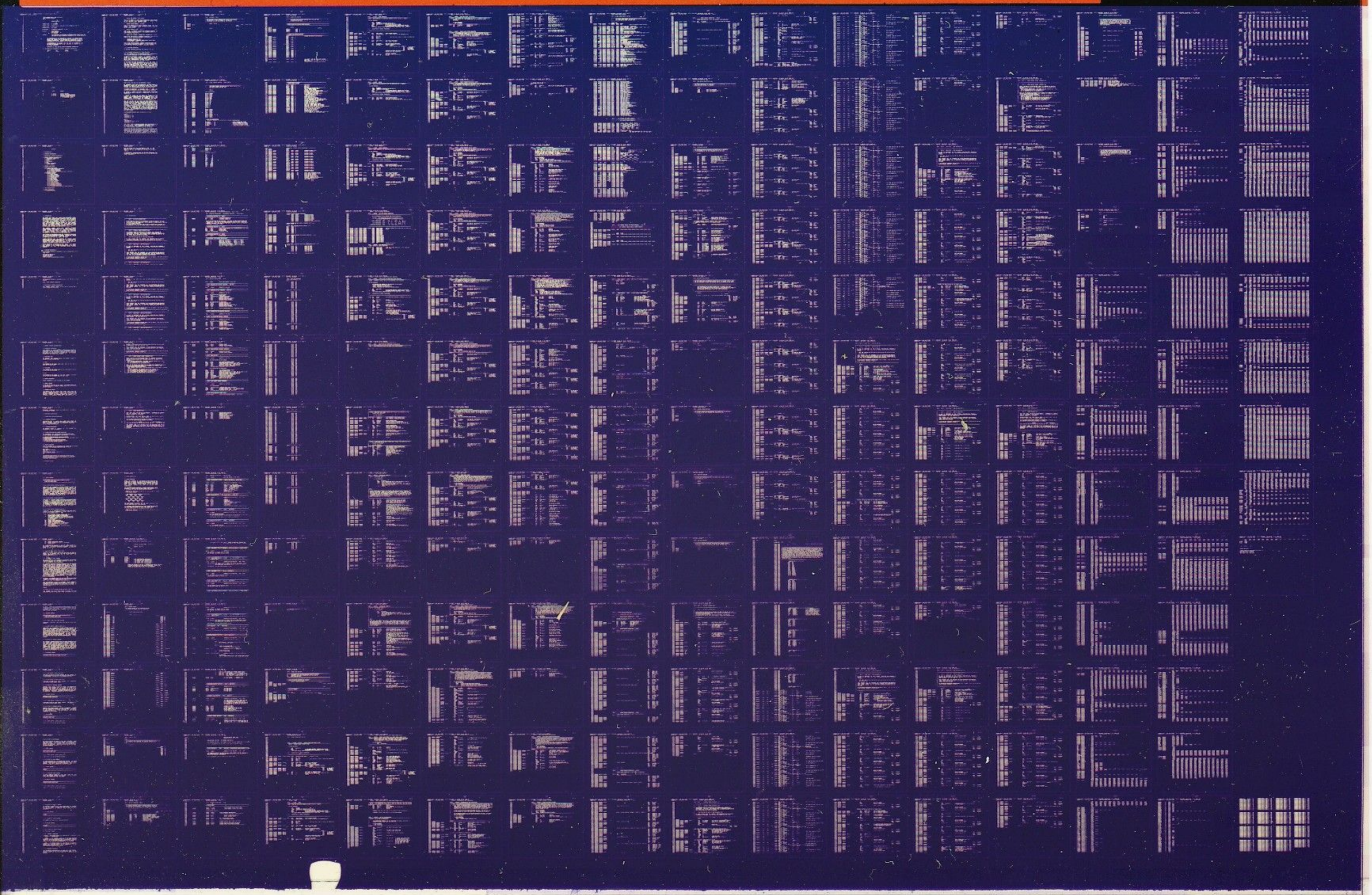
DMV11 DMV-11 LN UNT STC 3
CVOMECO

AH-S394C-MC
FICHE 01 OF 01

FEB 1985
digital
Made In USA



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

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

                        PRODUCT CODE:   AC-S392C-MC

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

                        PRODUCT DATE:   JULY 1983

                        MAINTAINER:     DIAGNOSTICS MERRIMACK  CC:38P

                        AUTHORS:        CHRIS BRIEMEN
                                        DAVE HOFFMAN
                                        RAY MARSHALL

                        PURPOSE:        THIS DIAGNOSTIC IS DESIGNED TO PERFORM STATIC LOGIC TESTS FOR
                                        THE M8053 OR M8L84 (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) 1981,1984 BY DIGITAL EQUIPMENT CORPORATION

                        THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                                DIGITAL         PDP             UNIBUS          MASSBUS
                                DEC             DI'CUS          DECTAPE
```

```
41
42
43
44
45
46                                   HISTORY
47                                   -------
48
49
50                          REV          DATE          REASON
51                          ---          ----          ------
52
53                           A           14-JAN-81     INITIAL RELEASE
54                           B           11-JUL-83     INSTALL OUTSTANDING PATCHES
55                           C           29-JUL-84     INCREASED TIMING PARAMETERS
56                                                     TO ALLOW PROGRAM TO RUN ON
57                                                     A J-11 PROCESSOR (ORION).
58
59
```

CONTENTS

```
107
108
109                                   1.0 INTRODUCTION
110
111                                   THE M8053 AND M8064 ARE SINGLE-LINE SYNCHRONOUS, MICRO-
112                                   PROCESSOR BASED COMMUNICATIONS INTERFACES WHICH CAN SUPPORT
113                                   BOTH CHARACTER-ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED
114                                   (SDLC, HDLC, ETC.) PROTOCOLS. THE PURPOSE OF THIS PROGRAM IS
115                                   TO PERFORM STATIC DIAGNOSTIC TESTING OF THE VIA, FIFO,
116                                   USYRT (BCP/BOP MODES), AND LINE DRIVERS ON THESE BOARDS.
117                                   NOTE THAT ALL EXTERNAL LOOPBACK (XLB) TESTS ARE CONTAINED
118                                   HERE. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: MODEM
119                                   LOOPBACK AND ASSORTED EXTERNAL LOOPBACK TESTS ( INCLUDING
120                                   BCP:CRC-16/ODD VRC/EVEN VRC; BOP:CRC-CCITT-1'S/0'S ).
121
122                                   THE  STATIC  LOGIC  TESTS  WILL  PROVIDE  EXTENSIVE
123                                   TROUBLESHOOTING CAPABILITIES, SUCH AS TIGHT SCOPE LOOPS,
124                                   SWITCH OPTIONS, AND ABILITY TO "LOCK" ONTO INTERMITTENT
125                                   ERRORS. IN ADDITION TESTS ARE DESIGNED AND STRUCTURED
126                                   TO  ACHIEVE  MAXIMUM  FAULT  RESOLUTION AND FACILITATE
127                                   REPLACEMENT OF THE SMALLEST FIELD REPLACEABLE UNIT.
128
129                                   THIS PROGRAM IS IMPLEMENTED USING THE  DIAGNOSTIC SUPERVISOR
130                                   AND  A STRUCTURED PROGRAMMING APPROACH.  BECAUSE THE  DESIGN
131                                   CONFORMS TO THE SUPERVISOR  (STANDALONE VERSION) THE PROGRAM
132                                   IS COMPATIBLE WITH ACT, APT, XXDP+, AND SLIDE.
133
134                                   THROUGH  DIALOGUE  WITH  THE  OPERATOR, THE  PROGRAM  ALLOWS
135                                   MODIFICATION  OF DEVICE PARAMETERS, SUCH AS LSI-BUS ADDRESS,
136                                   VECTOR ADDRESSES AND  DEVICE  PRIORITY.   IN  ADDITION,  THE
137                                   OPERATOR  CAN  SPECIFY  PARTICULAR  TESTS  TO  BE  RUN AND A
138                                   VARIETY OF LOOPING, RUNNING, AND REPORTING MODES.
139
140                                   DEVICE ERRORS WILL BE REPORTED AS THEY  OCCUR.   THE  REPORT
141                                   WILL  INCLUDE  A  TEST  NUMBER AND DESCRIPTION OF THE ERROR,
142                                   GOOD AND BAD  TEST  DATA,  AND  APPLICABLE  DEVICE  REGISTER
143                                   CONTENTS.
144
145
146                                   2.0 HARDWARE REQUIREMENTS
147
148                                   THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE M8053/8064 STATIC
149                                   LOGIC TESTS:
150
151                                   PDP-11/03 OR PDP-11/23
152                                   16K WORDS OF MEMORY
153                                   CONSOLE TERMINAL
154                                   M8053 OR M8064 COMMUNICATIONS INTERFACE
155
```

PROGRAM DOCUMENT

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

H3254, H3255 LOOPBACK CONNECTORS

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THIS PROGRAM (CVDME) SHOULD BE THE LAST OF THE FIVE DMV-11 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 30 SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

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

4.4 ACT/SLIDE

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

4.5 APT

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

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM.

4.7 ERROR LOGGING

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

5.0 PROGRAM LOAD MEDIA

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

DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY
THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CVDME-C-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

```
278
279
280                                   6.3.1 START COMMAND
281
282                                   ************************************************************
283                                   STA(RT)/TESTS:<TEST-LIST>/PASS:<PASS-CNT>/FLAGS:
284                                      <FLAG-LIST>/EOP:<INCR>
285                                   ************************************************************
286
287
288                                   6.3.1.1 TESTS SWITCH (/TESTS:<TEST-LIST>)
289
290                                   <TEST-LIST>  IS  A SEQUENCE OF DECIMAL NUMBERS (1:2 ETC.) OR
291                                   RANGES OF DECIMAL NUMBERS (1-5:8-10 ETC.) THAT  SPECIFY  THE
292                                   TESTS  TO BE EXECUTED.  THE NUMBERS ARE SEPARATED BY COLONS.
293                                   THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER  IN  THE
294                                   DIAGNOSTIC.  THEY MAY BE SPECIFIED IN ANY ORDER.  TESTS WILL
295                                   BE EXECUTED IN NUMERICAL ORDER REGARDLESS OF  THE  ORDER  OF
296                                   SPECIFICATION.   THE  JEFAULT  IS  TO EXECUTE ALL TESTS.  ON
297                                   THIS AND ALL SWITCHES, THE ANGLE BRACKETS <> ARE PUNCTUATION
298                                   USED  IN THE DEFINITION ONLY, AND ARE NOT TO BE TYPED BY THE
299                                   OPERATOR.  SEE EXAMPLE AT END OF 6.3.1.5.
300
301
302                                   6.3.1.2 PASS SWITCH (/PASS:<PASS-CNT>)
303
304                                   <PASS-CNT> IS A DECIMAL NUMBER INDICATING THE DESIRED NUMBER
305                                   OF  PASSES.   A PASS IS DEFINED AS THE EXECUTION OF THE FULL
306                                   DIAGNOSTIC (ALL SELECTED TESTS) AGAINST ALL UNITS SUBMITTED.
307                                   THE DEFAULT IS NON-ENDING EXECUTION.  IN THIS CASE EXIT FROM
308                                   THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C  OR
309                                   BY  OCCURANCE  OF AN ERROR WITH THE HALT ON ERROR FLAG BEING
310                                   SET.  THE EXIT IS A RETURN TO COMMAND MODE.  SEE EXAMPLE  AT
311                                   END OF 6.3.1.5.
312
313
314                                   6.3.1.3 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
315
316                                   <FLAG-LIST>  IS  A  SEQUENCE OF ELEMENTS OF THE FORM <FLAG>,
317                                   <FLAG=1>, OR <FLAG=0>, SEPARATED BY COLONS, WHERE <FLAG> HAS
318                                   ONE OF THE FOLLOWING VALUES:
319
320                                      HOE    HALT ON ERROR, CAUSING COMMAND MODE TO BE
321                                             ENTERED WHEN AN ERROR IS ENCOUNTERED
322                                      LOE    LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP
323                                             CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK
324                                             OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAIN-
325                                             ING THE ERROR
326                                      IER    INHIBIT ERROR REPORTING
327                                      IBE    INHIBIT BASIC ERROR REPORTS
328                                      IXE    INHIBIT EXTENDED ERROR REPORTS
329                                      PRI    DIRECT ALL MESSAGES TO A LINE PRINTER
330                                      PNT    PRINT NUMBER OF TEST BEING EXECUTED
331                                      BOE    BELL ON ERROR
332                                      UAM    RUN IN UNATTENDED MODE, BYPASSING MANUAL
333                                             INTERVENTION TESTS
```

```
                        ISR   INHIBIT STATISTICAL REPORTS
                        IDU   INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
                        LOT   LOOP ON TEST

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


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

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


6.3.1.5 EFFECT OF START COMMAND

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

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

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

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

EXAMPLE:

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

THIS  COMMAND  WILL CAUSE THREE PASSES TO BE MADE, EACH PASS
CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10  EXECUTED  AGAINST
```

ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

6.3.2.3 EFFECT OF RESTART COMMAND

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

6.3.3 CONTINUE COMMAND

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

```
446
447                                 6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)
448
449                                 <PASS-CNT> IS SAME AS IN START COMMAND, BUT THE  DEFAULT  IS
450                                 THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART.
451                                 IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.
452
453
454                                 6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)
455
456                                 <FLAG-LIST> IS SAME AS IN  START  COMMAND,  BUT  UNSPECIFIED
457                                 FLAGS RETAIN THEIR CURRENT VALUE.
458
459
460                                 6.3.3.3 EFFECT OF CONTINUE COMMAND
461
462                                 CONTINUE  MUST  FOLLOW  A START OR RESTART, AND COMMAND MODE
463                                 MUST HAVE  BEEN  ENTERED  DUE  TO  A  HALT  ON  ERROR  OR  A
464                                 CONTROL/C.  THE  EFFECT  OF  THE  COMMAND  IS  TO  GO TO THE
465                                 BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE  HALT
466                                 OR  CONTROL/C  TOOK PLACE.  SOFTWARE DIALOGUE MAY OPTIONALLY
467                                 BE REEXECUTED.  HARDWARE PARAMETERS MAY NOT BE CHANGED.
468
469
470                                 6.3.4 PROCEED COMMAND
471
472                                 ************************************************************
473                                 PRO(CEED)/FLAGS:<FLAG-LIST>
474                                 ************************************************************
475
476
477                                 6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
478
479                                 <FLAG-LIST> IS AS IN  THE  START  COMMAND,  BUT  UNSPECIFIED
480                                 FLAGS RETAIN THEIR CURRENT VALUE.
481
482
483                                 6.3.4.2 EFFECT OF PROCEED COMMAND
484
485                                 PROCEED  MUST FOLLOW A START, RESTART, OR CONTINUE.  COMMAND
486                                 MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR.  THE EFFECT
487                                 OF  THE  COMMAND  IS  TO  BEGIN  EXECUTION  AT  THE LOCATION
488                                 FOLLOWING THE ERROR CALL.  NEITHER  HARDWARE  NOR  SOFTWARE
489                                 PARAMETERS MAY BE ALTERED.
490
491
492                                 6.3.5 ADD COMMAND
493
494                                 ************************************************************
495                                 ADD/UNITS:<UNIT-LIST>
496                                 ************************************************************
497
498
499                                 6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>
500
501                                 <UNIT-LIST> IS AS IN THE RESTART COMMAND.
```

