  024136  105777  156270    24$:    TSTB    @BSEL6          ;SEE IF ANY ERRORS OCCURRED IN TEST
 4877  024142  001002                    BNE     26$             ;BR IF ERROR DETECTED
 4878  024144  000137  025400            JMP     90$             ;SUCCESSFUL COMPLETION OF TEST
 4879  024150  012737  000007  002336    MOV     #7,REGNUM       ;SET REG NO. FOR PRINTOUT
 4880  024156  005037  002324            CLR     GDATA           ;CLEAR EXPECTED DATA AREA
 4881  024162  005037  002326            CLR     BDATA           ;CLEAR ACTUAL DATA AREA
 4882                              ;CHK FOR ERROR 1
 4883  024166  127727  156240  000001    CMPB    @BSEL6,#1       ;CHK FOR ERROR 1
 4884  024174  001006                    BNE     28$             ;BR IF NOT
 4885                              ;REPORT TBMT NOT SET
 4886  024176                            GEDF    EM73,ERR12
 4887                                                    ;       'DEVICE FATAL'' ERROR # 40
 4888  024176  104455                                                    TRAP    C$ERDF
 4889  024200  000050                                                    .WORD   40
 4890  024202  014777                                                    .WORD   EM73
 4891  024204  020714                                                    .WORD   ERR12
 4892  024206                            ESCAPE  TST
 4893  024206  104410                                                    TRAP    C$ESCAPE
 4894  024210  001170                                                    .WORD   L10021-.
 4895                              ;CHK FOR ERROR 2
 4896  024212  127727  156214  000002  28$:    CMPB    @BSEL6,#2       ;CHK FOR ERROR 2
 4897  024220  001006                    BNE     30$             ;BR IF NOT
 4898                              ;REPORT TBMT NOT SET
 4899  024222                            GEDF    EM73,ERR12
 4900                                                    ;       'DEVICE FATAL'' ERROR # 41
 4901  024222  104455                                                    TRAP    C$ERDF
 4902  024224  000051                                                    .WORD   41
 4903  024226  014777                                                    .WORD   EM73
 4904  024230  020714                                                    .WORD   ERR12
 4905  024232                            ESCAPE  TST
 4906  024232  104410                                                    TRAP    C$ESCAPE
 4907  024234  001144                                                    .WORD   L10021-.
 4908                              ;CHK FOR ERROR 3
 4909  024236  127727  156170  000003  30$:    CMPB    @BSEL6,#3       ;CHK FOR ERROR 3
 4910  024244  001006                    BNE     31$             ;BR IF NOT
 4911                              ;REPORT TBMT NOT SET
 4912  024246                            GEDF    EM73,ERR12
```

```
4913
4914  024246  104455                                          ;       'DEVICE FATAL'' ERROR # 42
4915  024250  000052                                                  TRAP    C$ERDF
4916  024252  014777                                                  .WORD   42
4917  024254  020714                                                  .WORD   EM73
4918  024256                                                          .WORD   ERR12
4919  024256  104410          ESCAPE  TST
4920  024260  001120                                                  TRAP    C$ESCAPE
4921                                                                  .WORD   L10021-.
4922  024262  127727  156144  000004  ;CHK FOR ERROR 4
4923  024270  001011  31$:    CMPB    @BSEL6,#4       ;CHK FOR ERROR 4
4924                  ;REPORT BNE     32$             ;BR IF NOT
4925  024272  012737  000001  002336  CARRIER NOT SET
4926  024300                  MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
4927                          GEDF    EM80,ERR14
4928  024300  104455                                          ;       'DEVICE FATAL'' ERROR # 43
4929  024302  000053                                                  TRAP    C$ERDF
4930  024304  015161                                                  .WORD   43
4931  024306  021146                                                  .WORD   EM80
4932  024310                                                          .WORD   ERR14
4933  024310  104410          ESCAPE  TST
4934  024312  001066                                                  TRAP    C$ESCAPE
4935                                                                  .WORD   L10021-.
4936  024314  127727  156112  000005  ;CHK FOR ERROR 5
4937  024322  001006  32$:    CMPB    @BSEL6,#5       ;CHK FOR ERROR 5
4938                  ;REPORT BNE     34$             ;BR IF NOT
4939  024324                  TBMT NOT SET
4940                          GEDF    EM73,ERR12
4941  024324  104455                                          ;       'DEVICE FATAL'' ERROR # 44
4942  024326  000054                                                  TRAP    C$ERDF
4943  024330  014777                                                  .WORD   44
4944  024332  020714                                                  .WORD   EM73
4945  024334                                                          .WORD   ERR12
4946  024334  104410          ESCAPE  TST
4947  024336  001042                                                  TRAP    C$ESCAPE
4948                                                                  .WORD   L10021-.
4949  024340  127727  156066  000006  ;CHK FOR ERROR 6
4950  024346  001006  34$:    CMPB    @BSEL6,#6       ;CHK FOR ERROR 6
4951                  ;REPORT BNE     36$             ;BR IF NOT
4952  024350                  TBMT NOT SET
4953                          GEDF    EM73,ERR12
4954  024350  104455                                          ;       'DEVICE FATAL'' ERROR # 45
4955  024352  000055                                                  TRAP    C$ERDF
4956  024354  014777                                                  .WORD   45
4957  024356  020714                                                  .WORD   EM73
4958  024360                                                          .WORD   ERR12
4959  024360  104410          ESCAPE  TST
4960  024362  001016                                                  TRAP    C$ESCAPE
4961                                                                  .WORD   L10021-.
4962  024364  127727  156042  000007  ;CHK FOR ERROR 7
4963  024372  001006  36$:    CMPB    @BSEL6,#7       ;CHK FOR ERROR 7
4964                  ;REPORT BNE     38$             ;BR IF NOT
4965  024374                  TBMT NOT SET
4966                          GEDF    EM73,ERR12
4967  024374  104455                                          ;       'DEVICE FATAL'' ERROR # 46
4968  024376  000056                                                  TRAP    C$ERDF
                                                                      .WORD   46
```

```
4969  024400  014777                                                          .WORD   EM73
4970  024402  020714                                                          .WORD   ERR12
4971  024404
4972  024404  104410                                                          TRAP    C$ESCAPE
4973  024406  000772                                                          .WORD   L10021-.
4974                          ;CHK FOR ERROR 8
4975  024410  127727  156016  000010  38$:    CMPB    @BSEL6,#8.      ;CHK FOR ERROR 8
4976  024416  001006                  BNE     40$             ;BR IF NOT
4977                          ;REPORT TBMT NOT SET
4978  024420                          GEDF    EM73,ERR12
4979                                                  ;       'DEVICE FATAL'' ERROR # 47
4980  024420  104455                                                          TRAP    C$ERDF
4981  024422  000057                                                          .WORD   47
4982  024424  014777                                                          .WORD   EM73
4983  024426  020714                                                          .WORD   ERR12
4984  024430
4985  024430  104410                                                          TRAP    C$ESCAPE
4986  024432  000746                                                          .WORD   L10021-.
4987                          ;CHK FOR ERROR 9
4988  024434  127727  155772  000011  40$:    CMPB    @BSEL6,#9.      ;CHK FOR ERROR 9
4989  024442  001006                  BNE     42$             ;BR IF NOT
4990                          ;REPORT RDA NOT SET
4991  024444                          GEDF    EM75,ERR12
4992                                                  ;       'DEVICE FATAL'' ERROR # 48
4993  024444  104455                                                          TRAP    C$ERDF
4994  024446  000060                                                          .WORD   48
4995  024450  015035                                                          .WORD   EM75
4996  024452  020714                                                          .WORD   ERR12
4997  024454
4998  024454  104410                                                          TRAP    C$ESCAPE
4999  024456  000722                                                          .WORD   L10021-.
5000                          ;CHK FOR ERROR 10
5001  024460  127727  155746  000012  42$:    CMPB    @BSEL6,#10.     ;CHK FOR ERROR 10
5002  024466  001017                  BNE     44$             ;BR IF NOT
5003                          ;REPORT RCV'D DATA MISCOMPARE ERROR
5004  024470  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5005  024476  112737  000000  002324          MOVB    #000,GDATA      ;SET EXPECTED DATA
5006  024504  117737  155724  002326          MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5007  024512                          GEDF    EM34,ERR10
5008                                                  ;       'DEVICE FATAL'' ERROR # 49
5009  024512  104455                                                          TRAP    C$ERDF
5010  024514  000061                                                          .WORD   49
5011  024516  014502                                                          .WORD   EM34
5012  024520  020364                                                          .WORD   ERR10
5013  024522
5014  024522  104410                                                          TRAP    C$ESCAPE
5015  024524  000654                                                          .WORD   L10021-.
5016                          ;CHK FOR ERROR 11
5017  024526  127727  155700  000013  44$:    CMPB    @BSEL6,#11.     ;CHK FOR ERROR 11
5018  024534  001006                  BNE     46$             ;BR IF NOT
5019                          ;REPORT RDA NOT SET
5020  024536                          GEDF    EM75,ERR12
5021                                                  ;       'DEVICE FATAL'' ERROR # 50
5022  024536  104455                                                          TRAP    C$ERDF
5023  024540  000062                                                          .WORD   50
5024  024542  015035                                                          .WORD   EM75
```

```
024544  020714                                                                   .WORD   ERR12
024546                                            ESCAPE  TST
024546  104410                                                                   TRAP    C$ESCAPE
024550  000630                                                                   .WORD   L10021-.
                                          ;CHK FOR ERROR 12
024552  127727  155654  000014    46$:    CMPB    @BSEL6,#12.     ;CHK FOR ERROR 12
024560  001017                            BNE     48$             ;BR IF NOT
                                          ;REPORT RCV'D DATA MISCOMPARE ERROR
024562  012737  000000  002336            MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
024570  112737  000125  002324            MOVB    #125,GDATA      ;SET EXPECTED DATA
024576  117737  155632  002326            MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
024604                                    GEDF    EM34,ERR10
                                                                  ;       'DEVICE FATAL'' ERROR # 51
024604  104455                                                                   TRAP    C$ERDF
024606  000063                                                                   .WORD   51
024610  014502                                                                   .WORD   EM34
024612  020364                                                                   .WORD   ERR10
024614                                    ESCAPE  TST
024614  104410                                                                   TRAP    C$ESCAPE
024616  000562                                                                   .WORD   L10021-.
                                          ;CHK FOR ERROR 13
024620  127727  155606  000015    48$:    CMPB    @BSEL6,#13.     ;CHK FOR ERROR 13
024626  001006                            BNE     50$             ;BR IF NOT
                                          ;REPORT RDA NOT SET
024630                                    GEDF    EM75,ERR12
                                                                  ;       'DEVICE FATAL'' ERROR # 52
024630  104455                                                                   TRAP    C$ERDF
024632  000064                                                                   .WORD   52
024634  015035                                                                   .WORD   EM75
024636  020714                                                                   .WORD   ERR12
024640                                    ESCAPE  TST
024640  104410                                                                   TRAP    C$ESCAPE
024642  000536                                                                   .WORD   L10021-.
                                          ;CHK FOR ERROR 14
024644  127727  155562  000016    50$:    CMPB    @BSEL6,#14.     ;CHK FOR ERROR 14
024652  001017                            BNE     52$             ;BR IF NOT
                                          ;REPORT RCV'D DATA MISCOMPARE ERROR
024654  012737  000000  002336            MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
024662  112737  000252  002324            MOVB    #252,GDATA      ;SET EXPECTED DATA
024670  117737  155540  002326            MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
024676                                    GEDF    EM34,ERR10
                                                                  ;       'DEVICE FATAL'' ERROR # 53
024676  104455                                                                   TRAP    C$ERDF
024700  000065                                                                   .WORD   53
024702  014502                                                                   .WORD   EM34
024704  020364                                                                   .WORD   ERR10
024706                                    ESCAPE  TST
024706  104410                                                                   TRAP    C$ESCAPE
024710  000470                                                                   .WORD   L10021-.
                                          ;CHK FOR ERROR 15
024712  127727  155514  000017    52$:    CMPB    @BSEL6,#15.     ;CHK FOR ERROR 15
024720  001006                            BNE     54$             ;BR IF NOT
                                          ;REPORT RDA NOT SET
024722                                    GEDF    EM75,ERR12
                                                                  ;       'DEVICE FATAL'' ERROR # 54
024722  104455                                                                   TRAP    C$ERDF
```

```
5081  024724  000066                                                          .WORD   54
5082  024726  015035                                                          .WORD   EM75
5083  024730  020714                                                          .WORD   ERR12
5084  024732                      ESCAPE  TST
5085  024732  104410                                                          TRAP    C$ESCAPE
5086  024734  000444                                                          .WORD   L10021-.
5087                              ;CHK FOR ERROR 16
5088  024736  127727  155470  000020  54$:    CMPB    @BSEL6,#16.     ;CHK FOR ERROR 16
5089  024744  001017                        BNE     56$             ;BR IF NOT
5090                              ;REPORT RCV'D DATA MISCOMPARE ERROR
5091  024746  012737  000000  002336        MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5092  024754  112737  000377  002324        MOVB    #377,GDATA      ;SET EXPECTED DATA
5093  024762  117737  155446  002326        MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5094  024770                                GEDF    EM34,ERR10
5095                                                ;       'DEVICE FATAL'' ERROR # 55
5096  024770  104455                                                          TRAP    C$ERDF
5097  024772  000067                                                          .WORD   55
5098  024774  014502                                                          .WORD   EM34
5099  024776  020364                                                          .WORD   ERR10
5100  025000                      ESCAPE  TST
5101  025000  104410                                                          TRAP    C$ESCAPE
5102  025002  000376                                                          .WORD   L10021-.
5103                              ;CHK FOR ERROR 17
5104  025004  127727  155422  000021  56$:    CMPB    @BSEL6,#17.     ;CHK FOR ERROR 17
5105  025012  001006                        BNE     58$             ;BR IF NOT
5106                              ;REPORT RDA NOT SET
5107  025014                                GEDF    EM75,ERR12
5108                                                ;       'DEVICE FATAL'' ERROR # 56
5109  025014  104455                                                          TRAP    C$ERDF
5110  025016  000070                                                          .WORD   56
5111  025020  015035                                                          .WORD   EM75
5112  025022  020714                                                          .WORD   ERR12
5113  025024                      ESCAPE  TST
5114  025024  104410                                                          TRAP    C$ESCAPE
5115  025026  000352                                                          .WORD   L10021-.
5116                              ;CHK FOR ERROR 18
5117  025030  127727  155376  000022  58$:    CMPB    @BSEL6,#18.     ;CHK FOR ERROR 18
5118  025036  001017                        BNE     60$             ;BR IF NOT
5119                              ;REPORT RCV'D DATA MISCOMPARE ERROR
5120  025040  012737  000000  002336        MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5121  025046  112737  000000  002324        MOVB    #000,GDATA      ;SET EXPECTED DATA
5122  025054  117737  155354  002326        MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5123  025062                                GEDF    EM34,ERR10
5124                                                ;       'DEVICE FATAL'' ERROR # 57
5125  025062  104455                                                          TRAP    C$ERDF
5126  025064  000071                                                          .WORD   57
5127  025066  014502                                                          .WORD   EM34
5128  025070  020364                                                          .WORD   ERR10
5129  025072                      ESCAPE  TST
5130  025072  104410                                                          TRAP    C$ESCAPE
5131  025074  000304                                                          .WORD   L10021-.
5132                              ;CHK FOR ERROR 19
5133  025076  127727  155330  000023  60$:    CMPB    @BSEL6,#19.     ;CHK FOR ERROR 19
5134  025104  001006                        BNE     62$             ;BR IF NOT
5135                              ;REPORT RERR NOT SET
5136  025106                                GEDF    EM36,ERR10
```

```
5137
5138  025106  104455                                          ;       'DEVICE FATAL'' ERROR # 58
5139  025110  000072                                                  TRAP    C$ERDF
5140  025112  014551                                                  .WORD   58
5141  025114  020364                                                  .WORD   EM36
5142  025116                                                          .WORD   ERR10
5143  025116  104410                  ESCAPE  TST                     TRAP    C$ESCAPE
5144  025120  000260                                                  .WORD   L10021-.
5145                          ;CHK FOR ERROR 20
5146  025122  127727  155304  000024  62$:     CMPB    @BSEL6,#20.     ;CHK FOR ERROR 20
5147  025130  001006                  BNE     64$             ;BR IF NOT
5148                          ;REPORT RDA NOT SET
5149  025132                          GEDF    EM75,ERR12
5150                                                          ;       'DEVICE FATAL'' ERROR # 59
5151  025132  104455                                                  TRAP    C$ERDF
5152  025134  000073                                                  .WORD   59
5153  025136  015035                                                  .WORD   EM75
5154  025140  020714                                                  .WORD   ERR12
5155  025142                          ESCAPE  TST
5156  025142  104410                                                  TRAP    C$ESCAPE
5157  025144  000234                                                  .WORD   L10021-.
5158                          ;CHK FOR ERROR 21
5159  025146  127727  155260  000025  64$:     CMPB    @BSEL6,#21.     ;CHK FOR ERROR 21
5160  025154  001017                  BNE     66$             ;BR IF NOT
5161                          ;REPORT RCV'D DATA MISCOMPARE ERROR
5162  025156  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5163  025164  112737  000160  002324          MOVB    #160,GDATA      ;SET EXPECTED DATA
5164  025172  117737  155236  002326          MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5165  025200                          GEDF    EM34,ERR10
5166                                                          ;       'DEVICE FATAL'' ERROR # 60
5167  025200  104455                                                  TRAP    C$ERDF
5168  025202  000074                                                  .WORD   60
5169  025204  014502                                                  .WORD   EM34
5170  025206  020364                                                  .WORD   ERR10
5171  025210                          ESCAPE  TST
5172  025210  104410                                                  TRAP    C$ESCAPE
5173  025212  000166                                                  .WORD   L10021-.
5174                          ;CHK FOR ERROR 22
5175  025214  127727  155212  000026  66$:     CMPB    @BSEL6,#22.     ;CHK FOR ERROR 22
5176  025222  001006                  BNE     68$             ;BR IF NOT
5177                          ;REPORT RDA NOT SET
5178  025224                          GEDF    EM75,ERR12
5179                                                          ;       'DEVICE FATAL'' ERROR # 61
5180  025224  104455                                                  TRAP    C$ERDF
5181  025226  000075                                                  .WORD   61
5182  025230  015035                                                  .WORD   EM75
5183  025232  020714                                                  .WORD   ERR12
5184  025234                          ESCAPE  TST
5185  025234  104410                                                  TRAP    C$ESCAPE
5186  025236  000142                                                  .WORD   L10021-.
5187                          ;CHK FOR ERROR 23
5188  025240  127727  155166  000027  68$:     CMPB    @BSEL6,#23.     ;CHK FOR ERROR 23
5189  025246  001017                  BNE     70$             ;BR IF NOT
5190                          ;REPORT RCV'D DATA MISCOMPARE ERROR
5191  025250  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5192  025256  112737  000034  002324          MOVB    #034,GDATA      ;SET EXPECTED DATA
```

```
5193  025264  117737  155144  002326          MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5194  025272                                  GEDF    EM34,ERR10
5195                                                    ;       'DEVICE FATAL'' ERROR # 62
5196  025272  104455                                                   TRAP    C$ERDF
5197  025274  000076                                                   .WORD   62
5198  025276  014502                                                   .WORD   EM34
5199  025300  020364                                                   .WORD   ERR10
5200  025302                                  ESCAPE  TST
5201  025302  104410                                                   TRAP    C$ESCAPE
5202  025304  000074                                                   .WORD   L10021-.
5203                                  ;CHK FOR ERROR 24
5204  025306  127727  155120  000030  70$:     CMPB    @BSEL6,#24.     ;CHK FOR ERROR 24
5205  025314  001006                          BNE     72$             ;BR IF NOT
5206                                  ;REPORT RDA NOT SET
5207  025316                                  GEDF    EM75,ERR12
5208                                                    ;       'DEVICE FATAL'' ERROR # 63
5209  025316  104455                                                   TRAP    C$ERDF
5210  025320  000077                                                   .WORD   63
5211  025322  015035                                                   .WORD   EM75
5212  025324  020714                                                   .WORD   ERR12
5213  025326                                  ESCAPE  TST
5214  025326  104410                                                   TRAP    C$ESCAPE
5215  025330  000050                                                   .WORD   L10021-.
5216                                  ;CHK FOR ERROR 25
5217  025332  127727  155074  000031  72$:     CMPB    @BSEL6,#25.     ;CHK FOR ERROR 25
5218  025340  001011                          BNE     74$             ;BR IF NOT
5219                                  ;REPORT CARRIER NOT CLEARED
5220  025342  012737  000001  002336          MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
5221  025350                                  GEDF    EM81,ERR14
5222                                                    ;       'DEVICE FATAL'' ERROR # 64
5223  025350  104455                                                   TRAP    C$ERDF
5224  025352  000100                                                   .WORD   64
5225  025354  015201                                                   .WORD   EM81
5226  025356  021146                                                   .WORD   ERR14
5227  025360                                  ESCAPE  TST
5228  025360  104410                                                   TRAP    C$ESCAPE
5229  025362  000016                                                   .WORD   L10021-.
5230  025364  004737  004166          74$:     JSR     PC,GETWSR       ;GET CSR'S FOR PRINTOUT
5231                                  ;REPORT INVALID ERROR CODE FROM 6502
5232  025370                                  GEDF    EM82,ERR3
5233                                                    ;       'DEVICE FATAL'' ERROR # 65
5234  025370  104455                                                   TRAP    C$ERDF
5235  025372  000101                                                   .WORD   65
5236  025374  015225                                                   .WORD   EM82
5237  025376  020072                                                   .WORD   ERR3
5238
5239  025400                          90$:
5240  025400                          ENDTST
5241  025400                                                          L10021:
5242  025400  104401                                                   TRAP    C$ETST
```

```
5243  025402                MCODE:
5244                        ;
5245                        ;LINE# LOC   CODE        LINE
5246                        ;
5247                        ;0001  0000                    *=$0200                 ;START OF MICROCODE FOR INTEGRAL
5248                        ;0002  0200
5249                        ;0003  0200
5250                        ;0005  0200
5251                        ;0006  0200  ;*****************************************************************************
5252                        ;0007  0200  ; THIS IS THE 6502 MICROCODE WHICH IS LOADED INTO RAM AND EXECUTED FOR THE
5253                        ;0008  0200  ; PURPOSE OF TESTING THE INTEGRAL MODEM ON THE M8064, AT 56K BAUD. AFTER TH
5254                        ;0009  0200  ; LSI-11 PROGRAM DOES SOME INITIAL SETUP, IT TRANSFERS CONTROL TO THIS CODE
5255                        ;0010  0200  ; IN RAM, AND WAITS FOR COMPLETION OF THE TEST, AS INDICATED BY MRDY SET.
5256                        ;0011  0200  ; THIS CODE TRANSMITS, RECEIVES, AND CHECKS THE FOLLOWING CHARACTERS :
5257                        ;0012  0200  ; 2 SYNCH CHARACTERS, 5 DATA CHARACTERS 000, 125, 252, 377, 000, 2 CRC-16
5258                        ;0013  0200  ; CHARACTERS 160 AND 034, AND 2 TERMINATING SYNCHS. THE MESSAGE IS SENT USI
5259                        ;0014  0200  ; CHARACTER (DDCMP) MODE, THE SYNCH CHARACTER USED IS 226, STRIP SYNCH AND
5260                        ;0015  0200  ; IDLE MODES ARE SET, AND THE DATA CLOCK IS PROVIDED BY THE INTEGRAL MODEM.
5261                        ;0016  0200  ; ALL DATA AND CRC CHARACTERS ARE CHECKED AS THEY ARE RECEIVED, AND THE CRC
5262                        ;0017  0200  ; ERROR CHECK BIT IS CHECKED TO BE SET WITH RECEPTION OF THE LAST DATA
5263                        ;0018  0200  ; CHARACTER (000).
5264                        ;0019  0200  ;*****************************************************************************
5265                        ;0020  0200
5266                        ;0021  0200            ;EQUATES FOR BIT DEFINITIONS
5267                        ;0022  0200          BIT0   =a1
5268                        ;0023  0200          BIT1   =a2
5269                        ;0024  0200          BIT2   =a4
5270                        ;0025  0200          BIT3   =a10
5271                        ;0026  0200          BIT4   =a20
5272                        ;0027  0200          BIT5   =a40
5273                        ;0028  0200          BIT6   =a100
5274                        ;0029  0200          BIT7   =a200
5275                        ;0030  0200          BIT8   =a400
5276                        ;0031  0200          BIT9   =a1000
5277                        ;0032  0200          BIT10  =a2000
5278                        ;0033  0200          BIT11  =a4000
5279                        ;0034  0200          BIT12  =a10000
5280                        ;0035  0200          BIT13  =a20000
5281                        ;0036  0200          BIT14  =a40000
5282                        ;0037  0200          BIT15  =a100000
5283                        ;0038  0200
5284                        ;0039  0200
5285                        ;0040  0200
5286                        ;0041  0200            ;ADDRESS EQUATES FOR CSR REGISTERS
5287                        ;0042  0200          SEL0   =$10
5288                        ;0043  0200          BSEL0  =SEL0
5289                        ;0044  0200          BSEL1  =SEL0+1
5290                        ;0045  0200          SEL2   =SEL0+2
5291                        ;0046  0200          BSEL2  =SEL0+2
5292                        ;0047  0200          BSEL3  =SEL0+3
5293                        ;0048  0200          SEL4   =SEL0+4
5294                        ;0049  0200          BSEL4  =SEL0+4
5295                        ;0050  0200          BSEL5  =SEL0+5
5296                        ;0051  0200          SEL6   =SEL0+6
5297                        ;0052  0200          BSEL6  =SEL0+6
5298                        ;0053  0200          BSEL7  =SEL0+7
```

```
5299                    ;0054  0200
5300                    ;0055  0200
5301                    ;0056  0200
5302                    ;0057  0200          ;VERSATILE INTERFACE ADAPTER REGISTER EQUATES
5303                    ;0058  0200  OREGB  = $A000                    ;OUTPUT REGISTER B
5304                    ;0059  0200  OREGA  = OREGB+1                  ;OUTPUT REGISTER A
5305                    ;0060  0200  DDRB   = OREGB+2                  ;DATA DIRECTION REGISTER B
5306                    ;0061  0200  DDRA   = OREGB+3                  ;DATA DIRECTION REGISTER A
5307                    ;0062  0200  T1LL   = OREGB+6                  ;TIMER 1 LATCH LOW BITS
5308                    ;0063  0200  T1LH   = OREGB+7                  ;TIMER 1 LATCH HIGH BITS
5309                    ;0064  0200  ACR    = OREGB+$B                 ;AUXILIARY CONTROL REGISTER
5310                    ;0065  0200  PCR    = OREGB+$C                 ;PERIPHERAL CONTROL REGISTER
5311                    ;0066  0200
5312                    ;0067  0200
5313                    ;0068  0200
5314                    ;0069  0200          ;VIA OUTPUT REGISTER B BIT EQUATES
5315                    ;0070  0200  NULCLK = BIT7
5316                    ;0071  0200  RXEN   = BIT6
5317                    ;0072  0200  TXEN   = BIT5
5318                    ;0073  0200  DTR    = BIT4
5319                    ;0074  0200  RTSND  = BIT3
5320                    ;0075  0200  HDX    = BIT2
5321                    ;0076  0200  TTLOOP = BIT1
5322                    ;0077  0200  PRESET = BIT0
5323                    ;0078  0200
5324                    ;0079  0200
5325                    ;0080  0200
5326                    ;0081  0200          ;VIA OUTPUT REGISTER A BIT EQUATES
5327                    ;0082  0200  RING   = BIT7
5328                    ;0083  0200  CARIER = BIT6
5329                    ;0084  0200  MDMRDY = BIT5
5330                    ;0085  0200  BDRATE = BIT4
5331                    ;0086  0200  CTS    = BIT3
5332                    ;0087  0200  TM     = BIT2
5333                    ;0088  0200  RCVDAT = BIT1
5334                    ;0089  0200  UMAINT = BIT0
5335                    ;0090  0200
5336                    ;0091  0200
5337                    ;0092  0200          ;USYRT REGISTER ADDRESS EQUATES
5338                    ;0093  0200  RXDB   = $A100
5339                    ;0094  0200  RDSR   = RXDB+1
5340                    ;0095  0200  TXDB   = RXDB+2
5341                    ;0096  0200  TDSR   = RXDB+3
5342                    ;0097  0200  SAR    = RXDB+4
5343                    ;0098  0200  PCSAR  = RXDB+5
5344                    ;0099  0200  PCTLR  = RXDB+7
5345                    ;0100  0200
5346                    ;0101  0200
5347                    ;0102  0200
5348                    ;0103  0200          ;USYRT TDSR REGISTER BIT EQUATES
5349                    ;0104  0200  TEOM   = BIT1
5350                    ;0105  0200  TSOM   = BIT0
5351                    ;0106  0200
5352                    ;0107  0200
5353                    ;0108  0200
5354                    ;0109  0200          ;USYRT RDSR BIT EQUATES
```

```
                                  ;0110  0200                  RERR   = BIT7
                                  ;0111  0200
                                  ;0112  0200
                                  ;0113  0200
                                  ;0114  0200                   ;USYRT STATUS REGISTER EQUATES
                                  ;0115  0200                  USTATR = $A400
                                  ;0116  0200                  RDA    = BIT7
                                  ;0117  0200                  TBMT   = BIT6
                                  ;0118  0200                  RXACT  = BIT5
                                  ;0119  0200                  RSA    = BIT4
                                  ;0120  0200                  TSO    = BIT3
                                  ;0121  0200                  TXACT  = BIT2
                                  ;0122  0200                  TXUERR = BIT1
                                  ;0123  0200                  SYNFLG = BIT0
                                  ;0124  0200
                                  ;0125  0200
                                  ;0126  0200
                                  ;0127  0200                   ;MISCELLANEOUS EQUATES
                                  ;0128  0200                  SYNCH  = @226
                                  ;0129  0200
                                  ;0130  0200
                                  ;0131  0200  A0 00                    LDY     #0
025402   240   000                        .BYTE   240,000
                                  ;0132  0202  84 16                    STY     BSEL6          ;CLEAR BSEL6
025404   204   026                        .BYTE   204,026
                                  ;0133  0204  84 17                    STY     BSEL7          ;CLEAR BSEL7
025406   204   027                        .BYTE   204,027
                                  ;0134  0206                   ;TURN ON THE USYRT, CLOCK
                                  ;0135  0206  A2 60                    LDX     #TXEN!RXEN    ;ASSERT TXEN,RXEN,RTS,DTR
025410   242   140                        .BYTE   242,140
                                  ;0136  0208  8E 00 A0                 STX     OREGB          ;  AND RELEASE INT MODEM RESET
025412   216   000   240                  .BYTE   216,000,240
                                  ;0137  020B  A2 00                    LDX     #0             ;INIT TBMT TIME-OUT COUNTER
025415   242   000                        .BYTE   242,000
                                  ;0138  020D  2C 00 A4                 BIT     USTATR         ;SEE IF TBMT SET
025417   054   000   244                  .BYTE   054,000,244
                                  ;0139  0210  70 08                    BVS     *+10           ;BR IF TBMT SET
025422   160   010                        .BYTE   160,010
                                  ;0140  0212  E8                       INX                    ;INCREMENT TIME-OUT COUNTER
025424   350                              .BYTE   350
                                  ;0141  0213  D0 F8                    BNE     *-6            ;BR IF NO TIME-OUT
025425   320   370                        .BYTE   320,370
                                  ;0142  0215                   ; ***  ERROR 1  ***
                                  ;0143  0215  A0 01                    LDY     #1             ;SET CODE FOR TBMT TIME-OUT ERRO
025427   240   001                        .BYTE   240,001
                                  ;0144  0217  4C 90 03                 JMP     A100           ;GO TAKE ERROR EXIT
025431   114   220   003                  .BYTE   114,220,003
                                  ;0145  021A                   ;LOAD FIRST SYNCH CHAR INTO TRANSMITTER
                                  ;0146  021A  A2 96                    LDX     #SYNCH         ;LOAD FIRST SYNCH CHAR
025434   242   226                        .BYTE   242,226
                                  ;0147  021C  8E 02 A1                 STX     TXDB
025436   216   002   241                  .BYTE   216,002,241
                                  ;0148  021F  A2 00                    LDX     #0             ;INIT TBMT TIME-OUT COUNTER
025441   242   000                        .BYTE   242,000
                                  ;0149  0221  2C 00 A4                 BIT     USTATR         ;SEE IF TBMT SET
025443   054   000   244                  .BYTE   054,000,244
```