```
502
503
504                                   6.3.5.2 EFFECT OF ADD COMMAND
505
506                                   THE UNITS SPECIFIED ARE ADDED TO THE  TEST  SEQUENCE.   EACH
507                                   UNIT  MUST  HAVE  A  P-TABLE  IN  MEMORY  DUE  TO AN EARLIER
508                                   HARDWARE DIALOGUE.  THIS  COMMAND  MUST  BE  FOLLOWED  BY  A
509                                   RESTART  OR  CONTINUE.   THE UNITS SWITCH MUST BE SPECIFIED.
510                                   THE ADD COMMAND IS  MEANINGFUL  ONLY  FOR  UNITS  THAT  WERE
511                                   PREVIOUSLY DROPPED.
512
513
514                                   6.3.6 DROP COMMAND
515
516                                   ****************************************************************
517                                   DRO(P)/UNITS:<UNIT-LIST>
518                                   ****************************************************************
519
520
521                                   6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
522
523                                   <UNIT-LIST> IS AS IN THE RESTART COMMAND.
524
525
526                                   6.3.6.2 EFFECT OF DROP COMMAND
527
528                                   THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING.  THE UNITS
529                                   WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR  START
530                                   COMMAND.   THE  UNITS  SWITCH MUST BE ENTERED.  THIS COMMAND
531                                   MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.
532
533
534                                   6.3.7 PRINT COMMAND
535
536                                   ****************************************************************
537                                   PRI(NT)
538                                   ****************************************************************
539
540
541                                   6.3.7.1 EFFECT OF PRINT COMMAND
542
543                                   THE TOTAL NUMBER OF ERRORS FOR  EACH  UNIT  SINCE  THE  LAST
544                                   START  OR  RESTART  COMMAND  ARE  PRINTED.  THE ISR (INHIBIT
545                                   STATISTICAL REPORTING) FLAG IS CLEARED.
546
547
548                                   6.3.8 DISPLAY COMMAND
549
550                                   ****************************************************************
551                                   DIS(PLAY)/UNITS:<UNIT-LIST>
552                                   ****************************************************************
553
554
555                                   6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
556
557                                   <UNIT-LIST> IS AS IN THE RESTART COMMAND.
```

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

614
615 1. DEVICE CSR ADDRESS : (0) 160020?
616
617 THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE
618 ON THE LSI-BUS. THE ALLOWABLE RANGE IS 160020-177760
619 (OCTAL), AND THE DEFAULT VALUE IS 160020.
620
621 2. DEVICE VECTOR ADDRESS : (0) 300 ?
622
623 THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR FOR THIS
624 DEVICE. THE ALLOWABLE RANGE IS 000-674 (OCTAL), AND THE
625 DEFAULT VALUE IS 300.
626
627 3. DEVICE PRIORITY LEVEL : (0) 4 ?
628
629 THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS OF
630 THIS DEVICE WILL BE EXECUTED. THE ALLOWABLE RANGE IS 0-7,
631 AND THE DEFAULT VALUE IS 4.
632
633 4. BOARD TYPE (0=M8064, 1=M8053-V35, 2=M8053-EIA) : (0) 0 ?
634
635 THIS IS THE TYPE OF DMV-11 CURRENTLY INSTALLED. NOTE THAT THE
636 M8053 IS SWITCH SELECTABLE BETWEEN V.35 AND EIA.
637
638 5. TURNAROUND CONNECTOR TYPE .
639 (0=M3254/M3255, 1=INTEGRAL MODEM CABLE, 2=EIA CABLE,
640 3=V.35 CABLE, 4=NONE) : (0) 0 ?
641
642 THIS IS THE TYPE OF EXTERNAL LOOPBACK CONNECTOR BEING USED.
643 IF NO LOOPBACK CONNECTOR IS PRESENT (4), THE EXTERNAL LOOP-
644 BACK TESTS WILL ALL BE RUN USING TTL-INTERNAL LOOPBACK.
645
646
647 6.3.13 SOFTWARE PARAMETERS
648
649 NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY THIS TEST.
650
651
652 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
653
654 THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY
655 THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.
656
657 AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE
658 NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES.
659 ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A
660 ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER
661 QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.
662
663 ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN
664 ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN
665 LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR
666 QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE
667 GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH
668 THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED.
669 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
```

PROGRAM DOCUMENT

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).

```
736
737
738                                 7.0 TEST DESCRIPTIONS
739
740
741                                 ;*****************************************************************************
742                                 ;*      TEST 1  <RX DATA FLUSHING TEST>
743                                 ;*
744                                 ;*  IN BCP MODE/HALF DUPLEX IT IS DESIRABLE TO HAVE THE ABILITY TO FLUSH
745                                 ;*  THE USYRT OF ITS CRC CHARACTERS. THIS FLUSHING IS ACCOMPLISHED BY WRITING
746                                 ;*  TO THE VIA SHIFT REGISTER.
747                                 ;*  THIS TEST VERIFIES THAT WHEN THE VIA SR IS WRITTEN INTO, 8 PULSES WILL
748                                 ;*  BE GENERATED AT THE CB1 PIN (WHICH DIRECTLY FEEDS THE CHARACTER FIFO).
749                                 ;*
750                                 ;*****************************************************************************
751
752
753                                 ;*****************************************************************************
754                                 ;*      TEST 2  <INTEGRAL MODEM INTERFACE TEST>
755                                 ;*
756                                 ;*  THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM AND A MESSAGE IS
757                                 ;*  TRANSMITTED,  RECEIVED,  AND CHECKED USING A TURNAROUND CONNECTOR ON
758                                 ;*  THE  BOARD  OR  AT THE END OF A CABLE. THE FOLLOWING MESSAGE WILL BE
759                                 ;*  SENT IN BCP MODE WITH CRC-16 SPECIFIED:
760                                 ;*
761                                 ;*      SYNC SYNC 000 125 252 377 000 CRC1 CRC2 SYNC
762                                 ;*
763                                 ;*  IF  THE  P-TABLE  FOR  THE  CURRENT  UNIT INDICATES THAT NO EXTERNAL
764                                 ;*  TURNAROUND IS PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
765                                 ;*****************************************************************************
766
767
768                                 ;*****************************************************************************
769                                 ;*      TEST 3  <DATA TEST -- BCP XLB CRC-16>
770                                 ;*
771                                 ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
772                                 ;*  RECEIVE  IN  BCP  MODE  WITH  CRC-16  ERROR  DETECTION THE FOLLOWING
773                                 ;*  MESSAGE:
774                                 ;*
775                                 ;*     125  252  000 377 001 002 004 010 020 040 100 200 376 375 373 367
776                                 ;*     357 337 277 177
777                                 ;*
778                                 ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY 3 SYNC CHARACTERS AND REPEATED
779                                 ;*  THREE  TIMES WITH CRC'S FOLLOWING EACH ONE. THE LAST TRANSMISSION OF
780                                 ;*  THE  CRC  WILL BE FOLLOWED BY SEVERAL SYNC CHARACTERS BEFORE DROPPING
781                                 ;*  TXE & RXE. 8-BIT CHARACTER LENGTHS ARE ALSO UTILIZED.
782                                 ;*
783                                 ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
784                                 ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
785                                 ;*****************************************************************************
786
787
788                                 ;*****************************************************************************
789                                 ;*      TEST 4  <DATA TEST -- BCP XLB ODD VRC>
790                                 ;*
791                                 ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
```

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


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


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


                        ;**************************************************************************
                        ;*      TEST 7  <DATA TEST -- BOP XLB CRC-CCITT-0>
                        ;*
                        ;*  IF  XLB  IS  SPECIFIED  IN  THE  P-TABLE,  THIS TEST WILL TRANSMIT &
```

```
                        ;*  RECEIVE  IN  BOP MODE WITH CRC-CCITT-0 ERROR DETECTION THE FOLLOWING
                        ;*  SHORT MESSAGE:  125  252  000  377  001
                        ;*
                        ;*  THIS  MESSAGE  WILL  BE  PRECEDED  BY FLAG CHARACTERS AND REPEATED
                        ;*  THREE  TIMES WITH CRC AND FLAG'S FOLLOWING EACH ONE. 8-BIT CHARACTER
                        ;*  LENGTHS ARE ALSO UTILIZED.
                        ;*
                        ;*  IF XLB WAS NOT SPECIFIED (AND/OR BOARD TYPE IS M8064), THIS TEST MAY BE RUN
                        ;*  USING INTERNAL LOOPBACK (TTLOOP=1).
                        ;*********************************************************************************


                        ;*********************************************************************************
                        ;*      TEST 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 M3254 TEST CONNECTOR INSTALLED,
                        ;*  THE DMV-11 IN INITIALIZED AGAIN, (TTL LOOPBACK IS CLEARED), AND
                        ;*  THE FOLLOWING CHECKS ARE PERFORMED :
                        ;*  - RING, 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.
                        ;*********************************************************************************
```

```
;*****************************************************************************
;*      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.
;*****************************************************************************
```

```
905
906                                  8.0 ERROR INFORMATION
907
908
909                                  8.1 ERROR REPORTING
910
911                                  ERRORS  ARE  REPORTED  BY  THE PROGRAM AS THEY OCCUR (IF NOT
912                                  INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR
913                                  ERROR  REPORT  FORMAT,  AND CONSISTS OF A DESCRIPTION OF THE
914                                  ERROR, THE TEST NUMBER, SUBTEST  NUMBER,  PC  OF  THE  ERROR
915                                  CALL,  DEVICE  ADDRESS,  AND  BASIC  AND  EXTENDED  ERROR
916                                  INFORMATION.
917
918                                  THE FOLLOWING EXAMPLE PROVIDES A TYPICAL ERROR REPORT, WHICH
919                                  DESCRIBES A "MASTER CLEAR FAILURE" ERROR, AND PROVIDES THE PC
920                                  OF THE EPROR CALL AND THE DEVICE REGISTER CONTENTS :
921
922                                    CVDME DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
923                                    MASTER CLEAR FAILURE
924
925                                      THE CONTENTS OF ALL BYTE SELECT REG'S ARE:
926                                    BSEL0   BSEL1   BSEL2   BSEL3
927                                     000     000     000     000
928                                        BSEL4   BSEL5   BSEL6   BSEL7
929                                         000     000     121     000
930                                    BSEL10  BSEL11  BSEL12  BSEL13
931                                     000     000     000     000
932                                        BSEL14  BSEL15  BSEL16  BSEL17
933                                         000     000     000     000
934
935                                  FOR OTHER ERRORS, THE REPORT  MAY  BE  MORE  EXTENSIVE,  AND
936                                  REQUIRE ADDITIONAL DATA TO BE REPORTED.
937
938                                  IF  EXTENDED  ERROR INFORMATION HAD BEEN INHIBITED USING THE
939                                  IXE FLAG PRIOR TO RUNNING THE TEST, THE  ABOVE  ERROR  WOULD
940                                  HAVE BEEN REPORTED IN THE FOLLOWING SHORTENED FORM :
941
942                                    CVDME DVC FTL ERR 00001 ON UNIT 00 TST 002 SUB 000 PC: 021122
943                                    MASTER CLEAR FAILURE
944
945                                          0
946
```

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

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

                                .=2000

                                .MCALL  SVC
002000                          SVC     ; INITIALIZE SUPERVISOR MACROS

002000                          BGNMOD  LU1MOD

                        $LSTIN= 1
                        $LSTTAG= 1
                        SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
                        SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
                        SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
                        SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
                        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,C,0,15.,0
002000                                                          L$NAME::
002000  103                                                             .ASCII /C/
002001  126                                                             .ASCII /V/
002002  104                                                             .ASCII /D/
002003  115                                                             .ASCII /M/
002004  105                                                             .ASCII /E/
002005  000                                                             .BYTE   0
002006  000                                                             .BYTE   0
002007  000                                                             .BYTE   0
002010                                                          L$REV::
002010  103                                                             .ASCII  /C/
002011                                                          L$DEPO::
002011  060                                                             .ASCII  /0/
002012                                                          L$UNIT::
002012  000000                                                          .WORD   0
002014                                                          L$TIML::
002014  000017                                                          .WORD   15.
002016                                                          L$HPCP::
002016  033652                                                          .WORD   L$HARD
002020                                                          L$SPCP::
002020  000000                                                          .WORD   0
002022                                                          L$HPTP::
002022  002150                                                          .WORD   L$HW
002024                                                          L$SPTP::
002024  000000                                                          .WORD   0
002026                                                          L$LADP::
002026  034514                                                          .WORD   L$LAST
002030                                                          L$STA::
002030  000000                                                          .WORD   0
002032                                                          L$CO::
002032  000000                                                          .WORD   0
002034                                                          L$DTYP::
002034  000000                                                          .WORD   0
002036                                                          L$APT::
002036  000000                                                          .WORD   0
002040                                                          L$DTP::
002040  002124                                                          .WORD   L$DISPATCH
002042                                                          L$PRIO::
002042  000000                                                          .WORD   0
002044                                                          L$ENVI::
002044  000000                                                          .WORD   0
002046                                                          L$EXP1::
002046  000000                                                          .WORD   0
002050                                                          L$MREV::
002050     003                                                          .BYTE   C$REVISION
002051     003                                                          .BYTE   C$EDIT
```

```
1032  002052
1033  002052  000000                    L$EF::
                                                .WORD   0
1034  002054  000000                            .WORD   0
1035  002056                            L$SPC::
1036  002056  000000                            .WORD   0
1037  002060                            L$DEVP::
1038  002060  003264                            .WORD   L$DVTYP
1039  002062                            L$REPP::
1040  002062  000000                            .WORD   0
1041  002064                            L$EXP4::
1042  002064  000000                            .WORD   0
1043  002066                            L$EXP5::
1044  002066  000000                            .WORD   0
1045  002070                            L$AUT::
1046  002070  023230                            .WORD   L$AU
1047  002072                            L$DUT::
1048  002072  023224                            .WORD   L$DU
1049  002074                            L$LUN::
1050  002074  000000                            .WORD   0
1051  002076                            L$DESP::
1052  002076  003304                            .WORD   L$DESC
1053  002100                            L$LOAD::
1054  002100  104035                            EMT     E$LOAD
1055  002102                            L$ETP::
1056  002102  002172                            .WORD   L$ERRTBL
1057  002104                            L$ICP::
1058  002104  022554                            .WORD   L$INIT
1059  002106                            L$CCP::
1060  002106  023222                            .WORD   L$CLEAN
1061  002110                            L$ACP::
1062  002110  023076                            .WORD   L$AUTO
1063  002112                            L$PRT::
1064  002112  022546                            .WORD   L$PROT
1065  002114                            L$TEST::
1066  002114  000000                            .WORD   0
1067  002116                            L$DLY:.
1068  002116  000000                            .WORD   0
1069  002120                            L$HIME::
1070  002120  000000                            .WORD   0
1071
1072
1073                        .EVEN
```