```
                                  ;0150  0224  70 08              BVS    *+10       ;BR IF TBMT SET
025446  160  010                         .BYTE   160,010
                                  ;0151  0226  E8                 INX               ;INCREMENT TIME-OUT COUNTER
025450  350                              .BYTE   350
                                  ;0152  0227  D0 F8              BNE    *-6        ;BR IF NO TIME-OUT
025451  320  370                         .BYTE   320,370
                                  ;0153  0229                  ; ***  ERROR 2  ***
                                  ;0154  0229  A0 02              LDY    #2         ;SET CODE FOR TBMT TIME-OUT ERRO
025453  240  002                         .BYTE   240,002
                                  ;0155  022B  4C 90 03           JMP    A100       ;GO TAKE ERROR EXIT
025455  114  220  003                    .BYTE   114,220,003
                                  ;0156  022E                  ;LOAD SECOND SYNCH CHAR INTO TRANSMITTER
                                  ;0157  022E  A2 96              LDX    #SYNCH     ;LOAD SECOND SYNCH CHAR
025460  242  226                         .BYTE   242,226
                                  ;0158  0230  8E 02 A1           STX    TXDB
025462  216  002  241                    .BYTE   216,002,241
                                  ;0159  0233  A2 00              LDX    #0         ;INIT TBMT TIME-OUT COUNTER
025465  242  000                         .BYTE   242,000
                                  ;0160  0235  2C 00 A4           BIT    USTATR     ;SEE IF TBMT SET
025467  054  000  244                    .BYTE   054,000,244
                                  ;0161  0238  70 08              BVS    *+10       ;BR IF TBMT SET
025472  160  010                         .BYTE   160,010
                                  ;0162  023A  E8                 INX               ;INCREMENT TIME-OUT COUNTER
025474  350                              .BYTE   350
                                  ;0163  023B  D0 F8              BNE    *-6        ;BR IF NO TIME-OUT
025475  320  370                         .BYTE   320,370
                                  ;0164  023D                  ; ***  ERROR 3  ***
                                  ;0165  023D  A0 03              LDY    #3         ;SET CODE FOR TBMT TIME-OUT ERRO
025477  240  003                         .BYTE   240,003
                                  ;0166  023F  4C 90 03           JMP    A100       ;GO TAKE ERROR EXIT
025501  114  220  003                    .BYTE   114,220,003
                                  ;0167  0242                  ;CHECK FOR CARRIER SET
                                  ;0168  0242  2C 01 A0           BIT    OREGA      ;SEE IF CARRIER SET YET
025504  054  001  240                    .BYTE   054,001,240
                                  ;0169  0245  70 05              BVS    *+7        ;BR IF CARRIER SET
025507  160  005                         .BYTE   160,005
                                  ;0170  0247                  ; ***  ERROR 4  ***
                                  ;0171  0247  A0 04              LDY    #4         ;SET CODE FOR CARRIER NOT SET ER
025511  240  004                         .BYTE   240,004
                                  ;0172  0249  4C 90 03           JMP    A100       ;GO TAKE ERROR EXIT
025513  114  220  003                    .BYTE   114,220,003
                                  ;0173  024C                  ;LOAD TRANSMITTER WITH 000 CHAR
                                  ;0174  024C  A2 00              LDX    #000       ;CLEAR TSOM
025516  242  000                         .BYTE   242,000
                                  ;0175  024E  8E 03 A1           STX    TDSR
025520  216  003  241                    .BYTE   216,003,241
                                  ;0176  0251  8E 02 A1           STX    TXDB       ;LOAD 000 CHAR
025523  216  002  241                    .BYTE   216,002,241
                                  ;0177  0254  A2 00              LDX    #0         ;INIT TBMT TIME-OUT COUNTER
025526  242  000                         .BYTE   242,000
                                  ;0178  0256  2C 00 A4           BIT    USTATR     ;SEE IF TBMT SET
025530  054  000  244                    .BYTE   054,000,244
                                  ;0179  0259  70 08              BVS    *+10       ;BR IF TBMT SET
025533  160  010                         .BYTE   160,010
                                  ;0180  025B  E8                 INX               ;INCREMENT TIME-OUT COUNTER
025535  350                              .BYTE   350
```

```
                                    ;0181  025C  D0 F8            BNE     *-6             ;BR IF NO TIME-OUT
025536   320   370                         .BYTE   320,370
                                    ;0182  025E                 ; ***  ERROR 5  ***
                                    ;0183  025E  A0 05            LDY     #5              ;SET CODE FOR TBMT TIME-OUT ERRO
025540   240   005                         .BYTE   240,005
                                    ;0184  0260  4C 90 03         JMP     A100            ;GO TAKE ERROR EXIT
025542   114   220   003                   .BYTE   114,220,003
                                    ;0185  0263                 ;LOAD TRANSMITTER WITH 125 CHAR
                                    ;0186  0263  A2 55            LDX     #@125           ;LOAD 125 CHAR
025545   242   125                         .BYTE   242,125
                                    ;0187  0265  8E 02 A1         STX     TXDB
025547   216   002   241                   .BYTE   216,002,241
                                    ;0188  0268  A2 00            LDX     #0              ;INIT TBMT TIME-OUT COUNTER
025552   242   000                         .BYTE   242,000
                                    ;0189  026A  2C 00 A4         BIT     USTATR          ;SEE IF TBMT SET
025554   054   000   244                   .BYTE   054,000,244
                                    ;0190  026D  70 08            BVS     *+10            ;BR IF TBMT SET
025557   160   010                         .BYTE   160,010
                                    ;0191  026F  E8               INX                     ;INCREMENT TIME-OUT COUNTER
025561   350                               .BYTE   350
                                    ;0192  0270  D0 F8            BNE     *-6             ;BR IF NO TIME-OUT
025562   320   370                         .BYTE   320,370
                                    ;0193  0272                 ; ***  ERROR 6  ***
                                    ;0194  0272  A0 06            LDY     #6              ;SET CODE FOR TBMT TIME-OUT ERRO
025564   240   006                         .BYTE   240,006
                                    ;0195  0274  4C 90 03         JMP     A100            ;GO TAKE ERROR EXIT
025566   114   220   003                   .BYTE   114,220,003
                                    ;0196  0277                 ;LOAD TRANSMITTER WITH 252 CHAR
                                    ;0197  0277  A2 AA            LDX     #@252           ;LOAD 252 CHAR
025571   242   252                         .BYTE   242,252
                                    ;0198  0279  8E 02 A1         STX     TXDB
025573   216   002   241                   .BYTE   216,002,241
                                    ;0199  027C  A2 00            LDX     #0              ;INIT TBMT TIME-OUT COUNTER
025576   242   000                         .BYTE   242,000
                                    ;0200  027E  2C 00 A4         BIT     USTATR          ;SEE IF TBMT SET
025600   054   000   244                   .BYTE   054,000,244
                                    ;0201  0281  70 08            BVS     *+10            ;BR IF TBMT SET
025603   160   010                         .BYTE   160,010
                                    ;0202  0283  E8               INX                     ;INCREMENT TIME-OUT COUNTER
025605   350                               .BYTE   350
                                    ;0203  0284  D0 F8            BNE     *-6             ;BR IF NO TIME-OUT
025606   320   370                         .BYTE   320,370
                                    ;0204  0286                 ; ***  ERROR 7  ***
                                    ;0205  0286  A0 07            LDY     #7              ;SET CODE FOR TBMT TIME-OUT ERRO
025610   240   007                         .BYTE   240,007
                                    ;0206  0288  4C 90 03         JMP     A100            ;GO TAKE ERROR EXIT
025612   114   220   003                   .BYTE   114,220,003
                                    ;0207  028B                 ;LOAD TRANSMITTER WITH 377 CHAR AND END OF MESSAGE
                                    ;0208  028B  A2 FF            LDX     #@377           ;LOAD 377 CHAR
025615   242   377                         .BYTE   242,377
                                    ;0209  028D  8E 02 A1         STX     TXDB
025617   216   002   241                   .BYTE   216,002,241
                                    ;0210  0290  A2 00            LDX     #0              ;INIT TBMT TIME-OUT COUNTER
025622   242   000                         .BYTE   242,000
                                    ;0211  0292  2C 00 A4         BIT     USTATR          ;SEE IF TBMT SET
025624   054   000   244                   .BYTE   054,000,244
```

```
                                 ;0212  0295  70 08              BVS    *+10        ;BR IF TBMT SET
 025627   160   010                     .BYTE  160,010
                                 ;0213  0297  E8                 INX                ;INCREMENT TIME-OUT COUNTER
 025631   350                           .BYTE  350
                                 ;0214  0298  D0 F8              BNE    *-6         ;BR IF NO TIME-OUT
 025632   320   370                     .BYTE  320,370
                                 ;0215  029A               ; ***  ERROR 8  ***
                                 ;0216  029A  A0 08              LDY    #8          ;SET CODE FOR TBMT TIME-OUT ERRO
 025634   240   010                     .BYTE  240,010
                                 ;0217  029C  4C 90 03           JMP    A100        ;GO TAKE ERROR EXIT
 025636   114   220   003               .BYTE  114,220,003
                                 ;0218  029F               ;LOAD TRANSMITTER WITH 000 CHAR
                                 ;0219  029F  A2 00              LDX    #000        ;LOAD 000 CHAR
 025641   242   000                     .BYTE  242,000
                                 ;0220  02A1  8E 02 A1           STX    TXDB
 025643   216   002   241               .BYTE  216,002,241
                                 ;0221  02A4  A2 00              LDX    #0          ;INIT RDA TIME-OUT COUNTER
 025646   242   000                     .BYTE  242,000
                                 ;0222  02A6  2C 00 A4           BIT    USTATR      ;SEE IF RDA SET
 025650   054   000   244               .BYTE  054,000,244
                                 ;0223  02A9  30 08              BMI    *+10        ;BR IF RDA SET
 025653   060   010                     .BYTE  060,010
                                 ;0224  02AB  E8                 INX                ;INCREMENT TIME-OUT COUNTER
 025655   350                           .BYTE  350
                                 ;0225  02AC  D0 F8              BNE    *-6         ;BR IF NO TIME-OUT
 025656   320   370                     .BYTE  320,370
                                 ;0226  02AE               ; ***  ERROR 9  ***
                                 ;0227  02AE  A0 09              LDY    #9          ;SET CODE FOR RDA TIME-OUT ERROR
 025660   240   011                     .BYTE  240,011
                                 ;0228  02B0  4C 90 03           JMP    A100        ;GO TAKE ERROR EXIT
 025662   114   220   003               .BYTE  114,220,003
                                 ;0229  02B3               ;READ AND CHECK 000 CHAR
                                 ;0230  02B3  AD 00 A1           LDA    RXDB        ;READ RECEIVER BUFFER
 025665   255   000   241               .BYTE  255,000,241
                                 ;0231  02B6  C9 00              CMP    #@000       ;CHK FOR 000
 025670   311   000                     .BYTE  311,000
                                 ;0232  02B8  F0 05              BEQ    *+7         ;BR IF 000
 025672   360   005                     .BYTE  360,005
                                 ;0233  02BA               ; ***  ERROR 10  ***
                                 ;0234  02BA  A0 0A              LDY    #10         ;SET CODE FOR DATA MISCOMPARE ER
 025674   240   012                     .BYTE  240,012
                                 ;0235  02BC  4C 90 03           JMP    A100        ;GO TAKE ERROR EXIT
 025676   114   220   003               .BYTE  114,220,003
                                 ;0236  02BF  A2 00              LDX    #0          ;INIT RDA TIME-OUT COUNTER
 025701   242   000                     .BYTE  242,000
                                 ;0237  02C1  2C 00 A4           BIT    USTATR      ;SEE IF RDA SET
 025703   054   000   244               .BYTE  054,000,244
                                 ;0238  02C4  30 08              BMI    *+10        ;BR IF RDA SET
 025706   060   010                     .BYTE  060,010
                                 ;0239  02C6  E8                 INX                ;INCREMENT TIME-OUT COUNTER
 025710   350                           .BYTE  350
                                 ;0240  02C7  D0 F8              BNE    *-6         ;BR IF NO TIME-OUT
 025711   320   370                     .BYTE  320,370
                                 ;0241  02C9               ; ***  ERROR 11  ***
                                 ;0242  02C9  A0 0B              LDY    #11         ;SET CODE FOR RDA TIME-OUT ERROR
 025713   240   013                     .BYTE  240,013
```

```
                                ;0243  02CB  4C 90 03             JMP    A100         ;GO TAKE ERROR EXIT
025715   114   220   003               .BYTE  114,220,003
                                ;0244  02CE  A2 02                LDX    #TEOM        ;SET TEOM TO TERMINATE MSG
025720   242   002                     .BYTE  242,002
                                ;0245  02D0  8E 03 A1             STX    TDSR
025722   216   003   241               .BYTE  216,003,241
                                ;0246  02D3                   ;READ AND CHECK 125 CHAR
                                ;0247  02D3  AD 00 A1             LDA    RXDB         ;READ RECEIVER BUFFER
025725   255   000   241               .BYTE  255,000,241
                                ;0248  02D6  C9 55                CMP    #@125        ;CHK FOR 125
025730   311   125                     .BYTE  311,125
                                ;0249  02D8  F0 05                BEQ    *+7          ;BR IF 125
025732   360   005                     .BYTE  360,005
                                ;0250  02DA                   ; ***  ERROR 12  ***
                                ;0251  02DA  A0 0C                LDY    #12          ;SET CODE FOR DATA MISCOMPARE ER
025734   240   014                     .BYTE  240,014
                                ;0252  02DC  4C 90 03             JMP    A100         ;GO TAKE ERROR EXIT
025736   114   220   003               .BYTE  114,220,003
                                ;0253  02DF  A2 00                LDX    #0           ;INIT RDA TIME-OUT COUNTER
025741   242   000                     .BYTE  242,000
                                ;0254  02E1  2C 00 A4             BIT    USTATR       ;SEE IF RDA SET
025743   054   000   244               .BYTE  054,000,244
                                ;0255  02E4  30 08                BMI    *+10         ;BR IF RDA SET
025746   060   010                     .BYTE  060,010
                                ;0256  02E6  E8                   INX                 ;INCREMENT TIME-OUT COUNTER
025750   350                           .BYTE  350
                                ;0257  02E7  D0 F8                BNE    *-6          ;BR IF NO TIME-OUT
025751   320   370                     .BYTE  320,370
                                ;0258  02E9                   ; ***  ERROR 13  ***
                                ;0259  02E9  A0 0D                LDY    #13          ;SET CODE FOR RDA TIME-OUT ERROR
025753   240   015                     .BYTE  240,015
                                ;0260  02EB  4C 90 03             JMP    A100         ;GO TAKE ERROR EXIT
025755   114   220   003               .BYTE  114,220,003
                                ;0261  02EE                   ;READ AND CHECK 252 CHAR
                                ;0262  02EE  AD 00 A1             LDA    RXDB         ;READ RECEIVER BUFFER
025760   255   000   241               .BYTE  255,000,241
                                ;0263  02F1  C9 AA                CMP    #@252        ;CHK FOR 252
025763   311   252                     .BYTE  311,252
                                ;0264  02F3  F0 05                BEQ    *+7          ;BR IF 252
025765   360   005                     .BYTE  360,005
                                ;0265  02F5                   ; ***  ERROR 14  ***
                                ;0266  02F5  A0 0E                LDY    #14          ;SET CODE FOR DATA MISCOMPARE ER
025767   240   016                     .BYTE  240,016
                                ;0267  02F7  4C 90 03             JMP    A100         ;GO TAKE ERROR EXIT
025771   114   220   003               .BYTE  114,220,003
                                ;0268  02FA  A2 00                LDX    #0           ;INIT RDA TIME-OUT COUNTER
025774   242   000                     .BYTE  242,000
                                ;0269  02FC  2C 00 A4             BIT    USTATR       ;SEE IF RDA SET
025776   054   000   244               .BYTE  054,000,244
                                ;0270  02FF  30 08                BMI    *+10         ;BR IF RDA SET
026001   060   010                     .BYTE  060,010
                                ;0271  0301  E8                   INX                 ;INCREMENT TIME-OUT COUNTER
026003   350                           .BYTE  350
                                ;0272  0302  D0 F8                BNE    *-6          ;BR IF NO TIME-OUT
026004   320   370                     .BYTE  320,370
                                ;0273  0304                   ; ***  ERROR 15  ***
```

```
                          ;0274  0304  A0 0F          LDY    #15        ;SET CODE FOR RDA TIME-OUT ERROR
026006  240  017                 .BYTE  240,017
                          ;0275  0306  4C 90 03       JMP    A100       ;GO TAKE ERROR EXIT
026010  114  220  003            .BYTE  114,220,003
                          ;0276  0309              ;READ AND CHECK 377 CHAR
                          ;0277  0309  AD 00 A1       LDA    RXDB       ;READ RECEIVER BUFFER
026013  255  000  241            .BYTE  255,000,241
                          ;0278  030C  C9 FF          CMP    #@377      ;CHK FOR 377
026016  311  377                 .BYTE  311,377
                          ;0279  030E  F0 05          BEQ    *+7        ;BR IF 377
026020  360  005                 .BYTE  360,005
                          ;0280  0310              ; ***  ERROR 16  ***
                          ;0281  0310  A0 10          LDY    #16        ;SET CODE FOR DATA MISCOMPARE ER
026022  240  020                 .BYTE  240,020
                          ;0282  0312  4C 90 03       JMP    A100       ;GO TAKE ERROR EXIT
026024  114  220  003            .BYTE  114,220,003
                          ;0283  0315  A2 00          LDX    #0         ;INIT RDA TIME-OUT COUNTER
026027  242  000                 .BYTE  242,000
                          ;0284  0317  2C 00 A4       BIT    USTATR     ;SEE IF RDA SET
026031  054  000  244            .BYTE  054,000,244
                          ;0285  031A  30 08          BMI    *+10       ;BR IF RDA SET
026034  060  010                 .BYTE  060,010
                          ;0286  031C  E8             INX               ;INCREMENT TIME-OUT COUNTER
026036  350                      .BYTE  350
                          ;0287  031D  D0 F8          BNE    *-6        ;BR IF NO TIME-OUT
026037  320  370                 .BYTE  320,370
                          ;0288  031F              ; ***  ERROR 17  ***
                          ;0289  031F  A0 11          LDY    #17        ;SET CODE FOR RDA TIME-OUT ERROR
026041  240  021                 .BYTE  240,021
                          ;0290  0321  4C 90 03       JMP    A100       ;GO TAKE ERROR EXIT
026043  114  220  003            .BYTE  114,220,003
                          ;0291  0324              ;READ AND CHECK 000 CHAR
                          ;0292  0324  AD 00 A1       LDA    RXDB       ;READ RECEIVER BUFFER
026046  255  000  241            .BYTE  255,000,241
                          ;0293  0327  C9 00          CMP    #@000      ;CHK FOR 000
026051  311  000                 .BYTE  311,000
                          ;0294  0329  F0 05          BEQ    *+7        ;BR IF 000
026053  360  005                 .BYTE  360,005
                          ;0295  032B              ; ***  ERROR 18  ***
                          ;0296  032B  A0 12          LDY    #18        ;SET CODE FOR DATA MISCOMPARE ER
026055  240  022                 .BYTE  240,022
                          ;0297  032D  4C 90 03       JMP    A100       ;GO TAKE ERROR EXIT
026057  114  220  003            .BYTE  114,220,003
                          ;0298  0330  AE 01 A1       LDX    RDSR       ;CHECK FOR RERR BIT SET
026062  256  001  241            .BYTE  256,001,241
                          ;0299  0333  30 05          BMI    *+7        ;BR IF RERR BIT SET (NO CRC ERRO
026065  060  005                 .BYTE  060,005
                          ;0300  0335              ; ***  ERROR 19  ***
                          ;0301  0335  A0 13          LDY    #19
026067  240  023                 .BYTE  240,023
                          ;0302  0337  4C 90 03       JMP    A100       ;GO TAKE ERROR EXIT
026071  114  220  003            .BYTE  114,220,003
                          ;0303  033A  A2 00          LDX    #0         ;INIT RDA TIME-OUT COUNTER
026074  242  000                 .BYTE  242,000
                          ;0304  033C  2C 00 A4       BIT    USTATR     ;SEE IF RDA SET
026076  054  000  244            .BYTE  054,000,244
```

```
                                 ;0305  033F  30 08            BMI    *+10        ;BR IF RDA SET
026101  060  010                        .BYTE  060,010
                                 ;0306  0341  E8               INX                ;INCREMENT TIME-OUT COUNTER
026103  350                             .BYTE  350
                                 ;0307  0342  D0 F8            BNE    *-6         ;BR IF NO TIME-OUT
026104  320  370                        .BYTE  320,370
                                 ;0308  0344            ; ***  ERROR 20  ***
                                 ;0309  0344  A0 14            LDY    #20         ;SET CODE FOR RDA TIME-OUT ERROR
026106  240  024                        .BYTE  240,024
                                 ;0310  0346  4C 90 03         JMP    A100        ;GO TAKE ERROR EXIT
026110  114  220  003                   .BYTE  114,220,003
                                 ;0311  0349            ;READ AND CHECK FIRST CRC CHAR (160)
                                 ;0312  0349  AD 00 A1         LDA    RXDB        ;READ RECEIVER BUFFER
026113  255  000  241                   .BYTE  255,000,241
                                 ;0313  034C  C9 70            CMP    #@160       ;CHK FOR 160
026116  311  160                        .BYTE  311,160
                                 ;0314  034E  F0 05            BEQ    *+7         ;BR IF 160
026120  360  005                        .BYTE  360,005
                                 ;0315  0350            ; ***  ERROR 21  ***
                                 ;0316  0350  A0 15            LDY    #21         ;SET CODE FOR DATA MISCOMPARE ER
026122  240  025                        .BYTE  240,025
                                 ;0317  0352  4C 90 03         JMP    A100        ;GO TAKE ERROR EXIT
026124  114  220  003                   .BYTE  114,220,003
                                 ;0318  0355  A2 00            LDX    #0          ;INIT RDA TIME-OUT COUNTER
026127  242  000                        .BYTE  242,000
                                 ;0319  0357  2C 00 A4         BIT    USTATR      ;SEE IF RDA SET
026131  054  000  244                   .BYTE  054,000,244
                                 ;0320  035A  30 08            BMI    *+10        ;BR IF RDA SET
026134  060  010                        .BYTE  060,010
                                 ;0321  035C  E8               INX                ;INCREMENT TIME-OUT COUNTER
026136  350                             .BYTE  350
                                 ;0322  035D  D0 F8            BNE    *-6         ;BR IF NO TIME-OUT
026137  320  370                        .BYTE  320,370
                                 ;0323  035F            ; ***  ERROR 22  ***
                                 ;0324  035F  A0 16            LDY    #22         ;SET CODE FOR RDA TIME-OUT ERROR
026141  240  026                        .BYTE  240,026
                                 ;0325  0361  4C 90 03         JMP    A100        ;GO TAKE ERROR EXIT
026143  114  220  003                   .BYTE  114,220,003
                                 ;0326  0364            ;READ AND CHECK 2ND CRC CHAR (034)
                                 ;0327  0364  AD 00 A1         LDA    RXDB        ;READ RECEIVER BUFFER
026146  255  000  241                   .BYTE  255,000,241
                                 ;0328  0367  C9 1C            CMP    #@034       ;CHK FOR 034
026151  311  034                        .BYTE  311,034
                                 ;0329  0369  F0 05            BEQ    *+7         ;BR IF 034
026153  360  005                        .BYTE  360,005
                                 ;0330  036B            ; ***  ERROR 23  ***
                                 ;0331  036B  A0 17            LDY    #23         ;SET CODE FOR DATA MISCOMPARE ER
026155  240  027                        .BYTE  240,027
                                 ;0332  036D  4C 90 03         JMP    A100        ;GO TAKE ERROR EXIT
026157  114  220  003                   .BYTE  114,220,003
                                 ;0333  0370  A2 00            LDX    #0          ;INIT RDA TIME-OUT COUNTER
026162  242  000                        .BYTE  242,000
                                 ;0334  0372  2C 00 A4         BIT    USTATR      ;SEE IF RDA SET
026164  054  000  244                   .BYTE  054,000,244
                                 ;0335  0375  30 08            BMI    *+10        ;BR IF RDA SET
026167  060  010                        .BYTE  060,010
```

```
                                 ;0336  0377  E8               INX                       ;INCREMENT TIME-OUT COUNTER
026171   350                            .BYTE   350
                                 ;0337  0378  D0 F8            BNE      *-6              ;BR IF NO TIME-OUT
026172   320    370                     .BYTE   320,370
                                 ;0338  037A              ; ***  ERROR 24  ***
                                 ;0339  037A  A0 18            LDY      #24              ;SET CODE FOR RDA TIME-OUT ERROR
026174   240    030                     .BYTE   240,030
                                 ;0340  037C  4C 90 03         JMP      A100             ;GO TAKE ERROR EXIT
026176   114    220    003              .BYTE   114,220,003
                                 ;0341  037F              ;DROP RTS, CHECK FOR CARRIER TO DROP
                                 ;0342  037F  A2 68            LDX      #TXEN!RXEN!RTSND  ;DE-ASSERT RTS
026201   242    150                     .BYTE   242,150
                                 ;0343  0381  8E 00 A0         STX      OREGB
026203   216    000    240              .BYTE   216,000,240
                                 ;0344  0384  A2 00            LDX      #0               ;INIT CARRIER DROP TIME-OUT COUN
026206   242    000                     .BYTE   242,000
                                 ;0345  0386  2C 01 A0         BIT      OREGA            ;SEE IF CARRIER CLEARED
026210   054    001    240              .BYTE   054,001,240
                                 ;0346  0389  50 05            BVC      *+7              ;BR IF CARRIER CLEARED
026213   120    005                     .BYTE   120,005
                                 ;0347  038B  E8               INX                       ;INCREMENT TIME-OUT COUNTER
026215   350                            .BYTE   350
                                 ;0348  038C  D0 F8            BNE      *-6              ;BR IF NO TIME-OUT
026216   320    370                     .BYTE   320,370
                                 ;0349  038E              ; ***  ERROR 25  ***
                                 ;0350  038E  A0 19            LDY      #25              ;SET CODE FOR CARRIER DROP TIME-
026220   240    031                     .BYTE   240,031
                                 ;0351  0390              ;COME HERE FOR EXIT
                                 ;0352  0390  84 16     A100   STY      BSEL6            ;PUT ERROR NO. (IF ANY) INTO BSE
026222   204    026                     .BYTE   204,026
                                 ;0353  0392  85 17            STA      BSEL7            ;PUT BAD DATA (IF ANY) INTO BSEL
026224   205    027                     .BYTE   205,027
                                 ;0354  0394  60               RTS                       ;RETURN CONTROL TO LSI-11 PROGRA
026226   140                            .BYTE   140
                                 ;0355  0395
                                 ;0356  0395
                                 ;
                                 ;
                                 ;ERRORS = 0000
                                 ;
026227                           ENDCOD:
        026230                   .EVEN
```

```
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813  026230
5814  026230  012737  000003  002526
5815  026236  004737  005376
5816  026242  042737  001000  026320
5817  026250  012737  000011  027072
5818  026256  023727  002470  000000
5819  026264  001412
5820  026266  023727  002472  000004
5821  026274  001406
5822  026276  052737  001000  026320
5823  026304  052737  100000  027072
5824
5825  026312  004537  007234
5826  026316  065626
5827  026320  000000
5828  026322  103003
5829  026324
5830  026324  104460
5831  026326
5832  026326  104410
5833  026330  000554
5834
5835  026332  004537  010136
5836  026336  000226
5837  026340  000007
5838  026342  103003
5839  026344
5840  026344  104460
5841  026346
5842  026346  104410
5843  026350  000534
5844
```

```
.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
                                                                T3::
        MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
        JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
        BIC     #NOLOOP,1$      ;
        MOV     #9.,3$          ;*INIT ENTRAN COUNT/TTLOOP STATUS
        CMP     BRDTYP,#0       ;IS THIS AN M8064?
        BEQ     2$              ; YES: USE TTLOOP (NOT XLB).
        CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
        BEQ     2$              ;BR IF NO
        BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
        BIS     #BIT15,3$       ;      AND SET MSB OF ENTRAN STATUS (NOLOOP)
;-------------------------------------------------------------------------
2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
        DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
1$:     0                       ;USE 8 BIT CHARS
        BCC     .+8.            ;BR IF NO ERROR
        ERROR                   ;REPORT STACKED ERROR
                                                        TRAP    C$ERROR
        ESCAPE  TST             ;SKIP TO END OF TEST
                                                        TRAP    C$ESCAPE
                                                        .WORD   L10022-.

        JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
        SYNCH
        7.
        BCC     .+8.            ;BR IF NO ERROR
        ERROR                   ;REPORT STACKED ERROR
                                                        TRAP    C$ERROR
        ESCAPE  TST             ;SKIP TO END OF TEST
                                                        TRAP    C$ESCAPE
                                                        .WORD   L10022-.
```

```
5845  026352  004537  010136          JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
5846  026356  000226                  SYNCH
5847  026360  000010                  8.
5848  026362  103003                  BCC     .+8.            ;BR IF NO ERROR
5849  026364                          ERROR                   ;REPORT STACKED ERROR
5850  026364  104460                                                  TRAP    C$ERROR
5851  026366                          ESCAPE  TST             ;SKIP TO END OF TEST
5852  026366  104410                                                  TRAP    C$ESCAPE
5853  026370  000514                                                  .WORD   L10022-.
5854
5855  026372  004537  010250          JSR     R5,TXCTRL       ;CLEAR TSOM
5856  026376  000000                  000
5857  026400  000000                  0
5858
5859  026402  004537  010136          JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
5860  026406  000125                  125
5861  026410  000010                  8.
5862  026412  103003                  BCC     .+8.            ;BR IF NO ERROR
5863  026414                          ERROR                   ;REPORT STACKED ERROR
5864  026414  104460                                                  TRAP    C$ERROR
5865  026416                          ESCAPE  TST             ;SKIP TO END OF TEST
5866  026416  104410                                                  TRAP    C$ESCAPE
5867  026420  000464                                                  .WORD   L10022-.
5868
5869  026422  004537  010136          JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
5870  026426  000252                  252
5871  026430  000010                  8.
5872  026432  103003                  BCC     .+8.            ;BR IF NO ERROR
5873  026434                          ERROR                   ;REPORT STACKED ERROR
5874  026434  104460                                                  TRAP    C$ERROR
5875  026436                          ESCAPE  TST             ;SKIP TO END OF TEST
5876  026436  104410                                                  TRAP    C$ESCAPE
5877  026440  000444                                                  .WORD   L10022-.
5878
5879  026442  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 252(DATA2)
5880  026446  000000                  000
5881  026450  000010                  8.
5882  026452  103003                  BCC     .+8.            ;BR IF NO ERROR
5883  026454                          ERROR                   ;REPORT STACKED ERROR
5884  026454  104460                                                  TRAP    C$ERROR
5885  026456                          ESCAPE  TST             ;SKIP TO END OF TEST
5886  026456  104410                                                  TRAP    C$ESCAPE
5887  026460  000424                                                  .WORD   L10022-.
5888
5889  026462  004537  010136          JSR     R5,TXCHAR       ;LOAD 377(DATA4), TX 000(DATA3)
5890  026466  000377                  377
5891  026470  000010                  8.
5892  026472  103003                  BCC     .+8.            ;BR IF NO ERROR
5893  026474                          ERROR                   ;REPORT STACKED ERROR
5894  026474  104460                                                  TRAP    C$ERROR
5895  026476                          ESCAPE  TST             ;SKIP TO END OF TEST
5896  026476  104410                                                  TRAP    C$ESCAPE
5897  026500  000404                                                  .WORD   L10022-.
5898
5899  026502  004537  010136          JSR     R5,TXCHAR       ;LOAD 001(DATA5)
5900  026506  000001                  001
```

```
5901  026510  000000                    0
5902  026512  103003                    BCC     .+8.            ;BR IF NO ERROR
5903  026514                            ERROR                   ;REPORT STACKED ERROR
5904  026514  104460                                                    TRAP    C$ERROR
5905  026516                            ESCAPE  TST             ;SKIP TO END OF TEST
5906  026516  104410                                                    TRAP    C$ESCAPE
5907  026520  000364                                                    .WORD   L10022-.
5908
5909  026522  004537  011624            JSR     R5,RCV1ST       ;CLOCK AND RCV 125
5910  026526  000000                    0
5911  026530  103003                    BCC     .+8.            ;BR IF NO ERROR
5912  026532                            ERROR                   ;REPORT STACKED ERROR
5913  026532  104460                                                    TRAP    C$ERROR
5914  026534                            ESCAPE  TST             ;SKIP TO END OF TEST
5915  026534  104410                                                    TRAP    C$ESCAPE
5916  026536  000346                                                    .WORD   L10022-.
5917
5918  026540  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
5919  026544  000125                    125
5920  026546  000000                    0
5921  026550  000010                    8.
5922  026552  103003                    BCC     .+8.            ;BR IF NO ERROR
5923  026554                            ERROR                   ;REPORT STACKED ERROR
5924  026554  104460                                                    TRAP    C$ERROR
5925  026556                            ESCAPE  TST             ;SKIP TO END OF TEST
5926  026556  104410                                                    TRAP    C$ESCAPE
5927  026560  000324                                                    .WORD   L10022-.
5928
5929                            ;-------------------------------------------------------------
5930                            ; TRANSMIT THE BULK OF DATA OUT OF TABLE 'PATX'
5931  026562  012702  002646    ;-------------------------------------------------------------
5932  026566  112237  026624            MOV     #PATX+1,R2      ;SET UP TABLE POINTER
5933  026572  116237  000003  026604 5$: MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
5934                                    MOVB    3(R2),10$       ;SET UP TRANSMIT CHARACTER
5935  026600  004537  010136
5936  026604  000000            10$:    JSR     R5,TXCHAR       ;LOAD A CHARACTER
5937  026606  000000                    000                     ;** HOLE FOR NEXT TX CHARACTER
5938  026610  103003                    0
5939  026612                            BCC     .+8.            ;BR IF NO ERROR
5940  026612  104460                    ERROR                   ;REPORT STACKED ERROR
5941  026614                                                            TRAP    C$ERROR
5942  026614  104410                    ESCAPE  TST             ;SKIP TO END OF TEST
5943  026616  000266                                                    TRAP    C$ESCAPE
5944                                                                    .WORD   L10022-.
5945  026620  004537  010350
5946  026624  000000            20$:    JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
5947  026626  000000                    000                     ;** HOLE FOR EXPECTED CHARACTER
5948  026630  000010                    0
5949  026632  103003                    8.
5950  026634                            BCC     .+8.            ;BR IF NO ERROR
5951  026634  104460                    ERROR                   ;REPORT STACKED ERROR
5952  026636                                                            TRAP    C$ERROR
5953  026636  104410                    ESCAPE  TST             ;SKIP TO END OF TEST
5954  026640  000244                                                    TRAP    C$ESCAPE
5955                                                                    .WORD   L10022-.
5956  026642  022702  002665            CMP     #PATX+16.,R2    ;CHECK FOR 20TH CHARACTER OF TABLE
```

```
026646  001347                  BNE     5$              ;BR IF NOT DONE

                        ;------------------------------------------------------------------
026650  004537  010250          JSR     R5,TXCTRL       ;LOAD 1ST TEOM
026654  000002                  TEOM
026656  000000                  0
026660  004537  010350          JSR     R5,RXCHAR       ;READ/CHK 357(DATA17), RCV 337(DATA18)
026664  000357                  357
026666  000000                  0
026670  000010                  8.
026672  103003                  BCC     .+8.            ;BR IF NO ERROR
026674                          ERROR                   ;REPORT STACKED ERROR
026674  104460                                                                  TRAP    C$ERROR
026676                          ESCAPE  TST             ;SKIP TO END OF TEST
026676  104410                                                                  TRAP    C$ESCAPE
026700  000204                                                                  .WORD   L10022-.