M2


```
                                .SBTTL  DISPATCH TABLE

                                SLASH
                                ;;//////////////////////////////////////////////////////////////////
                                ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
                                ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
                                SLASH
                                ;;//////////////////////////////////////////////////////////////////

                                        DISPATCH 9.
002122  000011                                                         .WORD   9
002124                                                          L$DISPATCH::
002124  023232                                                          .WORD   T1
002126  023532                                                          .WORD   T2
002130  026236                                                          .WORD   T3
002132  027114                                                          .WORD   T4
002134  027656                                                          .WORD   T5
002136  030420                                                          .WORD   T6
002140  031102                                                          .WORD   T7
002142  031564                                                          .WORD   T8
002144  032530                                                          .WORD   T9
```

```
                        .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          ;M3254&M3255 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::

                                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
```

```
1192            000140                  PRIO3== 140
1193            000100                  PRIO2== 100
1194            000040                  PRIO1== 40
1195            000000                  PRIO0== 0
1196                                    ;
1197                                    ;OPERATOR FLAG BITS
1198                                    ;
1199            000004                  EVL==       4
1200            000010                  LOT==      10
1201            000020                  ADR==      20
1202            000040                  IDU==      40
1203            000100                  ISR==     100
1204            000200                  UAM==     200
1205            000400                  BOE==     400
1206            001000                  PNT==    1000
1207            002000                  PRI==    2000
1208            004000                  IXE==    4000
1209            010000                  IBE==   10000
1210            020000                  IER==   20000
1211            040000                  LOE==   40000
1212            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
```

```
1306            002000                  TXAB    = BIT10         ;TRANSMIT ABORT
1307            001000                  TXEOM   = BIT9          ;TRANSMIT END-OF-MESSAGE
1308            000400                  TXSOM   = BIT8          ;TRANSMIT START-OF-MESSAGE
1309
1310                                    ;;**************************************************************************
1311                                    ;* USYRT "SYNC/SECONDARY ADDRESS" REGISTER
1312                                    ;;**************************************************************************
1313
1314            120404                  PCSARL  = 120404        ;ADDRESS OF THIS REG
1315            000226                  SYNCH   = 226           ;STANDARD SYNCH CHARACTER
1316
1317                                    ;;**************************************************************************
1318                                    ;* USYRT "MODE CONTROL"
1319                                    ;;**************************************************************************
1320
1321            120405                  PCSARH  = 120405        ;ADDRESS OF THIS REG
1322
1323                                            ;BIT DEFINITIONS ON BYTE BASIS:
1324
1325            000200                  APA     = BIT7          ;"ALL PARTIES ADDRESS" ENABLE
1326            000100                  PROTO   = BIT6          ;SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
1327            000040                  STRIP   = BIT5          ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1328            000020                  SECAD   = BIT4          ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1329            000010                  IDLE    = BIT3          ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1330            000007                  XYZ     = BIT2!BIT1!BIT0 ;CRC/PARITY SELECTION CONTROL
1331
1332                                            ;BIT DEFINITIONS ON WORD BASIS:
1333
1334          100000                    APAD    = BIT15         ;"ALL PARTIES ADDRESS" ENABLE
1335          040000                    DDCMP   = BIT14         ;CODE FOR DDCMP MODE
1336          020000                    STRIPS  = BIT13         ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1337          010000                    SECADR  = BIT12         ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1338          004000                    IDLES   = BIT11         ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1339          000400                    CRCOS   = BIT8          ;CODE FOR CRC-CCITT-0 SELECTION
1340          001400                    CRC16   = BIT9!BIT8     ;CODE FOR CRC-16 SELECTION
1341          003400                    NOCHK   = BIT10!BIT9!BIT8 ;CODE FOR NO ERROR CHECKING
1342          002400                    EVRC    = BIT10!BIT8    ;CODE FOR VRC EVEN CHECK
1343          002000                    OVRC    = BIT10         ;CODE FOR VRC ODD CHECK
1344
1345                                    ;;**************************************************************************
1346                                    ;* USYRT "DATA LENGTH SELECT" REGISTER
1347                                    ;;**************************************************************************
1348
1349          120407                    PCR     = 120407        ;ADDRESS OF THIS REG
1350
1351                                            ;BIT DEFINITIONS:
1352
1353          000340                    TXDL    = BIT7!BIT6!BIT5 ;TRANSMIT DATA LENGTH SELECTION
1354          000020                    EXADD   = BIT4          ;EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
1355          000010                    EXCON   = BIT3          ;EXTENDED CONTROL FIELD -- NOT USED OR TESTED
1356          000007                    RXDL    = BIT2!BIT1!BIT0 ;RECEIVER DATA LENGTH SELECTION
1357
1358                                    ;;**************************************************************************
1359                                    ;* USYRT STATUS REGISTER (ADDR. A400)
1360                                    ;;**************************************************************************
1361          122000                    USTATR = 122000         ;USYRT STATUS REGISTER ADDRESS = A400 (HEX)
```

```
1362
1363                                    ;BIT DEFINITIONS:
1364
1365            000200          RDA     = BIT7          ;RECEIVER DATA AVAILABLE
1366            000100          TBMT    = BIT6          ;TRANSMITTER BUFFER EMPTY
1367            000040          RXACT   = BIT5          ;RECEIVER ACTIVE
1368            000020          RSA     = BIT4          ;RECEIVER STATUS AVAILABLE
1369            000010          TSO     = BIT3          ;TRANSMITTER SERIAL OUTPUT
1370            000004          TXACT   = BIT2          ;TRANSMITTER ACTIVE
1371            000002          TXU     = BIT1          ;TRANSMITTER UNDERRUN
1372            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                  ;"RXEN." -- 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
```

```
1429
1430                            ;       INITIALIZED TO 001 (HEX = 01) -- PORT A IS READ ONLY (EXCEPT FOR
1431                            ;                                       BIT0 WHICH ENABLES USYRT INTERNAL LOOPBACK).
1432
1433
1434
1435                            ;;*****************************************************************************
1436                            ;* TIMER 1 LOW ORDER (LATCH & COUNTER) -- "T1L-L" & "T1C-L" -- WRITE & READ
1437                            ;;*****************************************************************************
1438
1439    120004                  VIAT1A  = 120004                ;ADDRESS OF THIS REGISTER -- HEX = A0X4
1440
1441                            ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1442                            ;  WHEN READING, LOW ORDER COUNTER IS READ.
1443
1444
1445
1446                            ;;*****************************************************************************
1447                            ;* TIMER 1 HIGH ORDER COUNTER & TRIGGER -- "T1L-H AND TRIGGER" & "T1C-H"
1448                            ;*      -- WRITE & READ
1449                            ;;*****************************************************************************
1450
1451    120005                  VIAT1B  = 120005                ;ADDRESS OF THIS REGISTER -- HEX = A0X5
1452
1453                            ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1454                            ;                ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1455
1456                            ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1457
1458
1459
1460                            ;;*****************************************************************************
1461                            ;* TIMER 1 LOW ORDER LATCH -- "T1L-L" -- READ/WRITE
1462                            ;;*****************************************************************************
1463
1464    120006                  VIAT1C  = 120006                ;ADDRESS OF THIS REGISTER -- HEX = A0X6
1465
1466                            ;  THE LOW ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1467                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1468
1469
1470
1471                            ;;*****************************************************************************
1472                            ;* TIMER 1 HIGH ORDER LATCH -- "T1L-H" -- READ/WRITE
1473                            ;;*****************************************************************************
1474
1475    120007                  VIAT1D  = 120007                ;ADDRESS OF THIS REGISTER -- HEX = A0X7
1476
1477                            ;  THE HIGH ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1478                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1479
1480
1481
1482                            ;;*****************************************************************************
1483                            ;* TIMER 2 LOW ORDER (LATCH & COUNTER) -- "T2L-L" & "T2C-L" -- WRITE & READ
1484                            ;;*****************************************************************************
```

```
1485
1486          120010                  VIAT2A  = 120010                ;ADDRESS OF THIS REGISTER -- HEX = A0X8
1487
1488                                  ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1489                                  ;  WHEN READING, LOW ORDER COUNTER IS READ.
1490
1491
1492
1493                                  ;;*************************************************************************
1494                                  ;* TIMER 2 HIGH ORDER COUNTER & TRIGGER -- "T2L-H AND TRIGGER" & "T2C-H"
1495                                  ;*      -- WRITE & READ
1496                                  ;;*************************************************************************
1497
1498          120011                  VIAT2B  = 120011                ;ADDRESS OF THIS REGISTER -- HEX = A0X9
1499
1500                                  ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1501                                  ;                 ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1502
1503                                  ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1504
1505                                  ;;*************************************************************************
1506                                  ;* SHIFT REGISTER -- "SR" -- READ/WRITE
1507                                  ;;*************************************************************************
1508
1509          120012                  VIASR   = 120012                ;ADDRESS OF THIS REGISTER -- HEX = A0XA
1510
1511                                  ;  SHIFTING IS CONTROLLED BY THE SETTING OF VIASRC (ACR2 ---> ACR4) IN VIAACR
1512
1513
1514
1515                                  ;;*************************************************************************
1516                                  ;* AUXILIARY CONTROL REGISTER -- "ACR" -- READ/WRITE
1517                                  ;;*************************************************************************
1518
1519          120013                  VIAACR  = 120013                ;ADDRESS OF THIS REGISTER -- HEX = A0XB
1520
1521          000300                  T1MODE  = BIT7!BIT6             ;CONTROL THE MODE OF TIMER # 1
1522
1523                                                  ;BIT 7:
1524                                                  ; 0     PB7 DISABLED -- ONLY T1TO IN VIAIFR REFLECTS TIMEOUT
1525                                                  ; 1     PB7 & T1TO REFLECT TIMEOUT
1526
1527                                                  ;BIT 6:
1528                                                  ; 0     TIMER 1 IN ONE-SHOT MODE
1529                                                  ; 1     TIMER 1 IN CONTINUOUS SQUARE WAVE MODE
1530
1531          000040                  T2MODE  = BIT5                  ;CONTROLS THE MODE OF TIMER # 1
1532
1533                                                  ; 0     PULSE COUNTING MODE
1534                                                  ; 1     INTERVAL TIMER MODE
1535
1536          000034                  SRMODE  = BIT4!BIT3!BIT2        ;CONTROLS THE MODE OF THE SHIFT REGISTER
1537
1538                                                  ; 0     SR DISABLED
1539                                                  ; 1     SHIFT IN UNDER CONTROL OF T2, SHFT PULSES GEN'D ON CB1
1540                                                  ; 2     SHIFT IN AT SYS. CLOCK RATE, SHFT PULSES GEN'D ON CB1
```

```
1541                                    ; 3     SHIFT IN UNDER CONTROL OF EXTERNAL INPUT PULSES
1542                                    ; 4     SHIFT OUT -- FREE RUNNING -- RATE CONTROLLED BY T2
1543                                    ; 5     SHIFT OUT -- RATE CONTROLLED BY T2 -- PULSES ON CB1
1544                                    ; 6     SHIFT OUT -- SYS. CLOCK RATE -- PULSES ON CB1
1545                                    ; 7     SHIFT OUT -- UNDER CONTROL OF PULSES APPLIED TO CB1
1546
1547        000002          PBLENB  = BIT1                  ;PB LATCH CONTROL -- 1 ENABLES LATCH
1548        000001          PALENB  = BIT0                  ;PA LATCH CONTROL -- 1 ENABLES LATCH
1549
1550
1551
1552
1553                        ;;*************************************************************************
1554                        ;* PERIPHERAL CONTROL REGISTER -- "PCR" -- READ/WRITE
1555                        ;;*************************************************************************
1556
1557        120014          VIAPCR  = 120014                ;ADDRESS OF THIS REGISTER -- HEX = A0XC
1558
1559        000340          CB2CTL  = BIT7!BIT6!BIT5        ;CB2 MODE SELECT
1560        000020          CB1CTL  = BIT4                  ;CB1 MODE SELECT
1561        000016          CA2CTL  = BIT3!BIT2!BIT1        ;CA2 MODE SELECT
1562        000001          CA1CTL  = BIT0                  ;CA1 MODE SELECT
1563
1564
1565
1566                        ;;*************************************************************************
1567                        ;* INTERRUPT FLAG REGISTER -- "IFR" -- READ ONLY
1568                        ;;*************************************************************************
1569
1570        120015          VIAIFR  = 120015                ;ADDRESS OF THIS REGISTER -- HEX = A0XD
1571
1572        000200          FLGINQ  = BIT7                  ;SET WHEN A FLAG IN THIS REG. GOES HIGH AND
1573                                                        ;ITS CORRESPONDING BIT IN VIAIER IS SET.
1574                                                        ;(I.E. VIAIER IS THE ENABLE REGISTER FOR THE
1575                                                        ;FOR THE SETTING OF IRQ AND THE ISSUEANCE OF
1576                                                        ;AN INTERRUPT TO THE 6502 WHEN IRQ IS SET.)
1577
1578        000100          FLGT1   = BIT6                  ;TIMEOUT OF TIMER 1
1579        000040          FLGT2   = BIT5                  ;TIMEOUT OF TIMER 2
1580        000020          FLGCB1  = BIT4                  ;ACTIVE TRANSITION OF PIN 18 (CB1)
1581        000010          FLGCB2  = BIT3                  ;ACTIVE TRANSITION OF PIN 19 (CB2)
1582        000004          FLGSR   = BIT2                  ;COMPLETION OF 8 SHIFTS
1583        000002          FLGCA1  = BIT1                  ;ACTIVE TRANSITION OF PIN 40 (CA1)
1584        000001          FLGCA2  = BIT0                  ;ACTIVE TRANSITION OF PIN 39 (CA2)
1585
1586
1587
1588                        ;;*************************************************************************
1589                        ;* INTERRUPT ENABLE REGISTER -- "IER" -- READ/WRITE
1590                        ;;*************************************************************************
1591
1592        120016          VIAIER  = 120016                ;ADDRESS OF THIS REGISTER -- HEX = A0XE
1593
1594        000200          INTSC   = BIT7                  ;CONTROLS THE SETTING OR CLEARING OF BITS IN
1595                                                        ;THE REST OF IER.  IF = 0  THE OTHER BITS IN
1596                                                        ;THIS REG., IF SET, WILL CLEAR THEIR RESPECTIVE
```

```
1597                                                    ;BITS IN THE INT. ENAB. REG..  IF = 1, THE
1598                                                    ;RESPECTIVE BITS WILL BE SET.
1599
1600                            ;  WHEN WRITING THIS REG., THE COMMENT ABOVE HOLDS.
1601
1602                            ;  WHEN READING THIS REG., THE CURRENT STATE OF THE INT. ENABLE REG. IS RETURNED.
1603
1604                            ;  THE BIT ASSIGNMENTS ARE THE SAME AS FOR VIAIFR AS DEFINED ABOVE.
1605
1606
1607
1608                            ;;*****************************************************************************
1609                            ;* OUTPUT REGISTER A -- "ORA" -- READ ONLY (OR READ/WRITE UNDER CONTROL OF "DDPA")
1610                            ;;*****************************************************************************
1611
1612          120017            VIAORA  = 120017                ;ADDRESS OF THIS REGISTER -- HEX = A0XF
1613
1614                            ;  THIS ADDRESS ACCESSES THE SAME DATA AS "VIAMS" EXCEPT THAT NO "HANDSHAKING"
1615                            ;  WILL TAKE PLACE (I.E. THERE IS NO CHANGE IN IRQ OR CA2 AS A RESULT OF
1616                            ;  READING ORA THROUGH THIS ADDRESS)
1617
1618                            ;THE BIT ASSIGNMENTS ARE THE SAME AS FOR "VIAMS" ABOVE.
1619
1620
1621
```

```
1622                                    .SBTTL  REGISTER DEFINITIONS -- MISC
1623
1624                                    ;;*****************************************************************************
1625                                    ;* SWITCH PACKS
1626                                    ;;*****************************************************************************
1627
1628            121000                  SWPBOT  = 121000                ;"BOOT ADDRESS" SWITCH PACK [A200]
1629            121400                  SWPDDCMP = 121400               ;"DDCMP ADDRESS" SWITCH PACK [A300]
1630
1631                                    ;MISCELLANEOUS EQUATES
1632
1633            100000                  TCCHEK  = BIT15         ;FLAG TO REQUEST M3254,5 CHECK
1634            001000                  RAMADR  = 001000        ;STARTING ADRS OF RAM PAGE 2 (ADRS 0200 HEX)
1635
1636            000002                  EIAV35  = BIT1                  ;SELECT V.35 OR EIA 423/232C
1637            000001                  INTGRL  = BIT0                  ;SELECT INTEGRAL MODEM
1638
1639            040000                  NORXEN  = BIT14         ;KILL RXEN DURING "INITRN"
1640            001000                  NOLOOP  = BIT9          ;KILL TTLOOP DURING "INITRN"
1641
1642            000200                  NCTBMT  = BIT7          ;DISABLE INITIAL TBMT=0 CHECK IN TXCHAR
1643
1644            100000                  NOCRDA  = BIT15         ;DISABLE INITIAL RDA=0 CHECK IN RXCHAR
1645            040000                  NFCRDA  = BIT14         ;DISABLE FINAL RDA=1 CHECK IN RXCHAR
1646            020000                  NCRACT  = BIT13         ;DISABLE RXACT=1 CHECK AFTER CLOCKING (RXCHAR)
1647
```

```
                        .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  000000          ERRTYP::        .WORD   0
002174  000000          ERR@R::         .WORD   0
002176  000000          ERRMSG::        .WORD   0
002200  000000          ERRBLK::        .WORD   0

                        ;;*****************************************************************************
                        ;* STORAGE FOR DEVICE REGISTERS
                        ;;*****************************************************************************
002202                  USR0:                           ;STORAGE FOR DEVICE CSR REGISTERS
002202  000000          BSR0:   .WORD   0
002204                  USR2:
002204  000000          BSR1:   .WORD   0
002206                  USR4:
002206  000000          BSR2:   .WORD   0
002210                  USR6:
002210  000000          BSR3:   .WORD   0
002212                  USR10:
002212  000000          BSR4:   .WORD   0
002214                  USR12:
002214  000000          BSR5:   .WORD   0
002216                  USR14:
002216  000000          BSR6:   .WORD   0
002220                  USR16:
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
```

```
1698                                ;*************************************************************************
1699                                ;* MISCELLANEOUS STORAGE
1700                                ;*************************************************************************
1701  002322  000000  TDATA:  .WORD   0          ;TEST DATA
1702  002324  000000  GDATA:  .WORD   0          ;GOOD DATA
1703  002326  000000  BDATA:  .WORD   0          ;BAD DATA
1704  002330  000000  XDATA:  .WORD   0          ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
1705  002332  000000  SCRACH: .WORD   0          ;GEN'L PURPOSE SCRATCH WORD
1706  002334  000000  LOGDEV: .WORD   0          ;LOGICAL DEVICE NUMBER
1707  002336  000000  REGNUM: .WORD   0          ;CONTAINS A DEVICE REGISTER NUMBER
1708  002340  000000  PSTACK: .WORD   0          ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
1709  002342  000000  PRIOR:  .WORD   0          ;CPU PRIORITY FOR PRINTOUT
1710  002344  000000  SUBRPC: .WORD   0          ;PC OF SUBR CALL FOR ERROR REPORTS
1711  002346  000000  INTFLG: .WORD   0          ;INTERRUPT RECEIVED FLAGS
                                                 ;  BIT 0 FOR TX, BIT 1 FOR RCV
1712
1713  002350  000000  ERRFLG: .WORD   0          ;SUBROUTINE ERROR FLAG
1714  002352  000000  TIMFLG: .WORD   0          ;EVENT TIME-OUT FLAG
1715  002354  000000  RETADR: .WORD   0          ;SUBR ERROR RETURN ADDRESS
1716  002356  000000  REDBYT: .WORD   0          ;LO BYTE CONTAINS BYTE READ FROM LU REG
1717  002360  000000  WRIBYT: .WORD   0          ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
1718  002362  000000  LOADAT: .WORD   0          ;CONTAINS TEST DATA LOADED INTO REG
1719  002364  000000  GOODAT: .WORD   0          ;STORAGE FOR EXPECTED DATA
1720  002366  000000  BADDAT: .WORD   0          ;STORAGE FOR ACTUAL DATA
1721  002370  000000  FRSTIM: .WORD   0          ;FLAG=0 IF PROGRAM JUST LOADED
1722  002372  000000  SAVE4:  .WORD   0          ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
1723  002374  000000  SAVE6:  .WORD   0          ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
1724  002376  000000  ERROR1: .WORD   0          ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
1725  002400  000000  CHPTYP: .WORD   0          ;USYRT CHIP TYPE, =0 FOR SMC, ELSE =1
1726  002402  000000  SAVLEN: .WORD   0          ;SAVED TX AND RCV CHAR LENGTHS
1727  002404  000000  DEVMAP: .WORD   0          ;BIT MAP OF ACTIVE DEVICES
1728  002406  000000  DEVPTR: .WORD   0          ;DEVICE MAP BIT POINTER
1729  002410  000000  UNIT:   .WORD   0          ;CONTAINS UNIT NO. (1 TO N)
1730  002412  000000  STARES: .WORD   0          ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
1731  002414  000000  TSTNUM: .WORD   0          ;NO. OF CURRENT TEST (FOR SOME TESTS)
1732
```

```
1733
1734  002416                  ;;********** CURRENT DEVICE PARAMETERS **************
1735  002416                  BSEL0:
1736  002416  160020          SEL0:   MPCSR:  .WORD   160020          ;POINTER TO DMV11 CSR'S
1737  002420  160021                  BSEL1:  .WORD   160021          ;POINTER TO BSEL1
1738  002422                  BSEL2:
1739  002422  160022          SEL2:   BSEL3:  .WORD   160022          ;POINTER TO SEL2
1740  002424  160023                  BSEL3:  .WORD   160023          ;POINTER TO BSEL3
1741  002426                  BSEL4:
1742  002426  160024          SEL4:   BSEL5:  .WORD   160024          ;POINTER TO SEL4
1743  002430  160025                  BSEL5:  .WORD   160025          ;POINTER TO BSEL5
1744  002432                  BSEL6:
1745  002432  160026          SEL6:   BSEL7:  .WORD   160026          ;POINTER TO SEL6
1746  002434  160027                  BSEL7:  .WORD   160027          ;POINTER TO BSEL7
1747  002436                  BSEL10:
1748  002436  160030          SEL10:  BSEL11: .WORD   160030          ;POINTER TO SEL10
1749  002440  160031                  BSEL11: .WORD   160031          ;POINTER TO BSEL11
1750  002442                  BSEL12:
1751  002442  160032          SEL12:  BSEL13: .WORD   160032          ;POINTER TO SEL12
1752  002444  160033                  BSEL13: .WORD   160033          ;POINTER TO BSEL13
1753  002446                  BSEL14:
1754  002446  160034          SEL14:  BSEL15: .WORD   160034          ;POINTER TO SEL14
1755  002450  160035                  BSEL15: .WORD   160035          ;POINTER TO BSEL15
1756  002452                  BSEL16:
1757  002452  160036          SEL16:  BSEL17: .WORD   160036          ;POINTER TO SEL16
1758  002454  160037                  BSEL17: .WORD   160037          ;POINTER TO BSEL17
1759
1760  002456  000300          MPIVEC: .WORD      300          ;DMV11 INPUT INTERRUPT VECTOR
1761  002460  000304          MPOVEC: .WORD      304          ;DMV11 OUTPUT INTERRUPT VECTOR
1762  002462  000240          MPRIOR: .WORD      240          ;DMV11 DEVICE PRIORITY
1763  002464  000000          LUSWI1: .WORD        0          ;LINE UNIT SWITCH PACK #1
1764  002466  000000          LUSWI2: .WORD        0          ;LINE UNIT SWITCH PACK #2
1765  002470  000000          BRDTYP: .WORD        0          ;0=M8064, 1=M8053/V.35,2=M8053/EIA
1766  002472  000000          TSTCON: .WORD        0          ;TEST CONNECTOR INDICATOR
1767  002474  000001          BDRATE: .WORD        1          ;BAUD RATE = 56 K
1768                                                          ;        0 = 19.2 K
1769                                                          ;        1 = 56 K
```

```
                                GLOBAL DATA SECTION

                                ;TABLE OF USYRT REGISTER ADDRESSES
002476  120400                 USYREG: .WORD   120400          ;ADDRESS OF RDSRL
002500  120401                         .WORD   120401          ;ADDRESS OF RDSRH
002502  120402                         .WORD   120402          ;ADDRESS OF TDSRL
002504  120403                         .WORD   120403          ;ADDRESS OF TDSRH
002506  120404                         .WORD   120404          ;ADDRESS OF PCSARL
002510  120405                         .WORD   120405          ;ADDRESS OF PCSARH
002512  120407                         .WORD   120407          ;ADDRESS OF PCR
002514  122000                         .WORD   122000          ;ADDRESS OF USYRT STATUS REG

                                ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
002516  000010                 REDDAT: .BLKB   8.

                                ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
002526  000000                 REG0:   .WORD   0
002530  000000                 REG1:   .WORD   0
002532  000000                 REG2:   .WORD   0
002534  000000                 REG3:   .WORD   0
002536  000000                 REG4:   .WORD   0
002540  000000                 REG5:   .WORD   0
002542  000000                 REG6:   .WORD   0
002544  000000                 REG7:   .WORD   0

                                ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
002546  000000                 TMP0:   .WORD   0
002550  000000                 TMP1:   .WORD   0
002552  000000                 TMP2:   .WORD   0
002554  000000                 TMP3:   .WORD   0
002556  000000                 TMP4:   .WORD   0
002560  000000                 TMP5:   .WORD   0
002562  000000                 TMP6:   .WORD   0
002564  000000                 TMP7:   .WORD   0

                                ;***** UNIBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
002566                         UPBITS:
002566      377                         .BYTE   377             ;MASK FOR RDBR
002567      000                         .BYTE   000             ;MASK FOR RDSR
002570      000                         .BYTE   000             ;MASK FOR TDBR
002571      360                         .BYTE   360             ;MASK FOR TDSR
002572      000                         .BYTE   000             ;MASK FOR SSAR
002573      000                         .BYTE   000             ;MASK FOR PCSAR
002574      347                         .BYTE   347             ;MASK FOR PCR

002575      200                 TDSRNRW: .BYTE  200             ;TDSR NON-R/W BITS
```

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

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

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

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

```
 1870  002652  002                    .BYTE   002
 1871  002653  004                    .BYTE   004
 1872  002654  010                    .BYTE   010
 1873  002655  020                    .BYTE   020
 1874  002656  040                    .BYTE   040
 1875  002657  100                    .BYTE   100
 1876  002660  200                    .BYTE   200
 1877  002661  376                    .BYTE   376
 1878  002662  375                    .BYTE   375
 1879  002663  373                    .BYTE   373
 1880  002664  367                    .BYTE   367
 1881  002665  357                    .BYTE   357
 1882  002666  337                    .BYTE   337
 1883  002667  277                    .BYTE   277
 1884  002670  177                    .BYTE   177
 1885
 1886  002671  125                    .BYTE   125
 1887  002672  252                    .BYTE   252
 1888  002673  000                    .BYTE   000
 1889  002674  377                    .BYTE   377
 1890  002675  001                    .BYTE   001
 1891  002676  002                    .BYTE   002
 1892  002677  004                    .BYTE   004
 1893  002700  010                    .BYTE   010
 1894  002701  020                    .BYTE   020
 1895  002702  040                    .BYTE   040
 1896  002703  100                    .BYTE   100
 1897  002704  200                    .BYTE   200
 1898  002705  376                    .BYTE   376
 1899  002706  375                    .BYTE   375
 1900  002707  373                    .BYTE   373
 1901  002710  367                    .BYTE   367
 1902  002711  357                    .BYTE   357
 1903  002712  337                    .BYTE   337
 1904  002713  277                    .BYTE   277
 1905  002714  177                    .BYTE   177
 1906
 1907  002715  125                    .BYTE   125
 1908  002716  252                    .BYTE   252
 1909  002717  000                    .BYTE   000
 1910  002720  377                    .BYTE   377
 1911  002721  001                    .BYTE   001
 1912  002722  002                    .BYTE   002
 1913  002723  004                    .BYTE   004
 1914  002724  010                    .BYTE   010
 1915  002725  020                    .BYTE   020
 1916  002726  040                    .BYTE   040
 1917  002727  100                    .BYTE   100
 1918  002730  200                    .BYTE   200
 1919  002731  376                    .BYTE   376
 1920  002732  375                    .BYTE   375
 1921  002733  373                    .BYTE   373
 1922  002734  367                    .BYTE   367
 1923  002735  357                    .BYTE   357
 1924  002736  337                    .BYTE   337
 1925  002737  277                    .BYTE   277
```

```
1926  002740  177                 EPATX:  .BYTE   177
1927
1928                              ;***** DATA PATTERN I *****
1929  002741                      PATI:
1930  002741  000                         .BYTE   000
1931  002742  041                         .BYTE   041
1932  002743  102                         .BYTE   102
1933  002744  143                         .BYTE   143
1934  002745  204                         .BYTE   204
1935  002746  245                         .BYTE   245
1936  002747  306                         .BYTE   306
1937  002750  347                         .BYTE   347
1938  002751  000                         .BYTE   000
1939  002752  001                         .BYTE   001
1940  002753  002                         .BYTE   002
1941  002754  004                         .BYTE   004
1942  002755  040                         .BYTE   040
1943  002756  100                         .BYTE   100
1944  002757  200                         .BYTE   200
1945  002760  000                         .BYTE   000
1946  002761  346                         .BYTE   346
1947  002762  345                         .BYTE   345
1948  002763  343                         .BYTE   343
1949  002764  307                         .BYTE   307
1950  002765  247                         .BYTE   247
1951  002766  147                         .BYTE   147
1952  002767  347                         .BYTE   347
1953  002770  242                         .BYTE   242
1954  002771  105                         .BYTE   105
1955  002772  347                         .BYTE   347
1956  002773  010                         .BYTE   010
1957  002774  020                         .BYTE   020
1958  002775  367                         .BYTE   367
1959  002776  357                         .BYTE   357
1960  002777  030                         .BYTE   030
1961  003000  027                         .BYTE   027
1962  003001  377                         .BYTE   377
1963
1964                              ;***** DATA PATTERN J *****
1965  003002                      PATJ:
1966  003002  000                         .BYTE   000
1967  003003  000                         .BYTE   000
1968  003004  001                         .BYTE   001
1969  003005  002                         .BYTE   002
1970  003006  004                         .BYTE   004
1971  003007  020                         .BYTE   020
1972  003010  040                         .BYTE   040
1973  003011  010                         .BYTE   010
1974
1975                              ;***** DATA PATTERN K *****
1976  003012                      PATK:
1977  003012  000                         .BYTE   000
1978  003013  377                         .BYTE   377
1979  003014  376                         .BYTE   376
1980  003015  375                         .BYTE   375
1981  003016  373                         .BYTE   373
```

```
 1982  003017  376                    .BYTE   376
 1983  003020  177                    .BYTE   177
 1984  003021  377                    .BYTE   377
 1985  003022  000                    .BYTE   000
 1986  003023  001                    .BYTE   001
 1987  003024  002                    .BYTE   002
 1988  003025  004                    .BYTE   004
 1989  003026  010                    .BYTE   010
 1990  003027  200                    .BYTE   200
 1991  003030  125                    .BYTE   125
 1992  003031  252                    .BYTE   252
 1993  003032  000                    .BYTE   000
 1994
 1995                                 ;***** DATA PATTERN L *****
 1996  003033                         PATL:
 1997  003033  000                    .BYTE   000
 1998  003034  017                    .BYTE   017
 1999  003035  016                    .BYTE   016
 2000  003036  015                    .BYTE   015
 2001  003037  013                    .BYTE   013
 2002  003040  016                    .BYTE   016
 2003  003041  017                    .BYTE   017
 2004  003042  017                    .BYTE   017
 2005  003043  000                    .BYTE   000
 2006  003044  001                    .BYTE   001
 2007  003045  002                    .BYTE   002
 2008  003046  004                    .BYTE   004
 2009  003047  010                    .BYTE   010
 2010  003050  000                    .BYTE   000
 2011  003051  005                    .BYTE   005
 2012  003052  012                    .BYTE   012
 2013  003053  000                    .BYTE   000
```

```
2014
2015                                    ;***** DATA PATTERN Q *****
2016  003054  000                       PATQ:   .BYTE   000
2017  003055  002                               .BYTE   002
2018  003056  014                               .BYTE   014
2019  003057  060                               .BYTE   060
2020  003060  001                               .BYTE   001
2021  003061  007                               .BYTE   007
2022  003062  037                               .BYTE   037
2023  003063  177                               .BYTE   177
2024
2025
2026  003064                          ENDPAT:
2027                                  .EVEN
```

```
2028
2029
2030
2031                                    ;*** RECEIVED DATA BUFFER (64. WORDS) ***
2032  003064  000100                    RCVBUF: .BLKW   64.
2033
2034
2035
2036
```

```
                                .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  000070
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  001000                  MOV     #512.,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,GETMSR       ;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,@BSEL2      ;TELL M-LOOP TO GIVE US THE REQUESTED DATA

003466  010346                        MOV     R3,-(SP)
003470  012703  001000                MOV     #512.,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    #RDY,@BSEL2         ;DID THE M-LOOP FINISH
003506  001023                        BNE     5$                  ;YES,  GOOD.  RETURN

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

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