026702  004537  010250          JSR     R5,TXCTRL       ;LOAD 2ND TEOM
026706  000002                  TEOM
026710  000000                  0
026712  004537  010350          JSR     R5,RXCHAR       ;READ/CHK 337(DATA18), RCV 277(DATA19)
026716  000337                  337
026720  000000                  0
026722  000010                  8.
026724  103003                  BCC     .+8.            ;BR IF NO ERROR
026726                          ERROR                   ;REPORT STACKED ERROR
026726  104460                                                                  TRAP    C$ERROR
026730                          ESCAPE  TST             ;SKIP TO END OF TEST
026730  104410                                                                  TRAP    C$ESCAPE
026732  000152                                                                  .WORD   L10022-.

026734  004537  010350          JSR     R5,RXCHAR       ;READ/CHK 277(DATA19), RCV 177(DATA20)
026740  000277                  277
026742  000000                  0
026744  000010                  8.
026746  103003                  BCC     .+8.            ;BR IF NO ERROR
026750                          ERROR                   ;REPORT STACKED ERROR
026750  104460                                                                  TRAP    C$ERROR
026752                          ESCAPE  TST             ;SKIP TO END OF TEST
026752  104410                                                                  TRAP    C$ESCAPE
026754  000130                                                                  .WORD   L10022-.

026756  004537  010350          JSR     R5,RXCHAR       ;READ/CHK 177(DATA20), RCV FIRST CRC BYTE
026762  100177                  RXERR!177
026764  000001                  RERCHK
026766  000010                  8.
026770  103003                  BCC     .+8.            ;BR IF NO ERROR
026772                          ERROR                   ;REPORT STACKED ERROR
026772  104460                                                                  TRAP    C$ERROR
026774                          ESCAPE  TST             ;SKIP TO END OF TEST
026774  104410                                                                  TRAP    C$ESCAPE
026776  000106                                                                  .WORD   L10022-.

027000  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
027004  000156                  156
027006  000000                  0
027010  000010                  8.
```

```
6013  027012  103003  010350          BCC     .+8.            ;BR IF NO ERROR
6014  027014                          ERROR                   ;REPORT STACKED ERROR
6015  027014  104460                                                  TRAP    C$ERROR
6016  027016                          ESCAPE  TST             ;SKIP TO END OF TEST
6017  027016  104410                                                  TRAP    C$ESCAPE
6018  027020  000064                                                  .WORD   L10022-.
6019
6020  027022  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
6021  027026  000236                  236
6022  027030  000000                  0
6023  027032  000010                  8.
6024  027034  103003                  BCC     .+8.            ;BR IF NO ERROR
6025  027036                          ERROR                   ;REPORT STACKED ERROR
6026  027036  104460                                                  TRAP    C$ERROR
6027  027040                          ESCAPE  TST             ;SKIP TO END OF TEST
6028  027040  104410                                                  TRAP    C$ESCAPE
6029  027042  000042                                                  .WORD   L10022-.
6030                                  ;-----------------------------------------------------------------
6031  027044  005337  002526          DEC     REG0            ;DECREMENT COUNT
6032  027050  001406                  BEQ     40$             ;BR IF TRIPLE LOOP IS COMPLETED
6033
6034  027052  004537  010250          JSR     R5,TXCTRL       ;CLEAR TEOM, SET TSOM
6035  027056  000001                  TSOM
6036  027060  000001                  1
6037  027062  000137  026312          JMP     2$              ;AND RUN TX/RX AGAIN
6038
6039  027066  004537  011772  40$:    JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
6040  027072  000011          3$:     9.
6041  027074  103003                  BCC     .+8.            ;BR IF NO ERROR
6042  027076                          ERROR                   ;REPORT STACKED ERROR
6043  027076  104460                                                  TRAP    C$ERROR
6044  027100                          ESCAPE  TST             ;SKIP TO END OF TEST
6045  027100  104410                                                  TRAP    C$ESCAPE
6046  027102  000002                                                  .WORD   L10022-.
6047  027104                  ENDTST
6048  027104                                                  L10022:
6049  027104  104401                                                  TRAP    C$ETST
```

```
                                .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
027106                                                                          T4::
027106  004737  005376          ;       JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
027112  042737  001000  027170  ;       BIC     #NOLOOP,1$      ;
027120  012737  000011  027634  ;       MOV     #9.,3$          ;INIT ENTRAN COUNT/TTLOOP STATUS
027126  023727  002470  000000  ;       CMP     BRDTYP,#0       ;IS THIS AN M8064?
027134  001412                  ;       BEQ     2$              ; YES: USE TTLOOP (NOT XLB).
027136  023727  002472  000004  ;       CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
027144  001406                  ;       BEQ     2$              ;BR IF NO
027146  052737  001000  027170  ;       BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
027154  052737  100000  027634  ;       BIS     #BIT15,3$       ;       AND SPECIFY NOLOOP IN ENTRAN
                                ;-----------------------------------------------------------------------
027162  004537  007234          2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
027166  062226                  ;       DDCMP!STRIPS!OVRC!SYNCH ;SET DDCMP,STRIP SYNC,ODD VRC,SYNCH=226
027170  000347                  1$:     TXDL!RXDL               ;USE 7 BIT CHARS FOR RX & TX
027172  103003                  ;       BCC     .+8.            ;BR IF NO ERROR
027174                          ;       ERROR                   ;REPORT STACKED ERROR
027174  104460                                                                  TRAP    C$ERROR
027176                          ;       ESCAPE  TST             ;SKIP TO END OF TEST
027176  104410                                                                  TRAP    C$ESCAPE
027200  000446                                                                  .WORD   L10023-.

027202  004537  010136          ;       JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
027206  000226                  ;       SYNCH
027210  000007                  ;       7.
027212  103003                  ;       BCC     .+8.            ;BR IF NO ERROR
027214                          ;       ERROR                   ;REPORT STACKED ERROR
027214  104460                                                                  TRAP    C$ERROR
027216                          ;       ESCAPE  TST             ;SKIP TO END OF TEST
027216  104410                                                                  TRAP    C$ESCAPE
027220  000426                                                                  .WORD   L10023-.
```

```
6106  027222  004537  010136          JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
6107  027226  000226                  SYNCH
6108  027230  000010                  8.
6109  027232  103003                  BCC     .+8.            ;BR IF NO ERROR
6110  027234                          ERROR                   ;REPORT STACKED ERROR
6111  027234  104460                                                          TRAP    C$ERROR
6112  027236                          ESCAPE  TST             ;SKIP TO END OF TEST
6113  027236  104410                                                          TRAP    C$ESCAPE
6114  027240  000406                                                          .WORD   L10023-.
6115
6116  027242  004537  010250          JSR     R5,TXCTRL       ;CLEAR TSOM
6117  027246  000000                  000
6118  027250  000000                  0
6119
6120  027252  004537  010136          JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
6121  027256  000125                  125
6122  027260  000010                  8.
6123  027262  103003                  BCC     .+8.            ;BR IF NO ERROR
6124  027264                          ERROR                   ;REPORT STACKED ERROR
6125  027264  104460                                                          TRAP    C$ERROR
6126  027266                          ESCAPE  TST             ;SKIP TO END OF TEST
6127  027266  104410                                                          TRAP    C$ESCAPE
6128  027270  000356                                                          .WORD   L10023-.
6129
6130  027272  004537  010136          JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
6131  027276  000252                  252
6132  027300  000010                  8.
6133  027302  103003                  BCC     .+8.            ;BR IF NO ERROR
6134  027304                          ERROR                   ;REPORT STACKED ERROR
6135  027304  104460                                                          TRAP    C$ERROR
6136  027306                          ESCAPE  TST             ;SKIP TO END OF TEST
6137  027306  104410                                                          TRAP    C$ESCAPE
6138  027310  000336                                                          .WORD   L10023-.
6139
6140  027312  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(DATA3)
6141  027316  000000                  000
6142  027320  000000                  0
6143  027322  103003                  BCC     .+8.            ;BR IF NO ERROR
6144  027324                          ERROR                   ;REPORT STACKED ERROR
6145  027324  104460                                                          TRAP    C$ERROR
6146  027326                          ESCAPE  TST             ;SKIP TO END OF TEST
6147  027326  104410                                                          TRAP    C$ESCAPE
6148  027330  000316                                                          .WORD   L10023-.
6149
6150  027332  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV 125(DATA1)
6151  027336  000000                  0
6152  027340  103003                  BCC     .+8.            ;BR IF NO ERROR
6153  027342                          ERROR                   ;REPORT STACKED ERROR
6154  027342  104460                                                          TRAP    C$ERROR
6155  027344                          ESCAPE  TST             ;SKIP TO END OF TEST
6156  027344  104410                                                          TRAP    C$ESCAPE
6157  027346  000300                                                          .WORD   L10023-.
6158
6159  027350  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
6160  027354  000125                  125
6161  027356  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
```

```
6162  027360  000010                        8.
6163  027362  103003                        BCC     .+8.            ;BR IF NO ERROR
6164  027364                                ERROR                   ;REPORT STACKED ERROR
6165  027364  104460                                                                TRAP    C$ERROR
6166  027366                                ESCAPE  TST             ;SKIP TO END OF TEST
6167  027366  104410                                                                TRAP    C$ESCAPE
6168  027370  000256                                                                .WORD   L10023-.
6169                                  ;-------------------------------------------------------------
6170                                  ; TRANSMIT THE BULK OF DATA OUT OF TABLE 'PATX'
6171                                  ;-------------------------------------------------------------
6172  027372  012702  002646                MOV     #PATX+1,R2      ;SET UP TABLE POINTER
6173  027376  112237  027434        5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
6174  027402  116237  000001  027414        MOVB    1(R2),10$       ;SET UP TRANSMIT CHARACTER
6175
6176  027410  004537  010136                JSR     R5,TXCHAR       ;LOAD A CHARACTER
6177  027414  000000                10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
6178  027416  000000                        0
6179  027420  103003                        BCC     .+8.            ;BR IF NO ERROR
6180  027422                                ERROR                   ;REPORT STACKED ERROR
6181  027422  104460                                                                TRAP    C$ERROR
6182  027424                                ESCAPE  TST             ;SKIP TO END OF TEST
6183  027424  104410                                                                TRAP    C$ESCAPE
6184  027426  000220                                                                .WORD   L10023-.
6185
6186  027430  004537  010350                JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
6187  027434  000000                20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
6188  027436  000001                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6189  027440  000010                        8.
6190  027442  103003                        BCC     .+8.            ;BR IF NO ERROR
6191  027444                                ERROR                   ;REPORT STACKED ERROR
6192  027444  104460                                                                TRAP    C$ERROR
6193  027446                                ESCAPE  TST             ;SKIP TO END OF TEST
6194  027446  104410                                                                TRAP    C$ESCAPE
6195  027450  000176                                                                .WORD   L10023-.
6196
6197  027452  022702  002737                CMP     #EPATX-1,R2
6198  027456  001347                        BNE     5$
6199                                  ;-------------------------------------------------------------
6200  027460  004537  010250                JSR     R5,TXCTRL       ;LOAD TSOM
6201  027464  000001                        TSOM
6202  027466  000000                        0
6203  027470  103003                        BCC     .+8.            ;BR IF NO ERROR
6204  027472                                ERROR                   ;REPORT STACKED ERROR
6205  027472  104460                                                                TRAP    C$ERROR
6206  027474                                ESCAPE  TST             ;SKIP TO END OF TEST
6207  027474  104410                                                                TRAP    C$ESCAPE
6208  027476  000150                                                                .WORD   L10023-.
6209
6210  027500  004537  010350                JSR     R5,RXCHAR       ;READ & CHK 277, RCV 177
6211  027504  000277                        277
6212  027506  000001                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6213  027510  000010                        8.
6214  027512  103003                        BCC     .+8.            ;BR IF NO ERROR
6215  027514                                ERROR                   ;REPORT STACKED ERROR
6216  027514  104460                                                                TRAP    C$ERROR
6217  027516                                ESCAPE  TST             ;SKIP TO END OF TEST
```

```
6218  027516  104410                                                          TRAP    C$ESCAPE
6219  027520  000126                                                          .WORD   L10023-.
6220
6221  027522  004537  010250          JSR     R5,TXCTRL       ;LOAD 2ND TSOM
6222  027526  000001                  TSOM
6223  027530  000000                  0
6224  027532  103003                  BCC     .+8.            ;BR IF NO ERROR
6225  027534                          ERROR                   ;REPORT STACKED ERROR
6226  027534  104460                                                          TRAP    C$ERROR
6227  027536                          ESCAPE  TST             ;SKIP TO END OF TEST
6228  027536  104410                                                          TRAP    C$ESCAPE
6229  027540  000106                                                          .WORD   L10023-.
6230
6231  027542  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 177, RCV FIRST SYNC
6232  027546  000177                  177
6233  027550  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6234  027552  000010                  8.
6235  027554  103003                  BCC     .+8.            ;BR IF NO ERROR
6236  027556                          ERROR                   ;REPORT STACKED ERROR
6237  027556  104460                                                          TRAP    C$ERROR
6238  027560                          ESCAPE  TST             ;SKIP TO END OF TEST
6239  027560  104410                                                          TRAP    C$ESCAPE
6240  027562  000064                                                          .WORD   L10023-.
6241
6242  027564  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 1ST SYNC, RCV SECOND SYNC
6243  027570  000226                  SYNCH
6244  027572  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6245  027574  000010                  8.
6246  027576  103003                  BCC     .+8.            ;BR IF NO ERROR
6247  027600                          ERROR                   ;REPORT STACKED ERROR
6248  027600  104460                                                          TRAP    C$ERROR
6249  027602                          ESCAPE  TST             ;SKIP TO END OF TEST
6250  027602  104410                                                          TRAP    C$ESCAPE
6251  027604  000042                                                          .WORD   L10023-.
6252
6253  027606  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND SYNC, RCV NEXT ONE
6254  027612  000226                  SYNCH
6255  027614  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6256  027616  000010                  8.
6257  027620  103003                  BCC     .+8.            ;BR IF NO ERROR
6258  027622                          ERROR                   ;REPORT STACKED ERROR
6259  027622  104460                                                          TRAP    C$ERROR
6260  027624                          ESCAPE  TST             ;SKIP TO END OF TEST
6261  027624  104410                                                          TRAP    C$ESCAPE
6262  027626  000020                                                          .WORD   L10023-.
6263
6264  027630  004537  011772          JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
6265  027634  000011          3$:     9.
6266  027636  103003                  BCC     .+8.            ;BR IF NO ERROR
6267  027640                          ERROR                   ;REPORT STACKED ERROR
6268  027640  104460                                                          TRAP    C$ERROR
6269  027642                          ESCAPE  TST             ;SKIP TO END OF TEST
6270  027642  104410                                                          TRAP    C$ESCAPE
6271  027644  000002                                                          .WORD   L10023-.
6272  027646                  ENDTST
6273  027646                                                          L10023:
```

6274 027646 104401 TRAP CSETST

```
                                        .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
027650                                                                                  T5::
027650  004737  005376                          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
027654  042737  001000  027732                  BIC     #NOLOOP,1$      ;
027662  012737  000011  030376                  MOV     #9.,3$          ;INIT ENTRAN COUNT/STATUS
027670  023727  002470  000000                  CMP     BRDTYP,#0       ;IS BOARD TYPE M8064?
027676  001412                                  BEQ     2$              ; YES: SPECIFY TTLOOP (NOT XLB)
027700  023727  002472  000004                  CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
027706  001406                                  BEQ     2$              ;BR IF NO
027710  052737  001000  027732                  BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
027716  053737  100000  030376                  BIS     BIT15,3$        ;       AND SPECIFY NOLOOP IN ENTRAN
                                        ;-----------------------------------------------------------------
027724  004537  007234                  2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
027730  062626                                  DDCMP!STRIPS!EVRC!SYNCH ;SET DDCMP,STRIP SYNCH,EVEN VRC,SYNCH=226
027732  000347                          1$:     TXDL!RXDL               ;USE 7 BIT CHARS FOR TX & RX
027734  103003                                  BCC     .+8.            ;BR IF NO ERROR
027736                                          ERROR                   ;REPORT STACKED ERROR
027736  104460                                                                                  TRAP    C$ERROR
027740                                          ESCAPE  TST             ;SKIP TO END OF TEST
027740  104410                                                                                  TRAP    C$ESCAPE
027742  000446                                                                                  .WORD   L10024-.

027744  004537  010136                          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
027750  000226                                  SYNCH
027752  000007                                  7.
027754  103003                                  BCC     .+8.            ;BR IF NO ERROR
027756                                          ERROR                   ;REPORT STACKED ERROR
027756  104460                                                                                  TRAP    C$ERROR
027760                                          ESCAPE  TST             ;SKIP TO END OF TEST
027760  104410                                                                                  TRAP    C$ESCAPE
027762  000426                                                                                  .WORD   L10024-.

027764  004537  010136                          JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
```

```
6331  027770  000226                  SYNCH
6332  027772  000010                  8.
6333  027774  103003                  BCC     .+8.            ;BR IF NO ERROR
6334  027776                          ERROR                   ;REPORT STACKED ERROR
6335  027776  104460                                                  TRAP    C$ERROR
6336  030000                          ESCAPE  TST             ;SKIP TO END OF TEST
6337  030000  104410                                                  TRAP    C$ESCAPE
6338  030002  000406                                                  .WORD   L10024-.
6339
6340  030004  004537  010250          JSR     R5,TXCTRL       ;CLEAR TSOM
6341  030010  000000                  000
6342  030012  000000                  0
6343
6344  030014  004537  010136          JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
6345  030020  000125                  125
6346  030022  000010                  8.
6347  030024  103003                  BCC     .+8.            ;BR IF NO ERROR
6348  030026                          ERROR                   ;REPORT STACKED ERROR
6349  030026  104460                                                  TRAP    C$ERROR
6350  030030                          ESCAPE  TST             ;SKIP TO END OF TEST
6351  030030  104410                                                  TRAP    C$ESCAPE
6352  030032  000356                                                  .WORD   L10024-.
6353
6354  030034  004537  010136          JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
6355  030040  000252                  252
6356  030042  000010                  8.
6357  030044  103003                  BCC     .+8.            ;BR IF NO ERROR
6358  030046                          ERROR                   ;REPORT STACKED ERROR
6359  030046  104460                                                  TRAP    C$ERROR
6360  030050                          ESCAPE  TST             ;SKIP TO END OF TEST
6361  030050  104410                                                  TRAP    C$ESCAPE
6362  030052  000336                                                  .WORD   L10024-.
6363
6364  030054  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(DATA3)
6365  030060  000000                  000
6366  030062  000000                  0
6367  030064  103003                  BCC     .+8.            ;BR IF NO ERROR
6368  030066                          ERROR                   ;REPORT STACKED ERROR
6369  030066  104460                                                  TRAP    C$ERROR
6370  030070                          ESCAPE  TST             ;SKIP TO END OF TEST
6371  030070  104410                                                  TRAP    C$ESCAPE
6372  030072  000316                                                  .WORD   L10024-.
6373
6374  030074  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV 125(DATA1)
6375  030100  000000                  0
6376  030102  103003                  BCC     .+8.            ;BR IF NO ERROR
6377  030104                          ERROR                   ;REPORT STACKED ERROR
6378  030104  104460                                                  TRAP    C$ERROR
6379  030106                          ESCAPE  TST             ;SKIP TO END OF TEST
6380  030106  104410                                                  TRAP    C$ESCAPE
6381  030110  000300                                                  .WORD   L10024-.
6382
6383  030112  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
6384  030116  000125                  125
6385  030120  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6386  030122  000010                  8.
```

```
6387  030124  103003                  BCC     .+8.            ;BR IF NO ERROR
6388  030126                          ERROR                   ;REPORT STACKED ERROR
6389  030126  104460                                                  TRAP    C$ERROR
6390  030130                          ESCAPE  TST             ;SKIP TO END OF TEST
6391  030130  104410                                                  TRAP    C$ESCAPE
6392  030132  000256                                                  .WORD   L10024-.
6393                          ;-----------------------------------------------------------
6394                          ; TRANSMIT THE BULK OF DATA OUT OF TABLE 'PATX'
6395                          ;-----------------------------------------------------------
6396  030134  012702  002646          MOV     #PATX+1,R2      ;SET UP TABLE POINTER
6397  030140  112237  030176  5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
6398  030144  116237  000001  030156  MOVB    1(R2),10$       ;SET UP TRANSMIT CHARACTER
6399
6400  030152  004537  010136          JSR     R5,TXCHAR       ;LOAD A CHARACTER
6401  030156  000000  10$:     000             ;** HOLE FOR NEXT TX CHARACTER
6402  030160  000000                  0
6403  030162  103003                  BCC     .+8.            ;BR IF NO ERROR
6404  030164                          ERROR                   ;REPORT STACKED ERROR
6405  030164  104460                                                  TRAP    C$ERROR
6406  030166                          ESCAPE  TST             ;SKIP TO END OF TEST
6407  030166  104410                                                  TRAP    C$ESCAPE
6408  030170  000220                                                  .WORD   L10024-.
6409
6410  030172  004537  010350          JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
6411  030176  000000  20$:     000             ;** HOLE FOR EXPECTED CHARACTER
6412  030200  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6413  030202  000010                  8.
6414  030204  103003                  BCC     .+8.            ;BR IF NO ERROR
6415  030206                          ERROR                   ;REPORT STACKED ERROR
6416  030206  104460                                                  TRAP    C$ERROR
6417  030210                          ESCAPE  TST             ;SKIP TO END OF TEST
6418  030210  104410                                                  TRAP    C$ESCAPE
6419  030212  000176                                                  .WORD   L10024-.
6420
6421  030214  022702  002737          CMP     #EPATX-1,R2
6422  030220  001347                  BNE     5$
6423                          ;-----------------------------------------------------------
6424  030222  004537  010250          JSR     R5,TXCTRL       ;LOAD 1ST TSOM
6425  030226  000001                  TSOM
6426  030230  000000                  0
6427  030232  103003                  BCC     .+8.            ;BR IF NO ERROR
6428  030234                          ERROR                   ;REPORT STACKED ERROR
6429  030234  104460                                                  TRAP    C$ERROR
6430  030236                          ESCAPE  TST             ;SKIP TO END OF TEST
6431  030236  104410                                                  TRAP    C$ESCAPE
6432  030240  000150                                                  .WORD   L10024-.
6433
6434  030242  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 277, RCV 177
6435  030246  000277                  277
6436  030250  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6437  030252  000010                  8.
6438  030254  103003                  BCC     .+8.            ;BR IF NO ERROR
6439  030256                          ERROR                   ;REPORT STACKED ERROR
6440  030256  104460                                                  TRAP    C$ERROR
6441  030260                          ESCAPE  TST             ;SKIP TO END OF TEST
6442  030260  104410                                                  TRAP    C$ESCAPE
```

```
6443  030262  000126                                                          .WORD   L10024-.
6444
6445  030264  004537  010250          JSR     R5,TXCTRL       ;LOAD 2ND TSOM
6446  030270  000001                  TSOM
6447  030272  000000                  0
6448  030274  103003                  BCC     .+8.            ;BR IF NO ERROR
6449  030276                          ERROR                   ;REPORT STACKED ERROR
6450  030276  104460                                                          TRAP    C$ERROR
6451  030300                          ESCAPE  TST             ;SKIP TO END OF TEST
6452  030300  104410                                                          TRAP    C$ESCAPE
6453  030302  000106                                                          .WORD   L10024-.
6454
6455  030304  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 177, RCV FIRST SYNC
6456  030310  000177                  177
6457  030312  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6458  030314  000010                  8.
6459  030316  103003                  BCC     .+8.            ;BR IF NO ERROR
6460  030320                          ERROR                   ;REPORT STACKED ERROR
6461  030320  104460                                                          TRAP    C$ERROR
6462  030322                          ESCAPE  TST             ;SKIP TO END OF TEST
6463  030322  104410                                                          TRAP    C$ESCAPE
6464  030324  000064                                                          .WORD   L10024-.
6465
6466  030326  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 1ST SYNC, RCV SECOND SYNC
6467  030332  000226                  SYNCH
6468  030334  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6469  030336  000010                  8.
6470  030340  103003                  BCC     .+8.            ;BR IF NO ERROR
6471  030342                          ERROR                   ;REPORT STACKED ERROR
6472  030342  104460                                                          TRAP    C$ERROR
6473  030344                          ESCAPE  TST             ;SKIP TO END OF TEST
6474  030344  104410                                                          TRAP    C$ESCAPE
6475  030346  000042                                                          .WORD   L10024-.
6476
6477  030350  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND SYNC, RCV NEXT ONE
6478  030354  000226                  SYNCH
6479  030356  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6480  030360  000010                  8.
6481  030362  103003                  BCC     .+8.            ;BR IF NO ERROR
6482  030364                          ERROR                   ;REPORT STACKED ERROR
6483  030364  104460                                                          TRAP    C$ERROR
6484  030366                          ESCAPE  TST             ;SKIP TO END OF TEST
6485  030366  104410                                                          TRAP    C$ESCAPE
6486  030370  000020                                                          .WORD   L10024-.
6487
6488  030372  004537  011772          JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
6489  030376  000011          3$:     9.                      ;
6490  030400  103003                  BCC     .+8.            ;BR IF NO ERROR
6491  030402                          ERROR                   ;REPORT STACKED ERROR
6492  030402  104460                                                          TRAP    C$ERROR
6493  030404                          ESCAPE  TST             ;SKIP TO END OF TEST
6494  030404  104410                                                          TRAP    C$ESCAPE
6495  030406  000002                                                          .WORD   L10024-.
6496  030410                  ENDTST
6497  030410                                                          L10024:
6498  030410  104401                                                          TRAP    C$ETST
```

```
                        .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).
                        ;*
                        ;--*****************************************************************************
                        ;
030412                  ;       BGNTST                                                  T6::
030412  012737  000003  002526        MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
030420  004737  005376                JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
030424  042737  001000  030466        BIC     #NOLOOP,1$      ;
030432  023727  002470  000000        CMP     BRDTYP,#0       ;IS BOARD TYPE = M8064 ?
030440  001407                        BEQ     2$              ; YES: SPECIFY TTLOOP (NO XLB)
030442  023727  002472  000004        CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
030450  001403                        BEQ     2$              ;BR IF NO
030452  052737  001000  030466        BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
                        ;-----------------------------------------------------------------------
030460  004537  007234        2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030464  000000                        0                       ;SET BOP MODE,CRC-CCITT=>1'S CHECK
030466  000000                1$:     0                       ;USE 8 BIT CHARS
030470  103003                        BCC     .+8.            ;BR IF NO ERROR
030472                                  ERROR                   ;REPORT STACKED ERROR
030472  104460                                                                   TRAP    C$ERROR
030474                                  ESCAPE  TST             ;SKIP TO END OF TEST
030474  104410                                                                   TRAP    C$ESCAPE
030476  000374                                                                   .WORD   L10025-.

030500  004537  010250                JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
030504  000001                        TSOM
030506  000007                        7.
030510  004537  010250                JSR     R5,TXCTRL       ;LOAD 3RD FLAG, TX 2ND FLAG
030514  000001                        TSOM
030516  000010                        8.
030520  004537  010250                JSR     R5,TXCTRL       ;LOAD 4TH FLAG, TX 3RD FLAG
030524  000001                        TSOM
030526  000010                        8.
030530  004537  010250                JSR     R5,TXCTRL       ;CLEAR TSOM
030534  000000                        000
030536  000000                        0
030540  004537  010136                JSR     R5,TXCHAR       ;LOAD DATA1(125), TX 4TH FLAG
030544  000125                        125
030546  000010                        8.
030550  103003                        BCC     .+8.            ;BR IF NO ERROR
```

```
6555  030552
6556  030552  104460                  ERROR                   ;REPORT STACKED ERROR
                                                                      TRAP    C$ERROR
6557  030554
6558  030554  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                      TRAP    C$ESCAPE
6559  030556  000314                                                  .WORD   L10025-.
6560
6561  030560  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA2(252), TX DATA1(125)
6562  030564  000252                  252
6563  030566  000010                  8.
6564  030570  103003                  BCC     .+8.            ;BR IF NO ERROR
6565  030572                          ERROR                   ;REPORT STACKED ERROR
6566  030572  104460                                                TRAP    C$ERROR
6567  030574
6568  030574  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                      TRAP    C$ESCAPE
6569  030576  000274                                                  .WORD   L10025-.
6570
6571  030600  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA3(000), TX DATA2(252)
6572  030604  000000                  000
6573  030606  000010                  8.
6574  030610  103003                  BCC     .+8.            ;BR IF NO ERROR
6575  030612                          ERROR                   ;REPORT STACKED ERROR
6576  030612  104460                                                TRAP    C$ERROR
6577  030614
6578  030614  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                      TRAP    C$ESCAPE
6579  030616  000254                                                  .WORD   L10025-.
6580
6581  030620  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA4(377), TX DATA3(000)
6582  030624  000377                  377
6583  030626  000010                  8.
6584  030630  103003                  BCC     .+8.            ;BR IF NO ERROR
6585  030632                          ERROR                   ;REPORT STACKED ERROR
6586  030632  104460                                                TRAP    C$ERROR
6587  030634
6588  030634  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                      TRAP    C$ESCAPE
6589  030636  000234                                                  .WORD   L10025-.
6590
6591  030640  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA5(001), TX DATA4(377)
6592  030644  000001                  001
6593  030646  000011                  9.
6594  030650  103003                  BCC     .+8.            ;BR IF NO ERROR
6595  030652                          ERROR                   ;REPORT STACKED ERROR
6596  030652  104460                                                TRAP    C$ERROR
6597  030654
6598  030654  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                      TRAP    C$ESCAPE
6599  030656  000214                                                  .WORD   L10025-.
6600
6601  030660  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
6602  030664  000002                  TEOM
6603  030666  000000                  0
6604
6605  030670  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV DATA1(125)
6606  030674  000000                  0
6607  030676  103003                  BCC     .+8.            ;BR IF NO ERROR
6608  030700                          ERROR                   ;REPORT STACKED ERROR
6609  030700  104460                                                TRAP    C$ERROR
6610  030702                          ESCAPE  TST             ;SKIP TO END OF TEST
```

```
6611  030702  104410                                                          TRAP    C$ESCAPE
6612  030704  000166                                                          .WORD   L10025-.
6613
6614  030706  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA1(125), RCV DATA2(252)
6615  030712  000525                RXSOM!125               ; & CHECK RSOM=1
6616  030714  000000                0
6617  030716  000010                8.
6618  030720  103003                BCC     .+8.            ;BR IF NO ERROR
6619  030722                        ERROR                   ;REPORT STACKED ERROR
6620  030722  104460                                                          TRAP    C$ERROR
6621  030724                        ESCAPE  TST             ;SKIP TO END OF TEST
6622  030724  104410                                                          TRAP    C$ESCAPE
6623  030726  000144                                                          .WORD   L10025-.
6624
6625  030730  004537  010250        JSR     R5,TXCTRL       ;SET TEOM
6626  030734  000002                TEOM
6627  030736  000000                0
6628
6629  030740  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA2(252), RCV DATA3(000)
6630  030744  000252                252
6631  030746  000000                0
6632  030750  000010                8.
6633  030752  103003                BCC     .+8.            ;BR IF NO ERROR
6634  030754                        ERROR                   ;REPORT STACKED ERROR
6635  030754  104460                                                          TRAP    C$ERROR
6636  030756                        ESCAPE  TST             ;SKIP TO END OF TEST
6637  030756  104410                                                          TRAP    C$ESCAPE
6638  030760  000112                                                          .WORD   L10025-.
6639
6640  030762  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA3(000), RCV DATA4(377)
6641  030766  000000                000
6642  030770  000000                0
6643  030772  000010                8.
6644  030774  103003                BCC     .+8.            ;BR IF NO ERROR
6645  030776                        ERROR                   ;REPORT STACKED ERROR
6646  030776  104460                                                          TRAP    C$ERROR
6647  031000                        ESCAPE  TST             ;SKIP TO END OF TEST
6648  031000  104410                                                          TRAP    C$ESCAPE
6649  031002  000070                                                          .WORD   L10025-.
6650
6651  031004  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA4(377), RCV DATA5(001)
6652  031010  000377                377
6653  031012  000000                0
6654  031014  020010                NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
6655  031016  103003                BCC     .+8.            ;BR IF NO ERROR
6656  031020                        ERROR                   ;REPORT STACKED ERROR
6657  031020  104460                                                          TRAP    C$ERROR
6658  031022                        ESCAPE  TST             ;SKIP TO END OF TEST
6659  031022  104410                                                          TRAP    C$ESCAPE
6660  031024  000046                                                          .WORD   L10025-.
6661
6662  031026  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA5(001), RCV FIRST FLAG
6663  031032  001001                RXEOM!001               ; & CHECK REOM
6664  031034  000001                RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
6665  031036  060000                NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
6666  031040  103003                BCC     .+8.            ;BR IF NO ERROR
```

```
031042                          ERROR                       ;REPORT STACKED ERROR
031042  104460                                                                      TRAP    C$ERROR
031044                          ESCAPE  TST                 ;SKIP TO END OF TEST
031044  104410                                                                      TRAP    C$ESCAPE
031046  000024                                                                      .WORD   L10025-.

                        ;-------------------------------------------------------------------------
031050  005337  002526          DEC     REG0                ;DECREMENT COUNT
031054  001406                  BEQ     40$                 ;BR IF TRIPLE LOOP IS COMPLETED

031056  004537  010250          JSR     R5,TXCTRL           ;CLEAR TEOM, SET TSOM
031062  000001                  TSOM
031064  000001                  1
031066  000137  030460          JMP     2$                  ;AND RUN TX/RX AGAIN
031072                  40$:
031072                  ENDTST
031072                                                                   L10025:
031072  104401                                                                      TRAP    C$ETST
```

```
                                .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
031074                                                                          T7::
031074  012737  000003  002526        MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
031102  004737  005376              JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
031106  042737  001000  031150      BIC     #NOLOOP,1$      ;
031114  023727  002470  000000      CMP     BRDTYP,#0       ;BOARD TYPE = M8064 ?
031122  001407                      BEQ     2$              ; YES: SPECIFY TTLOOP (NOT XLB)
031124  023727  002472  000004      CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
031132  001403                      BEQ     2$              ;BR IF NO
031134  052737  001000  031150      BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
                                ;-----------------------------------------------------------------------
031142  004537  007234        2$:   JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
031146  000400                      CRCOS                   ;SET BOP MODE,CRC-CCITT=>0'S CHECK
031150  000000                1$:   0                       ;USE 8 BIT CHARS
031152  103003                      BCC     .+8.            ;BR IF NO ERROR
031154                              ERROR                   ;REPORT STACKED ERROR
031154  104460                                                              TRAP    C$ERROR
031156                              ESCAPE  TST             ;SKIP TO END OF TEST
031156  104410                                                              TRAP    C$ESCAPE
031160  000374                                                              .WORD   L10026-.
                                ;
031162  004537  010250              JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
031166  000001                      TSOM
031170  000007                      7.
031172  004537  010250              JSR     R5,TXCTRL       ;LOAD 3RD FLAG, TX 2ND FLAG
031176  000001                      TSOM
031200  000010                      8.
031202  004537  010250              JSR     R5,TXCTRL       ;LOAD 4TH FLAG, TX 3RD FLAG
031206  000001                      TSOM
031210  000010                      8.
031212  004537  010250              JSR     R5,TXCTRL       ;CLEAR TSOM
031216  000000                      000
031220  000000                      0
031222  004537  010136              JSR     R5,TXCHAR       ;LOAD DATA1(125), TX 4TH FLAG
031226  000125                      125
031230  000010                      8.
031232  103003                      BCC     .+8.            ;BR IF NO ERROR
```

```
6740  031234
6741  031234  104460                  ERROR                   ;REPORT STACKED ERROR
                                                                TRAP    C$ERROR
6742  031236
6743  031236  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                TRAP    C$ESCAPE
6744  031240  000314                                          .WORD   L10026-.
6745
6746  031242  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA2(252), TX DATA1(125)
6747  031246  000252                  252
6748  031250  000010                  8.
6749  031252  103003                  BCC     .+8.            ;BR IF NO ERROR
6750  031254                          ERROR                   ;REPORT STACKED ERROR
6751  031254  104460                                          TRAP    C$ERROR
6752  031256
6753  031256  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                TRAP    C$ESCAPE
6754  031260  000274                                          .WORD   L10026-.
6755
6756  031262  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA3(000), TX DATA2(252)
6757  031266  000000                  000
6758  031270  000010                  8.
6759  031272  103003                  BCC     .+8.            ;BR IF NO ERROR
6760  031274                          ERROR                   ;REPORT STACKED ERROR
6761  031274  104460                                          TRAP    C$ERROR
6762  031276
6763  031276  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                TRAP    C$ESCAPE
6764  031300  000254                                          .WORD   L10026-.
6765
6766  031302  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA4(377), TX DATA3(000)
6767  031306  000377                  377
6768  031310  000010                  8.
6769  031312  103003                  BCC     .+8.            ;BR IF NO ERROR
6770  031314                          ERROR                   ;REPORT STACKED ERROR
6771  031314  104460                                          TRAP    C$ERROR
6772  031316
6773  031316  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                TRAP    C$ESCAPE
6774  031320  000234                                          .WORD   L10026-.
6775
6776  031322  004537  010136          JSR     R5,TXCHAR       ;LOAD DATA5(001), TX DATA4(377)
6777  031326  000001                  001
6778  031330  000011                  9.
6779  031332  103003                  BCC     .+8.            ;BR IF NO ERROR
6780  031334                          ERROR                   ;REPORT STACKED ERROR
6781  031334  104460                                          TRAP    C$ERROR
6782  031336
6783  031336  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                TRAP    C$ESCAPE
6784  031340  000214                                          .WORD   L10026-.
6785
6786  031342  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
6787  031346  000002                  TEOM
6788  031350  000000                  0
6789  031352  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV DATA1(125)
6790  031356  000000                  0
6791  031360  103003                  BCC     .+8.            ;BR IF NO ERROR
6792  031362                          ERROR                   ;REPORT STACKED ERROR
6793  031362  104460                                          TRAP    C$ERROR
6794  031364
6795  031364  104410                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                TRAP    C$ESCAPE
```

```
6796  031366  000166                                                          .WORD   L10026-.
6797
6798  031370  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA1(125), RCV DATA2(252)
6799  031374  000525                RXSOM!125               ; & CHECK RSOM=1
6800  031376  000000                0
6801  031400  000010                8.
6802  031402  103003                BCC     .+8.            ;BR IF NO ERROR
6803  031404                        ERROR                   ;REPORT STACKED ERROR
6804  031404  104460                                                          TRAP    C$ERROR
6805  031406                        ESCAPE  TST             ;SKIP TO END OF TEST
6806  031406  104410                                                          TRAP    C$ESCAPE
6807  031410  000144                                                          .WORD   L10026-.
6808
6809  031412  004537  010250        JSR     R5,TXCTRL       ;SET TEOM
6810  031416  000002                TEOM
6811  031420  000000                0
6812  031422  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA2(252), RCV DATA3(000)
6813  031426  000252                252
6814  031430  000000                0
6815  031432  000010                8.
6816  031434  103003                BCC     .+8.            ;BR IF NO ERROR
6817  031436                        ERROR                   ;REPORT STACKED ERROR
6818  031436  104460                                                          TRAP    C$ERROR
6819  031440                        ESCAPE  TST             ;SKIP TO END OF TEST
6820  031440  104410                                                          TRAP    C$ESCAPE
6821  031442  000112                                                          .WORD   L10026-.
6822
6823  031444  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA3(000), RCV DATA4(377)
6824  031450  000000                000
6825  031452  000000                0
6826  031454  000010                8.
6827  031456  103003                BCC     .+8.            ;BR IF NO ERROR
6828  031460                        ERROR                   ;REPORT STACKED ERROR
6829  031460  104460                                                          TRAP    C$ERROR
6830  031462                        ESCAPE  TST             ;SKIP TO END OF TEST
6831  031462  104410                                                          TRAP    C$ESCAPE
6832  031464  000070                                                          .WORD   L10026-.
6833
6834  031466  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA4(377), RCV DATA5(001)
6835  031472  000377                377
6836  031474  000000                0
6837  031476  020010                NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
6838  031500  103003                BCC     .+8.            ;BR IF NO ERROR
6839  031502                        ERROR                   ;REPORT STACKED ERROR
6840  031502  104460                                                          TRAP    C$ERROR
6841  031504                        ESCAPE  TST             ;SKIP TO END OF TEST
6842  031504  104410                                                          TRAP    C$ESCAPE
6843  031506  000046                                                          .WORD   L10026-.
6844
6845  031510  004537  010350        JSR     R5,RXCHAR       ;READ/CHK DATA5(001), RCV FIRST FLAG
6846  031514  001001                RXEOM!001               ; & CHECK REOM
6847  031516  000001                RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
6848  031520  060000                NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
6849  031522  103003                BCC     .+8.            ;BR IF NO ERROR
6850  031524                        ERROR                   ;REPORT STACKED ERROR
6851  031524  104460                                                          TRAP    C$ERROR
```

```
CVDMEA.P11    10-DEC-80 09:16      TEST 7 -- DATA TEST -- BOP XLB CRC-CCITT-0

  6852  031526
  6853  031526  104410                      ESCAPE  TST             ;SKIP TO END OF TEST
  6854  031530  000024                                                              TRAP    C$ESCAPE
  6855                                                                              .WORD   L10026-.
  6856  031532  005337  002526      ;-------------------------------------------------------------
                                            DEC     REG0            ;DECREMENT COUNT
  6857  031536  001406                      BEQ     40$             ;BR IF TRIPLE LOOP IS COMPLETED
  6858
  6859  031540  004537  010250              JSR     R5,TXCTRL       ;CLEAR TEOM, SET TSOM
  6860  031544  000001                      TSOM
  6861  031546  000001                      1
  6862  031550  000137  031142              JMP     2$              ;AND RUN TX/RX AGAIN
  6863  031554                      40$:
  6864  031554                      ENDTST
  6865  031554                                                                L10026:
  6866  031554  104401                                                               TRAP    C$ETST
```

```
                        .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
031556                                                                  T8::
031556  012737  000010  002414          MOV     #8.,TSTNUM      ;SET TEST NO. FOR POSSIBLE PRINTOUT
031564  004537  007532                  JSR     R5,CKLPBK       ;CHK FOR H3254/5 INSTALLED
031570  100000                          TCCHEK
031572  103002                          BCC     2$              ;BR IF YES, TO RUN TEST
031574                                  EXIT    TST             ;NO TEST CONNECTOR, SKIP TEST
031574  104432                                                          TRAP    C$EXIT
031576  000722                                                          .WORD   L10027-.
031600                          2$:
                                        BGNSUB
031600                                                                  T8.1:
031600  104402                                                          TRAP    C$BSUB
                        ;INIT DMV, SET TTL LOOPBACK, CHK MODEM STATUS
031602  004737  003352                  JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR TO INIT DMV11
031606  004537  003712                  JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
031612  120002                          VIADPB
031614  000377                          377
031616  004537  003712                  JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
031622  120003                          VIADPA
031624  000000                          000
031626  004537  003712                  JSR     R5,WRITEI       ;SET TTL LOOPBACK
031632  120000                          VIAORB
031634  000002                          TTLOOP
031636  012737  000001  002336          MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
                        ;CHK FOR RING, CARRIER, MODEM RDY, CTS, TEST MODE, = 1
031644  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
031650  120017                          VIAORA
031652  000000                  4$:     .WORD   0
031654  142737  000023  031652          BICB    #SPEED!RCVDAT!UMAINT,4$ ;CLEAR UNNEEDED BITS
031662  123727  031652  000354          CMPB    4$,#RING!CARIER!MDMRDY!CTS!TM ;CHK FOR BITS SET
```

```
                        BEQ     8$                      ;BR IF ALL BITS SET
                        MOV     #RING!CARIER!MDMRDY!CTS!TM,GDATA ;SET EXPECTED DATA
                        MOV     4$,BDATA                ;SET ACTUAL DATA
                ;REPORT 'MODEM STATUS INCORRECT'
                        GEDF    EM83,ERR11
                                                ;       'DEVICE FATAL'' ERROR # 66
                                                                        TRAP    C$ERDF
                                                                        .WORD   66
                                                                        .WORD   EM83
                                                                        .WORD   ERR11
                        ESCAPE  SUB
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10030-.
                ;DE-ASSERT RTS, CHK FOR CTS = 0
                8$:     JSR     R5,WRITEI               ;DE-ASSERT RTS
                        VIAORB
                        RTSND!TTLOOP
                        JSR     R5,READI                ;READ MODEM STATUS
                        VIAORA
                10$:    .WORD   0
                        BITB    #CTS,10$                ;CHK FOR CTS = 0
                        BEQ     12$                     ;BR IF YES
                ;REPORT CTS NOT CLEARED
                        GEDF    EM84,ERR14
                                                ;       'DEVICE FATAL'' ERROR # 67
                                                                        TRAP    C$ERDF
                                                                        .WORD   67
                                                                        .WORD   EM84
                                                                        .WORD   ERR14
                        ESCAPE  SUB
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10030-.
                ;ASSERT RTS, CHK FOR CTS = 1
                12$:    JSR     R5,WRITEI               ;ASSERT RTS
                        VIAORB
                        TTLOOP
                        JSR     R5,READI                ;READ MODEM STATUS
                        VIAORA
                14$:    .WORD   0
                        BITB    #CTS,14$                ;CHK FOR CTS = 1
                        BNE     15$                     ;BR IF YES
                ;REPORT CTS NOT SET
                        GEDF    EM85,ERR14
                                                ;       'DEVICE FATAL'' ERROR # 68
                                                                        TRAP    C$ERDF
                                                                        .WORD   68
                                                                        .WORD   EM85
                                                                        .WORD   ERR14
                15$:
                        ENDSUB
                                                                L10030:
                                                                        TRAP    C$ESUB
                ;SEE IF BOARD IS M8053 WITH H3254 INSTALLED
                        TST     BRDTYP                  ;SEE IF M8053
                        BNE     17$                     ;BR IF YES
                16$:    JMP     A1                      ;SKIP THIS SECTION OF CODE
```

```
                        TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

 6979 032042 023727 002472 000000 17$:    CMP     TSTCON,#0       ;SEE IF H3254 INSTALLED
 6980 032050 001372                       BNE     16$             ;BR IF NOT, TO SKIP CODE
 6981 032052                              BGNSUB
 6982 032052                                                                      T8.2:
 6983 032052 104402                                                                TRAP    C$BSUB
 6984                       ;INIT DMV, (TTL LOOPBACK IS CLEARED), CHK MODEM STATUS
 6985 032054 004737 003352                JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR TO INIT DMV11
 6986 032060 004537 003712                JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
 6987 032064 120002                       VIADPB
 6988 032066 000377                       377
 6989 032070 004537 003712                JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
 6990 032074 120003                       VIADPA
 6991 032076 000000                       000
 6992 032100 004537 003712                JSR     R5,WRITEI       ;DISABLE TTLOOP
 6993 032104 120000                       VIAORB
 6994 032106 000000                       000
 6995 032110 012737 000001 002336         MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
 6996                       ;-----------------------------------------------------------
 6997                       ;CHK FOR RING (IF EIA), CARRIER, MODEM RDY, CTS = 1, TEST MODE, = 0
 6998                       ;-----------------------------------------------------------
 6999 032116 004537 003566                JSR     R5,READI        ;READ MODEM STATUS
 7000 032122 120017                       VIAORA
 7001 032124 000000         18$:          .WORD   0
 7002
 7003 032126 023727 002470 000001         CMP     BRDTYP,#1       ;IS V.35 THE SELECTED I/F ?
 7004 032134 001013                       BNE     21$             ; NO: BR TO DO CHECK WITH RING
 7005                                                             ;YES: REMOVE RING BEFORE CHECKING
 7006 032136 142737 000223 032124         BICB    #RING!SPEED!RCVDAT!UMAINT,18$ ;CLEAR UNNEEDED BITS
 7007 032144 123727 032124 000150         CMPB    18$,#CARIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
 7008 032152 001427                       BEQ     20$             ;BR IF STATUS CORRECT
 7009 032154 012737 000150 002324         MOV     #CARIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
 7010 032162 000412                       BR      19$
 7011                                                             ; DO CHECK WITH RING....
 7012 032164 142737 000023 032124 21$:    BICB    #SPEED!RCVDAT!UMAINT,18$ ;CLEAR UNNEEDED BITS
 7013 032172 123727 032124 000350         CMPB    18$,#RING!CARIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
 7014 032200 001414                       BEQ     20$             ;BR IF STATUS CORRECT
 7015 032202 012737 000350 002324         MOV     #RING!CARIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
 7016
 7017                       ;REPORT 'MODEM STATUS INCORRECT''
 7018 032210 013737 032124 002326 19$:    MOV     18$,BDATA       ;SET ACTUAL DATA
 7019 032216                              GEDF    EM83,ERR11
 7020                                                             ;       'DEVICE FATAL'' ERROR # 69
 7021 032216 104455                                                                TRAP    C$ERDF
 7022 032220 000105                                                                .WORD   69
 7023 032222 015262                                                                .WORD   EM83
 7024 032224 020540                                                                .WORD   ERR11
 7025 032226                              ESCAPE  SUB
 7026 032226 104410                                                                TRAP    C$ESCAPE
 7027 032230 000266                                                                .WORD   L10031-.
 7028                       ;DE-ASSERT RTS, CHK FOR CTS,CARRIER = 0
 7029 032232 004537 003712 20$:    JSR     R5,WRITEI       ;DE-ASSERT RTS
 7030 032236 120000                VIAORB
 7031 032240 000010                RTSND
 7032 032242 004537 003566         JSR     R5,READI        ;READ MODEM STATUS
 7033 032246 120017                VIAORA
 7034 032250 000000         22$:   .WORD   0
```

```
7035  032252  132737  000010  032250          BITB    #CTS,22$        ;CHK FOR CTS = 0
7036  032260  001406                          BEQ     24$             ;BR IF YES
7037                                  ;REPORT CTS NOT CLEARED
7038  032262                                  GEDF    EM84,ERR14
7039                                                  ;       'DEVICE FATAL'' ERROR # 70
7040  032262  104455                                                          TRAP    C$ERDF
7041  032264  000106                                                          .WORD   70
7042  032266  015311                                                          .WORD   EM84
7043  032270  021146                                                          .WORD   ERR14
7044  032272                                  ESCAPE  SUB
7045  032272  104410                                                          TRAP    C$ESCAPE
7046  032274  000222                                                          .WORD   L10031-.
7047  032276  132737  000100  032250  24$:     BITB    #CARIER,22$     ;CHK FOR CARRIER = 0
7048  032304  001406                          BEQ     26$             ;BR IF YES
7049                                  ;REPORT CARRIER NOT CLEARED
7050  032306                                  GEDF    EM86,ERR14
7051                                                  ;       'DEVICE FATAL'' ERROR # 71
7052  032306  104455                                                          TRAP    C$ERDF
7053  032310  000107                                                          .WORD   71
7054  032312  015342                                                          .WORD   EM86
7055  032314  021146                                                          .WORD   ERR14
7056  032316                                  ESCAPE  SUB
7057  032316  104410                                                          TRAP    C$ESCAPE
7058  032320  000176                                                          .WORD   L10031-.
7059                                  ;ASSERT RTS, CHK FOR CTS,CARRIER = 1
7060  032322  004537  003712          26$:    JSR     R5,WRITEI       ;ASSERT RTS
7061  032326  120000                          VIAORB
7062  032330  000000                          000
7063  032332  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
7064  032336  120017                          VIAORA
7065  032340  000000                  28$:    .WORD   0
7066  032342  132737  000010  032340          BITB    #CTS,28$        ;CHK FOR CTS = 1
7067  032350  001006                          BNE     30$             ;BR IF YES
7068                                  ;REPORT CTS NOT SET
7069  032352                                  GEDF    EM85,ERR14
7070                                                  ;       'DEVICE FATAL'' ERROR # 72
7071  032352  104455                                                          TRAP    C$ERDF
7072  032354  000110                                                          .WORD   72
7073  032356  015326                                                          .WORD   EM85
7074  032360  021146                                                          .WORD   ERR14
7075  032362                                  ESCAPE  SUB
7076  032362  104410                                                          TRAP    C$ESCAPE
7077  032364  000132                                                          .WORD   L10031-.
7078  032366  132737  000100  032340  30$:     BITB    #CARIER,28$     ;CHK FOR CARRIER = 1
7079  032374  001006                          BNE     32$             ;BR IF YES
7080                                  ;REPORT CARRIER NOT SET
7081  032376                                  GEDF    EM87,ERR14
7082                                                  ;       'DEVICE FATAL'' ERROR # 73
7083  032376  104455                                                          TRAP    C$ERDF
7084  032400  000111                                                          .WORD   73
7085  032402  015363                                                          .WORD   EM87
7086  032404  021146                                                          .WORD   ERR14
7087  032406                                  ESCAPE  SUB
7088  032406  104410                                                          TRAP    C$ESCAPE
7089  032410  000106                                                          .WORD   L10031-.
7090                                  ;DE-ASSERT DTR, CHK FOR MODEM READY = 0
```

```
CVDMEA.P11    10-DEC-80 09:16          TEST 8 -- MODEM CONTROL SIGNAL LOOPBACK TEST

 7091  032412  004537  003712          32$:    JSR     R5,WRITEI       ;DE-ASSERT DTR
 7092  032416  120000                          VIAORB
 7093  032420  000020                          DTR
 7094  032422  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
 7095  032426  120017                          VIAORA
 7096  032430  000000                  34$:    .WORD   0
 7097  032432  132737  000040  032430          BITB    #MDMRDY,34$     ;CHK FOR MODEM READY = 0
 7098  032440  001406                          BEQ     36$             ;BR IF YES
 7099                                  ;REPORT MODEM READY NOT CLEARED
 7100  032442                                  GEDF    EM88,ERR14
 7101                                                          ;       'DEVICE FATAL'' ERROR # 74
 7102  032442  104455                                                          TRAP    C$ERDF
 7103  032444  000112                                                          .WORD   74
 7104  032446  015403                                                          .WORD   EM88
 7105  032450  021146                                                          .WORD   ERR14
 7106  032452                                  ESCAPE  SUB
 7107  032452  104410                                                          TRAP    C$ESCAPE
 7108  032454  000042                                                          .WORD   L10031-.
 7109                                  ;ASSERT DTR, CHK FOR MODEM READY = 1
 7110  032456  004537  003712          36$:    JSR     R5,WRITEI       ;ASSERT DTR
 7111  032462  120000                          VIAORB
 7112  032464  000000                          000
 7113  032466  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
 7114  032472  120017                          VIAORA
 7115  032474  000000                  38$:    .WORD   0
 7116  032476  132737  000040  032474          BITB    #MDMRDY,38$     ;CHK FOR MODEM READY = 1
 7117  032504  001004                          BNE     40$             ;BR IF YES
 7118                                  ;REPORT MODEM READY NOT SET
 7119  032506                                  GEDF    EM89,ERR14
 7120                                                          ;       'DEVICE FATAL'' ERROR # 75
 7121  032506  104455                                                          TRAP    C$ERDF
 7122  032510  000113                                                          .WORD   75
 7123  032512  015426                                                          .WORD   EM89
 7124  032514  021146                                                          .WORD   ERR14
 7125  032516                          40$:
 7126  032516                                  ENDSUB
 7127  032516                                                          L10031:
 7128  032516  104403                                                          TRAP    C$ESUB
 7129  032520                          A1:
 7130  032520                          ENDTST
 7131  032520                                                          L10027:
 7132  032520  104401                                                          TRAP    C$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
032522                                                                          T9::
032522  012737  000003  002526          MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
032530  004737  005376                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
032534  042737  001000  032576          BIC     #NOLOOP,1$      ;
032542  023727  002470  000000          CMP     BRDTYP,#0       ;BOARD TYPE = M8064 ?
032550  001407                          BEQ     2$              ; YES: SPECIFY TTLOOP (NOT XLB)
032552  023727  002472  000004          CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
032560  001403                          BEQ     2$              ;BR IF NO
032562  052737  001000  032576          BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
                                ;-----------------------------------------------------------------------
032570  004537  007234          2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
032574  065626                          DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
032576  000000          1$:     0               ;USE 8 BIT CHARS
032600  103003                  BCC     .+8.    ;BR IF NO ERROR
032602                          ERROR           ;REPORT STACKED ERROR
032602  104460                                                          TRAP    C$ERROR
032604                          ESCAPE  TST     ;SKIP TO END OF TEST
032604  104410                                                          TRAP    C$ESCAPE
032606  001032                                                          .WORD   L10032-.

032610  004537  010250                  JSR     R5,TXCTRL       ;SET TSOM, TX 1ST SYNCH
032614  000001                          TSOM
032616  000007                          7.
032620  004537  010250                  JSR     R5,TXCTRL       ;TX 2ND SYNCH
032624  000001                          TSOM
032626  000010                          8.
032630  004537  010250                  JSR     R5,TXCTRL       ;CLEAR TSOM
032634  000000                          000
032636  000000                          0
```

```
7189  032640  004537  010136          JSR     R5,TXCHAR       ;LOAD 201(HEADR1), TX 3RD SYNCH
7190  032644  000201                  201
7191  032646  000010                  8.
7192  032650  103003                  BCC     .+8.            ;BR IF NO ERROR
7193  032652                          ERROR                   ;REPORT STACKED ERROR
7194  032652  104460                                                          TRAP    C$ERROR
7195  032654                          ESCAPE  TST             ;SKIP TO END OF TEST
7196  032654  104410                                                          TRAP    C$ESCAPE
7197  032656  000762                                                          .WORD   L10032-.
7198
7199  032660  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(HEADR2), TX HEADR1
7200  032664  000000                  000
7201  032666  000010                  8.
7202  032670  103003                  BCC     .+8.            ;BR IF NO ERROR
7203  032672                          ERROR                   ;REPORT STACKED ERROR
7204  032672  104460                                                          TRAP    C$ERROR
7205  032674                          ESCAPE  TST             ;SKIP TO END OF TEST
7206  032674  104410                                                          TRAP    C$ESCAPE
7207  032676  000742                                                          .WORD   L10032-.
7208
7209  032700  004537  010136          JSR     R5,TXCHAR       ;LOAD 075(HEADR3), TX HEADR2
7210  032704  000075                  075
7211  032706  000010                  8.
7212  032710  103003                  BCC     .+8.            ;BR IF NO ERROR
7213  032712                          ERROR                   ;REPORT STACKED ERROR
7214  032712  104460                                                          TRAP    C$ERROR
7215  032714                          ESCAPE  TST             ;SKIP TO END OF TEST
7216  032714  104410                                                          TRAP    C$ESCAPE
7217  032716  000722                                                          .WORD   L10032-.
7218
7219  032720  004537  010136          JSR     R5,TXCHAR       ;LOAD 003(HEADR4), TX HEADR3
7220  032724  000003                  003
7221  032726  000010                  8.
7222  032730  103003                  BCC     .+8.            ;BR IF NO ERROR
7223  032732                          ERROR                   ;REPORT STACKED ERROR
7224  032732  104460                                                          TRAP    C$ERROR
7225  032734                          ESCAPE  TST             ;SKIP TO END OF TEST
7226  032734  104410                                                          TRAP    C$ESCAPE
7227  032736  000702                                                          .WORD   L10032-.
7228
7229  032740  004537  010136          JSR     R5,TXCHAR       ;LOAD 002(HEADR5)
7230  032744  000002                  002
7231  032746  000000                  0
7232  032750  103003                  BCC     .+8.            ;BR IF NO ERROR
7233  032752                          ERROR                   ;REPORT STACKED ERROR
7234  032752  104460                                                          TRAP    C$ERROR
7235  032754                          ESCAPE  TST             ;SKIP TO END OF TEST
7236  032754  104410                                                          TRAP    C$ESCAPE
7237  032756  000662                                                          .WORD   L10032-.
7238
7239  032760  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV 201
7240  032764  000000                  0
7241  032766  103003                  BCC     .+8.            ;BR IF NO ERROR
7242  032770                          ERROR                   ;REPORT STACKED ERROR
7243  032770  104460                                                          TRAP    C$ERROR
7244  032772                          ESCAPE  TST             ;SKIP TO END OF TEST
```

```
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10032-.

032776  004537  010350        JSR     R5,RXCHAR       ;READ & CHK 201(HEADR1), RCV HEADR2
033002  000201                201
033004  000000                0
033006  000010                8.
033010  103003                BCC     .+8.            ;BR IF NO ERROR
033012                        ERROR                   ;REPORT STACKED ERROR
033012  104460                                                TRAP    C$ERROR
033014                        ESCAPE  TST             ;SKIP TO END OF TEST
033014  104410                                                TRAP    C$ESCAPE
033016  000622                                                .WORD   L10032-.

033020  004537  010136        JSR     R5,TXCHAR       ;LOAD 001(HEADR6)
033024  000001                001
033026  000000                0
033030  103003                BCC     .+8.            ;BR IF NO ERROR
033032                        ERROR                   ;REPORT STACKED ERROR
033032  104460                                                TRAP    C$ERROR
033034                        ESCAPE  TST             ;SKIP TO END OF TEST
033034  104410                                                TRAP    C$ESCAPE
033036  000602                                                .WORD   L10032-.

033040  004537  010350        JSR     R5,RXCHAR       ;READ & CHK 000(HEADR2), RCV HEADR3
033044  000000                000
033046  000000                0
033050  000010                8.
033052  103003                BCC     .+8.            ;BR IF NO ERROR
033054                        ERROR                   ;REPORT STACKED ERROR
033054  104460                                                TRAP    C$ERROR
033056                        ESCAPE  TST             ;SKIP TO END OF TEST
033056  104410                                                TRAP    C$ESCAPE
033060  000560                                                .WORD   L10032-.

033062  004537  010250        JSR     R5,TXCTRL       ;SET TEOM
033066  000002                TEOM                    ;(STARTS CRC-16 CHARACTER)
033070  000000                0
033072  004537  010350        JSR     R5,RXCHAR       ;READ & CHK 075(HEADR3), RCV HEADR4
033076  000075                075
033100  000000                0
033102  000010                8.
033104  103003                BCC     .+8.            ;BR IF NO ERROR
033106                        ERROR                   ;REPORT STACKED ERROR
033106  104460                                                TRAP    C$ERROR
033110                        ESCAPE  TST             ;SKIP TO END OF TEST
033110  104410                                                TRAP    C$ESCAPE
033112  000526                                                .WORD   L10032-.

033114  004537  010350        JSR     R5,RXCHAR       ;READ & CHK 003(HEADR4), RCV HEADR5
033120  000003                003
033122  000000                0
033124  000010                8.
033126  103003                BCC     .+8.            ;BR IF NO ERROR
033130                        ERROR                   ;REPORT STACKED ERROR
033130  104460                                                TRAP    C$ERROR
```

```
7301  033132                        ESCAPE  TST             ;SKIP TO END OF TEST
7302  033132  104410                                                    TRAP    C$ESCAPE
7303  033134  000504                                                    .WORD   L10032-.
7304
7305  033136  004537  010250        JSR     R5,TXCTRL       ;CLEAR TEOM
7306  033142  000000                000
7307  033144  000000                0
7308                                ;-----------------------------------------------------------------
7309  033146  004537  010136        JSR     R5,TXCHAR       ;LOAD 000(DATA1)
7310  033152  000000                000
7311  033154  000000                0
7312  033156  103003                BCC     .+8.            ;BR IF NO ERROR
7313  033160                        ERROR                   ;REPORT STACKED ERROR
7314  033160  104460                                                    TRAP    C$ERROR
7315  033162                        ESCAPE  TST             ;SKIP TO END OF TEST
7316  033162  104410                                                    TRAP    C$ESCAPE
7317  033164  000454                                                    .WORD   L10032-.
7318
7319  033166  004537  010350        JSR     R5,RXCHAR       ;READ & CHK 002(HEADR5), RCV HEADR6
7320  033172  000002                002
7321  033174  000000                0
7322  033176  000010                8.
7323  033200  103003                BCC     .+8.            ;BR IF NO ERROR
7324  033202                        ERROR                   ;REPORT STACKED ERROR
7325  033202  104460                                                    TRAP    C$ERROR
7326  033204                        ESCAPE  TST             ;SKIP TO END OF TEST
7327  033204  104410                                                    TRAP    C$ESCAPE
7328  033206  000432                                                    .WORD   L10032-.
7329
7330  033210  004537  010136        JSR     R5,TXCHAR       ;LOAD 377(DATA2)
7331  033214  000377                377
7332  033216  000000                0
7333  033220  103003                BCC     .+8.            ;BR IF NO ERROR
7334  033222                        ERROR                   ;REPORT STACKED ERROR
7335  033222  104460                                                    TRAP    C$ERROR
7336  033224                        ESCAPE  TST             ;SKIP TO END OF TEST
7337  033224  104410                                                    TRAP    C$ESCAPE
7338  033226  000412                                                    .WORD   L10032-.
7339
7340  033230  004537  010350        JSR     R5,RXCHAR       ;READ/CHK 001(HEADR6), 'RCV' FIRST CRC BYTE
7341  033234  100001                RXERR!001
7342  033236  000001                RERCHK
7343  033240  000010                8.
7344  033242  103003                BCC     .+8.            ;BR IF NO ERROR
7345  033244                        ERROR                   ;REPORT STACKED ERROR
7346  033244  104460                                                    TRAP    C$ERROR
7347  033246                        ESCAPE  TST             ;SKIP TO END OF TEST
7348  033246  104410                                                    TRAP    C$ESCAPE
7349  033250  000370                                                    .WORD   L10032-.
7350
7351  033252  004537  010136        JSR     R5,TXCHAR       ;LOAD 376(DATA3)
7352  033256  000376                376
7353  033260  000000                0
7354  033262  103003                BCC     .+8.            ;BR IF NO ERROR
7355  033264                        ERROR                   ;REPORT STACKED ERROR
7356  033264  104460                                                    TRAP    C$ERROR
```

```
7357  033266
7358  033266  104410                        ESCAPE  TST             ;SKIP TO END OF TEST
7359  033270  000350                                                                TRAP    C$ESCAPE
7360                                                                               .WORD   L10032-.
7361  033272  004537  010350                JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
7362  033276  000043                        043
7363  033300  000000                        0
7364  033302  000010                        8.
7365  033304  103003                        BCC     .+8.            ;BR IF NO ERROR
7366  033306                                ERROR                   ;REPORT STACKED ERROR
7367  033306  104460                                                                TRAP    C$ERROR
7368  033310                                ESCAPE  TST             ;SKIP TO END OF TEST
7369  033310  104410                                                                TRAP    C$ESCAPE
7370  033312  000326                                                                .WORD   L10032-.
7371
7372  033314  004537  010136                JSR     R5,TXCHAR       ;LOAD 375(DATA4)
7373  033320  000375                        375
7374  033322  000000                        0
7375  033324  103003                        BCC     .+8.            ;BR IF NO ERROR
7376  033326                                ERROR                   ;REPORT STACKED ERROR
7377  033326  104460                                                                TRAP    C$ERROR
7378  033330                                ESCAPE  TST             ;SKIP TO END OF TEST
7379  033330  104410                                                                TRAP    C$ESCAPE
7380  033332  000306                                                                .WORD   L10032-.
7381
7382  033334  004537  010350                JSR     R5,RXCHAR       ;READ & CHK SECOND CRC BYTE; RCV DATA1
7383  033340  000035                        035
7384  033342  000000                        0
7385  033344  000010                        8.
7386  033346  103003                        BCC     .+8.            ;BR IF NO ERROR
7387  033350                                ERROR                   ;REPORT STACKED ERROR
7388  033350  104460                                                                TRAP    C$ERROR
7389  033352                                ESCAPE  TST             ;SKIP TO END OF TEST
7390  033352  104410                                                                TRAP    C$ESCAPE
7391  033354  000264                                                                .WORD   L10032-.
7392
7393                                ;-------------------------------------------------------------
7394                                ; TRANSMIT THE BULK OF DATA OUT OF TABLE 'PATK'
7395                                ;-------------------------------------------------------------
7396  033356  012702  003012                MOV     #PATK,R2        ;SET UP TABLE POINTER
7397  033362  112237  033420        5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED RX CHARACTER
7398  033366  116237  000003  033400        MOVB    3(R2),10$       ;SET UP TRANSMIT CHARACTER
7399
7400  033374  004537  010136                JSR     R5,TXCHAR       ;LOAD A CHARACTER
7401  033400  000000                10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
7402  033402  000000                        0
7403  033404  103003                        BCC     .+8.            ;BR IF NO ERROR
7404  033406                                ERROR                   ;REPORT STACKED ERROR
7405  033406  104460                                                                TRAP    C$ERROR
7406  033410                                ESCAPE  TST             ;SKIP TO END OF TEST
7407  033410  104410                                                                TRAP    C$ESCAPE
7408  033412  000226                                                                .WORD   L10032-.
7409
7410  033414  004537  010350                JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
7411  033420  000000                20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
7412  033422  000000                        0
```

```
7413  033424  000010                    8.
7414  033426  103003                    BCC     .+8.            ;BR IF NO ERROR
7415  033430                            ERROR                   ;REPORT STACKED ERROR
7416  033430  104460                                                            TRAP    C$ERROR
7417  033432                            ESCAPE  TST             ;SKIP TO END OF TEST
7418  033432  104410                                                            TRAP    C$ESCAPE
7419  033434  000204                                                            .WORD   L10032-.
7420
7421  033436  022702  003027            CMP     #PATK+13.,R2    ;CHECK FOR 14TH CHARACTER OF TABLE
7422  033442  001347                    BNE     5$              ;BR IF NOT DONE
7423                                    ;-----------------------------------------------------------
7424  033444  004537  010250            JSR     R5,TXCTRL       ;SET TEOM
7425  033450  000002                    TEOM
7426  033452  000000                    0
7427  033454  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 200(DATA13), RCV 125(DATA14)
7428  033460  000200                    200
7429  033462  000000                    0
7430  033464  000010                    8.
7431  033466  103003                    BCC     .+8.            ;BR IF NO ERROR
7432  033470                            ERROR                   ;REPORT STACKED ERROR
7433  033470  104460                                                            TRAP    C$ERROR
7434  033472                            ESCAPE  TST             ;SKIP TO END OF TEST
7435  033472  104410                                                            TRAP    C$ESCAPE
7436  033474  000144                                                            .WORD   L10032-.
7437
7438  033476  004537  010250            JSR     R5,TXCTRL       ;SET TEOM
7439  033502  000002                    TEOM
7440  033504  000000                    0
7441  033506  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 125(DATA14), RCV 252(DATA15)
7442  033512  000125                    125
7443  033514  000000                    0
7444  033516  000010                    8.
7445  033520  103003                    BCC     .+8.            ;BR IF NO ERROR
7446  033522                            ERROR                   ;REPORT STACKED ERROR
7447  033522  104460                                                            TRAP    C$ERROR
7448  033524                            ESCAPE  TST             ;SKIP TO END OF TEST
7449  033524  104410                                                            TRAP    C$ESCAPE
7450  033526  000112                                                            .WORD   L10032-.
7451
7452  033530  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 252(DATA15), RCV 000(DATA16)
7453  033534  000252                    252
7454  033536  000000                    0
7455  033540  000010                    8.
7456  033542  103003                    BCC     .+8.            ;BR IF NO ERROR
7457  033544                            ERROR                   ;REPORT STACKED ERROR
7458  033544  104460                                                            TRAP    C$ERROR
7459  033546                            ESCAPE  TST             ;SKIP TO END OF TEST
7460  033546  104410                                                            TRAP    C$ESCAPE
7461  033550  000070                                                            .WORD   L10032-.
7462
7463  033552  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 000(DATA16), RCV FIRST CRC BYTE
7464  033556  100000                    RXERR!000
7465  033560  000001                    RERCHK
7466  033562  000010                    8.
7467  033564  103003                    BCC     .+8.            ;BR IF NO ERROR
7468  033566                            ERROR                   ;REPORT STACKED ERROR
```

```
033566  104460                                                                  TRAP    C$ERROR
033570                          ESCAPE  TST             ;SKIP TO END OF TEST
033570  104410                                                                  TRAP    C$ESCAPE
033572  000046                                                                  .WORD   L10032-.