```
                                .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$:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;---------------------------------------------------------------------------------------
                                READI:
003566  012577  176634                MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003572  112777  000001  176622        MOVB    @REDLOC,@BSEL2  ;TELL M-LOOP TO GIVE US THE REQUESTED DATA
                                ;
003600  010346                        MOV     R3,-(SP)
003602  012703  001000                MOV     #512.,R3                ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003606  077301                1$:     SOB     R3,1$
003610  012603                        MOV     (SP)+,R3
                                ;
003612  132777  000200  176602        BITB    @RDY,@BSEL2     ;DID THE M-LOOP FINISH
003620  001023                        BNE     5$              ;YES,  GOOD.  RETURN
                                ;
003622  004737  004166                JSR     PC,GETMSR       ;GET BYTE SELECT REGISTERS
003626  012737  000001  002324        MOV     @REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
003634                                GTDF    EM4,ERR4        ;"RDY" 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
```

```
CVDMEC0 DMV11 LINE UNIT DIAG3   MACY11 30A(1052)  12-JUL-84  11:12  PAGE 55
CVDMEC.P11    12-JUL-84 10:56           ....M-LOOP -- WRITE

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

003700  012577  176522          WRITE:  MOV     (R5)+,@SEL4         ;SETUP SOURCE POINTER
003704  113577  176522                  MOVB    @(R5)+,@SEL6        ;MAKE DATA AVAILABLE TO M-LOOP
003710  000404                        BR      MLWRI               ;THE REST OF THIS ROUTINE IS THE SAME AS "WRITEI"
```

```
2236                                    .SBTTL ....M-LOOP -- WRITE IMMEDIATE
2237                                    ;**************************************************************************
2238                                    ; WRITEI - WRITE IMMEDIATE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
2239                                    ;
2240                                    ; CALLING SEQUENCE:
2241                                    ;
2242                                    ;       JSR     R5,WRITEI
2243                                    ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
2244                                    ;       .WORD   <DATA FIELD -- DATA TO BE WRITTEN IN DMV-11>
2245                                    ;       BCC     N0              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2246                                    ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
2247                                    ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
2248                                    ;
2249                                    ; N0:   <RESUMPTION OF NORMAL PROCESSING>
2250                                    ;
2251                                    ;--**************************************************************************
2252
2253  003712                            WRITEI:
2254  003712  012577  176510                   MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
2255  003716  012577  176510                   MOV     (R5)+,@SEL6     ;MAKE DATA AVAILABLE TO M-LOOP
2256  003722  112777  000002  176472  MLWRI:  MOVB    @WRILOC,@SEL2   ;TELL M-LOOP TO WRITE THE DATA
2257
2258  003730  010346                   MOV     R3,-(SP)
2259  003732  012703  001000                   MOV     #512.,R3                ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2260  003736  077301                   1$:     SOB     R3,1$
2261  003740  012603                   MOV     (SP)+,R3
2262
2263  003742  132777  000200  176452          BITB    @#RDY,@SEL2     ;DID THE M-LOOP FINISH
2264  003750  001023                   BNE     5$              ;YES,  GOOD.  RETURN
2265  003752  004737  004166                   JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
2266  003756  012737  000002  002324          MOV     @WRILOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
2267  003764                           GTDF    EM4,ERR4        ;"RDY" TIMEOUT
2268                                    ;       QUEUE "DEVICE FATAL" ERROR # 4
2269  003764  012737  000001  002172                                   MOV     #T.EDF,ERRTYP
2270  003772  012737  000004  002174                                   MOV     #4,ERRNBR
2271  004000  012737  014307  002176                                   MOV     #EM4,ERRMSG
2272  004006  012737  020120  002200                                   MOV     #ERR4,ERRBLK
2273  004014  000261                   SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
2274  004016  000401                   BR      6$              ;RETURN WITH THAT INDICATION
2275
2276  004020  000241                   5$:     CLC                     ;INDICATE "NO ERROR"
2277  004022  000205                   6$:     RTS     R5              ;RETURN
2278
2279
2280
2281
```

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

004024  117737  176366  002202  GETBSR: MOVB    @#SEL0,#SR0     ;PUT THE CURRENT CSR VALUES INTO THE PRINT-OUT
004032  117737  176362  002204          MOVB    @#SEL1,#SR1     ;TABLE
004040  117737  176356  002206          MOVB    @#SEL2,#SR2
004046  117737  176352  002210          MOVB    @#SEL3,#SR3
004054  117737  176346  002212          MOVB    @#SEL4,#SR4
004062  117737  176342  002214          MOVB    @#SEL5,#SR5
004070  117737  176336  002216          MOVB    @#SEL6,#SR6
004076  117737  176332  002220          MOVB    @#SEL7,#SR7
004104  117737  176326  002222          MOVB    @#SEL10,#SR10
004112  117737  176322  002224          MOVB    @#SEL11,#SR11
004120  117737  176316  002226          MOVB    @#SEL12,#SR12
004126  117737  176312  002230          MOVB    @#SEL13,#SR13
004134  117737  176306  002232          MOVB    @#SEL14,#SR14
004142  117737  176302  002234          MOVB    @#SEL15,#SR15
004150  117737  176276  002236          MOVB    @#SEL16,#SR16
004156  117737  176272  002240          MOVB    @#SEL17,#SR17
004164  000207                          RTS     PC              ;RETURN TO CALLER

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

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

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

004250  010037  004264  STUREG: MOV     R0,2$           ;PUT SPECIFIED REGISTER'S ADDRESS IN I/O CALLS
004254  010037  004302          MOV     R0,4$

004260  004537  003700          JSR     R5,WRITE        ;WRITE IT
004264  000000          2$:     .WORD   0               ;*** MODIFIED FROM ABOVE ***
004266  002322                  .WORD   TDATA           ;
004270  103431                  BCS     10$             ;ON ERROR, EXIT

004272  005037  002326          CLR     BDATA           ;CLEAR BOTH BYTES -- JUST IN CASE....
004276  004537  003454          JSR     R5,READ         ;READ IT BACK AGAIN
004302  000000          4$:     .WORD   0               ;*** MODIFIED FROM ABOVE ***
004304  002326                  .WORD   BDATA
004306  103422                  BCS     10$             ;ON ERROR, EXIT

004310  123737  002324  002326  CMPB    GDATA,BDATA     ;DID WE READ WHAT WE WROTE?
004316  000241          CLC                             ; (THIS ISN'T NEEDED FOR THE ERROR TEST BUT
                                                        ; MUST BE CLEARED ON EXIT IF NO ERROR OCCURED)
004320  001415          BEQ     10$                     ;YES, EXIT FROM SUBTEST
004322                  GTDF    EM25,ERR7A              ;REPORT READ/WRITE ERROR
                                                        ;       QUEUE "DEVICE FATAL" ERROR # 5
004322  012737  000001  002172                                  MOV     #T.EDF,ERRTYP
004330  012737  000005  002174                                  MOV     #5,ERRNBR
004336  012737  014360  002176                                  MOV     #EM25,ERRMSG
004344  012737  020244  002200                                  MOV     #ERR7A,ERRBLK
004352  000261          SEC                             ;INDICATE THAT AN ERROR WAS DETECTED
004354  000207          10$:    RTS     PC
```

```
                                .SBTTL ....STALL -- DELAY FOR 10.5 MICRO-SEC'S (ON LSI-11)
                                ;++*************************************************************************
                                ; STALL -- THIS SUBROUTINE STALLS FOR ABOUT 10.5 MICRO-SECONDS
                                ;--*************************************************************************

004356  000207                STALL:  RTS     PC
```

```
                                        .SBTTL

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

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

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

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

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

004456  000207                          RTS     PC              ;RETURN


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

```
2436                                    .SBTTL ....INITT1 -- INITIALIZE TIMER #1
2437                                    ;**************************************************************************
2438                                    ;* INITT1 - INITIALIZE TIMER # 1
2439                                    ;*
2440                                    ;*        CALLING SEQUENCE:
2441                                    ;*
2442                                    ;*              JSR     R5,INITT1
2443                                    ;*              .WORD   <VALUE LOADED INTO THE T1 LATCH @ VIAT1C & VIAT1D>
2444                                    ;*              .WORD   <VALUE LOADED INTO "T1L-L" & "T1C-H">
2445                                    ;*              .BYTE   <BITS 6 & 7 WILL BE LOADED INTO "ACR", BIT 5 WILL BE
2446                                    ;*                       USED TO SET OR CLEAR BIT 6 ("T1") OF THE INTERRUPT
2447                                    ;*                       ENABLE REGISTER ("IER")>
2448                                    ;*              .BYTE   <UNUSED>
2449                                    ;*
2450                                    ;*
2451                                    ;* NOTE:
2452                                    ;*
2453                                    ;* BEFORE LOADING AND STARTING THE COUNTER, THE LATCH REGISTER (ACCESSED THRU
2454                                    ;* "VIAT1C") IS LOADED.  THEN, T1L-L IS LOADED AND NEXT, T1C-H.  THIS LAST
2455                                    ;* LOAD WILL RESET THE TIMEOUT BIT AND COUNTER LOGIC.  IT IS EXPECTED AT THIS
2456                                    ;* TIME (5/25/79) THAT THE INTERRUPT FACILITY OF THE VIA CHIP WILL NOT BE USED
2457                                    ;* -- HOWEVER, ACCESS TO THE INTERRUPT ENABLE BIT IS GIVEN THROUGH THE THIRD
2458                                    ;* PARAMETER IN THE CALLING SEQUENCE (BIT 5 = 0 WILL CAUSE THIS ROUTINE TO
2459                                    ;* CLEAR THE ENABLE BIT ("T1") IN "IER".)
2460                                    ;*
2461                                    ;**************************************************************************
2462
2463  004534  010146                    INITT1: MOV     R1,-(SP)        ;SAVE THE REGISTER WE WILL BE USING
2464  004536  012537  004660                    MOV     (R5)+,7$        ;SETUP VALUE TO BE WRITTEN IN LATCH
2465  004542  012537  004706                    MOV     (R5)+,10$       ;SETUP VALUE TO BE WRITTEN IN COUNTER
2466  004546  111501                            MOVB    (R5),R1         ;GET & PROCESS BITS FOR ACR 6 & 7
2467  004550  143701  000077                    BICB    077,R1
2468  004554  010137  004650                    MOV     R1,4$           ;SETUP CALL SET ACR'S BITS 6 & 7
2469  004560  112501                            MOVB    (R5)+,R1        ;NOW, GET THE BIT TO BE USED IN SETTING OR
2470                                                                    ;CLEARING BIT 6 OF "IER"
2471  004562  106301                            ASLB    R1              ;THE PASSED BIT IS IN THE WRONG POSITION
2472  004564  106301                            ASLB    R1              ;BUT, THE PASSED BIT SHOULD CONTROL THE OPERATION.
2473                                                                    ;WE KNOW WE ARE SETTING OR CLEARING BIT 6 --
2474                                                                    ;THUS, THE PASSED BIT WILL BECOME THE CONTROLLING
2475                                                                    ;BIT 7 AND WE WILL "OR" IN THE BIT WE WISH TO
2476                                                                    ;BE CONTROLLED (BIT 6).
2477  004566  143701  000177                    BICB    177,R1          ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
2478  004572  153701  000100                    BISB    100,R1          ;THEN SET BIT 6
2479  004576  010137  004610                    MOV     R1,2$           ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE
2480
2481  004602  004537  003712                    JSR     R5,WRITEI       ;WRITE TO
2482  004606  120016                            VIAIER                  ;THE VIA'S IER
2483  004610  000000                    2$:     .WORD   0               ;INTERRUPT ENABLE/DISABLE INFORMATION
2484
2485  004612  004537  003566                    JSR     R5,READI        ;READ THE CURRENT SETTING OF
2486  004616  120013                            VIAACR                  ;THE VIA'S ACR
2487  004620  000000                    3$:     .WORD   0               ;INTO "3$"
2488
2489  004622  013701  004620                    MOV     3$,R1           ;GET THAT VALUE
2490  004626  143701  000300                    BICB    300,R1          ;CLEAR THE CURRENT SETTING OF BITS 6 & 7
2491  004632  053701  004650                    BIS     4$,R1           ;SET THEM ACCORDING TO THE PASSED VALUES
```