033574  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
033600  000231                  231
033602  000000                  0
033604  000010                  8.
033606  103003                  BCC     .+8.            ;BR IF NO ERROR
033610                          ERROR                   ;REPORT STACKED ERROR
033610  104460                                                                  TRAP    C$ERROR
033612                          ESCAPE  TST             ;SKIP TO END OF TEST
033612  104410                                                                  TRAP    C$ESCAPE
033614  000024                                                                  .WORD   L10032-.

033616  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
033622  000176                  176
033624  000000                  0
033626  000010                  8.
033630  103003                  BCC     .+8.            ;BR IF NO ERROR
033632                          ERROR                   ;REPORT STACKED ERROR
033632  104460                                                                  TRAP    C$ERROR
033634                          ESCAPE  TST             ;SKIP TO END OF TEST
033634  104410                                                                  TRAP    C$ESCAPE
033636  000002                                                                  .WORD   L10032-.
033640                  ENDTST
033640                                                                  L10032:
033640  104401                                                                  TRAP    C$ETST
```

```
                        .SBTTL  HARDWARE PARAMETER CODING SECTION


                        ;////////////////////////////////////////////////////////////////////////
                        ;/ THE HARDWARE 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.
                        ;////////////////////////////////////////////////////////////////////////

033642                          BGNHRD
033642  000027                                                  .WORD L10033-L$HARD/2
033644                                                  L$HARD::

033644                          GPRMA   ADDRES,0,0,160020,177776,YES
033644  000031                                                  .WORD   T$CODE
033646  033722                                                  .WORD   ADDRES
033650  160020                                                  .WORD   T$LOLIM
033652  177776                                                  .WORD   T$HILIM
033654                          GPRMA   VECTOR,2,0,0,674,YES
033654  001031                                                  .WORD   T$CODE
033656  033750                                                  .WORD   VECTOR
033660  000000                                                  .WORD   T$LOLIM
033662  000674                                                  .WORD   T$HILIM
033664                          GPRMD   PRIRTY,4,0,7000,4,7,YES
033664  002032                                                  .WORD   T$CODE
033666  034001                                                  .WORD   PRIRTY
033670  007000                                                  .WORD   7000
033672  000004                                                  .WORD   T$LOLIM
033674  000007                                                  .WORD   T$HILIM
033676                          GPRMD   BDTY.M,12,0,7,0,2,YES
033676  005032                                                  .WORD   T$CODE
033700  034032                                                  .WORD   BDTY.M
033702  000007                                                  .WORD   7
033704  000000                                                  .WORD   T$LOLIM
033706  000002                                                  .WORD   T$HILIM
033710                          GPRMD   TCON.M,14,0,7,0,4,YES
033710  006032                                                  .WORD   T$CODE
033712  034115                                                  .WORD   TCON.M
033714  000007                                                  .WORD   7
033716  000000                                                  .WORD   T$LOLIM
033720  000004                                                  .WORD   T$HILIM

033722                          ENDHRD
                                                                .EVEN
033722                                                  L10033:

                                .NLIST  BEX
033722  042504  044526  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
033750  042504  044526  042503  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
034001     104  053105  041511  PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
034032  047502  051101  020104  BDTY.M: .ASCIZ  /BOARD TYPE (0=M8064, 1=M8053-V.35, 2=M8053-EIA) : /
034115     124  051125  040516  TCON.M: .ASCII  /TURNAROUND CONNECTOR TYPE -/<15><12>
```

```
        034152  030050  044075  031063          .ASCII  /(0=H3254&H3255, 1=INTEGRAL MODEM CABLE, 2=EIA CABLE,/<15><12>
        034240  031440  053075  031456          .ASCIZ  / 3=V.35 CABLE, 4=NONE) : /
                                          .LIST   BEX
  7549                                    .EVEN
```

```
                        .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.
                        ;////////////////////////////////////////////////////////////////////////

7562  034272                BGNSFT
7563  034272  000000                                          .WORD L10034-L$SOFT/2
7564  034274                                          L$SOFT::
7565
7566  034274                ENDSFT
7567                                                          .EVEN
7568  034274                                          L10034:
```

CVDMEA.P11 10-DEC-80 09:16

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

                                .SBTTL ****** PATCH AREA FOR DEBUG ******

7569
7570
7571  034274                    PATCH:
7572            034474                  .=.+200
7573  034474    000240                  NOP
7574  034476    000240                  NOP
7575  034500    000240                  NOP
7576                            ;*************************************
7577
7578                            .SBTTL  'ENDMOD'' STATEMENT
7579
7580  034502                            ENDMOD
7581
7582                            .SBTTL  'LASTAD'' STATEMENT & END OF PROGRAM
7583  034502                            LASTAD
7584                                                                    .EVEN
7585  034502    000000                                                  .WORD   0
7586  034504    000000                                                  .WORD   0
7587  034506                    L$LAST::
7588
7589            000001          .END
```

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- USER SYMBOLS

```
ABC    = 000160          1265#
ADDRES   033722          7517    7548#
ADR    = 000020 G        1197#
AD.HIT   023206          4531    4566#
AD.OK    023202          4555    4560#
APA    = 000200          1321#
APAD   = 100000          1330#
ASSEMB= 000010            955
A1       032520          6978    7129#
BADDAT   002366          1716#
BDATA    002326          1699#   2352*   2355    2358    2632*   2735*   2736*   3503*   3504*   3521*   3522*   3915    3964
                         4009    4080    4142    4798*   4799*   4881*   5006*   5035*   5064*   5093*   5122*   5164*   5193*
                         6925*   7018*
BDRATE   002474          1763#   4502*
BDTY.M   034032          7533    7548#
BIT0   = 000001 G        1170#   1215    1227    1269    1278    1297    1326    1352    1368    1386    1402    1544    1558
                         1580    1633    2766    2811    2856    2901    2946    2989    3036    3103    4468
BIT00  = 000001 G        1159#   1170
BIT01  = 000002 G        1158#   1169
BIT02  = 000004 G        1157#   1168
BIT03  = 000010 G        1156#   1167
BIT04  = 000020 G        1155#   1166
BIT05  = 000040 G        1154#   1165
BIT06  = 000100 G        1153#   1164
BIT07  = 000200 G        1152#   1163
BIT08  = 000400 G        1151#   1162
BIT09  = 001000 G        1150#   1161
BIT1   = 000002 G        1169#   1268    1296    1326    1352    1367    1385    1401    1543    1557    1579    1632    3060
BIT10  = 002000 G        1149#   1274    1302    1337    1338    1339
BIT11  = 004000 G        1148#   1273    1301    1334
BIT12  = 010000 G        1147#   1333
BIT13  = 020000 G        1146#   1332    1642
BIT14  = 040000 G        1145#   1331    1635    1641
BIT15  = 100000 G        1144#   1272    1300    1330    1629    1640    5823    6083    6308
BIT2   = 000004 G        1168#   1267    1295    1326    1352    1366    1384    1400    1532    1557    1578
BIT3   = 000010 G        1167#   1266    1294    1325    1351    1365    1383    1399    1532    1557    1577
BIT4   = 000020 G        1166#   1214    1265    1324    1350    1364    1382    1398    1532    1556    1576
BIT5   = 000040 G        1165#   1265    1323    1349    1363    1381    1397    1527    1555    1575    3645
BIT6   = 000100 G        1164#   1226    1265    1322    1349    1362    1380    1396    1517    1555    1574    3665
BIT7   = 000200 G        1163#   1225    1233    1264    1293    1321    1349    1361    1379    1395    1517    1555    1568
                         1590    1638    3482    3483    3654
BIT8   = 000400 G        1162#   1276    1304    1335    1336    1337    1338
BIT9   = 001000 G        1161#   1275    1303    1336    1337    1636
BOE    = 000400 G        1201#
BRDTYP   002470          1761#   3263    3274    3282    3292    4500*   5818    6078    6303    6523    6708    6976    7003
                         7164
BSEL0    002416          1730#   2294    4538    4544
BSEL1    002420          1733#   2090*   2295
BSEL10   002436          1743#   2302
BSEL11   002440          1745#   2303
BSEL12   002442          1746#   2304
BSEL13   002444          1748#   2305
BSEL14   002446          1749#   2306
BSEL15   002450          1751#   2307
BSEL16   002452          1752#   2308
BSEL17   002454          1754#   2309    4490
```

```
BSEL2   002422  1734# 2097  2133* 2140  2180* 2187  2252* 2259  2296  4855* 4861
BSEL3   002424  1736# 2297
BSEL4   002426  1737# 2298
BSEL5   002430  1739# 2299
BSEL6   002432  1740# 2155  2300  4876  4883  4896  4909  4922  4936  4949  4962  4975  4988
                5001  5017  5030  5046  5059  5075  5088  5104  5117  5133  5146  5159  5175
                5188  5204  5217
BSEL7   002434  1742# 2301  5006  5035  5064  5093  5122  5164  5193
BSR0    002202  1666# 2294* 4164
BSR1    002204  1668# 2295* 4163
BSR10   002222  1681# 2302* 4198
BSR11   002224  1682# 2303* 4197
BSR12   002226  1683# 2304* 4196
BSR13   002230  1684# 2305* 4195
BSR14   002232  1685# 2306* 4215
BSR15   002234  1686# 2307* 4214
BSR16   002236  1687# 2308* 4213
BSR17   002240  1688# 2309* 4212
BSR2    002206  1670# 2296* 4162
BSR3    002210  1672# 2297* 4161
BSR4    002212  1674# 2298* 4181
BSR5    002214  1676# 2299* 4180
BSR6    002216  1678# 2300* 4179
BSR7    002220  1680# 2301* 4178
CARIER= 000100  1396# 6922  6924  7007  7009  7013  7015  7047  7078
CA1CTL= 000001  1558#
CA2CTL= 000016  1557#
CB1CTL= 000020  1556#
CB2CTL= 000340  1555#
CHKTSO  007074  3098#
CHPTYP  002400  1721# 4417*
CKLPBK  007532  3242# 4767  6895
CKRACT  005654  2806# 3647  3701  3728  3760  3771
CKRDA   006154  2896# 3656  3667  3704  3720  3725
CKROR   006454  2984#
CKRSA   006314  2941#
CKSEOM  006614  3031# 3707
CKTACT  005514  2761# 3213  3379  3431  3757  3768
CKTBMT  006014  2851# 3196  3210  3388  3395  3436  3443  4843
CKUSTS  005410  2726# 3179  4829
CRC0S = 000400  1335# 6715
CRC16 = 001400  1336# 4824  5826  7171
CTS   = 000010  1399# 6922  6924  6943  6962  7007  7009  7013  7015  7035  7066
C$AU  = 000052   955# 4610
C$AUTO= 000061   955# 4564
C$BRK = 000022   955#
C$BSEG= 000004   955#
C$BSUB= 000002   955# 6904  6983
C$CEFG= 000045   955#
C$CLCK= 000062   955#
C$CLEA= 000012   955# 4582
C$CLOS= 000035   955#
C$CLP1= 000006   955#
C$CVEC= 000036   955# 4553
C$DCLN= 000044   955#
C$DODU= 000051   955# 4558
```

```
C$DRPT= 000024   955#
C$DU  = 000053   955#  4597
C$EDIT= 000003   955#  1027
C$ERDF= 000055   955#  4803  4868  4888  4901  4914  4928  4941  4954  4967  4980  4993  5009
                 5022  5038  5051  5067  5080  5096  5109  5125  5138  5151  5167  5180  5196
                 5209  5223  5234  6929  6948  6967  7021  7040  7052  7071  7083  7102  7121
C$ERHR= 000056   955#
C$ERRO= 000060   955#  4636  4646  4668  4680  4693  4703  4716  4727  4738  4833  4847  5830
                 5840  5850  5864  5874  5884  5894  5904  5913  5924  5940  5951  5968  5982
                 5993  6004  6015  6026  6043  6091  6101  6111  6125  6135  6145  6154  6165
                 6181  6192  6205  6216  6226  6237  6248  6259  6268  6315  6325  6335  6349
                 6359  6369  6378  6389  6405  6416  6429  6440  6450  6461  6472  6483  6492
                 6534  6556  6566  6576  6586  6596  6609  6620  6635  6646  6657  6668  6719
                 6741  6751  6761  6771  6781  6793  6804  6818  6829  6840  6851  7175  7194
                 7204  7214  7224  7234  7243  7254  7264  7275  7289  7300  7314  7325  7335
                 7346  7356  7367  7377  7388  7405  7416  7433  7447  7458  7469  7480  7491
C$ERSF= 000054   955#
C$ERSO= 000057   955#
C$ESCA= 000010   955#  4638  4648  4670  4682  4695  4705  4718  4729  4740  4808  4835  4849
                 4873  4893  4906  4919  4933  4946  4959  4972  4985  4998  5014  5027  5043
                 5056  5072  5085  5101  5114  5130  5143  5156  5172  5185  5201  5214  5228
                 5832  5842  5852  5866  5876  5886  5896  5906  5915  5926  5942  5953  5970
                 5984  5995  6006  6017  6028  6045  6093  6103  6113  6127  6137  6147  6156
                 6167  6183  6194  6207  6218  6228  6239  6250  6261  6270  6317  6327  6337
                 6351  6361  6371  6380  6391  6407  6418  6431  6442  6452  6463  6474  6485
                 6494  6536  6558  6568  6578  6588  6598  6611  6622  6637  6648  6659  6670
                 6721  6743  6753  6763  6773  6783  6795  6806  6820  6831  6842  6853  6934
                 6953  7026  7045  7057  7076  7088  7107  7177  7196  7206  7216  7226  7236
                 7245  7256  7266  7277  7291  7302  7316  7327  7337  7348  7358  7369  7379
                 7390  7407  7418  7435  7449  7460  7471  7482  7493
C$ESEG= 000005   955#
C$ESUB= 000003   955#  6974  7128
C$ETST= 000001   955#  4744  5242  6049  6274  6498  6683  6866  7132  7497
C$EXIT= 000032   955#  4771  6899
C$GETB= 000026   955#
C$GETW= 000027   955#
C$GMAN= 000043   955#
C$GPHR= 000042   955#  4476
C$GPLO= 000030   955#
C$GPRI= 000040   955#
C$INIT= 000011   955#  4509
C$INLP= 000020   955#
C$MANI= 000050   955#
C$MEM = 000031   955#
C$MSG = 000023   955#  3857  3896  3930  3974  4019  4054  4089  4124
C$OPEN= 000034   955#
C$PNTB= 000014   955#  3853  3910  3920  3926  3943  3949  3959  3969  3988  3994  4004  4014
                 4033  4039  4049  4068  4075  4085  4103  4109  4119  4267
C$PNTF= 000017   955#  3304  3315  3326  3344
C$PNTS= 000016   955#
C$PNTX= 000015   955#  3873  3889  4158  4168  4175  4185  4192  4202  4209  4219  4234  4244
                 4251  4261  4283  4293  4300  4310  4317  4327  4334  4344  4360  4370  4377
                 4387
C$QIO = 000377   955#
C$RDBU= 000007   955#
C$REFG= 000047   955#  4434  4441  4448  4455
```

```
C$RESE= 000033          955#  4505  4594
C$REVI= 000003          955#  1026
C$RFLA= 000021          955#
C$RPT = 000025          955#
C$SEFG= 000046          955#
C$SPRI= 000041          955#
C$SVEC= 000037          955#  4534
C$TPRI= 000013          955#
DDCMP = 040000         1331#  4632  4824  5826  6087  6311  7171
DEVMAP  002404         1723#  4464* 4482*
DEVPTR  002406         1724#  4468* 4480* 4482  4483*
DFPTBL  002150 G       1103#
DIAGMC= 000000          955
DOTBMT= 000007         1243#
DTR   = 000020         1382#  2614  2617  2690  3152  3155  4817  4820  7093
DTRL  = 000000         1387#
D.BUG = 000000         1644#
EF.CON= 000036 G       1177#  4454
EF.NEW= 000035 G       1178#  4447
EF.PWR= 000034 G       1179#
EF.RES= 000037 G       1176#  4440
EF.STA= 000040 G       1175#  4433
EIAV35= 000002         1632#  3258
EM100   015536         3112   3812#
EM101   015556         3124   3812#
EM14    014324         3569   3812#
EM16    014340         3559   3812#
EM2     014214         2638   3812#
EM25    014360         2366   3812#
EM28    014406         3056   3628   3812#
EM29    014427         3045   3638   3812#
EM3     014263         2105   3812#
EM30    014444         3080   3605   3812#
EM31    014465         3069   3615   3812#
EM34    014502         3510   3812#  5011  5040  5069  5098  5127  5169  5198
EM35    014530         3536   3812#
EM36    014551         3546   3812#  5140
EM39    014566         3582   3812#
EM4     014307         2149   2196   2267  3812#  4870
EM40    014610         3592   3812#
EM54    014626         3497   3812#
EM68    014650         2742   3812#
EM69    014677         2775   3812#
EM70    014715         2786   3812#
EM71    014737         2820   3812#
EM72    014755         2831   3812#
EM73    014777         2865   3812#  4890  4903  4916  4943  4956  4969  4982
EM74    015014         2876   3812#
EM75    015035         2910   3812#  4995  5024  5053  5082  5111  5153  5182  5211
EM76    015051         2921   3812#
EM77    015071         2955   3812#
EM78    015105         2966   3812#
EM79    015125         3812#  4805
EM80    015161         3812#  4930
EM81    015201         3812#  5225
EM82    015225         3812#  5236
```

```
EM83    015262          3812#   6931    7023
EM84    015311          3812#   6950    7042
EM85    015326          3812#   6969    7073
EM86    015342          3812#   7054
EM87    015363          3812#   7085
EM88    015403          3812#   7104
EM89    015426          3812#   7123
EM90    015450          2998    3812#
EM91    015501          3009    3812#
ENDCOD  026227          4783    4811    5787#
ENDEMB  012116          3812#   3850    3923    4264
ENDIT   023066          4457    4506#
ENDPAT  003064          2022#
ENTRAN  011772          3751#   6039    6264    6488
EPATX   002740          1922#   6197    6421
ERRBLK  002200 G        1660#   2106*   2150*   2197*   2268*   2367*   2639*   2743*   2776*   2787*   2821*   2832*   2866*
                        2877*   2911*   2922*   2956*   2967*   2999*   3010*   3046*   3057*   3070*   3081*   3113*   3125*
                        3498*   3511*   3537*   3547*   3560*   3570*   3583*   3593*   3606*   3616*   3629*   3639*
ERRFLG  002350          1709#
ERRMSG  002176 G        1659#   2105*   2149*   2196*   2267*   2366*   2638*   2742*   2775*   2786*   2820*   2831*   2865*
                        2876*   2910*   2921*   2955*   2966*   2998*   3009*   3045*   3056*   3069*   3080*   3112*   3124*
                        3497*   3510*   3536*   3546*   3559*   3569*   3582*   3592*   3605*   3615*   3628*   3638*
ERRNBR  002174 G        1658#   2104*   2148*   2195*   2266*   2365*   2637*   2741*   2774*   2785*   2819*   2830*   2864*
                        2875*   2909*   2920*   2954*   2965*   2997*   3008*   3044*   3055*   3068*   3079*   3111*   3123*
                        3496*   3509*   3535*   3545*   3558*   3568*   3581*   3591*   3604*   3614*   3627*   3637*
ERROR1  002376          1720#   4418*
ERRTYP  002172 G        1657#   2103*   2147*   2194*   2265*   2364*   2636*   2740*   2773*   2784*   2818*   2829*   2863*
                        2874*   2908*   2919*   2953*   2964*   2996*   3007*   3043*   3054*   3067*   3078*   3110*   3122*
                        3495*   3508*   3534*   3544*   3557*   3567*   3580*   3590*   3603*   3613*   3626*   3636*
ERR10   020364 G        2639    2743    3511    3936#   5012    5041    5070    5099    5128    5141    5170    5199
ERR11   020540 G        3981#   6932    7024
ERR11$  022032          4016    4121    4276#
ERR12   020714 G        2776    2787    2821    2832    2866    2877    2911    2922    2956    2967    2999    3010    3046
                        3057    3070    3081    3113    3125    3498    3537    3547    3560    3570    3583    3593    3606
                        3616    3629    3639    4026#   4891    4904    4917    4944    4957    4970    4983    4996    5025
                        5054    5083    5112    5154    5183    5212
ERR12$  022364          3971    4051    4353#
ERR13   021030 G        4061#   4806
ERR14   021146 G        4096#   4931    5226    6951    6970    7043    7055    7074    7086    7105    7124
ERR3    020072 G        3847#   5237
ERR4    020120 G        2106    2150    2197    2268    3862#   4871
ERR4$   021306          3848    4152#
ERR5$   021634          3893    4227#
ERR7A   020244 G        2367    3901#
EVL     = 000004 G      1195#
EVRC    = 002400        1338#   6311
EXADD   = 000020        1350#
EXCON   = 000010        1351#
EXECUT= 000005          1242#   4855    4864
E$END   = 002100         955#
E$LOAD= 000035           955#   1050
FLGCA1= 000002          1579#
FLGCA2= 000001          1580#
FLGCB1= 000020          1576#
FLGCB2= 000010          1577#
FLGIRQ= 000200          1568#
```

```
FLGSR = 000004          1578#
FLGT1 = 000100          1574#
FLGT2 = 000040          1575#
FMT10   012560          3812#
FMT10A  012634          3812#
FMT11   012655          3812#  4241  4258
FMT12   012674          3812#
FMT13   012703          3812#
FMT14   012751          3812#
FMT15   012766          3812#  3907
FMT15A  013020          3812#
FMT16   013072          3812#
FMT16A  013115          3812#
FMT17   013157          3812#
FMT17A  013236          3812#
FMT17B  013317          3812#
FMT17C  013372          3812#
FMT19   013452          3341   3812#
FMT2    012126          3812#
FMT21   013503          3812#  3940  3985  4030  4065  4100
FMT22   013513          3812#  3946  3991  4036  4106
FMT23   013535          3812#  3966  4011  4082
FMT24   013614          3812#  4280  4357
FMT25   013627          3812#  4290  4324  4367
FMT26   013657          3812#  4307  4341  4384
FMT27   013712          3812#  3956  4001  4046  4116
FMT28   013721          3812#
FMT29   013755          3812#  4297  4314  4331  4374
FMT3    012163          3812#  3917
FMT30   013762          3301   3812#
FMT31   014017          3312   3812#
FMT32   014065          3323   3812#
FMT39   014117          3812#
FMT4    012247          3812#  4155  4231
FMT4A   012305          3812#  4165  4199
FMT4B   012340          3812#  4172  4189  4206  4248
FMT4C   012345          3812#  4182  4216
FMT40   014150          3812#  4072
FMT5    012400          3812#  3870
FMT5A   012443          3812#  3886
FMT7    012530          3812#
FRSTIM  002370          1717#  4420  4429*
FSAU  = 000015           955#  4607  4609
FSAUTO= 000020           955#  4527  4563
FSBGN = 000040           955#   958  3847  3862  3901  3936  3981  4026  4061  4096  4400  4413  4527
                        4577  4591  4607  4627  4638  4648  4670  4682  4695  4705  4718  4729  4740
                        4743  4764  4771  4808  4835  4849  4873  4893  4906  4919  4933  4946  4959
                        4972  4985  4998  5014  5027  5043  5056  5072  5085  5101  5114  5130  5143
                        5156  5172  5185  5201  5214  5228  5241  5813  5832  5842  5852  5866  5876
                        5886  5896  5906  5915  5926  5942  5953  5970  5984  5995  6006  6017  6028
                        6045  6048  6074  6093  6103  6113  6127  6137  6147  6156  6167  6183  6194
                        6207  6218  6228  6239  6250  6261  6270  6273  6299  6317  6327  6337  6351
                        6361  6371  6380  6391  6407  6418  6431  6442  6452  6463  6474  6485  6494
                        6497  6519  6536  6558  6568  6578  6588  6598  6611  6622  6637  6648  6659
                        6670  6682  6704  6721  6743  6753  6763  6773  6783  6795  6806  6820  6831
                        6842  6853  6865  6893  6899  6903  6934  6953  6973  6982  7026  7045  7057
```

```
                    7076    7088    7107    7127    7131    7160    7177    7196    7206    7216    7226    7236    7245
                    7256    7266    7277    7291    7302    7316    7327    7337    7348    7358    7369    7379    7390
                    7407    7418    7435    7449    7460    7471    7482    7493    7496    7512    7563    7581
F$CLEA= 000007       955#    4577    4581
F$DU  = 000016       955#    4591    4596
F$END = 000041       955#     958    3858    3897    3931    3975    4020    4055    4090    4125    4510    4565    4583
                    4598    4611    4627    4638    4648    4670    4682    4695    4705    4718    4729    4740    4743
                    4745    4764    4771    4808    4835    4849    4873    4893    4906    4919    4933    4946    4959
                    4972    4985    4998    5014    5027    5043    5056    5072    5085    5101    5114    5130    5143
                    5156    5172    5185    5201    5214    5228    5241    5243    5813    5832    5842    5852    5866
                    5876    5886    5896    5906    5915    5926    5942    5953    5970    5984    5995    6006    6017
                    6028    6045    6048    6050    6074    6093    6103    6113    6127    6137    6147    6156    6167
                    6183    6194    6207    6218    6228    6239    6250    6261    6270    6273    6275    6299    6317
                    6327    6337    6351    6361    6371    6380    6391    6407    6418    6431    6442    6452    6463
                    6474    6485    6494    6497    6499    6519    6536    6558    6568    6578    6588    6598    6611
                    6622    6637    6648    6659    6670    6682    6684    6704    6721    6743    6753    6763    6773
                    6783    6795    6806    6820    6831    6842    6853    6865    6867    6893    6899    6903    6934
                    6953    6973    6975    6982    7026    7045    7057    7076    7088    7107    7127    7129    7131
                    7133    7160    7177    7196    7206    7216    7226    7236    7245    7256    7266    7277    7291
                    7302    7316    7327    7337    7348    7358    7369    7379    7390    7407    7418    7435    7449
                    7460    7471    7482    7493    7496    7498    7547    7569    7581
F$HARD= 000004       955#    7512    7545
F$HW  = 000013       955#    1101    1117
F$INIT= 000006       955#    4413    4508
F$JMP = 000050       955#    4771    6899
F$MOD = 000000       955#     958    7581
F$MSG = 000011       955#    3847    3856    3862    3895    3901    3929    3936    3973    3981    4018    4026    4053
                    4061    4088    4096    4123
F$PROT= 000021       955#    4400    4405
F$PWR = 000017       955#
F$RPT = 000012       955#
F$SEG = 000003       955#
F$SOFT= 000005       955#    7563    7567
F$SRV = 000010       955#
F$SUB = 000002       955#    6904    6973    6983    7127
F$SW  = 000014       955#    1126    1131
F$TEST= 000001       955#    4628    4743    4765    5241    5814    6048    6075    6273    6300    6497    6520    6682
                    6705    6865    6894    7131    7161    7496
GDATA   002324      1698#   2100*   2144*   2191*   2262*   2358    2630*   2631*   2734*   3501*   3502*   3519*   3520*
                    3863*   3865    3885    3916    3965    4010    4081    4141    4796*   4797*   4864*   4880*   5005*
                    5034*   5063*   5092*   5121*   5163*   5192*   6924*   7009*   7015*
GETBSR  004024      2294#
GETPRM  022740      4458    4472#   4481
GETURS  004360      2390#   4353
GETVRS  004460      2421#   4276
GETWSR  004166      2099    2143    2190    2261    2315#   4863    5230
GOODAT  002364      1715#
G$CNTO= 000200       955#
G$DELM= 000372       955#
G$DISP= 000003       955#
G$EXCP= 000400       955#
G$HILI= 000002       955#
G$LOLI= 000001       955#
G$NO  = 000000       955#
G$OFFS= 000400       955#    7516    7521    7526    7532    7538
G$OFSI= 000376       955#    7516    7521    7526    7532    7538
```

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- USER SYMBOLS

```
G$PRMA= 000001            955#  7516  7521
G$PRMD= 000002            955#  7526  7532  7538
G$PRML= 000000            955#
G$RADA= 000140            955#
G$RADB= 000000            955#
G$RADD= 000040            955#
G$RADL= 000120            955#
G$RADO= 000020            955#  7516  7521  7526  7532  7538
G$XFER= 000004            955#
G$YES = 000010            955#  7516  7521  7526  7532  7538
HDX   = 000004           1384#  4690
HELP  = 000000            946#   980  1068  1092  2069
HOE   = 100000 G          1208#
IBE   = 010000 G          1205#
IDLE  = 000010            1325#
IDLES = 004000            1334#  4824  5826  7171
IDU   = 000040 G          1198#
IEI   = 000001            1215#
IEO   = 000020            1214#
IER   = 020000 G          1206#
INIDMV  005376            2712#  4629  4766  5815  6075  6300  6521  6706  7162
INITRN  007234            3148#  4631  5825  6086  6310  6529  6714  7170
INITT1  004534            2459#
INITT2  004734            2546#
INTFLG  002346            1707#
INTGRL= 000001            1633#  3256  4768
INTSC = 000200            1590#
ISR   = 000100 G          1199#
IXE   = 004000 G          1204#
I$AU  = 000041             955#  4607#  4611#
I$AUTO= 000041             955#  4527#  4565#
I$CLN = 000041             955#  4577#  4583#
I$DU  = 000041             955#  4591#  4598#
I$HRD = 000041            7512#  7547#
I$INIT= 000041             955#  4413#  4510#
I$MOD = 000041             955#   958#  7581#
I$MSG = 000041             955#  3847#  3858#  3862#  3897#  3901#  3931#  3936#  3975#  3981#  4020#  4026#  4055#
                          4061#  4090#  4096#  4125#
I$PROT= 000040             955#  4400#
I$PTAB= 000041             955#
I$PWR = 000041             955#
I$RPT = 000041             955#
I$SEG = 000041             955#  4627   4764  5813  6074  6299  6519  6704  6893  6903  6982  7160
I$SETU= 000041             955#
I$SFT = 000041            7563#  7569#
I$SRV = 000041             955#
I$SUB = 000041             955#  4627  4764  5813  6074  6299  6519  6704  6893  6903#  6934  6953  6973#
                          6975#  6982#  7026  7045  7057  7076  7088  7107  7127#  7129#  7160
I$TST = 000041             955#  4627#  4638  4648  4670  4682  4695  4705  4716  4729  4740  4743#  4745#
                          4764#  4771  4808  4835  4849  4873  4893  4906  4919  4933  4946  4959  4972
                          4985  4998  5014  5027  5043  5056  5072  5085  5101  5114  5130  5143  5156
                          5172  5185  5201  5214  5228  5241#  5243#  5813#  5832  5842  5852  5866  5876
                          5886  5896  5906  5915  5926  5942  5953  5970  5984  5995  6006  6017  6028
                          6045  6048#  6050#  6074#  6093  6103  6113  6127  6137  6147  6156  6167  6183
                          6194  6207  6218  6228  6239  6250  6261  6270  6273#  6275#  6299#  6317  6327
                          6337  6351  6361  6371  6380  6391  6407  6418  6431  6442  6452  6463  6474
```

```
                    6485   6494   6497#  6499#  6519#  6536   6558   6568   6578   6588   6598   6611   6622
                    6637   6648   6659   6670   6682#  6684#  6704#  6721   6743   6753   6763   6773   6783
                    6795   6806   6820   6831   6842   6853   6865#  6867#  6893#  6899   6903   6982   7131#
                    7133#  7160#  7177   7196   7206   7216   7226   7236   7245   7256   7266   7277   7291
                    7302   7316   7327   7337   7348   7358   7369   7379   7390   7407   7418   7435   7449
                    7460   7471   7482   7493   7496#  7498#
J$JMP = 000167        955#
LOADAT  002362       1714#
LOE   = 040000 G     1207#
LOGDEV  002334       1702#  4466*  4473*  4475   4557
LOT   = 000010 G     1196#
LUSWI1  002464       1759#  4498*
LUSWI2  002466       1760#  4499*
LU1MOD  002000 G      958#
L$ACP   002110 G     1057#
L$APT   002036 G     1015#
L$AU    023222 G     1042   4607#
L$AUT   002070 G     1041#
L$AUTO  023070 G     1058   4527#
L$CCP   002106 G     1055#
L$CLEA  023214 G     1056   4577#
L$CO    002032 G     1011#
L$DEPO  002011 G      993#
L$DESC  003304 G     1048   2057#
L$DESP  002076 G     1047#
L$DEVP  002060 G     1033#
L$DISP  002124 G     1018   1081#
L$DLY   002116 G     1063#
L$DTP   002040 G     1017#
L$DTYP  002034 G     1013#
L$DU    023216 G     1044   4591#
L$DUT   002072 G     1043#
L$DVTY  003264 G     1034   2045#
L$EF    002052 G     1028#
L$ENVI  002044 G     1021#
L$ERRT  002172 G     1052   1656#
L$ETP   002102 G     1051#
L$EXP1  002046 G     1023#
L$EXP4  002064 G     1037#
L$EXP5  002066 G     1039#
L$HARD  033644 G     1000   7512   7513#
L$HIME  002120 G     1065#
L$HPCP  002016 G      999#
L$HPTP  002022 G     1003#
L$HW    002150 G     1004   1101   1102#
L$ICP   002104 G     1053#
L$INIT  022554 G     1054   4413#
L$LADP  002026 G     1007#
L$LAST  034506 G     1008   7587#
L$LOAD  002100 G     1049#
L$LUN   002074 G     1045#
L$MREV  002050 G     1025#
L$NAME  002000 G      982#
L$PRIO  002042 G     1019#
L$PROT  022546 G     1060   4400#
L$PRT   002112 G     1059#
```

```
L$REPP  002062 G        1035#
L$REV   002010 G         991#
L$SOFT  034274 G        7563    7564#
L$SPC   002056 G        1031#
L$SPCP  002020 G        1001#
L$SPTP  002024 G        1005#
L$STA   002030 G        1009#
L$SW    002172 G        1126    1127#
L$TEST  002114 G        1061#
L$TIML  002014 G         997#
L$UNIT  002012 G         995#
L10000  002170          1101    1117#
L10001  002172          1126    1131#
L10002  020116          3856#
L10003  020242          3895#
L10004  020362          3929#
L10005  020536          3973#
L10006  020712          4018#
L10007  021026          4053#
L10010  021144          4088#
L10011  021260          4123#
L10013  023066          4508#
L10014  023204          4563#
L10015  023214          4581#
L10016  023220          4596#
L10017  023222          4609#
L10020  023522          4639    4649    4671    4683    4696    4706    4719    4730    4741    4743#
L10021  025400          4772    4809    4836    4850    4874    4894    4907    4920    4934    4947    4960    4973    4986
                        4999    5015    5028    5044    5057    5073    5086    5102    5115    5131    5144    5157    5173
                        5186    5202    5215    5229    5241#
L10022  027104          5833    5843    5853    5867    5877    5887    5897    5907    5916    5927    5943    5954    5971
                        5985    5996    6007    6018    6029    6046    6048#
L10023  027646          6094    6104    6114    6128    6138    6148    6157    6168    6184    6195    6208    6219    6229
                        6240    6251    6262    6271    6273#
L10024  030410          6318    6328    6338    6352    6362    6372    6381    6392    6408    6419    6432    6443    6453
                        6464    6475    6486    6495    6497#
L10025  031072          6537    6559    6569    6579    6589    6599    6612    6623    6638    6649    6660    6671    6682#
L10026  031554          6722    6744    6754    6764    6774    6784    6796    6807    6821    6832    6843    6854    6865#
L10027  032520          6900    7131#
L10030  032026          6935    6954    6973#
L10031  032516          7027    7046    7058    7077    7089    7108    7127#
L10032  033640          7178    7197    7207    7217    7227    7237    7246    7257    7267    7278    7292    7303    7317
                        7328    7338    7349    7359    7370    7380    7391    7408    7419    7436    7450    7461    7472
                        7483    7494    7496#
L10033  033722          7512    7546#
L10034  034274          7563    7568#
MCLR  = 000100          1226#   1228    2090    2100
MCODE   025402          4775    4787    5243#
MDMRDY= 000040          1397#   6922    6924    7007    7009    7013    7015    7097    7116
MLWRI   003722          2227    2252#
MPCSR   002416          1732#   3938    3983    4028    4063    4098    4486
MPIVEC  002456          1756#   4494*
MPOVEC  002460          1757#   4495*   4496*
MPRIOR  002462          1758#   4497*
MRDY  = 000200          1233#   2097    2140    2187    2259    4861
MREQ  = 000001          1227#   1228    2090    2100
```

```
MSTCLR  003352          2090#   2712    6906    6985
NCRACT= 020000          1642#   6654    6665    6837    6848
NCTBMT= 000200          1638#   3385
NEWLIN  012123          3812#
NEWST   022720          4450    4465#
NFCRDA= 040000          1641#   6665    6848
NOCHK = 003400          1337#   4632
NOCRDA= 100000          1640#   4713    4724    4735
NOLOOP= 001000          1636#   5816    5822    6076    6082    6301    6307    6522    6527    6707    6712    7163    7168
NORXEN= 040000          1635#
NULCLK= 000200          1379#
OVRC  = 002000          1339#   6087
OSAPTS= 000000           955#   1009
OSAU  = 000001           955#    979#   1041
OSBGNR= 000000           955#   1035
OSBGNS= 000000           955#   1001
OSDU  = 000001           955#    979#   1043
OSERRT= 000001           955#    979#   1051
OSGNSW= 000000           955#   1005
OSPOIN= 000001           955#    979#   1067
OSSETU= 000000           955#    995    7585
PALENB= 000001          1544#
PATCH   034274          7571#
PATE    002576          1812#
PATF    002606          1823#   2620
PATG    002616          1834#
PATI    002741          1925#
PATJ    003002          1961#
PATK    003012          1972#   7396    7421
PATL    003033          1992#
PATQ    003054          2012#
PATX    002645          1860#   5931    5956    6172    6396
PBLENB= 000002          1543#
PCR   = 120407          1345#   3168
PCSARH= 120405          1317#   3162    4827
PCSARL= 120404          1310#   3158    4822
PNT   = 001000 G        1202#
PRESET= 000001          1386#   2614    3152    4817
PRI   = 002000 G        1203#
PRIOR   002342          1705#
PRIRTY  034001          7527    7548#
PRI00 = 000000 G        1191#
PRI01 = 000040 G        1190#
PRI02 = 000100 G        1189#
PRI03 = 000140 G        1188#
PRI04 = 000200 G        1187#
PRI05 = 000240 G        1186#
PRI06 = 000300 G        1185#
PRI07 = 000340 G        1184#
PROTO = 000100          1322#
PSTACK  002340          1704#   4415*
RABGA = 000004          1267#   3573    3575    3585
RAMADR= 001000          1630#   4776    4788    4854
RCVBUF  003064          2028#
RCVDAT= 000002          1401#   6921    7006    7012
RCV1ST  011624          3695#   5909    6150    6374    6605    6789    7239
```

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- USER SYMBOLS

```
RDA    = 000200  1361# 2903  2914  3716
RDSRH  = 120401  1261# 2987  3034  3473
RDSRL  = 120400  1255# 3476
READ     003454  2132# 2353  2394  2399  2424
READI    003566  2178# 2481  2568  2622  2727  2763  2808  2853  2898  2943  2986  3033  3100
                 3202  3472  3475  3486  3713  4698  4790  6918  6940  6959  6999  7032  7063
                 7094  7113
REDBYT   002356  1712#
REDDAT   002516  1777#
REDLOC=  000001  1238# 2133  2144  2180  2191
REDPAG=  000003  1240#
REGNUM   002336  1703# 2628* 2629* 2733* 2762* 2807* 2852* 2897* 2942* 2985* 3032* 3099* 3491*
                 3500* 3518* 3523* 3902  3951  3996  4041  4071  4111  4795* 4879* 4925* 5004*
                 5033* 5062* 5091* 5120* 5162* 5191* 5220* 6916* 6995*
REG0     002526  1780# 5814* 6031* 6520* 6673* 6705* 6856* 7161*
REG1     002530  1781#
REG2     002532  1782#
REG3     002534  1783#
REG4     002536  1784#
REG5     002540  1785#
REG6     002542  1786#
REG7     002544  1787#
REOM   = 000002  1268# 3062  3073  3596  3598  3608
RERCHK=  000001  1278# 3524  6000  6161  6188  6212  6233  6244  6255  6385  6412  6436  6457
                 6468  6479  6664  6847  7342  7465
RERR   = 000200  1264# 3527  3529  3539
RETADR   002354  1711#
RING   = 000200  1395# 6922  6924  7006  7013  7015
ROR    = 000010  1266# 299'  3002  3550  3552  3562
RSA    = 000020  1364# 2948  2959
RSOM   = 000001  1269# 3038  3049  3619  3621  3631
RSTCHK   005104  2608#
RTSND  = 000010  1383# 2614  2617  2690  3152  3155  4690  4817  4820  6939  7031
RUN    = 000200  1225# 1228  2090  2100
RXABGA=  002000  1274#
RXACT  = 000040  1363# 2813  2824
RXCHAR   010350  3469# 4710  4721  4732  5918  5945  5962  5976  5987  5998  6009  6020  6159
                 6186  6210  6231  6242  6253  6383  6410  6434  6455  6466  6477  6614  6629
                 6640  6651  6662  6798  6812  6823  6834  6845  7248  7269  7283  7294  7319
                 7340  7361  7382  7410  7427  7441  7452  7463  7474  7485
RXDL   = 000007  1352# 3480  6088  6312
RXEN   = 000100  1380# 3188  3218  4677  4690
RXEOM  = 001000  1275# 6663  6846
RXERR  = 100000  1272# 5999  7341  7464
RXOR   = 004000  1273#
RXSOM  = 000400  1276# 6615  6799
SAVE4    002372  1718# 4422* 4426
SAVE6    002374  1719# 4423* 4427
SAVLEN   002402  1722# 3165* 3166* 3480  4419*
SCRACH   002332  1701#
SECAD  = 000020  1324#
SECADR=  010000  1333#
SEL0     002416  1731# 2315
SEL10    002436  1744# 2319
SEL12    002442  1747# 2320
SEL14    002446  1750# 2321
```

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SEL16   002452          1753#  2322
SEL2    002422          1735#  2316
SEL4    002426          1738#  2132*  2179*  2225*  2250*  2317   4853*  4854*
SEL6    002432          1741#  2202   2226*  2251*  2318
SETVIA  005304          2678#  2713
SFPTBL  002172 G        1128#
SFR   = 000001          1368#
SPEED = 000020          1398#  6921   7006   7012
SRMODE= 000034          1532#
STALL   004356          2377#
STARES  002412          1726#  3298   3309   3320   3337   4461*  4467*
STARST  022710          4436   4443   4460#
STEPLU  012072          3200   3391   3439   3660   3710   3766   3789#  4685
STRIP = 000040          1323#
STRIPS= 020000          1332#  4632   4824   5826   6087   6311   7171
STRTML= 000301          1228#
STUREG  004250          2344#
SUBRPC  002344          1706#  4416*
SVCGBL= 000000           955#   958   965#   982   991   993   995   997   999  1001  1003  1005  1007
                        1009  1011  1013  1015  1017  1019  1021  1023  1025  1028  1031  1033  1035
                        1037  1039  1041  1043  1045  1047  1049  1051  1053  1055  1057  1059  1061
                        1063  1065  1081  1102  1103  1127  1128  1656  2045  2057  3847  3862  3901
                        3936  3981  4026  4061  4096  4400  4413  4527  4577  4591  4607  7513  7564
                        7587#  7588
SVCINS= 000001           955#   962#   983   984   985   986   987   988   989   990   992   994   996
                         998  1000  1002  1004  1006  1008  1010  1012  1014  1016  1018  1020  1022
                        1024  1026  1027  1029  1030  1032  1034  1036  1038  1040  1042  1044  1046
                        1048  1050  1052  1054  1056  1058  1060  1062  1064  1066  1080  1082  1083
                        1084  1085  1086  1087  1088  1089  1090  1101  1126  2046  2049  2058  2065
                        2103  2104  2105  2106  2147  2148  2149  2150  2194  2195  2196  2197  2265
                        2266  2267  2268  2364  2365  2366  2367  2636  2637  2638  2639  2740  2741
                        2742  2743  2773  2774  2775  2776  2784  2785  2786  2787  2818  2819  2820
                        2821  2829  2830  2831  2832  2863  2864  2865  2866  2874  2875  2876  2877
                        2908  2909  2910  2911  2919  2920  2921  2922  2953  2954  2955  2956  2964
                        2965  2966  2967  2996  2997  2998  2999  3007  3008  3009  3010  3043  3044
                        3045  3046  3054  3055  3056  3057  3067  3068  3069  3070  3078  3079  3080
                        3081  3110  3111  3112  3113  3122  3123  3124  3125  3301  3302  3303  3304
                        3305  3312  3313  3314  3315  3316  3323  3324  3325  3326  3327  3340  3341
                        3342  3343  3344  3345  3495  3496  3497  3498  3508  3509  3510  3511  3534
                        3535  3536  3537  3544  3545  3546  3547  3557  3558  3559  3560  3567  3568
                        3569  3570  3580  3581  3582  3583  3590  3591  3592  3593  3603  3604  3605
                        3606  3613  3614  3615  3616  3626  3627  3628  3629  3636  3637  3638  3639
                        3850  3851  3852  3853  3854  3857  3869  3870  3871  3872  3873  3874  3884
                        3885  3886  3887  3888  3889  3890  3896  3905  3906  3907  3908  3909  3910
                        3911  3914  3915  3916  3917  3918  3919  3920  3921  3923  3924  3925  3926
                        3927  3930  3938  3939  3940  3941  3942  3943  3944  3946  3947  3948  3949
                        3950  3954  3955  3956  3957  3958  3959  3960  3963  3964  3965  3966  3967
                        3968  3969  3970  3974  3983  3984  3985  3986  3987  3988  3989  3991  3992
                        3993  3994  3995  3999  4000  4001  4002  4003  4004  4005  4008  4009  4010
                        4011  4012  4013  4014  4015  4019  4028  4029  4030  4031  4032  4033  4034
                        4036  4037  4038  4039  4040  4044  4045  4046  4047  4048  4049  4050  4054
                        4063  4064  4065  4066  4067  4068  4069  4071  4072  4073  4074  4075  4076
                        4079  4080  4081  4082  4083  4084  4085  4086  4089  4098  4099  4100  4101
                        4102  4103  4104  4106  4107  4108  4109  4110  4114  4115  4116  4117  4118
                        4119  4120  4124  4153  4154  4155  4156  4157  4158  4159  4161  4162  4163
                        4164  4165  4166  4167  4168  4169  4171  4172  4173  4174  4175  4176  4178
```

```
4179  4180  4181  4182  4183  4184  4185  4186  4188  4189  4190  4191  4192
4193  4195  4196  4197  4198  4199  4200  4201  4202  4203  4205  4206  4207
4208  4209  4210  4212  4213  4214  4215  4216  4217  4218  4219  4220  4229
4230  4231  4232  4233  4234  4235  4237  4238  4239  4240  4241  4242  4243
4244  4245  4247  4248  4249  4250  4251  4252  4254  4255  4256  4257  4258
4259  4260  4261  4262  4264  4265  4266  4267  4268  4278  4279  4280  4281
4282  4283  4284  4286  4287  4288  4289  4290  4291  4292  4293  4294  4296
4297  4298  4299  4300  4301  4303  4304  4305  4306  4307  4308  4309  4310
4311  4313  4314  4315  4316  4317  4318  4320  4321  4322  4323  4324  4325
4326  4327  4328  4330  4331  4332  4333  4334  4335  4337  4338  4339  4340
4341  4342  4343  4344  4345  4355  4356  4357  4358  4359  4360  4361  4363
4364  4365  4366  4367  4368  4369  4370  4371  4373  4374  4375  4376  4377
4378  4380  4381  4382  4383  4384  4385  4386  4387  4388  4433  4434  4436
4440  4441  4443  4447  4448  4450  4454  4455  4457  4475  4476  4477  4479
4505  4509  4530  4531  4532  4533  4534  4535  4552  4553  4557  4558  4564
4582  4594  4597  4610  4636  4638  4639  4646  4648  4649  4668  4670  4671
4680  4682  4683  4693  4695  4696  4703  4705  4706  4716  4718  4719  4727
4729  4730  4738  4740  4741  4744  4771  4772  4803  4804  4805  4806  4808
4809  4833  4835  4836  4847  4849  4850  4868  4869  4870  4871  4873  4874
4888  4889  4890  4891  4893  4894  4901  4902  4903  4904  4906  4907  4914
4915  4916  4917  4919  4920  4928  4929  4930  4931  4933  4934  4941  4942
4943  4944  4946  4947  4954  4955  4956  4957  4959  4960  4967  4968  4969
4970  4972  4973  4980  4981  4982  4983  4985  4986  4993  4994  4995  4996
4998  4999  5009  5010  5011  5012  5014  5015  5022  5023  5024  5025  5027
5028  5038  5039  5040  5041  5043  5044  5051  5052  5053  5054  5056  5057
5067  5068  5069  5070  5072  5073  5080  5081  5082  5083  5085  5086  5096
5097  5098  5099  5101  5102  5109  5110  5111  5112  5114  5115  5125  5126
5127  5128  5130  5131  5138  5139  5140  5141  5143  5144  5151  5152  5153
5154  5156  5157  5167  5168  5169  5170  5172  5173  5180  5181  5182  5183
5185  5186  5196  5197  5198  5199  5201  5202  5209  5210  5211  5212  5214
5215  5223  5224  5225  5226  5228  5229  5234  5235  5236  5237  5242  5830
5832  5833  5840  5842  5843  5850  5852  5853  5864  5866  5867  5874  5876
5877  5884  5886  5887  5894  5896  5897  5904  5906  5907  5913  5915  5916
5924  5926  5927  5940  5942  5943  5951  5953  5954  5968  5970  5971  5982
5984  5985  5993  5995  5996  6004  6006  6007  6015  6017  6018  6026  6028
6029  6043  6045  6046  6049  6091  6093  6094  6101  6103  6104  6111  6113
6114  6125  6127  6128  6135  6137  6138  6145  6147  6148  6154  6156  6157
6165  6167  6168  6181  6183  6184  6192  6194  6195  6205  6207  6208  6216
6218  6219  6226  6228  6229  6237  6239  6240  6248  6250  6251  6259  6261
6262  6268  6270  6271  6274  6315  6317  6318  6325  6327  6328  6335  6337
6338  6349  6351  6352  6359  6361  6362  6369  6371  6372  6378  6380  6381
6389  6391  6392  6405  6407  6408  6416  6418  6419  6429  6431  6432  6440
6442  6443  6450  6452  6453  6461  6463  6464  6472  6474  6475  6483  6485
6486  6492  6494  6495  6498  6534  6536  6537  6556  6558  6559  6566  6568
6569  6576  6578  6579  6586  6588  6589  6596  6598  6599  6609  6611  6612
6620  6622  6623  6635  6637  6638  6646  6648  6649  6657  6659  6660  6668
6670  6671  6683  6719  6721  6722  6741  6743  6744  6751  6753  6754  6761
6763  6764  6771  6773  6774  6781  6783  6784  6793  6795  6796  6804  6806
6807  6818  6820  6821  6829  6831  6832  6840  6842  6843  6851  6853  6854
6866  6899  6900  6904  6929  6930  6931  6932  6934  6935  6948  6949  6950
6951  6953  6954  6967  6968  6969  6970  6974  6983  7021  7022  7023  7024
7026  7027  7040  7041  7042  7043  7045  7046  7052  7053  7054  7055  7057
7058  7071  7072  7073  7074  7076  7077  7083  7084  7085  7086  7088  7089
7102  7103  7104  7105  7107  7108  7121  7122  7123  7124  7128  7132  7175
7177  7178  7194  7196  7197  7204  7206  7207  7214  7216  7217  7224  7226
7227  7234  7236  7237  7243  7245  7246  7254  7256  7257  7264  7266  7267
```

```
                    7275  7277  7278  7289  7291  7292  7300  7302  7303  7314  7316  7317  7325
                    7327  7328  7335  7337  7338  7346  7348  7349  7356  7358  7359  7367  7369
                    7370  7377  7379  7380  7388  7390  7391  7405  7407  7408  7416  7418  7419
                    7433  7435  7436  7447  7449  7450  7458  7460  7461  7469  7471  7472  7480
                    7482  7483  7491  7493  7494  7497  7512  7516  7517  7518  7519  7521  7522
                    7523  7524  7526  7527  7528  7529  7530  7532  7533  7534  7535  7536  7538
                    7539  7540  7541  7542  7545  7563  7567  7584  7585  7586
SVCSUB= 000001        955#  964#  6903  6982
SVCTAG= 000001        955#  966#  1073  1077  1117  1131  3856  3895  3929  3973  4018  4053  4088
                    4123  4508  4563  4581  4596  4609  4743  5241  6048  6273  6497  6682  6865
                    6973  7127  7131  7496  7546  7568
SVCTST= 000001        955#  963#  4627  4764  5813  6074  6299  6519  6704  6893  7160
SWPBOT= 121000       1624#
SWPDDC= 121400       1625#
SYNCH = 000226       1311#  3195  4632  5826  5836  5846  6087  6097  6107  6243  6254  6311  6321
                    6331  6467  6478  7171
SSLSYM= 010000         955#  1118#  1132#  3857#  3896#  3930#  3974#  4019#  4054#  4089#  4124#  4509#  4564#
                    4582#  4597#  4610#  4744#  5242#  6049#  6274#  6498#  6683#  6866#  6974#  7128#  7132#
                    7497#  7547#  7569#
TAB   = 000004       1295#
TBMT  = 000100       1362#  2858  2869  3205
TCCHEK= 100000       1629#  3243  6896
TCON.M  034115       7539  7548#
TDATA   002322       1697#  2349
TDSRH = 120403       1290#  3191  3426  4838
TDSRL = 120402       1284#  3194  3373  4841
TDSRNR  002575       1809#
TEOM  = 000002       1296#  5960  5974  6602  6626  6787  6810  7281  7425  7439
TERR  = 000200       1293#
TGA   = 000010       1294#
TIMFLG  002352       1710#
TM    = 000004       1400#  6922  6924
TMP0    002546       1790#  4536*  4554  4566*
TMP1    002550       1791#
TMP2    002552       1792#
TMP3    002554       1793#
TMP4    002556       1794#
TMP5    002560       1795#
TMP6    002562       1796#
TMP7    002564       1797#
TSO   = 000010       1365#  3105  3117
TSOM  = 000001       1297#  3192  4652  4656  4839  6035  6201  6222  6425  6446  6540  6543  6546
                    6677  6725  6728  6731  6860  7181  7184
TSTCON  002472       1762#  3245  3250  3266  3268  3277  3280  3286  4501*  5820  6080  6305  6525
                    6710  6979  7166
TSTNUM  002414       1727#  3340  4765*  6894*
TTLOOP= 000002       1385#  3188  3218  3752  3765  4677  4690  6915  6939  6958
TXAB  = 002000       1302#
TXACT = 000004       1366#  2768  2779
TXCHAR  010136       3368#  4663  5835  5845  5859  5869  5879  5889  5899  5935  6096  6106  6120
                    6130  6140  6176  6320  6330  6344  6354  6364  6400  6551  6561  6571  6581
                    6591  6736  6746  6756  6766  6776  7189  7199  7209  7219  7229  7259  7309
                    7330  7351  7372  7400
TXCTRL  010250       3421#  4651  4655  4659  5855  5959  5973  6034  6116  6200  6221  6340  6424
                    6445  6539  6542  6545  6548  6601  6625  6676  6724  6727  6730  6733  6786
                    6809  6859  7180  7183  7186  7280  7305  7424  7438
```

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TXDL  = 000340          1349#   3480    6088    6312
TXEN  = 000040          1381#   3188    3218    4677    4690
TXEOM = 001000          1303#
TXERR = 100000          1300#
TXGA  = 004000          1301#
TXSOM = 000400          1304#
TXTMLT  017744          3821#   3884
TXTML0  017154          3812#   3821
TXTML1  017160          3812#   3821
TXTML2  017174          3812#   3821
TXTML3  017211          3812#   3821
TXTML4  017233          3812#   3821
TXTML5  017254          3812#   3821
TXTML6  017304          3812#   3821
TXTML7  017316          3812#   3821
TXTNP   017500          3812#   3833
TXTNPT  020030          3834#
TXTNP0  017505          3812#   3834
TXTNP1  017515          3812#   3834
TXTNP2  017525          3812#   3834
TXTNP3  017535          3812#   3834
TXTNP4  017552          3812#   3834
TXTNP5  017567          3812#   3834
TXTNP6  017604          3812#   3834
TXTNP7  017620          3812#   3834
TXTNP8  017634          3812#   3834
TXTNUL  017152          3812#
TXTUR   017650          3812#   3906    3955    4045
TXTURT  020052          3837#   3905    3954    4044
TXTUR0  017663          3812#   3837
TXTUR1  017671          3812#   3837
TXTUR2  017677          3812#   3837
TXTUR3  017705          3812#   3837
TXTUR4  017713          3812#   3837
TXTUR5  017722          3812#   3837
TXTUR6  017731          3812#   3837
TXTUR7  017735          3812#   3837
TXTVR   017353          3812#   3825    4000    4115
TXTVRA  017451          3812#   3829
TXTVRB  017454          3812#   3829
TXTVRC  017460          3812#   3829
TXTVRD  017464          3812#   3829
TXTVRE  017470          3812#   3829
TXTVRF  017474          3812#   3829
TXTVRT  017766          3826#   3999    4114
TXTVR0  017371          3812#   3826
TXTVR1  017375          3812#   3826
TXTVR2  017401          3812#   3826
TXTVR3  017406          3812#   3826
TXTVR4  017413          3812#   3826
TXTVR5  017420          3812#   3826
TXTVR6  017425          3812#   3826
TXTVR7  017432          3812#   3826
TXTVR8  017437          3812#   3829
TXTVR9  017444          3812#   3829
TXT1    015602          3812#   4153
```

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TXT10   016402          3812#
TXT11   016422          3812#
TXT11A  016474          3812#
TXT11B  016532          3812#
TXT12   016602          3812#   3939    3984    4029    4064    4099
TXT13   016630          3812#   4356
TXT14   016645          3812#   4355
TXT15   016703          3812#   4373
TXT16   016745          3812#   4279
TXT17   016760          3812#   4278
TXT18   017015          3812#   4296
TXT19   017056          3812#   4313
TXT2    015640          3812#   4171
TXT2A   015702          3812#   4188
TXT2B   015741          3812#   4205
TXT20   017112          3812#   4330
TXT3    016004          3812#   4154
TXT4    016034          3812#   4229
TXT4A   016074          3812#   4247
TXT5    016135          3812#
TXT6    016137          3812#   4230
TXT7    016162          3812#
TXT7A   016252          3812#
TXT8    016342          3812#
TXT9    016362          3812#
TXU   = 000002          1367#   3489
T$ARGC= 000005           983#    984#    985#    986#    987#    988#   3301#   3305    3312#   3316    3323#   3327    3340#
                        3345    3850#   3854    3869#   3874    3884#   3890    3905#   3911    3914#   3921    3923#   3927
                        3938#   3944    3946#   3950    3954#   3960    3963#   3970    3983#   3989    3991#   3995    3999#
                        4005    4008#   4015    4028#   4034    4036#   4040    4044#   4050    4063#   4069    4071#   4076
                        4079#   4086    4098#   4104    4106#   4110    4114#   4120    4153#   4159    4161#   4169    4171#
                        4176    4178#   4186    4188#   4193    4195#   4203    4205#   4210    4212#   4220    4229#   4235
                        4237#   4245    4247#   4252    4254#   4262    4264#   4268    4278#   4284    4286#   4294    4296#
                        4301    4303#   4311    4313#   4318    4320#   4328    4330#   4335    4337#   4345    4355#   4361
                        4363#   4371    4373#   4378    4380#   4388
T$CODE= 006032          7516#   7521#   7526#   7532#   7538#
T$ERRN= 000113           955#   4804#   4869#   4889#   4902#   4915#   4929#   4942#   4955#   4968#   4981#   4994#   5010#
                        5023#   5039#   5052#   5068#   5081#   5097#   5110#   5126#   5139#   5152#   5168#   5181#   5197#
                        5210#   5224#   5235#   6930#   6949#   6968#   7022#   7041#   7053#   7072#   7084#   7103#   7122#
T$EXCP= 000000          7516#   7520    7521#   7525    7526#   7531    7532#   7537    7538#   7543
T$FLAG= 000040          4638#   4648#   4670#   4682#   4695#   4705#   4718#   4729#   4740#   4771#   4808#   4835#   4849#
                        4873#   4893#   4906#   4919#   4933#   4946#   4959#   4972#   4985#   4998#   5014#   5027#   5043#
                        5056#   5072#   5085#   5101#   5114#   5130#   5143#   5156#   5172#   5185#   5201#   5214#   5228#
                        5832#   5842#   5852#   5866#   5876#   5886#   5896#   5906#   5915#   5926#   5942#   5953#   5970#
                        5984#   5995#   6006#   6017#   6028#   6045#   6093#   6103#   6113#   6127#   6137#   6147#   6156#
                        6167#   6183#   6194#   6207#   6218#   6228#   6239#   6250#   6261#   6270#   6317#   6327#   6337#
                        6351#   6361#   6371#   6380#   6391#   6407#   6418#   6431#   6442#   6452#   6463#   6474#   6485#
                        6494#   6536#   6558#   6568#   6578#   6588#   6598#   6611#   6622#   6637#   6648#   6659#   6670#
                        6721#   6743#   6753#   6763#   6773#   6783#   6795#   6806#   6820#   6831#   6842#   6853#   6899#
                        6934#   6953#   7026#   7045#   7057#   7076#   7088#   7107#   7177#   7196#   7206#   7216#   7226#
                        7236#   7245#   7256#   7266#   7277#   7291#   7302#   7316#   7327#   7337#   7348#   7358#   7369#
                        7379#   7390#   7407#   7418#   7435#   7449#   7460#   7471#   7482#   7493#
T$GMAN= 000000           955#
T$HILI= 000004          7516#   7519    7521#   7524    7526#   7530    7532#   7536    7538#   7542
T$LAST= 000001           955#   7585#
T$LOLI= 000000          7516#   7518    7521#   7523    7526#   7529    7532#   7535    7538#   7541
```

```
T$LSYM= 010000   955#  1118  1132  3857  3896  3930  3974  4019  4054  4089  4124  4509  4564
                4582  4597  4610  4744  5242  6049  6274  6498  6683  6866  6974  7128  7132
                7497  7547  7569
T$LTNO= 000011  7588#
T$NEST= 177777   955#   958#  1101#  1117#  1126#  1131#  3847#  3856#  3862#  3895#  3901#  3929#  3936#
                3973#  3981#  4018#  4026#  4053#  4061#  4088#  4096#  4123#  4400#  4405#  4413#  4508#
                4527#  4563#  4577#  4581#  4591#  4596#  4607#  4609#  4628#  4743#  4765#  5241#  5814#
                6048#  6075#  6273#  6300#  6497#  6520#  6682#  6705#  6865#  6894#  6904#  6973#  6983#
                7127#  7131#  7161#  7496#  7512#  7545#  7563#  7567#  7581#
T$NS0 = 000000    958#  7581
T$NS1 = 000005   1101#  1117  1126#  1131  3847#  3856  3862#  3895  3901#  3929  3936#  3973  3981#
                4018  4026#  4053  4061#  4088  4096#  4123  4400#  4405  4413#  4508  4527#  4563
                4577#  4581  4591#  4596  4607#  4609  4628#  4743  4765#  5241  5814#  6048  6075#
                6273  6300#  6497  6520#  6682  6705#  6865  6894#  7131  7161#  7496  7512#  7545
                7563#  7567
T$NS2 = 000002  6904#  6973  6983#  7127
T$PTNU= 000000   955#
T$SAVL= 177777   955#
T$SEGL= 177777   955#
T$SUBN= 000000   955#  4627#  4764#  5813#  6074#  6299#  6519#  6704#  6893#  6903#  6982#  7160#
T$TAGL= 177777   955#
T$TAGN= 010035   955#  1101#  1126#  3847#  3862#  3901#  3936#  3981#  4026#  4061#  4096#  4400#  4413#
                4527#  4577#  4591#  4607#  4628#  4765#  5814#  6075#  6300#  6520#  6705#  6894#  6904#
                6983#  7161#  7512#  7563#
T$TEMP= 000000  1073#  1077#  1082#  1083#  1084#  1085#  1086#  1087#  1088#  1089#  1090#  1091#  1117#
                1131#  3856#  3895#  3929#  3973#  4018#  4053#  4088#  4123#  4405#  4508#  4563#  4581#
                4596#  4609#  4638#  4639  4648#  4649  4670#  4671  4682#  4683  4695#  4696  4705#
                4706  4718#  4719  4729#  4730  4740#  4741  4743#  4771#  4772  4808#  4809  4835#
                4836  4849#  4850  4873#  4874  4893#  4894  4906#  4907  4919#  4920  4933#  4934
                4946#  4947  4959#  4960  4972#  4973  4985#  4986  4998#  4999  5014#  5015  5027#
                5028  5043#  5044  5056#  5057  5072#  5073  5085#  5086  5101#  5102  5114#  5115
                5130#  5131  5143#  5144  5156#  5157  5172#  5173  5185#  5186  5201#  5202  5214#
                5215  5228#  5229  5241#  5832#  5833  5842#  5843  5852#  5853  5866#  5867  5876#
                5877  5886#  5887  5896#  5897  5906#  5907  5915#  5916  5926#  5927  5942#  5943
                5953#  5954  5970#  5971  5984#  5985  5995#  5996  6006#  6007  6017#  6018  6028#
                6029  6045#  6046  6048#  6093#  6094  6103#  6104  6113#  6114  6127#  6128  6137#
                6138  6147#  6148  6156#  6157  6167#  6168  6183#  6184  6194#  6195  6207#  6208
                6218#  6219  6228#  6229  6239#  6240  6250#  6251  6261#  6262  6270#  6271  6273#
                6317#  6318  6327#  6328  6337#  6338  6351#  6352  6361#  6362  6371#  6372  6380#
                6381  6391#  6392  6407#  6408  6418#  6419  6431#  6432  6442#  6443  6452#  6453
                6463#  6464  6474#  6475  6485#  6486  6494#  6495  6497#  6536#  6537  6558#  6559
                6568#  6569  6578#  6579  6588#  6589  6598#  6599  6611#  6612  6622#  6623  6637#
                6638  6648#  6649  6659#  6660  6670#  6671  6682#  6721#  6722  6743#  6744  6753#
                6754  6763#  6764  6773#  6774  6783#  6784  6795#  6796  6806#  6807  6820#  6821
                6831#  6832  6842#  6843  6853#  6854  6865#  6899#  6900  6934#  6935  6953#  6954
                6973#  7026#  7027  7045#  7046  7057#  7058  7076#  7077  7088#  7089  7107#  7108
                7127#  7131#  7177#  7178  7196#  7197  7206#  7207  7216#  7217  7226#  7227  7236#
                7237  7245#  7246  7256#  7257  7266#  7267  7277#  7278  7291#  7292  7302#  7303
                7316#  7317  7327#  7328  7337#  7338  7348#  7349  7358#  7359  7369#  7370  7379#
                7380  7390#  7391  7407#  7408  7418#  7419  7435#  7436  7449#  7450  7460#  7461
                7471#  7472  7482#  7483  7493#  7494  7496#  7516#  7521#  7526#  7532#  7538#  7545#
                7567#  7581#
T$TEST= 000011   955#  4627#  4764#  5813#  6074#  6299#  6519#  6704#  6893#  6903  6982  7160#  7588
T$TSTM= 177777   955#  3304  3315  3326  3344  3853  3857  3873  3889  3896  3910  3920  3926
                3930  3943  3949  3959  3969  3974  3988  3994  4004  4014  4019  4033  4039
                4049  4054  4068  4075  4085  4089  4103  4109  4119  4124  4158  4168  4175
```