```
2492  004636  010137  004650                  MOV     R1,4$           ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
2493
2494  004642  004537  003712                  JSR     R5,WRITEI       ;WRITE TO
2495  004646  120013                          VIAACR                  ;THE VIA'S ACR
2496  004650  000000              4$:         .WORD   0               ;THE NEW REGISTER SETTING
2497
2498  004652  004537  003712                  JSR     R5,WRITEI       ;WRITE TO
2499  004656  120006                          VIAT1C                  ;LOW ORDER LATCH REGISTER (T1L-L)
2500  004660  000000              7$:         .WORD   0               ;THE VALUE PASSED
2501
2502  004662  113737  004661  004676          MOVB    7$+1,8$         ;SETUP FOR AND
2503  004670  004537  003712                  JSR     R5,WRITEI       ;WRITE TO
2504  004674  120007                          VIAT1D                  ;HIGH ORDER LATCH REGESTER (T1L-H)
2505  004676  000000              8$:         .WORD   0               ;THE VALUE PASSED
2506
2507  004700  004537  003712                  JSR     R5,WRITEI       ;WRITE TO
2508  004704  120004                          VIAT1A                  ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2509  004706  000000              10$:        .WORD   0               ;THE VALUE PASSED
2510
2511  004710  113737  004707  004724          MOVB    10$+1,11$       ;SETUP FOR AND
2512  004716  004537  003712                  JSR     R5,WRITEI       ;WRITE TO
2513  004722  120005                          VIAT1B                  ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2514  004724  000000              11$:        .WORD   0               ;THE VALUE PASSED
2515
2516                                  ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2517
2518  004726  012601                          MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2519  004730  005205                          INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2520                                                                  ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2521
2522  004732  000205                          RTS     R5              ;NOW, RETURN
2523
2524
```

```
                            .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
```

```
                        ....INITT2 -- INITIALIZE TIMER #2

2581  005040  004537  003712            JSR     R5,WRITEI       ;WRITE TO
2582  005044  120013                    VIAACR                  ;THE VIA'S ACR
2583  005046  000000            4$:     .WORD   0               ;THE NEW REGISTER SETTING
2584
2585  005050  004537  003712            JSR     R5,WRITEI       ;WRITE TO
2586  005054  120010                    VIAT2A                  ;LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
2587  005056  000000            10$:    .WORD   0               ;THE VALUE PASSED
2588
2589  005060  113737  005057  005074    MOVB    10$+1,11$       ;SETUP FOR AND
2590  005066  004537  003712            JSR     R5,WRITEI       ;WRITE TO
2591  005072  120011                    VIAT2B                  ;HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
2592  005074  000000            11$:    .WORD   0               ;THE VALUE PASSED
2593
2594                            ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2595
2596  005076  012601                    MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2597  005100  005205                    INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2598                                                            ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2599
2600  005102  000205                    RTS     R5              ;THEN RETURN
2601
```

```
                        .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$,BDATA        ;GET ACTUAL DATA
                        ;STACK "USYRT NOT CLEANED BY PROGRAM RESET" MSG
005212                          GTDF    EM2,ERR10
                                                        ;       QUEUE "DEVICE FATAL" ERROR @ 6
005212  012737  000001  002172                                                    MOV     @T.EDF,ERRTYP
005220  012737  000006  002174                                                    MOV     @6,E[illegible]R
005226  012737  014214  002176                                                    MOV     @EM2,E[illegible]MSG
005234  012737  020364  002200                                                    MOV     @ERR10,ERRBLK
005242  000261                   SEC                     ;SET C BIT TO FLAG ERROR
005244  000406                   BR      10$             ;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
```

```
                        ....RSTCHK -- RESET USYRT/VERIFY ALL USYRT REGS & RESET STATE

                        ;*************************************************************************
                        ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
                        ;*************************************************************************
005270  010146          WAIT50: MOV     R1,-(SP)        ;SAVE R1
005272  012701  000764          MOV     #500.,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
```

```
2707                                    .SBTTL ....INIDMV -- INIT DMV (MCLR, VIA SETUP)
2708                                    ;**************************************************************************
2709                                    ;* INIDMV - THIS SUBROUTINE INITIALIZES THE DMV-11, BY DOING A MASTER CLEAR,
2710                                    ;*       ENTERING THE M-LOOP, AND PROGRAMMING THE VIA REGS FOR DEFAULT
2711                                    ;*       OPERATION.
2712                                    ;*
2713                                    ;*       CALLING SEQUENCE :
2714                                    ;*          JSR  PC,INIDMV
2715                                    ;**************************************************************************
2716  005376  004737  003352           INIDMV: JSR     PC,MSTCLR       ;MASTER CLR, M-LOOP
2717  005402  004737  005304                   JSR     PC,SETVIA       ;PROGRAM VIA
2718  005406  000207                           RTS     PC              ;RETURN
2719
2720
2721
2722
2723                                    .SBTTL ....CKUSTS -- CHECK USYRT STATUS REGISTERS
2724                                    ;**************************************************************************
2725                                    ;* CKUSTS - THIS SUBROUTINE CHECKS THE USYRT STATUS BY READING THE USYRT
2726                                    ;*       STATUS REGISTER AND COMPARING IT TO THE LOW BYTE OF THE WORD FOLLOWING
2727                                    ;*       THE CALL. IF THERE IS A MISMATCH, THE SUBROUTINE STACKS THE ERROR
2728                                    ;*       INFORMATION, AND SETS THE "C" BIT AND RETURNS.
2729                                    ;**************************************************************************
2730  005410                            CKUSTS:
2731  005410  004537  003566                   JSR     R5,READI        ;READ USYRT STATUS REGISTER
2732  005414  122000                           USTATR
2733  005416  000000                   1$:     .WORD   0
2734  005420  122537  005416                   CMPB    (R5)+,1$        ;SEE IF STATUS MATCHES EXPECTED
2735  005424  000241                           CLC                     ;CLEAR C BIT
2736  005426  001430                           BEQ     2$              ;BR IF STATUS OK
2737  005430  012737  000007  002336           MOV     #7,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
2738  005436  016537  177777  002324           MOV     -1(R5),GDATA    ;GET EXPECTED DATA
2739  005444  005037  002326                   CLR     BDATA           ;GET ACTUAL DATA
2740  005450  113737  005416  002326           MOVB    1$,BDATA
2741                                    ;STACK "USYRT STATUS INCORRECT" ERROR
2742  005456                                    GTDF    EM68,ERR10
2743                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 7
2744  005456  012737  000001  002172                                                            MOV     #T.EDF,ERRTYP
2745  005464  012737  000007  002174                                                            MOV     #7,ERRNBR
2746  005472  012737  014650  002176                                                            MOV     #EM68,ERRMSG
2747  005500  012737  020364  002200                                                            MOV     #ERR10,ERRBLK
2748  005506  000261                           SEC                     ;SET C BIT FOR ERROR
2749  005510  005205                   2$:     INC     R5              ;INCREMENT R5 PAST ARGUMENT
2750  005512  000205                           RTS     R5              ;RETURN
2751
2752
2753
2754
```

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

```
2900                                    .SBTTL ....CKRACT -- CHECK RECEIVER ACTIVE (RXACT)
2901                                    ;**************************************************************************
2902                                    ;* CKRACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RXACT IN THE USYRT
2903                                    ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
2904                                    ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
2905                                    ;*
2906                                    ;*      CALLING SEQUENCE :
2907                                    ;*      JSR     R5,CKRACT
2908                                    ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF RXACT>
2909                                    ;**************************************************************************
2910  005654                            CKRACT:
2911  005654  012737  000007  002336            MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
2912  005662  004537  003566                    JSR     R5,READI        ;READ USYRT STATUS
2913  005666  122000                            USTATR
2914  005670  000000                    1$:     .WORD   0
2915  005672  032725  000001                    BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RXACT
2916  005676  001422                            BEQ     2$              ;BR IF EXPECTED RXACT = 0
2917  005700  132737  000040  005670            BITB    #RXACT,1$       ;SEE IF RXACT = 1
2918  005706  001040                            BNE     3$              ;BR IF RXACT = 1
2919                                    ;STACK "RXACT NOT SET" MSG
2920  005710                                    GTDF    EM71,ERR12
2921                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 10
2922  005710  012737  000001  002172                                                                  MOV     #T.EDF,ERRTYP
2923  005716  012737  000012  002174                                                                  MOV     #10,ERRNBR
2924  005724  012737  014737  002176                                                                  MOV     #EM71,ERRMSG
2925  005732  012737  020714  002200                                                                  MOV     #ERR12,ERRBLK
2926  005740  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2927  005742  000423                            BR      4$              ;TAKE ERROR EXIT
2928  005744  132737  000040  005670  2$:     BITB    #RXACT,1$       ;SEE IF RXACT = 0
2929  005752  001416                            BEQ     3$              ;BR IF RXACT = 0
2930                                    ;STACK "RXACT NOT CLEARED" MSG
2931  005754                                    GTDF    EM72,ERR12
2932                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 11
2933  005754  012737  000001  002172                                                                  MOV     #T.EDF,ERRTYP
2934  005762  012737  000013  002174                                                                  MOV     #11,ERRNBR
2935  005770  012737  014755  002176                                                                  MOV     #EM72,ERRMSG
2936  005776  012737  020714  002200                                                                  MOV     #ERR12,ERRBLK
2937  006004  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2938  006006  000401                            BR      4$              ;TAKE ERROR EXIT
2939  006010  000241                    3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
2940  006012  000205                    4$:     RTS     R5              ;RETURN
2941
2942
2943
2944
```

```
                        .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  000014  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
```

```
3078  007032  001416                        BEQ     5$              ;BR IF REOM = 0
3079                                  ;STACK "REOM NOT CLEARED" MSG
3080  007034                                GTDF    EM30,ERR12
3081                                                                  ;       QUEUE "DEVICE FATAL" ERROR # 23
3082  007034  012737  000001  002172                                                        MOV     #T.EDF,ERRTYP
3083  007042  012737  000027  002174                                                        MOV     #23,ERRNBR
3084  007050  012737  014444  002176                                                        MOV     #EM30,ERRMSG
3085  007056  012737  020714  002200                                                        MOV     #ERR12,ERRBLK
3086  007064  000261                        SEC                     ;SET C BIT TO FLAG ERROR
3087  007066  000401                        BR      6$              ;TAKE ERROR EXIT
3088  007070  000241                5$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
3089  007072  000205                6$:     RTS     R5              ;RETURN
3090
3091
```

```
                                .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
```

```
 3191  007412  120000                        VIAORB
 3192  007414  000142              13$:      .AEN!RXEN!TTLOOP        ;* THIS VALUE MIGHT BE MODIFIED ABOVE
 3193
 3194  007416  004537  003712                JSR     R5,WRITEI       ;SET TSOM IN USYRT
 3195  007422  120403                        TDSRH
 3196  007424  000001                        TSOM
 3197  007426  004537  003712                JSR     R5,WRITEI       ;LOAD SYNCH CHAR INTO TX BUF
 3198  007432  120402                        TDSRL
 3199  007434  000226                        SYNCH
 3200  007436  004537  006014                JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
 3201  007442  000000                        0
 3202  007444  103427                        BCS     7$              ;IF ERROR, EXIT SUBROUTINE
 3203  007446  005001                        CLR     R1              ;INIT CYCLE COUNTER
 3204  007450  004537  012072          4$:   JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
 3205  007454  000001                        1
 3206  007456  004537  003566                JSR     R5,READI        ;READ USYRT STATUS REG
 3207  007462  122000                        USTATR
 3208  007464  000000                  5$:   .WORD   0
 3209  007466  132737  000100  007464        BITB    #TBMT,5$        ;SEE IF TBMT IS SET YET
 3210  007474  001010                        BNE     6$              ;BR IF YES
 3211  007476  005201                        INC     R1              ;INCR CYCLE COUNTER
 3212  007500  020127  000003                CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
 3213  007504  002761                        BLT     4$              ;BR IF LESS THAN 3 CYCLES
 3214  007506  004537  006014                JSR     R5,CKTBMT       ;GO STACK "TBMT NOT SET" MSG
 3215  007512  000001                        1
 3216  007514  103403                        BCS     7$              ;IF ERROR, EXIT SUBROUTINE
 3217  007516  004537  005514          6$:   JSR     R5,CKTACT       ;CHK FOR TXACT = 1
 3218  007522  000001                        1
 3219  007524  012601                  7$:   MOV     (SP)+,R1        ;RESTORE R1
 3220  007526  000205                        RTS     R5              ;RETURN (IF C = 1, WE HAD AN ERROR)
 3221
 3222  007530  000142                  20$:  TXEN!RXEN!TTLOOP        ;DEFAULT VALUE FOR VIAORB: ENABLE
 3223                                                                ;TX AND RX ON USYRT, ASSERT RTS, DTR
 3224
```

```
                                .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 PROV'IDED 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 M3254 AND M3255 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 M3254.5 CHECK IS DESIRED
007536  001407                          BEQ     2$              ;BR IF NOT
007540  005737  002472                  TST     TSTCON          ;SEE IF M3254.5 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$:    TST     TSTCON          ;SEE IF M3254, M3255 USED
007634  001406                          BEQ     12$             ;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
```

```
3281  007670  005737  002472          18$:    TST     TSTCON          ;SEE IF M3254, M3255 USED
3282  007674  001002                          BNE     23$             ;BR IF NOT
3283  007676  000137  010072          20$:    JMP     46$             ;GO RUN THE TEST
3284  007702  023727  002472  000003  23$:    CMP     TSTCON,#3       ;SEE IF OPERATOR SPEC'D V.35
3285  007710  001006                          BNE     28$             ;BR IF NO
3286  007712  023727  002470  000001          CMP     BRDTYP,#1       ;TSTCOM MATCH BRDTYP?
3287  007720  001766                          BEQ     20$             ;YES: RUN TEST
3288  007722  000137  010006                  JMP     40$             ;WRONG INTERFACE, GO SKIP TEST
3289
3290  007726  023727  002472  000002  28$:    CMP     TSTCON,#2       ;SEE IF OPERATOR SPEC'D EIA
3291  007734  001760                          BEQ     20$             ;BR IF YES, TO RUN EIA
3292  007736  000137  010006                  JMP     40$             ;WRONG INTERFACE, GO SKIP TEST
3293
3294                                  ;*** NO INTERFACE REQUESTED - FIND ONE TO TEST ***
3295
3296  007742  005737  002470          32$:    TST     BRDTYP          ;SEE IF INTEGRAL MODEM SELECTED
3297  007746  001343                          BNE     16$             ;BR IF NOT (TEST FOR V35/EIA)
3298  007750  000137  007652                  JMP     12$             ;SEE IF INTEGRAL MODEM CAN BE RUN
3299
3300                                  ;PRINT "FOR BAUD RATE SPECIFIED."
3301  007754                          38$:
3302  007754  023727  002412  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3303  007762  001063                          BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3304  007764                          PRINTF  #FMT30
3305  007764  012746  013762                                                  MOV     #FMT30,-(SP)
3306  007770  012746  000001                                                  MOV     #1,-(SP)
3307  007774  010600                                                          MOV     SP,R0
3308  007776  104417                                                          TRAP    C$PNTF
3309  010000  062706  000004                                                  ADD     #4 SP
3310  010004  000434                          BR      48$             ;GO TO PRINT "TEST NOT RUN"
3311                                  ;PRINT "IMPROPER CONNECTOR TYPE SPECIFIED"
3312  010006                          40$:
3313  010006  023727  002412  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3314  010014  001046                          BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3315  010016                          PRINTF  #FMT31          ; *********
3316  010016  012746  014017                                                  MOV     #FMT31,-(SP)
3317  010022  012746  000001                                                  MOV     #1,-(SP)
3318  010026  010600                                                          MOV     SP,R0
3319  010030  104417                                                          TRAP    C$PNTF
3320  010032  062706  000004                                                  ADD     #4,SP
3321  010036  000417                          BR      48$             ;GO TO PRINT "TEST 'OT RUN"
3322
3323                                  ;PRINT "FOR OPTION SPECIFIED."
3324  010040  023727  002412  000001  42$:    CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3325  010046  001031                          BNE     50$             ;BR IF NOT TO SKIP PRINTING
3326  010050                          PRINTF  #FMT32
3327  010050  012746  014065                                                  MOV     #FMT32,-(SP)
3328  010054  012746  000001                                                  MOV     #1,-(SP)
3329  010060  010600                                                          MOV     SP,R0
3330  010062  104417                                                          TRAP    C$PNTF
3331  010064  062706  000004                                                  ADD     #4,SP
3332  010070  000402                          BR      48$             ;GO TO PRINT "TEST NOT RUN"
3333
3334                                  ;*** BRANCH HERE TO RUN TEST ***
3335  010072  000241          46$:    CLC                             ;CLEAR C BIT TO RUN TEST
3336  010074  000417                  BR      52$                     ;EXIT
```

```
3337
3338                                    ;*** BRANCH HERE TO SKIP TEST ***
3339                                    ;PRINT "TEST XX NOT RUN"
3340   010076                           48$:
3341   010076  023727  002412  000001          CMP     STARES,#1       ;SEE IF THIS IS FIRST PASS SINCE STA OR RES
3342   010104  001012                          BNE     50$             ;BR IF NOT, TO SKIP PRINTING
3343   010106                                  PRINTF  #FMT19,TSTNUM
3344   010106  013746  002414                                                          MOV     TSTNUM,-(SP)
3345   010112  012746  013452                                                          MOV     #FMT19,-(SP)
3346   010116  012746  000002                                                          MOV     #2,-(SP)
3347   010122  010600                                                                  MOV     SP,R0
3348   010124  104417                                                                  TRAP    C$PNTF
3349   010126  062706  000006                                                          ADD     #6,SP
3350   010132  000261                  50$:    SEC                     ;SET C BIT TO SKIP TEST
3351   010134  000205                  52$:    RTS     R5              ;RETURN
3352
3353
3354
```

```
                                .SBTTL ....TXCHAR -- TRANSMIT A CHARACTER
                                ;*****************************************************************************
                                ;* TXCHAR - THIS SUBROUTINE INITIATES TRANSMISSION OF A CHAR BY LOADING
                                ;*      THE USYRT TDSRL WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
                                ;*      FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
                                ;*      PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
                                ;*      MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
                                ;*      IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                ;*      IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                ;*      DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,TXCHAR
                                ;*      .WORD   <DATA FOR TDSRL IN LO BYTE>
                                ;*      .WORD   <NUMBER OF CYCLES TO CLOCK (IN LO BYTE)>
                                ;*              <SWITCH TO DISABLE INITIAL TBMT=0 CHECK (MSB IN HI BYTE)>
                                ;*****************************************************************************
010136                          TXCHAR:
010136  010146                          MOV     R1,-(SP)        ;SAVE R1
010140  010246                          MOV     R2,-(SP)        ;SAVE R2
010142  012537  010154                  MOV     (R5)+,1$        ;GET DATA FOR TDSRL
010146  004537  003712                  JSR     R5,WRITEI       ;LOAD DATA INTO TDSRL
010152  120402                          TDSRL
010154  000000                  1$:     .WORD   0
010156  005001                          CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
010160  005002                          CLR     R2              ;CLEAR REQ'D CYCLE COUNT
010162  112502                          MOVB    (R5)+,R2        ;GET DESIRED NO. OF CYCLES
010164  001425                          BEQ     6$              ;BR IF NO CLOCKING DONE
010166  004537  005514          3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
010172  000001                          1
010174  103421                          BCS     6$              ;BR TO EXIT IF ERROR
010176  020102                          CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
010200  001414                          BEQ     5$              ;BR IF YES
                                
010202  131527  000200                  BITB    (R5),@NCTBMT    ;* CHECK FOR "TBMT=0 CHECK" DISABLE
010206  001004                          BNE     7$              ;* BR IF MSB IS NOT SET
                                
010210  004537  006014                  JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
010214  000000                          0
010216  103410                          BCS     6$              ;BR TO EXIT IF ERROR
010220  004537  012072          7$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
010224  000001                          1
010226  005201                          INC     R1              ;INCR CYCLE COUNT
010230  000756                          BR      3$              ;KEEP CLOCKING
010232  004537  006014          5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
010236  000001                          1
010240  012602                  6$:     MOV     (SP)+,R2        ;RESTORE R2
010242  012601                          MOV     (SP)+,R1        ;RESTORE R1
010244  005205                          INC     R5              ;ADJUST R5 FOR SANE RETURN
010246  000205                          RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

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

```
                                .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
```

```
3510  010542                                        GTDF    EM34,ERR10
3511                                                ;       QUEUE "DEVICE FATAL" ERROR # 27
3512  010542  012737  000001  002172                                    MOV     #T.EDF,ERRTYP
3513  010550  012737  000033  002174                                    MOV     #27,ERRNBR
3514  010556  012737  014502  002176                                    MOV     #EM34,ERRMSG
3515  010564  012737  020364  002200                                    MOV     #ERR10,ERRBLK
3516  010572  000137  011610                        JMP     20$             ;TAKE ERROR EXIT
3517  010576  116501  000001              6$:       MOVB    1(R5),R1        ;GET RDSRH
3518  010602  042701  177400                        BIC     #177400,R1      ;MASK OFF UNUSED BITS
3519  010606  123701  010362                        CMPB    2$,R1           ;COMPARE RCV'D STATUS TO EXPECTED
3520  010612  001016                                BNE     7$              ;BR IF MISMATCH
3521  010614  000137  011474                        JMP     17$             ;CONTINUE
3522  010620  012737  000001  002336                MOV     #1,REGNUM       ;SET USYRT REG NO. FOR RDSRH
3523  010626  005037  002324                        CLR     GDATA           ;SET EXPECTED DATA
3524  010632  110137  002324                        MOVB    R1,GDATA
3525  010636  005037  002326                        CLR     BDATA           ;SET ACTUAL DATA
3526  010642  113737  010362  002326                MOVB    2$,BDATA
3527  010650  012737  000001  002336  7$:       MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
3528  010656  032765  000001  000002                BIT     #RERCHK,2(R5)   ;SEE IF RCV ERROR BIT SHOULD BE IGNORED
3529  010664  001447                                BEQ     9$              ;BR IF YES
3530                                        ;CHECK RERR BIT
3531  010666  132701  000200                        BITB    #RERR,R1        ;SEE IF EXPECTED BIT = 1
3532  010672  001022                                BNE     8$              ;BR IF YES
3533  010674  132737  000200  010362                BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 0
3534  010702  001440                                BEQ     9$              ;BR IF YES
3535                                        ;STACK "RERR NOT CLEARED" MSG
3536  010704                                        GTDF    EM35,ERR12
3537                                                ;       QUEUE "DEVICE FATAL" ERROR # 28
3538  010704  012737  000001  002172                                    MOV     #T.EDF,ERRTYP
3539  010712  012737  000034  002174                                    MOV     #28,ERRNBR
3540  010720  012737  014530  002176                                    MOV     #EM35,ERRMSG
3541  010726  012737  020714  002200                                    MOV     #ERR12,ERRBLK
3542  010734  000137  011610                        JMP     20$             ;TAKE ERROR EXIT
3543  010740  132737  000200  010362  8$:       BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 1
3544  010746  001016                                BNE     9$              ;BR IF YES
3545                                        ;STACK "RERR NOT SET" MSG
3546  010750                                        GTDF    EM36,ERR12
3547                                                ;       QUEUE "DEVICE FATAL" ERROR # 29
3548  010750  012737  000001  002172                                    MOV     #T.EDF,ERRTYP
3549  010756  012737  000035  002174                                    MOV     #29,ERRNBR
3550  010764  012737  014551  002176                                    MOV     #EM36,ERRMSG
3551  010772  012737  020714  002200                                    MOV     #ERR12,ERRBLK
3552  011000  000137  011610                        JMP     20$             ;TAKE ERROR EXIT
3553                                        ;CHECK ROR BIT
3554  011004  132701  000010              9$:       BITB    #ROR,R1         ;SEE IF EXPECTED BIT = 1
3555  011010  001022                                BNE     10$             ;BR IF YES
3556  011012  132737  000010  010362                BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 0
3557  011020  001440                                BEQ     11$             ;BR IF YES
3558                                        ;STACK "ROR NOT CLEARED" MSG
3559  011022                                        GTDF    EM16,ERR12
3560                                                ;       QUEUE "DEVICE FATAL" ERROR # 30
3561  011022  012737  000001  002172                                    MOV     #T.EDF,ERRTYP
3562  011030  012737  000036  002174                                    MOV     #30,ERRNBR
3563  011036  012737  014340  002176                                    MOV     #EM16,ERRMSG
3564  011044  012737  020714  002200                                    MOV     #ERR12,ERRBLK
3565  011052  000137  011610                        JMP     20$             ;TAKE ERROR EXIT
```

```
 3566  011056  132737  000010  010362  10$:    BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 1
 3567  011064  001016                          BNE     11$             ;BR IF YES
 3568                                  ;STACK "ROR NOT SET" MSG
 3569  011066                                  GTDF    EM14,ERR12
 3570                                                  ;       QUEUE "DEVICE FATAL" ERROR # 31
 3571  011066  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
 3572  011074  012737  000037  002174                                          MOV     #31,ERRNBR
 3573  011102  012737  014324  002176                                          MOV     #EM14,ERRMSG
 3574  011110  012737  020714  002200                                          MOV     #ERR12,ERRBLK
 3575  011116  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
 3576                                  ;CHECK RABGA BIT
 3577  011122  132701  000004          11$:    BITB    #RABGA,R1       ;SEE IF EXPECTED BIT = 1
 3578  011126  001022                          BNE     12$             ;BR IF YES
 3579  011130  132737  000004  010362          BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 0
 3580  011136  001440                          BEQ     13$             ;BR IF YES
 3581                                  ;STACK "RABGA NOT CLEARED" MSG
 3582  011140                                  GTDF    EM39,ERR12
 3583                                                  ;       QUEUE "DEVICE FATAL" ERROR # 32
 3584  011140  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
 3585  011146  012737  000040  002174                                          MOV     #32,ERRNBR
 3586  011154  012737  014566  002176                                          MOV     #EM39,ERRMSG
 3587  011162  012737  020714  002200                                          MOV     #ERR12,ERRBLK
 3588  011170  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
 3589  011174  132737  000004  010362  12$:    BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 1
 3590  011202  001016                          BNE     13$             ;BR IF YES
 3591                                  ;STACK "RABGA NOT SET" MSG
 3592  011204                                  GTDF    EM40,ERR12
 3593                                                  ;       QUEUE "DEVICE FATAL" ERROR # 33
 3594  011204  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
 3595  011212  012737  000041  002174                                          MOV     #33,ERRNBR
 3596  011220  012737  014610  002176                                          MOV     #EM40,ERRMSG
 3597  011226  012737  020714  002200                                          MOV     #ERR12,ERRBLK
 3598  011234  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
 3599                                  ;CHECK REOM BIT
 3600  011240  132701  000002          13$:    BITB    #REOM,R1        ;SEE IF EXPECTED BIT = 1
 3601  011244  001022                          BNE     14$             ;BR IF YES
 3602  011246  132737  000002  010362          BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 0
 3603  011254  001440                          BEQ     15$             ;BR IF YES
 3604                                  ;STACK "REOM NOT CLEARED" MSG
 3605  011256                                  GTDF    EM30,ERR12
 3606                                                  ;       QUEUE "DEVICE FATAL" ERROR # 34
 3607  011256  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
 3608  011264  012737  000042  002174                                          MOV     #34,ERRNBR
 3609  011272  012737  014444  002176                                          MOV     #EM30,ERRMSG
 3610  011300  012737  020714  002200                                          MOV     #ERR12,ERRBLK
 3611  011306  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
 3612  011312  132737  000002  010362  14$:    BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 1
 3613  011320  001016                          BNE     15$             ;BR IF YES
 3614                                  ;STACK "REOM NOT SET" MSG
 3615  011322                                  GTDF    EM31,ERR12
 3616                                                  ;       QUEUE "DEVICE FATAL" ERROR # 35
 3617  011322  012737  000001  002172                                          MOV     #T.EDF,ERRTYP
 3618  011330  012737  000043  002174                                          MOV     #35,ERRNBR
 3619  011336  012737  014465  002176                                          MOV     #EM31,ERRMSG
 3620  011344  012737  020714  002200                                          MOV     #ERR12,ERRBLK
 3621  011352  000137  011610                  JMP     20$             ;TAKE ERROR EXIT
```

```
3622                                    ;CHECK RSOM BIT
3623  011356  132701  000001            15$:    BITB    @RSOM,R1        ;SEE IF EXPECTED BIT = 1
3624  011362  001022                            BNE     16$             ;BR IF YES
3625  011364  132737  000001  010362            BITB    @RSOM,2$        ;SEE IF ACTUAL BIT = 0
3626  011372  001440                            BEQ     17$             ;BR IF YES
3627                                    ;STACK "RSOM NOT CLEARED" MSG
3628  011374                                    GTDF    EM28,ERR12
3629                                                    ;       QUEUE "DEVICE FATAL" ERROR # 36
3630  011374  012737  000001  002172                                                    MOV     #T.EDF,ERRTYP
3631  011402  012737  000044  002174                                                    MOV     #36,ERRNUM
3632  011410  012737  014406  002176                                                    MOV     #EM28,ERRMSG
3633  011416  012737  020714  002200                                                    MOV     #ERR12,ERRBLK
3634  011424  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3635  011430  132737  000001  010362  16$:     BITB    @RSOM,2$        ;SEE IF ACTUAL BIT = 1
3636  011436  001016                            BNE     17$             ;BR IF YES
3637                                    ;STACK "RSOM NOT SET" MSG
3638  011440                                    GTDF    EM29,ERR12
3639                                                    ;       QUEUE "DEVICE FATAL" ERROR # 37
3640  011440  012737  000001  002172                                                    MOV     #T.EDF,ERRTYP
3641  011446  012737  000045  002174                                                    MOV     #37,ERRNBR
3642  011454  012737  014427  002176                                                    MOV     #EM29,ERRMSG
3643  011462  012737  020714  002200                                                    MOV     #ERR12,ERRBLK
3644  011470  000137  011610                    JMP     20$             ;TAKE ERROR EXIT
3645
3646  011474  116502  000004            17$:    MOVB    4(R5),R2        ;GET DESIRED NO. OF CYCLES
3647  011500  005001                            CLR     R1              ;INIT CYCLE COUNT
3648
3649  011502  136527  000005  000040  18$:     BITB    5(R5),@BIT5     ;* IS RXACT CHECK TO BE DISABLED ?
3650  011510  001004                            BNE     31$             ;* BR IF YES
3651  011512  004537  005654                    JSR     R5,CKRACT       ;CHK FOR RACT = 1
3652  011516  000001                            1
3653  011520  103433                            BCS     20$             ;BR TO EXIT IF ERROR
3654
3655  011522  020102                    31$:    CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3656  011524  001415                            BEQ     19$             ;BR IF YES
3657
3658  011526  136527  000005  000200            BITB    5(R5),@BIT7     ;* SEE IF INITIAL RDA CHECK DESIRED
3659  011534  001004                            BNE     22$             ;* BR IF NO
3660  011536  004537  006154                    JSR     R5,CKRDA        ;CHK FOR RDA = 0
3661  011542  000000                            0
3662  011544  103421                            BCS     20$             ;BR TO EXIT IF ERROR
3663
3664  011546  004537  012072            22$:    JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3665  011552  000001                            1
3666  011554  005201                            INC     R1              ;INCR CYCLE COUNT
3667  011556  000751                            BR      18$             ;CONTINUE CLOCKING
3668
3669  011560  136527  000005  000100  19$:     BITB    5(R5),@BIT6     ;* IS FINAL RDA CHECK TO BE SKIPPED ?
3670  011566  001004                            BNE     30$             ;* BR IF YES
3671  011570  004537  006154                    JSR     R5,CKRDA        ;CHK RDA = 1
3672  011574  000001                            1
3673  011576  103404                            BCS     20$             ;BR IF ERROR
3674
3675  011600  062705  000006            30$:    ADD     #6,R5           ;FIX UP RETURN ADRS
3676  011604  000241                            CLC                     ;SET C = 0 FOR NO ERROR
3677  011606  000403                            BR      21$             ;TAKE ERROR-FREE EXIT
```

```
 3678  011610  062705  000006          20$:    ADD     #6,R5           ;FIX UP RETURN ADDRESS
 3679  011614  000261                          SEC                     ;SET C BIT FOR ERROR
 3680  011616  012602                  21$:    MOV     (SP)+,R2        ;RESTORE R2
 3681  011620  012601                          MOV     (SP)+,R1        ;RESTORE R1
 3682  011622  000205                          RTS     R5              ;RETURN
 3683
```

```
                                .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
```

```
 3732  011752  004537  005654          4$:     JSR     R5,CKRACT       ;CHK FOR RXACT = 1
 3733  011756  000001                          1
 3734  011760  103401                          BCS     6$              ;BR TO EXIT IF ERROR
 3735  011762  000241                  5$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
 3736  011764  012602                  6$:     MOV     (SP)+,R2        ;RESTORE R2
 3737  011766  012601                          MOV     (SP)+,R1        ;RESTORE R1
 3738  011770  000205                          RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
 3739
 3740
```

```
                                        .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          1$:     MOV     #TTLOOP,1$      ;INIT DEFAULT VIAORB (TTLOOP=1)
012000  112537  012050                  2$:     MOVB    (R5)+,2$        ;GET DESIRED # OF TICKS (LOWER BYTE)
012004  105725                          4$:     TSTB    (R5)+           ;SEE IF MSB SET (TTLOOP DISABLE BIT)
012006  100002                                  BPL     4$              ;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
```

```
                                .SBTTL ....STEPLU -- CLOCK THE USYRT N TIMES
                                ;**************************************************************************
                                ;* STEPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NUMBER OF CYCLES
                                ;*      PASSED IN THE WORD FOLLOWING THE CALL. THE VIA ACR MUST BE PREVIOUSLY
                                ;*      SET UP FOR T1 ONE-SHOT MODE, AND THE T1 LATCHES MUST BE PREVIOUSLY SET
                                ;*      TO CONTROL THE WIDTH OF THE CLOCK PULSE. ALL THAT THIS SUBROUTINE
                                ;*      DOES IS TO LOAD 000 INTO THE HI BYTE OF THE T1 COUNTER, FOR THE
                                ;*      DESIRED NUMBER OF TIMES.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,STEPLU
                                ;*      .WORD   <NUMBER OF CYCLES TO CLOCK>
                                ;**************************************************************************
012072                          STEPLU:
012072  010146                          MOV     R1,-(SP)        ;SAVE R1
012074  012501                          MOV     (R5)+,R1        ;INIT CYCLE COUNTER
012076  004537  003712          1$:     JSR     R5,WRITEI       ;LOAD T1C-H, START COUNTER, CLOCK 1 CYCLE
012102  120005                          VIAT1B
012104  000000                          000
012106  005301                          DEC     R1              ;DECR CYCLE COUNTER
012110  001372                          BNE     1$              ;BR IF ALL CYCLES NOT DONE YET
012112  012601                          MOV     (SP)+,R1        ;RESTORE R1
012114  000205                          RTS     R5              ;RETURN
```

```
                                        GLOBAL ERROR REPORT SECTION

                                        .SBTTL  GLOBAL ERROR REPORT SECTION

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

                                        .NLIST  BEX
012116  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%A%SEL%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%S%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 6%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  /%   %T/
012703     045  022516  022524  FMT13:  .ASCIZ  /%[illegible]%S2%03%S2%03%S2%03%S2%03%S2%03/
012751     045  031123  047445  FMT14:  .ASCIZ  /%S2%[illegible]2%03/
012766  040445  020040  042504  FMT15:  .ASCIZ  /%A  DETECTED IN %T%T%A --/
013020  040445  020040  042504  FMT15A: .ASCIZ  /%A  DETECTED @ TEST PATTERN ELEMENT @ %02/
013072  047045  052045  047445  FMT16:  .ASCIZ  /%N%T%03%S4%03%S%03/
013115     045  022516  022524  FMT16A: .ASCIZ  /%N%T%03%S%0[illegible]%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 %05%A ERRORS:/

013452  047045  040445  042524  FMT19:  .ASCIZ  /%N%ATEST %02%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  /%04%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 %02%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  /MOR 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  /RFNR NOT CLEARED/
014551     122  051105  020122  EM36:   .ASCIZ  /R  R 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  /CA IER NOT SET/
015403     115  042117  046505  EM88:   .ASCIZ  /MY  EM RDY NOT CLRD/
015426  047515  042504  020115  EM89:   .ASCIZ  /ML_EM 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  /  LO   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  /
```

```
                        ....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  051.17  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

017166  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/
```