```
                    4185  4192  4202  4209  4219  4234  4244  4251  4261  4267  4283  4293  4300
                    4310  4317  4327  4334  4344  4360  4370  4377  4387  4434  4441  4448  4455
                    4476  4505  4509  4534  4553  4558  4564  4582  4594  4597  4610  4636  4638
                    4646  4648  4668  4670  4680  4682  4693  4695  4703  4705  4716  4718  4727
                    4729  4738  4740  4744  4771  4803  4808  4833  4835  4847  4849  4868  4873
                    4888  4893  4901  4906  4914  4919  4928  4933  4941  4946  4954  4959  4967
                    4972  4980  4985  4993  4998  5009  5014  5022  5027  5038  5043  5051  5056
                    5067  5072  5080  5085  5096  5101  5109  5114  5125  5130  5138  5143  5151
                    5156  5167  5172  5180  5185  5196  5201  5209  5214  5223  5228  5234  5242
                    5830  5832  5840  5842  5850  5852  5864  5866  5874  5876  5884  5886  5894
                    5896  5904  5906  5913  5915  5924  5926  5940  5942  5951  5953  5968  5970
                    5982  5984  5993  5995  6004  6006  6015  6017  6026  6028  6043  6045  6049
                    6091  6093  6101  6103  6111  6113  6125  6127  6135  6137  6145  6147  6154
                    6156  6165  6167  6181  6183  6192  6194  6205  6207  6216  6218  6226  6228
                    6237  6239  6248  6250  6259  6261  6268  6270  6274  6315  6317  6325  6327
                    6335  6337  6349  6351  6359  6361  6369  6371  6378  6380  6389  6391  6405
                    6407  6416  6418  6429  6431  6440  6442  6450  6452  6461  6463  6472  6474
                    6483  6485  6492  6494  6498  6534  6536  6556  6558  6566  6568  6576  6578
                    6586  6588  6596  6598  6609  6611  6620  6622  6635  6637  6646  6648  6657
                    6659  6668  6670  6683  6719  6721  6741  6743  6751  6753  6761  6763  6771
                    6773  6781  6783  6793  6795  6804  6806  6818  6820  6829  6831  6840  6842
                    6851  6853  6866  6899  6904  6929  6934  6948  6953  6967  6974  6983  7021
                    7026  7040  7045  7052  7057  7071  7076  7083  7088  7102  7107  7121  7128
                    7132  7175  7177  7194  7196  7204  7206  7214  7216  7224  7226  7234  7236
                    7243  7245  7254  7256  7264  7266  7275  7277  7289  7291  7300  7302  7314
                    7316  7325  7327  7335  7337  7346  7348  7356  7358  7367  7369  7377  7379
                    7388  7390  7405  7407  7416  7418  7433  7435  7447  7449  7458  7460  7469
                    7471  7480  7482  7491  7493  7497
T$TSTS= 000001        955#  4628#  4765#  5814#  6075#  6300#  6520#  6705#  6894#  7161#
T$$AU  = 010017      4607#  4609
T$$AUT= 010014       4527#  4563
T$$CLE= 010015       4577#  4581
T$$DU  = 010016      4591#  4596
T$$HAR= 010033       7512#  7546
T$$HW  = 010000      1101#  1117
T$$INI= 010013       4413#  4508
T$$MSG= 010011       3847#  3856  3862#  3895  3901#  3929  3936#  3973  3981#  4018  4026#  4053  4061#
                     4088  4096#  4123
T$$PRO= 010012       4400#
T$$SOF= 010034       7563#  7568
T$$SUB= 010031       6904#  6934  6953  6973  6983#  7026  7045  7057  7076  7088  7107  7127
T$$SW  = 010001      1126#  1131
T$$TES= 010032       4628#  4638  4648  4670  4682  4695  4705  4718  4729  4740  4743  4765#  4771
                     4808  4835  4849  4873  4893  4906  4919  4933  4946  4959  4972  4985  4998
                     5014  5027  5043  5056  5072  5085  5101  5114  5130  5143  5156  5172  5185
                     5201  5214  5228  5241  5814#  5832  5842  5852  5866  5876  5886  5896  5906
                     5915  5926  5942  5953  5970  5984  5995  6006  6017  6028  6045  6048  6075#
                     6093  6103  6113  6127  6137  6147  6156  6167  6183  6194  6207  6218  6228
                     6239  6250  6261  6270  6273  6300#  6317  6327  6337  6351  6361  6371  6380
                     6391  6407  6418  6431  6442  6452  6463  6474  6485  6494  6497  6520#  6536
                     6558  6568  6578  6588  6598  6611  6622  6637  6648  6659  6670  6682  6705#
                     6721  6743  6753  6763  6773  6783  6795  6806  6820  6831  6842  6853  6865
                     6894#  6899  7131  7161#  7177  7196  7206  7216  7226  7236  7245  7256  7266
                     7277  7291  7302  7316  7327  7337  7348  7358  7369  7379  7390  7407  7418
                     7435  7449  7460  7471  7482  7493  7496
T.EDF  = 000001      1644#  2103  2147  2194  2265  2364  2636  2740  2773  2784  2818  2829  2863
```

```
                    2874    2908    2919    2953    2964    2996    3007    3043    3054    3067    3078    3110    3122
                    3495    3508    3534    3544    3557    3567    3580    3590    3603    3613    3626    3636
T.EHRD= 000002      1644#
T.ESF = 000000      1644#
T.ESFT= 000003      1644#
T1      023224 G    1082    4627#
T1MODE= 000300      1517#
T2      023524 G    1083    4764#
T2MODE= 000040      1527#
T3      026230 G    1084    5813#
T4      027106 G    1085    6074#
T5      027650 G    1086    6299#
T6      030412 G    1087    6519#
T7      031074 G    1088    6704#
T8      031556 G    1089    6893#
T8.1    031600      6903#
T8.2    032052      6982#
T9      032522 G    1090    7160#
UAM   = 000200 G    1200#
UMAINT= 000001      1402#   6921    7006    7012
UNIT    002410      1725#
UPBITS  002566      1800#
UREGS   002242      1690#   2390    2393*   2396    4363    4364    4365    4366    4380    4381    4382    4383
USTATR= 122000      1357#   2395    2728    2764    2809    2854    2899    2944    3101    3203    3487    3714
USYREG  002476      1767#   2621
USYRT = 120400      1249#   2391    2404    2408
VECTOR  033750      7522    7548#
VIA   = 120000      1371#   2422    2429
VIAACR= 120013      1515#   2482    2491    2569    2578    2692    3171    4642
VIADPA= 120003      1421#   2683    6911    6990
VIADPB= 120002      1409#   2680    6908    6987
VIAIER= 120016      1588#   2478    2565    2698
VIAIFR= 120015      1566#
VIAMS = 120001      1393#
VIAORA= 120017      1608#   2686    6919    6941    6960    7000    7033    7064    7095    7114
VIAORB= 120000      1377#   2613    2616    2689    3151    3154    3187    3764    4676    4689    4816    4819    6914
                    6938    6957    6993    7030    7061    7092    7111
VIAPCR= 120014      1553#   2695
VIASR = 120012      1505#   4699
VIAT1A= 120004      1435#   2504
VIAT1B= 120005      1447#   2509    3793
VIAT1C= 120006      1460#   2495    3174
VIAT1D= 120007      1471#   2500    3177
VIAT2A= 120010      1482#   2582
VIAT2B= 120011      1494#   2587
VREGS   002262      1693#   2421    4286    4287    4288    4289    4303    4304    4305    4306    4320    4321    4322
                    4323    4337    4338    4339    4340
WAIT50  005270      2655#
WRIBYT  002360      1713#
WRILOC= 000002      1239#   2252    2262
WRIPAG= 000004      1241#
WRITE   003700      2225#   2347
WRITEI  003712      2249#   2477    2490    2494    2499    2503    2508    2564    2577    2581    2586    2612    2615
                    2679    2682    2685    2688    2691    2694    2697    3150    3153    3157    3161    3167    3170
                    3173    3176    3186    3190    3193    3372    3425    3763    3792    4641    4675    4688    4779
                    4815    4818    4821    4826    4837    4840    6907    6910    6913    6937    6956    6986    6989
```

```
                        6992    7029    7060    7091    7110
WSR0    002202          1665#   2315*   4240
WSR10   002212          1673#   2319*   4257
WSR12   002214          1675#   2320*   4256
WSR14   002216          1677#   2321*   4255
WSR16   002220          1679#   2322*   4254
WSR2    002204          1667#   2316*   4239
WSR4    002206          1669#   2317*   4238
WSR6    002210          1671#   2318*   4237
XDATA   002330          1700#   3914    3963    4008    4079    4142*   4143*
XORGB   021262          3912    3961    4006    4077    4140#
XYZ   = 000007          1326#
X$ALWA= 000000           955#
X$FALS= 000040           955#
X$OFFS= 000400           955#
X$TRUE= 000020           955#
$E    = 000113          1644#   2102#   2146#   2193#   2264#   2363#   2635#   2739#   2772#   2783#   2817#   2828#   2862#
                        2873#   2907#   2918#   2952#   2963#   2995#   3006#   3042#   3053#   3066#   3077#   3109#   3121#
                        3494#   3507#   3533#   3543#   3556#   3566#   3579#   3589#   3602#   3612#   3625#   3635#   4802#
                        4867#   4887#   4900#   4913#   4927#   4940#   4953#   4966#   4979#   4992#   5008#   5021#   5037#
                        5050#   5066#   5079#   5095#   5108#   5124#   5137#   5150#   5166#   5179#   5195#   5208#   5222#
                        5233#   6928#   6947#   6966#   7020#   7039#   7051#   7070#   7082#   7101#   7120#
$LSTIN= 000001           960#
$LSTTA= 000001           961#
$T    = 000011          1644#   4611#   4745#   5789#   6050#   6275#   6499#   6684#   6867#   7133#
.     = 034506           951#   1690#   1693#   1777#   2028#   2049#   2065#   3813#   4634    4639    4644    4649    4666
                        4671    4678    4683    4691    4696    4701    4706    4714    4719    4725    4730    4736    4741
                        4772    4809    4831    4836    4845    4850    4874    4894    4907    4920    4934    4947    4960
                        4973    4986    4999    5015    5028    5044    5057    5073    5086    5102    5115    5131    5144
                        5157    5173    5186    5202    5215    5229    5788#   5828    5833    5838    5843    5848    5853
                        5862    5867    5872    5877    5882    5887    5892    5897    5902    5907    5911    5916    5922
                        5927    5938    5943    5949    5954    5966    5971    5980    5985    5991    5996    6002    6007
                        6013    6018    6024    6029    6041    6046    6089    6094    6099    6104    6109    6114    6123
                        6128    6133    6138    6143    6148    6152    6157    6163    6168    6179    6184    6190    6195
                        6203    6208    6214    6219    6224    6229    6235    6240    6246    6251    6257    6262    6266
                        6271    6313    6318    6323    6328    6333    6338    6347    6352    6357    6362    6367    6372
                        6376    6381    6387    6392    6403    6408    6414    6419    6427    6432    6438    6443    6448
                        6453    6459    6464    6470    6475    6481    6486    6490    6495    6532    6537    6554    6559
                        6564    6569    6574    6579    6584    6589    6594    6599    6607    6612    6618    6623    6633
                        6638    6644    6649    6655    6660    6666    6671    6717    6722    6739    6744    6749    6754
                        6759    6764    6769    6774    6779    6784    6791    6796    6802    6807    6816    6821    6827
                        6832    6838    6843    6849    6854    6900    6935    6954    7027    7046    7058    7077    7089
                        7108    7173    7178    7192    7197    7202    7207    7212    7217    7222    7227    7232    7237
                        7241    7246    7252    7257    7262    7267    7273    7278    7287    7292    7298    7303    7312
                        7317    7323    7328    7333    7338    7344    7349    7354    7359    7365    7370    7375    7380
                        7386    7391    7403    7408    7414    7419    7431    7436    7445    7450    7456    7461    7467
                        7472    7478    7483    7489    7494    7572#
```

BCOMPL	1#	955#	4435	4442	4449	4456	4478				
BERROR	1#	955#									
BGNAU	1#	955#	4606								
BGNAUT	1#	955#	4526								
BGNCLN	1#	955#	4576								
BGNDU	1#	955#	4590								
BGNHRD	1#	955#	7511								
BGNHW	1#	955#	1100								
BGNINI	1#	955#	4412								
BGNMOD	1#	955#	957								
BGNMSG	1#	955#	3846	3861	3900	3935	3980	4025	4060	4095	
BGNPRO	1#	955#	4399								
BGNPTA	1#	955#									
BGNRPT	1#	955#									
BGNSEG	1#	955#									
BGNSET	1#	955#									
BGNSFT	1#	955#	7562								
BGNSRV	1#	955#									
BGNSUB	1#	955#	6902	6981							
BGNSW	1#	955#	1125								
BGNTST	1#	955#	4627	4764	5813	6074	6299	6519	6704	6893	7160
BNCOMP	1#	955#									
BNERRO	1#	955#									
BREAK	1#	955#									
BRESET	1#	955#	4504	4593							
CKLOOP	1#	955#									
CLOCK	1#	955#									
CLOSE	1#	955#									
CLRVEC	1#	955#	4551								
COMMEN	1#	955#									
DELAY	1#	955#									
DESCRI	1#	955#	2056								
DEVTYP	1#	955#	2044								
DISPAT	1#	955#	1079								
DISPLA	1#	955#									
DOCLN	1#	955#									
DODU	1#	955#	4556								
DORPT	1#	955#									
ENDAU	1#	955#	4608								
ENDAUT	1#	955#	4562								
ENDCLN	1#	955#	4580								
ENDCOM	1#	955#									
ENDDU	1#	955#	4595								
ENDHRD	1#	955#	7544								
ENDHW	1#	955#	1116								
ENDINI	1#	955#	4507								
ENDMOD	1#	955#	7580								
ENDMSG	1#	955#	3855	3894	3928	3972	4017	4052	4087	4122	
ENDPRO	1#	955#	4404								
ENDPTA	1#	955#									
ENDRPT	1#	955#									
ENDSEG	1#	955#									
ENDSET	1#	955#									
ENDSFT	1#	955#	7566								
ENDSRV	1#	955#									
ENDSUB	1#	955#	6972	7126							

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- MACRO NAMES

```
ENDSW       1#    955#  1130
ENDTST      1#    955#  4742  5240  6047  6272  6496  6681  6864  7130  7495
EQUALS      1#    955#  1140
ERRDF       1#    955#  4803  4868  4888  4901  4914  4928  4941  4954  4967  4980  4993  5009  5022
         5038  5051  5067  5080  5096  5109  5125  5138  5151  5167  5180  5196  5209  5223  5234
         6929  6948  6967  7021  7040  7052  7071  7083  7102  7121
ERRHRD      1#    955#
ERROR       1#    955#  4635  4645  4667  4679  4692  4702  4715  4726  4737  4832  4846  5829  5839
         5849  5863  5873  5883  5893  5903  5912  5923  5939  5950  5967  5981  5992  6003  6014
         6025  6042  6090  6100  6110  6124  6134  6144  6153  6164  6180  6191  6204  6215  6225
         6236  6247  6258  6267  6314  6324  6334  6348  6358  6368  6377  6388  6404  6415  6428
         6439  6449  6460  6471  6482  6491  6533  6555  6565  6575  6585  6595  6608  6619  6634
         6645  6656  6667  6718  6740  6750  6760  6770  6780  6792  6803  6817  6828  6839  6850
         7174  7193  7203  7213  7223  7233  7242  7253  7263  7274  7288  7299  7313  7324  7334
         7345  7355  7366  7376  7387  7404  7415  7432  7446  7457  7468  7479  7490
ERRSF       1#    955#
ERRSOF      1#    955#
ERRTBL      1#    955#  1655
ESCAPE      1#    955#  4637  4647  4669  4681  4694  4704  4717  4728  4739  4807  4834  4848  4872
         4892  4905  4918  4932  4945  4958  4971  4984  4997  5013  5026  5042  5055  5071  5084
         5100  5113  5129  5142  5155  5171  5184  5200  5213  5227  5831  5841  5851  5865  5875
         5885  5895  5905  5914  5925  5941  5952  5969  5983  5994  6005  6016  6027  6044  6092
         6102  6112  6126  6136  6146  6155  6166  6182  6193  6206  6217  6227  6238  6249  6260
         6269  6316  6326  6336  6350  6360  6370  6379  6390  6406  6417  6430  6441  6451  6462
         6473  6484  6493  6535  6557  6567  6577  6587  6597  6610  6621  6636  6647  6658  6669
         6720  6742  6752  6762  6772  6782  6794  6805  6819  6830  6841  6852  6933  6952  7025
         7044  7056  7075  7087  7106  7176  7195  7205  7215  7225  7235  7244  7255  7265  7276
         7290  7301  7315  7326  7336  7347  7357  7368  7378  7389  7406  7417  7434  7448  7459
         7470  7481  7492
EXIT        1#    955#  4770  6898
FEQUAL      1#    955#
GEDF     1644#    4801  4866  4886  4899  4912  4926  4939  4952  4965  4978  4991  5007  5020  5036
         5049  5065  5078  5094  5107  5123  5136  5149  5165  5178  5194  5207  5221  5232  6927
         6946  6965  7019  7038  7050  7069  7081  7100  7119
GEHRD    1644#
GESF     1644#
GESFT    1644#
GETBYT      1#    955#
GETPRI      1#    955#
GETWOR      1#    955#
GMANIA      1#    955#
GMANID      1#    955#
GMANIL      1#    955#
GPHARD      1#    955#  4474
GPRMA       1#    955#  7515  7520
GPRMD       1#    955#  7525  7531  7537
GPRML       1#    955#
GTDF     1644#    2101  2145  2192  2263  2362  2634  2738  2771  2782  2816  2827  2861  2872  2906
         2917  2951  2962  2994  3005  3041  3052  3065  3076  3108  3120  3493  3506  3532  3542
         3555  3565  3578  3588  3601  3611  3624  3634
GTHRD    1644#
GTSF     1644#
GTSFT    1644#
HEADER      1#    955#   981
INLOOP      1#    955#
IOSETU      1#    955#
```

```
IOSTAR      1#    955#
KT11        1#    955#
LASTAD      1#    955#   7583
MANUAL      1#    955#
MEMORY      1#    955#
MSG      4611#  4617   4745#  4751   5789#  5795   6050#  6056   6275#  6281   6499#  6505   6684#  6690   6867#
         6873   7133#  7139
MSBYTE      1#    955#   982#   988    989    990
MSCHEC      1#    955#  4771#  6899#
MSCNTO      1#    955#  7516#  7521#  7526#  7532#  7538#
MSCOUN      1#    955#  3301#  3312#  3323#  3340#  3850#  3869#  3884#  3905#  3914#  3923#  3938#  3946#  3954#
         3963#  3983#  3991#  3999#  4008#  4028#  4036#  4044#  4063#  4071#  4079#  4098#  4106#  4114#  4153#
         4161#  4171#  4178#  4188#  4195#  4205#  4212#  4229#  4237#  4247#  4254#  4264#  4278#  4286#  4296#
         4303#  4313#  4320#  4330#  4337#  4355#  4363#  4373#  4380#
MSDATA      1#    955#   982#   991    993    995    997    999   1001   1003   1005   1007   1009   1011   1013
         1015   1017   1019   1021#  1023   1025   1028   1031   1033   1035   1037   1039   1041   1043   1045
         1047   1049   1051   1053   1055   1057   1059   1061   1063   1065   2045#  2057#
MSDECR      1#    955#  1117#  1131#  3856#  3895#  3929#  3973#  4018#  4053#  4088#  4123#  4405#  4508#  4563#
         4581#  4596#  4609#  4743#  5241#  6048#  6273#  6497#  6682#  6865#  6973#  7127#  7131#  7496#  7545#
         7567#  7581#
MSDEFA      1#    955#  7516#  7521#  7526#  7532#  7538#
MSENDE      1#    955#  1117#  1131#  3856#  3895#  3929#  3973#  4018#  4053#  4088#  4123#  4508#  4563#  4581#
         4596#  4609#  4743#  5241#  6048#  6273#  6497#  6682#  6865#  6973#  7127#  7131#  7496#  7545#  7567#
         7581#
MSERRI      1#    955#  4803#  4868#  4888#  4901#  4914#  4928#  4941#  4954#  4967#  4980#  4993#  5009#  5022#
         5038#  5051#  5067#  5080#  5096#  5109#  5125#  5138#  5151#  5167#  5180#  5196#  5209#  5223#  5234#
         6929#  6948#  6967#  7021#  7040#  7052#  7071#  7083#  7102#  7121#
MSESCA      1#    955#  4638#  4639   4648#  4649   4670#  4671   4682#  4683   4695#  4696   4705#  4706   4718#
         4719   4729#  4730   4740#  4741   4808#  4809   4835#  4836   4849#  4850   4873#  4874   4893#  4894
         4906#  4907   4919#  4920   4933#  4934   4946#  4947   4959#  4960   4972#  4973   4985#  4986   4998#
         4999   5014#  5015   5027#  5028   5043#  5044   5056#  5057   5072#  5073   5085#  5086   5101#  5102
         5114#  5115   5130#  5131   5143#  5144   5156#  5157   5172#  5173   5185#  5186   5201#  5202   5214#
         5215   5228#  5229   5832#  5833   5842#  5843   5852#  5853   5866#  5867   5876#  5877   5886#  5887
         5896#  5897   5906#  5907   5915#  5916   5926#  5927   5942#  5943   5953#  5954   5970#  5971   5984#
         5985   5995#  5996   6006#  6007   6017#  6018   6028#  6029   6045#  6046   6093#  6094   6103#  6104
         6113#  6114   6127#  6128   6137#  6138   6147#  6148   6156#  6157   6167#  6168   6183#  6184   6194#
         6195   6207#  6208   6218#  6219   6228#  6229   6239#  6240   6250#  6251   6261#  6262   6270#  6271
         6317#  6318   6327#  6328   6337#  6338   6351#  6352   6361#  6362   6371#  6372   6380#  6381   6391#
         6392   6407#  6408   6418#  6419   6431#  6432   6442#  6443   6452#  6453   6463#  6464   6474#  6475
         6485#  6486   6494#  6495   6536#  6537   6558#  6559   6568#  6569   6578#  6579   6588#  6589   6598#
         6599   6611#  6612   6622#  6623   6637#  6638   6648#  6649   6659#  6660   6670#  6671   6721#  6722
         6743#  6744   6753#  6754   6763#  6764   6773#  6774   6783#  6784   6795#  6796   6806#  6807   6820#
         6821   6831#  6832   6842#  6843   6853#  6854   6934#  6935   6953#  6954   7026#  7027   7045#  7046
         7057#  7058   7076#  7077   7088#  7089   7107#  7108   7177#  7178   7196#  7197   7206#  7207   7216#
         7217   7226#  7227   7236#  7237   7245#  7246   7256#  7257   7266#  7267   7277#  7278   7291#  7292
         7302#  7303   7316#  7317   7327#  7328   7337#  7338   7348#  7349   7358#  7359   7369#  7370   7379#
         7380   7390#  7391   7407#  7408   7418#  7419   7435#  7436   7449#  7450   7460#  7461   7471#  7472
         7482#  7483   7493#  7494
MSESCS      1#    955#  4638#  4648#  4670#  4682#  4695#  4705#  4718#  4729#  4740#  4808#  4835#  4849#  4873#
         4893#  4906#  4919#  4933#  4946#  4959#  4972#  4985#  4998#  5014#  5027#  5043#  5056#  5072#  5085#
         5101#  5114#  5130#  5143#  5156#  5172#  5185#  5201#  5214#  5228#  5832#  5842#  5852#  5866#  5876#
         5886#  5896#  5906#  5915#  5926#  5942#  5953#  5970#  5984#  5995#  6006#  6017#  6028#  6045#  6093#
         6103#  6113#  6127#  6137#  6147#  6156#  6167#  6183#  6194#  6207#  6218#  6228#  6239#  6250#  6261#
         6270#  6317#  6327#  6337#  6351#  6361#  6371#  6380#  6391#  6407#  6418#  6431#  6442#  6452#  6463#
         6474#  6485#  6494#  6536#  6558#  6568#  6578#  6588#  6598#  6611#  6622#  6637#  6648#  6659#  6670#
         6721#  6743#  6753#  6763#  6773#  6783#  6795#  6806#  6820#  6831#  6842#  6853#  6934#  6953#  7026#
```

```
                7045#  7057#  7076#  7088#  7107#  7177#  7196#  7206#  7216#  7226#  7236#  7245#  7256#  7266#  7277#
                7291#  7302#  7316#  7327#  7337#  7348#  7358#  7369#  7379#  7390#  7407#  7418#  7435#  7449#  7460#
                7471#  7482#  7493#
M$EXCP             1#   955#  7516#  7521#  7526#  7532#  7538#
M$EXIT             1#   955#  4771#  4772   6899#  6900
M$EXSE             1#   955#  4771#  6899#
M$EXTJ             1#   955#  4771#  6899#
M$GEN              1#   955#   958#   982#   991#   993#   995#   997#   999#  1001#  1003#  1005#  1007#  1009#  1011#
                1013#  1015#  1017#  1019#  1021#  1023#  1025#  1028#  1031#  1033#  1035#  1037#  1039#  1041#  1043#
                1045#  1047#  1049#  1051#  1053#  1055#  1057#  1059#  1061#  1063#  1065#  1081#  1102#  1103#  1117#
                1127#  1128#  1131#  1656#  2045#  2057#  3847#  3856#  3862#  3895#  3901#  3929#  3936#  3973#  3981#
                4018#  4026#  4053#  4061#  4088#  4096#  4123#  4400#  4413#  4508#  4527#  4563#  4577#  4581#  4591#
                4596#  4607#  4609#  4627#  4743#  4764#  5241#  5813#  6048#  6074#  6273#  6299#  6497#  6519#  6682#
                6704#  6865#  6893#  6903#  6973#  6982#  7127#  7131#  7160#  7496#  7513#  7546#  7564#  7568#  7587#
M$GENB             1#   955#
M$GETS             1#   955#  1117#  1131#  3856#  3895#  3929#  3973#  4018#  4053#  4088#  4123#  4405#  4508#  4563#
                4581#  4596#  4609#  4743#  5241#  6048#  6273#  6497#  6682#  6865#  6973#  7127#  7131#  7496#  7545#
                7567#  7581#
M$GETT             1#   955#  4638#  4648#  4670#  4682#  4695#  4705#  4718#  4729#  4740#  4771#  4808#  4835#  4849#
                4873#  4893#  4906#  4919#  4933#  4946#  4959#  4972#  4985#  4998#  5014#  5027#  5043#  5056#  5072#
                5085#  5101#  5114#  5130#  5143#  5156#  5172#  5185#  5201#  5214#  5228#  5832#  5842#  5852#  5866#
                5876#  5886#  5896#  5906#  5915#  5926#  5942#  5953#  5970#  5984#  5995#  6006#  6017#  6028#  6045#
                6093#  6103#  6113#  6127#  6137#  6147#  6156#  6167#  6183#  6194#  6207#  6218#  6228#  6239#  6250#
                6261#  6270#  6317#  6327#  6337#  6351#  6361#  6371#  6380#  6391#  6407#  6418#  6431#  6442#  6452#
                6463#  6474#  6485#  6494#  6536#  6558#  6568#  6578#  6588#  6598#  6611#  6622#  6637#  6648#  6659#
                6670#  6721#  6743#  6753#  6763#  6773#  6783#  6795#  6806#  6820#  6831#  6842#  6853#  6899#  6934#
                6953#  7026#  7045#  7057#  7076#  7088#  7107#  7177#  7196#  7206#  7216#  7226#  7236#  7245#  7256#
                7266#  7277#  7291#  7302#  7316#  7327#  7337#  7348#  7358#  7369#  7379#  7390#  7407#  7418#  7435#
                7449#  7460#  7471#  7482#  7493#
M$GNGB             1#   955#   958#   982#   991#   993#   995#   997#   999#  1001#  1003#  1005#  1007#  1009#  1011#
                1013#  1015#  1017#  1019#  1021#  1023#  1025#  1028#  1031#  1033#  1035#  1037#  1039#  1041#  1043#
                1045#  1047#  1049#  1051#  1053#  1055#  1057#  1059#  1061#  1063#  1065#  1080#  1081   1101#  1102
                1103   1126#  1127   1128   1656#  2045#  2057#  3847#  3862#  3901#  3936#  3981#  4026#  4061#  4096#
                4400#  4413#  4527#  4577#  4591#  4607#  7512#  7513   7563#  7564   7584#  7587
M$GNIN             1#   955#   982#   983    984    985    986    987    988#   989#   990#   991#   992    993#   994
                995#   996    997#   998    999#  1000   1001#  1002   1003#  1004   1005#  1006   1007#  1008   1009#
                1010   1011#  1012   1013#  1014   1015#  1016   1017#  1018   1019#  1020   1021#  1022   1023#  1024
                1025#  1026   1027   1028#  1029   1030#  1031#  1032   1033#  1034   1035#  1036   1037#  1038   1039#
                1040   1041#  1042   1043#  1044   1045#  1046   1047#  1048   1049#  1050   1051#  1052   1053#  1054
                1055#  1056   1057#  1058   1059#  1060   1061#  1062   1063#  1064   1065#  1066   1080#  1082#  1083#
                1084#  1085#  1086#  1087#  1088#  1089#  1090#  1101#  1126#  2045#  2046   2049   2057#  2058   2065
                2103   2104   2105   2106   2147   2148   2149   2150   2194   2195   2196   2197   2265   2266   2267
                2268   2364   2365   2366   2367   2636   2637   2638   2639   2740   2741   2742   2743   2773   2774
                2775   2776   2784   2785   2786   2787   2818   2819   2820   2821   2829   2830   2831   2832   2863
                2864   2865   2866   2874   2875   2876   2877   2908   2909   2910   2911   2919   2920   2921   2922
                2953   2954   2955   2956   2964   2965   2966   2967   2996   2997   2998   2999   3007   3008   3009
                3010   3043   3044   3045   3046   3054   3055   3056   3057   3067   3068   3069   3070   3078   3079
                3080   3081   3110   3111   3112   3113   3122   3123   3124   3125   3301#  3302#  3303   3304#  3305
                3312#  3313#  3314   3315#  3316   3323#  3324#  3325   3326#  3327   3340#  3341#  3342#  3343   3344#
                3345   3495   3496   3497   3498   3508   3509   3510   3511   3534   3535   3536   3537   3544   3545
                3546   3547   3557   3558   3559   3560   3567   3568   3569   3570   3580   3581   3582   3583   3590
                3591   3592   3593   3603   3604   3605   3606   3613   3614   3615   3616   3626   3627   3628   3629
                3636   3637   3638   3639   3850#  3851#  3852   3853#  3854   3857#  3869#  3870#  3871#  3872   3873#
                3874   3884#  3885#  3886#  3887#  3888   3889#  3890   3896#  3905#  3906#  3907#  3908#  3909   3910#
                3911   3914#  3915#  3916#  3917#  3918#  3919   3920#  3921   3923#  3924#  3925   3926#  3927   3930#
                3938#  3939#  3940#  3941#  3942   3943#  3944   3946#  3947#  3948   3949#  3950   3954#  3955#  3956#
```

3957# 3958 3959# 3960 3963# 3964# 3965# 3966# 3967# 3968 3969# 3970 3974# 3983# 3984#
3985# 3986# 3987 3988# 3989 3991# 3992# 3993 3994# 3995 3999# 4000# 4001# 4002# 4003
4004# 4005 4008# 4009# 4010# 4011# 4012# 4013 4014# 4015 4019# 4028# 4029# 4030# 4031#
4032 4033# 4034 4036# 4037# 4038 4039# 4040 4044# 4045# 4046# 4047# 4048 4049# 4050
4054# 4063# 4064# 4065# 4066# 4067 4068# 4069 4071# 4072# 4073# 4074 4075# 4076 4079#
4080# 4081# 4082# 4083# 4084 4085# 4086 4089# 4098# 4099# 4100# 4101# 4102 4103# 4104
4106# 4107# 4108 4109# 4110 4114# 4115# 4116# 4117# 4118 4119# 4120 4124# 4153# 4154#
4155# 4156# 4157 4158# 4159 4161# 4162# 4163# 4164# 4165# 4166# 4167 4168# 4169 4171#
4172# 4173# 4174 4175# 4176 4178# 4179# 4180# 4181# 4182# 4183# 4184 4185# 4186 4188#
4189# 4190# 4191 4192# 4193 4195# 4196# 4197# 4198# 4199# 4200# 4201 4202# 4203 4205#
4206# 4207# 4208 4209# 4210 4212# 4213# 4214# 4215# 4216# 4217# 4218 4219# 4220 4229#
4230# 4231# 4232# 4233 4234# 4235 4237# 4238# 4239# 4240# 4241# 4242# 4243 4244# 4245
4247# 4248# 4249# 4250 4251# 4252 4254# 4255# 4256# 4257# 4258# 4259# 4260 4261# 4262
4264# 4265# 4266 4267# 4268 4278# 4279# 4280# 4281# 4282 4283# 4284 4286# 4287# 4288#
4289# 4290# 4291# 4292 4293# 4294 4296# 4297# 4298# 4299 4300# 4301 4303# 4304# 4305#
4306# 4307# 4308# 4309 4310# 4311 4313# 4314# 4315# 4316 4317# 4318 4320# 4321# 4322#
4323# 4324# 4325# 4326 4327# 4328 4330# 4331# 4332# 4333 4334# 4335 4337# 4338# 4339#
4340# 4341# 4342# 4343 4344# 4345 4355# 4356# 4357# 4358# 4359 4360# 4361 4363# 4364#
4365# 4366# 4367# 4368# 4369 4370# 4371 4373# 4374# 4375# 4376 4377# 4378 4380# 4381#
4382# 4383# 4384# 4385# 4386 4387# 4388 4433# 4434# 4436# 4440# 4441# 4443# 4447# 4448#
4450# 4454# 4455# 4457# 4475# 4476# 4477# 4479# 4505# 4509# 4530# 4531# 4532# 4533# 4534#
4535 4552# 4553# 4557# 4558# 4564# 4582# 4594# 4597# 4610# 4636# 4638# 4639# 4646# 4648#
4649# 4668# 4670# 4671# 4680# 4682# 4683# 4693# 4695# 4696# 4703# 4705# 4706# 4716# 4718#
4719# 4727# 4729# 4730# 4738# 4740# 4741# 4744# 4771# 4772# 4803# 4804# 4805# 4806# 4808#
4809# 4833# 4835# 4836# 4847# 4849# 4850# 4868# 4869# 4870# 4871# 4873# 4874# 4888# 4889#
4890# 4891# 4893# 4894# 4901# 4902# 4903# 4904# 4906# 4907# 4914# 4915# 4916# 4917# 4919#
4920# 4928# 4929# 4930# 4931# 4933# 4934# 4941# 4942# 4943# 4944# 4946# 4947# 4954# 4955#
4956# 4957# 4959# 4960# 4967# 4968# 4969# 4970# 4972# 4973# 4980# 4981# 4982# 4983# 4985#
4986# 4993# 4994# 4995# 4996# 4998# 4999# 5009# 5010# 5011# 5012# 5014# 5015# 5022# 5023#
5024# 5025# 5027# 5028# 5038# 5039# 5040# 5041# 5043# 5044# 5051# 5052# 5053# 5054# 5056#
5057# 5067# 5068# 5069# 5070# 5072# 5073# 5080# 5081# 5082# 5083# 5085# 5086# 5096# 5097#
5098# 5099# 5101# 5102# 5109# 5110# 5111# 5112# 5114# 5115# 5125# 5126# 5127# 5128# 5130#
5131# 5138# 5139# 5140# 5141# 5143# 5144# 5151# 5152# 5153# 5154# 5156# 5157# 5167# 5168#
5169# 5170# 5172# 5173# 5180# 5181# 5182# 5183# 5185# 5186# 5196# 5197# 5198# 5199# 5201#
5202# 5209# 5210# 5211# 5212# 5214# 5215# 5223# 5224# 5225# 5226# 5228# 5229# 5234# 5235#
5236# 5237# 5242# 5830# 5832# 5833# 5840# 5842# 5843# 5850# 5852# 5853# 5864# 5866# 5867#
5874# 5876# 5877# 5884# 5886# 5887# 5894# 5896# 5897# 5904# 5906# 5907# 5913# 5915# 5916#
5924# 5926# 5927# 5940# 5942# 5943# 5951# 5953# 5954# 5968# 5970# 5971# 5982# 5984# 5985#
5993# 5995# 5996# 6004# 6006# 6007# 6015# 6017# 6018# 6026# 6028# 6029# 6043# 6045# 6046#
6049# 6091# 6093# 6094# 6101# 6103# 6104# 6111# 6113# 6114# 6125# 6127# 6128# 6135# 6137#
6138# 6145# 6147# 6148# 6154# 6156# 6157# 6165# 6167# 6168# 6181# 6183# 6184# 6192# 6194#
6195# 6205# 6207# 6208# 6216# 6218# 6219# 6226# 6228# 6229# 6237# 6239# 6240# 6248# 6250#
6251# 6259# 6261# 6262# 6268# 6270# 6271# 6274# 6315# 6317# 6318# 6325# 6327# 6328# 6335#
6337# 6338# 6349# 6351# 6352# 6359# 6361# 6362# 6369# 6371# 6372# 6378# 6380# 6381# 6389#
6391# 6392# 6405# 6407# 6408# 6416# 6418# 6419# 6429# 6431# 6432# 6440# 6442# 6443# 6450#
6452# 6453# 6461# 6463# 6464# 6472# 6474# 6475# 6483# 6485# 6486# 6492# 6494# 6495# 6498#
6534# 6536# 6537# 6556# 6558# 6559# 6566# 6568# 6569# 6576# 6578# 6579# 6586# 6588# 6589#
6596# 6598# 6599# 6609# 6611# 6612# 6620# 6622# 6623# 6635# 6637# 6638# 6646# 6648# 6649#
6657# 6659# 6660# 6668# 6670# 6671# 6683# 6719# 6721# 6722# 6741# 6743# 6744# 6751# 6753#
6754# 6761# 6763# 6764# 6771# 6773# 6774# 6781# 6783# 6784# 6793# 6795# 6796# 6804# 6806#
6807# 6818# 6820# 6821# 6829# 6831# 6832# 6840# 6842# 6843# 6851# 6853# 6854# 6866# 6899#
6900# 6904# 6929# 6930# 6931# 6932# 6934# 6935# 6948# 6949# 6950# 6951# 6953# 6954# 6967#
6968# 6969# 6970# 6974# 6983# 7021# 7022# 7023# 7024# 7026# 7027# 7040# 7041# 7042# 7043#
7045# 7046# 7052# 7053# 7054# 7055# 7057# 7058# 7071# 7072# 7073# 7074# 7076# 7077# 7083#
7084# 7085# 7086# 7088# 7089# 7102# 7103# 7104# 7105# 7107# 7108# 7121# 7122# 7123# 7124#
7128# 7132# 7175# 7177# 7178# 7194# 7196# 7197# 7204# 7206# 7207# 7214# 7216# 7217# 7224#

```
                7226#  7227#  7234#  7236#  7237#  7243#  7245#  7246#  7254#  7256#  7257#  7264#  7266#  7267#  7275#
                7277#  7278#  7289#  7291#  7292#  7300#  7302#  7303#  7314#  7316#  7317#  7325#  7327#  7328#  7335#
                7337#  7338#  7346#  7348#  7349#  7356#  7358#  7359#  7367#  7369#  7370#  7377#  7379#  7380#  7388#
                7390#  7391#  7405#  7407#  7408#  7416#  7418#  7419#  7433#  7435#  7436#  7447#  7449#  7450#  7458#
                7460#  7461#  7469#  7471#  7472#  7480#  7482#  7483#  7491#  7493#  7494#  7497#  7512#  7516#  7517
                7518   7519   7521#  7522   7523   7524   7526#  7527   7528   7529   7530   7532#  7533   7534   7535
                7536   7538#  7539   7540   7541   7542   7545#  7563#  7567#  7584#  7585#  7586#
MSGNLS     1#   955#
MSGNSU     1#   955#   6903#  6982#
MSGNTA     1#   955#   1117#  1131#  3856#  3895#  3929#  3973#  4018#  4053#  4088#  4123#  4508#  4563#  4581#
           4596#  4609#  4743#  5241#  6048#  6273#  6497#  6682#  6865#  6973#  7127#  7131#  7496#  7545#  7546
           7567#  7568
MSGNTE     1#   955#   4627#  4764#  5813#  6074#  6299#  6519#  6704#  6893#  7160#
MSHAPT     1#   955#   982#
MSHNAP     1#   955#   982#   1021
MSINCR     1#   955#   958#   1101#  1126#  3304#  3315#  3326#  3344#  3847#  3853#  3857#  3862#  3873#  3889#
           3896#  3901#  3910#  3920#  3926#  3930#  3936#  3943#  3949#  3959#  3969#  3974#  3981#  3988#  3994#
           4004#  4014#  4019#  4026#  4033#  4039#  4049#  4054#  4061#  4068#  4075#  4085#  4089#  4096#  4103#
           4109#  4119#  4124#  4158#  4168#  4175#  4185#  4192#  4202#  4209#  4219#  4234#  4244#  4251#  4261#
           4267#  4283#  4293#  4300#  4310#  4317#  4327#  4334#  4344#  4360#  4370#  4377#  4387#  4400#  4413#
           4434#  4441#  4448#  4455#  4476#  4505#  4509#  4527#  4534#  4553#  4558#  4564#  4577#  4582#  4591#
           4594#  4597#  4607#  4610#  4627#  4628#  4636#  4638#  4646#  4648#  4668#  4670#  4680#  4682#  4693#
           4695#  4703#  4705#  4716#  4718#  4727#  4729#  4738#  4740#  4744#  4764#  4765#  4771#  4803#  4808#
           4833#  4835#  4847#  4849#  4868#  4873#  4888#  4893#  4901#  4906#  4914#  4919#  4928#  4933#  4941#
           4946#  4954#  4959#  4967#  4972#  4980#  4985#  4993#  4998#  5009#  5014#  5022#  5027#  5038#  5043#
           5051#  5056#  5067#  5072#  5080#  5085#  5096#  5101#  5109#  5114#  5125#  5130#  5138#  5143#  5151#
           5156#  5167#  5172#  5180#  5185#  5196#  5201#  5209#  5214#  5223#  5228#  5234#  5242#  5813#  5814#
           5830#  5832#  5840#  5842#  5850#  5852#  5864#  5866#  5874#  5876#  5884#  5886#  5894#  5896#  5904#
           5906#  5913#  5915#  5924#  5926#  5940#  5942#  5951#  5953#  5968#  5970#  5982#  5984#  5993#  5995#
           6004#  6006#  6015#  6017#  6026#  6028#  6043#  6045#  6049#  6074#  6075#  6091#  6093#  6101#  6103#
           6111#  6113#  6125#  6127#  6135#  6137#  6145#  6147#  6154#  6156#  6165#  6167#  6181#  6183#  6192#
           6194#  6205#  6207#  6216#  6218#  6226#  6228#  6237#  6239#  6248#  6250#  6259#  6261#  6268#  6270#
           6274#  6299#  6300#  6315#  6317#  6325#  6327#  6335#  6337#  6349#  6351#  6359#  6361#  6369#  6371#
           6378#  6380#  6389#  6391#  6405#  6407#  6416#  6418#  6429#  6431#  6440#  6442#  6450#  6452#  6461#
           6463#  6472#  6474#  6483#  6485#  6492#  6494#  6498#  6519#  6520#  6534#  6536#  6556#  6558#  6566#
           6568#  6576#  6578#  6586#  6588#  6596#  6598#  6609#  6611#  6620#  6622#  6635#  6637#  6646#  6648#
           6657#  6659#  6668#  6670#  6683#  6704#  6705#  6719#  6721#  6741#  6743#  6751#  6753#  6761#  6763#
           6771#  6773#  6781#  6783#  6793#  6795#  6804#  6806#  6818#  6820#  6829#  6831#  6840#  6842#  6851#
           6853#  6866#  6893#  6894#  6899#  6903#  6904#  6929#  6934#  6948#  6953#  6967#  6974#  6982#  6983#
           7021#  7026#  7040#  7045#  7052#  7057#  7071#  7076#  7083#  7088#  7102#  7107#  7121#  7128#  7132#
           7160#  7161#  7175#  7177#  7194#  7196#  7204#  7206#  7214#  7216#  7224#  7226#  7234#  7236#  7243#
           7245#  7254#  7256#  7264#  7266#  7275#  7277#  7287#  7291#  7300#  7302#  7314#  7316#  7325#  7327#
           7335#  7337#  7346#  7348#  7356#  7358#  7367#  7369#  7377#  7379#  7388#  7390#  7405#  7407#  7416#
           7418#  7433#  7435#  7447#  7449#  7458#  7460#  7469#  7471#  7480#  7482#  7491#  7493#  7497#  7512#
           7563#
MSIOSE     1#   955#
MSLDR0     1#   955#   4433#  4440#  4447#  4454#  4475#  4552#  4557#
MSMASK     1#   955#
MSMCHI     1#   955#
MSMCLO     1#   955#
MSMSK1     1#   955#
MSPOP      1#   955#   1117#  1131#  3856#  3895#  3929#  3973#  4018#  4053#  4088#  4123#  4405#  4508#  4563#
           4581#  4596#  4609#  4743#  5241#  6048#  6273#  6497#  6682#  6865#  6973#  7127#  7131#  7496#  7545#
           7567#  7581#
MSPRIN     1#   955#   3301#  3312#  3323#  3340#  3850#  3869#  3884#  3905#  3914#  3923#  3938#  3946#  3954#
           3963#  3983#  3991#  3999#  4008#  4028#  4036#  4044#  4063#  4071#  4079#  4098#  4106#  4114#  4153#
```

```
        4161#  4171#  4178#  4188#  4195#  4205#  4212#  4229#  4237#  4247#  4254#  4264#  4278#  4286#  4296#
        4303#  4313#  4320#  4330#  4337#  4355#  4363#  4373#  4380#
MSPUSH     1#   955#   958#  1101#  1126#  3847#  3862#  3901#  3936#  3981#  4026#  4061#  4096#  4400#  4413#
        4527#  4577#  4591#  4607#  4627#  4628   4764#  4765   5813#  5814   6074#  6075   6299#  6300   6519#
        6520   6704#  6705   6893#  6894   6903#  6904   6982#  6983   7160#  7161   7512#  7563#
MSPUT      1#   955#  3301#  3312#  3323#  3340#  3850#  3869#  3884#  3905#  3914#  3923#  3938#  3946#  3954#
        3963#  3983#  3991#  3999#  4008#  4028#  4036#  4044#  4063#  4071#  4079#  4098#  4106#  4114#  4153#
        4161#  4171#  4178#  4188#  4195#  4205#  4212#  4229#  4237#  4247#  4254#  4264#  4278#  4286#  4296#
        4303#  4313#  4320#  4330#  4337#  4355#  4363#  4373#  4380#  4530#
MSPUT1     1#   955#  3301#  3302   3312#  3313   3323#  3324   3340#  3341   3342   3850#  3851   3869#  3870
        3871   3884#  3885   3886   3887   3905#  3906   3907   3908   3914#  3915   3916   3917   3918   3923#
        3924   3938#  3939   3940   3941   3946#  3947   3954#  3955   3956   3957   3963#  3964   3965   3966
        3967   3983#  3984   3985   3986   3991#  3992   3999#  4000   4001   4002   4008#  4009   4010   4011
        4012   4028#  4029   4030   4031   4036#  4037   4044#  4045   4046   4047   4063#  4064   4065   4066
        4071#  4072   4073   4079#  4080   4081   4082   4083   4098#  4099   4100   4101   4106#  4107   4114#
        4115   4116   4117   4153#  4154   4155   4156   4161#  4162   4163   4164   4165   4166   4171#  4172
        4173   4178#  4179   4180   4181   4182   4183   4188#  4189   4190   4195#  4196   4197   4198   4199
        4200   4205#  4206   4207   4212#  4213   4214   4215   4216   4217   4229#  4230   4231   4232   4237#
        4238   4239   4240   4241   4242   4247#  4248   4249   4254#  4255   4256   4257   4258   4259   4264#
        4265   4278#  4279   4280   4281   4286#  4287   4288   4289   4290   4291   4296#  4297   4298   4303#
        4304   4305   4306   4307   4308   4313#  4314   4315   4320#  4321   4322   4323   4324   4325   4330#
        4331   4332   4337#  4338   4339   4340   4341   4342   4355#  4356   4357   4358   4363#  4364   4365
        4366   4367   4368   4373#  4374   4375   4380#  4381   4382   4383   4384   4385   4530#  4531   4532
        4533
MSRADI     1#   955#  7516#  7521#  7526#  7532#  7538#
MSRBR0     1#   955#
MSRNR0     1#   955#  4475#  4477
MSSETS     1#   955#   958#  1101#  1126#  3847#  3862#  3901#  3936#  3981#  4026#  4061#  4096#  4400#  4413#
        4527#  4577#  4591#  4607#  4628#  4765#  5814#  6075#  6300#  6520#  6705#  6894#  6904#  6983#  7161#
        7512#  7563#
MSSTAR     1#   955#
MSSVC      1#   955#  3301#  3304   3312#  3315   3323#  3326   3340#  3344   3850#  3853   3856#  3857   3869#
        3873   3884#  3889   3895#  3896   3905#  3910   3914#  3920   3923#  3926   3929#  3930   3938#  3943
        3946#  3949   3954#  3959   3963#  3969   3973#  3974   3983#  3988   3991#  3994   3999#  4004   4008#
        4014   4018#  4019   4028#  4033   4036#  4039   4044#  4049   4053#  4054   4063#  4068   4071#  4075
        4079#  4085   4088#  4089   4098#  4103   4106#  4109   4114#  4119   4123#  4124   4153#  4158   4161#
        4168   4171#  4175   4178#  4185   4188#  4192   4195#  4202   4205#  4209   4212#  4219   4229#  4234
        4237#  4244   4247#  4251   4254#  4261   4264#  4267   4278#  4283   4286#  4293   4296#  4300   4303#
        4310   4313#  4317   4320#  4327   4330#  4334   4337#  4344   4355#  4360   4363#  4370   4373#  4377
        4380#  4387   4433#  4434   4440#  4441   4447#  4448   4454#  4455   4475#  4476   4505#  4508#  4509
        4530#  4534   4552#  4553   4557#  4558   4563#  4564   4581#  4582   4594#  4596#  4597   4609#  4610
        4636#  4638#  4646#  4648#  4668#  4670#  4680#  4682#  4693#  4695#  4703#  4705#  4716#  4718#  4727#
        4729#  4738#  4740#  4743#  4744   4771#  4803   4808#  4833#  4835#  4847#  4849#  4868   4873#  4888
        4893#  4901   4906#  4914   4919#  4928   4933#  4941   4946#  4954   4959#  4967   4972#  4980   4985#
        4993   4998#  5009   5014#  5022   5027#  5038   5043#  5051   5056#  5067   5072#  5080   5085#  5096
        5101#  5109   5114#  5125   5130#  5138   5143#  5151   5156#  5167   5172#  5180   5185#  5196   5201#
        5209   5214#  5223   5228#  5234   5241#  5242   5830#  5832#  5840#  5842#  5850#  5852#  5864#  5866#
        5874#  5876#  5884#  5886#  5894#  5896#  5904#  5906#  5913#  5915#  5924#  5926#  5940#  5942#  5951#
        5953#  5968#  5970#  5982#  5984#  5993#  5995#  6004#  6006#  6015#  6017#  6026#  6028#  6043#  6045#
        6048#  6049   6091#  6093#  6101#  6103#  6111#  6113#  6125#  6127#  6135#  6137#  6145#  6147#  6154#
        6156#  6165#  6167#  6181#  6183#  6192#  6194#  6205#  6207#  6216#  6218#  6226#  6228#  6237#  6239#
        6248#  6250#  6259#  6261#  6268#  6270#  6273#  6274   6315#  6317#  6325#  6327#  6335#  6337#  6349#
        6351#  6359#  6361#  6369#  6371#  6378#  6380#  6389#  6391#  6405#  6407#  6416#  6418#  6429#  6431#
        6440#  6442#  6450#  6452#  6461#  6463#  6472#  6474#  6483#  6485#  6492#  6494#  6497#  6498   6534#
        6536#  6556#  6558#  6566#  6568#  6576#  6578#  6586#  6588#  6596#  6598#  6609#  6611#  6620#  6622#
        6635#  6637#  6646#  6648#  6657#  6659#  6668#  6670#  6682#  6683   6719#  6721#  6741#  6743#  6751#
```

```
                6753#  6761#  6763#  6771#  6773#  6781#  6783#  6793#  6795#  6804#  6806#  6818#  6820#  6829#  6831#
                6840#  6842#  6851#  6853#  6865#  6866   6899#  6903#  6904   6929   6934#  6948   6953#  6967   6973#
                6974   6982#  6983   7021   7026#  7040   7045#  7052   7057#  7071   7076#  7083   7088#  7102   7107#
                7121   7127#  7128   7131#  7132   7175#  7177#  7194#  7196#  7204#  7206#  7214#  7216#  7224#  7226#
                7234#  7236#  7243#  7245#  7254#  7256#  7264#  7266#  7275#  7277#  7289#  7291#  7300#  7302#  7314#
                7316#  7325#  7327#  7335#  7337#  7346#  7348#  7356#  7358#  7367#  7369#  7377#  7379#  7388#  7390#
                7405#  7407#  7416#  7418#  7433#  7435#  7447#  7449#  7458#  7460#  7469#  7471#  7480#  7482#  7491#
                7493#  7496#  7497
MSTLAB             1#   955#  3304#  3315#  3326#  3344#  3853#  3857#  3873#  3889#  3896#  3910#  3920#  3926#  3930#
                3943#  3949#  3959#  3969#  3974#  3988#  3994#  4004#  4014#  4019#  4033#  4039#  4049#  4054#  4068#
                4075#  4085#  4089#  4103#  4109#  4119#  4124#  4158#  4168#  4175#  4185#  4192#  4202#  4209#  4219#
                4234#  4244#  4251#  4261#  4267#  4283#  4293#  4300#  4310#  4317#  4327#  4334#  4344#  4360#  4370#
                4377#  4387#  4434#  4441#  4448#  4455#  4476#  4505#  4509#  4534#  4553#  4558#  4564#  4582#  4594#
                4597#  4610#  4636#  4638#  4646#  4648#  4668#  4670#  4680#  4682#  4693#  4695#  4703#  4705#  4716#
                4718#  4727#  4729#  4738#  4740#  4744#  4771#  4803#  4808#  4833#  4835#  4847#  4849#  4868#  4873#
                4888#  4893#  4901#  4906#  4914#  4919#  4928#  4933#  4941#  4946#  4954#  4959#  4967#  4972#  4980#
                4985#  4993#  4998#  5009#  5014#  5022#  5027#  5038#  5043#  5051#  5056#  5067#  5072#  5080#  5085#
                5096#  5101#  5109#  5114#  5125#  5130#  5138#  5143#  5151#  5156#  5167#  5172#  5180#  5185#  5196#
                5201#  5209#  5214#  5223#  5228#  5234#  5242#  5830#  5832#  5840#  5842#  5850#  5852#  5864#  5866#
                5874#  5876#  5884#  5886#  5894#  5896#  5904#  5906#  5913#  5915#  5924#  5926#  5940#  5942#  5951#
                5953#  5968#  5970#  5982#  5984#  5993#  5995#  6004#  6006#  6015#  6017#  6026#  6028#  6043#  6045#
                6049#  6091#  6093#  6101#  6103#  6111#  6113#  6125#  6127#  6135#  6137#  6145#  6147#  6154#  6156#
                6165#  6167#  6181#  6183#  6192#  6194#  6205#  6207#  6216#  6218#  6226#  6228#  6237#  6239#  6248#
                6250#  6259#  6261#  6268#  6270#  6274#  6315#  6317#  6325#  6327#  6335#  6337#  6349#  6351#  6359#
                6361#  6369#  6371#  6378#  6380#  6389#  6391#  6405#  6407#  6416#  6418#  6429#  6431#  6440#  6442#
                6450#  6452#  6461#  6463#  6472#  6474#  6483#  6485#  6492#  6494#  6498#  6534#  6536#  6556#  6558#
                6566#  6568#  6576#  6578#  6586#  6588#  6596#  6598#  6609#  6611#  6620#  6622#  6635#  6637#  6646#
                6648#  6657#  6659#  6668#  6670#  6683#  6719#  6721#  6741#  6743#  6751#  6753#  6761#  6763#  6771#
                6773#  6781#  6783#  6793#  6795#  6804#  6806#  6818#  6820#  6829#  6831#  6840#  6842#  6851#  6853#
                6866#  6899#  6904#  6929#  6934#  6948#  6953#  6967#  6974#  6983#  7021#  7026#  7040#  7045#  7052#
                7057#  7071#  7076#  7083#  7088#  7102#  7107#  7121#  7128#  7132#  7175#  7177#  7194#  7196#  7204#
                7206#  7214#  7216#  7224#  7226#  7234#  7236#  7243#  7245#  7254#  7256#  7264#  7266#  7275#  7277#
                7289#  7291#  7300#  7302#  7314#  7316#  7325#  7327#  7335#  7337#  7346#  7348#  7356#  7358#  7367#
                7369#  7377#  7379#  7388#  7390#  7405#  7407#  7416#  7418#  7433#  7435#  7447#  7449#  7458#  7460#
                7469#  7471#  7480#  7482#  7491#  7493#  7497#
MSTSTL             1#   955#  3304#  3315#  3326#  3344#  3853#  3857#  3873#  3889#  3896#  3910#  3920#  3926#  3930#
                3943#  3949#  3959#  3969#  3974#  3988#  3994#  4004#  4014#  4019#  4033#  4039#  4049#  4054#  4068#
                4075#  4085#  4089#  4103#  4109#  4119#  4124#  4158#  4168#  4175#  4185#  4192#  4202#  4209#  4219#
                4234#  4244#  4251#  4261#  4267#  4283#  4293#  4300#  4310#  4317#  4327#  4334#  4344#  4360#  4370#
                4377#  4387#  4434#  4441#  4448#  4455#  4476#  4505#  4509#  4534#  4553#  4558#  4564#  4582#  4594#
                4597#  4610#  4636#  4638#  4646#  4648#  4668#  4670#  4680#  4682#  4693#  4695#  4703#  4705#  4716#
                4718#  4727#  4729#  4738#  4740#  4744#  4771#  4803#  4808#  4833#  4835#  4847#  4849#  4868#  4873#
                4888#  4893#  4901#  4906#  4914#  4919#  4928#  4933#  4941#  4946#  4954#  4959#  4967#  4972#  4980#
                4985#  4993#  4998#  5009#  5014#  5022#  5027#  5038#  5043#  5051#  5056#  5067#  5072#  5080#  5085#
                5096#  5101#  5109#  5114#  5125#  5130#  5138#  5143#  5151#  5156#  5167#  5172#  5180#  5185#  5196#
                5201#  5209#  5214#  5223#  5228#  5234#  5242#  5830#  5832#  5840#  5842#  5850#  5852#  5864#  5866#
                5874#  5876#  5884#  5886#  5894#  5896#  5904#  5906#  5913#  5915#  5924#  5926#  5940#  5942#  5951#
                5953#  5968#  5970#  5982#  5984#  5993#  5995#  6004#  6006#  6015#  6017#  6026#  6028#  6043#  6045#
                6049#  6091#  6093#  6101#  6103#  6111#  6113#  6125#  6127#  6135#  6137#  6145#  6147#  6154#  6156#
                6165#  6167#  6181#  6183#  6192#  6194#  6205#  6207#  6216#  6218#  6226#  6228#  6237#  6239#  6248#
                6250#  6259#  6261#  6268#  6270#  6274#  6315#  6317#  6325#  6327#  6335#  6337#  6349#  6351#  6359#
                6361#  6369#  6371#  6378#  6380#  6389#  6391#  6405#  6407#  6416#  6418#  6429#  6431#  6440#  6442#
                6450#  6452#  6461#  6463#  6472#  6474#  6483#  6485#  6492#  6494#  6498#  6534#  6536#  6556#  6558#
                6566#  6568#  6576#  6578#  6586#  6588#  6596#  6598#  6609#  6611#  6620#  6622#  6635#  6637#  6646#
                6648#  6657#  6659#  6668#  6670#  6683#  6719#  6721#  6741#  6743#  6751#  6753#  6761#  6763#  6771#
                6773#  6781#  6783#  6793#  6795#  6804#  6806#  6818#  6820#  6829#  6831#  6840#  6842#  6851#  6853#
```

```
        6866#  6899#  6904#  6929#  6934#  6948#  6953#  6967#  6974#  6983#  7021#  7026#  7040#  7045#  7052#
        7057#  7071#  7076#  7083#  7088#  7102#  7107#  7121#  7128#  7132#  7175#  7177#  7194#  7196#  7204#
        7206#  7214#  7216#  7224#  7226#  7234#  7236#  7243#  7245#  7254#  7256#  7264#  7266#  7275#  7277#
        7289#  7291#  7300#  7302#  7314#  7316#  7325#  7327#  7335#  7337#  7346#  7348#  7356#  7358#  7367#
        7369#  7377#  7379#  7388#  7390#  7405#  7407#  7416#  7418#  7433#  7435#  7447#  7449#  7458#  7460#
        7469#  7471#  7480#  7482#  7491#  7493#  7497#
MSWORD     1#   955#  1021#  1030   1080#  1082   1083   1084   1085   1086   1087   1088   1089   1090   4771#
        4803#  4804   4805   4806   4868#  4869   4870   4871   4888#  4889   4890   4891   4901#  4902   4903
        4904   4914#  4915   4916   4917   4928#  4929   4930   4931   4941#  4942   4943   4944   4954#  4955
        4956   4957   4967#  4968   4969   4970   4980#  4981   4982   4983   4993#  4994   4995   4996   5009#
        5010   5011   5012   5022#  5023   5024   5025   5038#  5039   5040   5041   5051#  5052   5053   5054
        5067#  5068   5069   5070   5080#  5081   5082   5083   5096#  5097   5098   5099   5109#  5110   5111
        5112   5125#  5126   5127   5128   5138#  5139   5140   5141   5151#  5152   5153   5154   5167#  5168
        5169   5170   5180#  5181   5182   5183   5196#  5197   5198   5199   5209#  5210   5211   5212   5223#
        5224   5225   5226   5234#  5235   5236   5237   6899#  6929#  6930   6931   6932   6948#  6949   6950
        6951   6967#  6968   6969   6970   7021#  7022   7023   7024   7040#  7041   7042   7043   7052#  7053
        7054   7055   7071#  7072   7073   7074   7083#  7084   7085   7086   7102#  7103   7104   7105   7121#
        7122   7123   7124   7516#  7521#  7526#  7532#  7538#  7585   7586
MSXFER     1#   955#
NEWTST  1644#  4611   4745   5789   6050   6275   6499   6684   6867   7133
NTST    1644#  4611   4745   5789   6050   6275   6499   6684   6867   7133
OPEN       1#   955#
POINTE     1#   955#   978
PRINTB     1#   955#  3849   3904   3913   3922   3937   3945   3953   3962   3982   3990   3998   4007   4027
        4035   4043   4062   4070   4078   4097   4105   4113   4263
PRINTF     1#   955#  3300   3311   3322   3339
PRINTS     1#   955#
PRINTX     1#   955#  3868   3883   4152   4160   4170   4177   4187   4194   4204   4211   4228   4236   4246
        4253   4277   4285   4295   4302   4312   4319   4329   4336   4354   4362   4372   4379
READBU     1#   955#
READEF     1#   955#  4432   4439   4446   4453
RFLAGS     1#   955#
SETDF   1644#  2103   2147   2194   2265   2364   2636   2740   2773   2784   2818   2829   2863   2874   2908
        2919   2953   2964   2996   3007   3043   3054   3067   3078   3110   3122   3495   3508   3534   3544
        3557   3567   3580   3590   3603   3613   3626   3636
SETHRD  1644#
SETPRI     1#   955#
SETSF   1644#
SETSFT  1644#
SETVEC     1#   955#  4529
SLASH      1#   955#  1072   1076
STARS      1#   955#
SVC        1#   953#   954
T$GEN   1644#  2103   2147   2194   2265   2364   2636   2740   2773   2784   2818   2829   2863   2874   2908
        2919   2953   2964   2996   3007   3043   3054   3067   3078   3110   3122   3495   3508   3534   3544
        3557   3567   3580   3590   3603   3613   3626   3636
XFER       1#   955#  4771#  6399#
XFERF      1#   955#
XFERT      1#   955#
$GEDF   1644#  4802   4867   4887   4900   4913   4927   4940   4953   4966   4979   4992   5008   5021   5037
        5050   5066   5079   5095   5108   5124   5137   5150   5166   5179   5195   5208   5222   5233   6928
        6947   6966   7020   7039   7051   7070   7082   7101   7120
$GEHRD  1644#
$GESF   1644#
$GESFT  1644#
$GTDF   1644#  2102   2146   2193   2264   2363   2635   2739   2772   2783   2817   2828   2862   2873   2907
```

CVDMEA.P11 10-DEC-80 09:16 CROSS REFERENCE TABLE -- MACRO NAMES

```
          2918    2952    2963    2995    3006    3042    3053    3066    3077    3109    3121    3494    3507    3533    3543
          3556    3566    3579    3589    3602    3612    3625    3635
$GTHRD    1644#
$GTSF     1644#
$GTSFT    1644#


. ABS.  034506     000

 ERRORS DETECTED:  0

 CVDMEA.BIN,CVDMEA.SEQ/CRF/SOL=SVC34R.MAC,CVDMEA.P11
 RUN-TIME: 32 41 4 SECONDS
 RUN-TIME RATIO: 164/78=2.0
 CORE USED:  21K  (41 PAGES)
```