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

 017650  051525  051131  020124  TXTUR:  .ASCIZ  /USYR1 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
3817        017744               .EVEN
3818
3819
3820                             .SBTTL ....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- "TXT__T"
3821                             ;-------------------------------------------------------------------
3822                             ;------------ TEXT ADDRESS TABLES USED BY ERROR HANDLERS ------------
3823                             ;-------------------------------------------------------------------
3824
3825  017744  017154  017160  017174  TXTMLT: .WORD   TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7
3826  017752  017211  017233  017254
3827  017760  017304  017316
3828
3829  017764  017353                           .WORD   TXTVR
3830  017766  017371  017375  017401  TXTVRT: .WORD   TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7
3831  017774  017406  017413  017420
3832  020002  017425  017432
3833  020006  017437  017444  017451           .WORD   TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF
3834  020014  017454  017460  017464
3835  020022  017470  017474
3836
3837  020026  017500                           .WORD   TXTNP
3838  020030  017505  017515  017525  TXTNPT: .WORD   TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8
3839  020036  017535  017552  017567
3840  020044  017604  017620  017634
3841  020052  017663  017671  017677  TXTURT: .WORD   TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7
3842  020060  017705  017713  017722
3843  020066  017731  017735
3844
3845
3846
```

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

                        ;-------------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR3 -- DUMP THE BYTE SELECT REGISTERS
                        ;-------------------------------------------------------------------------
                                BGNMSG  ERR3
                                                                                        ERR3::
 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
                                                                                        ERR4::
 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
 020242                                 ENDMSG
 020242                                                                                 L10003:
 020242  104423                                                                        TRAP    C$MSG
                        ;-------------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS
```

```
3903                                ;-------------------------------------------------------------------------
3904  020244                        BGNMSG  ERR7A
3905  020244                                                                          ERR7A::
3906  020244  113701  002336        MOVB    REGNUM,R1
3907  020250  006301                ASL     R1              ;AS PASSED, THIS WAS A BYTE OFFSET
3908  020252                        PRINTB  #FMT15,#TXTUR,TXTURT(R1)
3909  020252  016146  020052                                                          MOV     TXTURT(R1),-(SP)
3910  020256  012746  017650                                                          MOV     #TXTUR,-(SP)
3911  020262  012746  012766                                                          MOV     #FMT15,-(SP)
3912  020266  012746  000003                                                          MOV     #3,-(SP)
3913  020272  010600                                                                  MOV     SP,R0
3914  020274  104414                                                                  TRAP    C$PNTB
3915  020276  062706  000010                                                          ADD     #10,SP
3916  020302  004737  021262        JSR     PC,XORGR
3917  020306                        PRINTB  #FMT3,GDATA,BDATA,XDATA
3918  020306  013746  002330                                                          MOV     XDATA,-(SP)
3919  020312  013746  002326                                                          MOV     BDATA,-(SP)
3920  020316  013746  002324                                                          MOV     GDATA,-(SP)
3921  020322  012746  012163                                                          MOV     #FMT3,-(SP)
3922  020326  012746  000004                                                          MOV     #4,-(SP)
3923  020332  010600                                                                  MOV     SP,R0
3924  020334  104414                                                                  TRAP    C$PNTB
3925  020336  062706  000012                                                          ADD     #12,SP
3926  020342                        PRINTB  #ENDEMB
3927  020342  012746  012116                                                          MOV     #ENDEMB,-(SP)
3928  020346  012746  000001                                                          MOV     #1,-(SP)
3929  020352  010600                                                                  MOV     SP,R0
3930  020354  104414                                                                  TRAP    C$PNTB
3931  020356  062706  000004                                                          ADD     #4,SP
3932  020362                        ENDMSG
3933  020362                                                                  L10004:
3934  020362  104423                                                                  TRAP    C$MSG
3935                                ;-------------------------------------------------------------------------
3936                                .SBTTL ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)
3937                                ;-------------------------------------------------------------------------
3938
3939  020364                        BGNMSG  ERR10
3940  020364                                                                          ERR10::
3941  020364                        PRINTB  #FMT21,#TXT12,MPCSR
3942  020364  013746  002416                                                          MOV     MPCSR,-(SP)
3943  020370  012746  016602                                                          MOV     #TXT12,-(SP)
3944  020374  012746  013503                                                          MOV     #FMT21,-(SP)
3945  020400  012746  000003                                                          MOV     #3,-(SP)
3946  020404  010600                                                                  MOV     SP,R0
3947  020406  104414                                                                  TRAP    C$PNTB
3948  020410  062706  000010                                                          ADD     #10,SP
3949  020414                        PRINTB  #FMT22
3950  020414  012746  013513                                                          MOV     #FMT22,-(SP)
3951  020420  012746  000001                                                          MOV     #1,-(SP)
3952  020424  010600                                                                  MOV     SP,R0
3953  020426  104414                                                                  TRAP    C$PNTB
3954  020430  062706  000004                                                          ADD     #4,SP
3955  020434  013701  002336        MOV     REGNUM,R1
3956  020440  006301                ASL     R1              ;GET PTR TO USYRT REG ASCII
3957  020442                        PRINTB  #FMT27,#TXTUR,TXTURT(R1)
3958  020442  016146  020052                                                          MOV     TXTURT(R1),-(SP)
```

```
3959  020446  012746  017650                                                  MOV     #TXTUR,-(SP)
3960  020452  012746  013712                                                  MOV     #FMT27,-(SP)
3961  020456  012746  000003                                                  MOV     #3,-(SP)
3962  020462  010600                                                          MOV     SP,R0
3963  020464  104414                                                          TRAP    C$PNTB
3964  020466  062706  000010                                                  ADD     #10,SP
3965  020472  004737  021262          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
3966  020476                         PRINTB  #FMT23,GDATA,BDATA,XDATA
3967  020476  013746  002330                                                  MOV     XDATA,-(SP)
3968  020502  013746  002326                                                  MOV     BDATA,-(SP)
3969  020506  013746  002324                                                  MOV     GDATA,-(SP)
3970  020512  012746  013535                                                  MOV     #FMT23,-(SP)
3971  020516  012746  000004                                                  MOV     #4,-(SP)
3972  020522  010600                                                          MOV     SP,R0
3973  020524  104414                                                          TRAP    C$PNTB
3974  020526  062706  000012                                                  ADD     #12,SP
3975  020532  004737  022364          JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
3976  020536                         ENDMSG
3977  020536                                                          L10005:
3978  020536  104423                                                          TRAP    C$MSG
3979
3980                                 ;-------------------------------------------------------------------------
3981                                 .SBTTL ....ERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT)
3982                                 ;-------------------------------------------------------------------------
3983
3984  020540                         BGNMSG  ERR11
3985  020540                                                          ERR11::
3986  020540                         PRINTB  #FMT21,#TXT12,MPCSR
3987  020540  013746  002416                                                  MOV     MPCSR,-(SP)
3988  020544  012746  016602                                                  MOV     #TXT12,-(SP)
3989  020550  012746  013503                                                  MOV     #FMT21,-(SP)
3990  020554  012746  000003                                                  MOV     #3,-(SP)
3991  020560  010600                                                          MOV     SP,R0
3992  020562  104414                                                          TRAP    C$PNTB
3993  020564  062706  000010                                                  ADD     #10,SP
3994  020570                         PRINTB  #FMT22
3995  020570  012746  013513                                                  MOV     #FMT22,-(SP)
3996  020574  012746  000001                                                  MOV     #1,-(SP)
3997  020600  010600                                                          MOV     SP,R0
3998  020602  104414                                                          TRAP    C$PNTB
3999  020604  062706  000004                                                  ADD     #4,SP
4000  020610  013701  002336          MOV     REGNUM,R1
4001  020614  006301                  ASL     R1                      ;GET PTR TO VIA REG ASCII
4002  020616                         PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
4003  020616  016146  017766                                                  MOV     TXTVRT(R1),-(SP)
4004  020622  012746  017353                                                  MOV     #TXTVR,-(SP)
4005  020626  012746  013712                                                  MOV     #FMT27,-(SP)
4006  020632  012746  000003                                                  MOV     #3,-(SP)
4007  020636  010600                                                          MOV     SP,R0
4008  020640  104414                                                          TRAP    C$PNTB
4009  020642  062706  000010                                                  ADD     #10,SP
4010  020646  004737  021262          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
4011  020652                         PRINTB  #FMT23,GDATA,BDATA,XDATA
4012  020652  013746  002330                                                  MOV     XDATA,-(SP)
4013  020656  013746  002326                                                  MOV     BDATA,-(SP)
4014  020662  013746  002324                                                  MOV     GDATA,-(SP)
```

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

 4015  020666  012746  013535                                                  MOV     #FMT23,-(SP)
 4016  020672  012746  000004                                                  MOV     #4,-(SP)
 4017  020676  010600                                                          MOV     SP,R0
 4018  020700  104414                                                          TRAP    C$PNTB
 4019  020702  062706  000012                                                  ADD     #12,SP
 4020  020706  004737  022032          JSR     PC,ERR11$               ;GET & PRINT VIA REGISTERS
 4021  020712                          ENDMSG
 4022  020712                                                          L10006:
 4023  020712  104423                                                          TRAP    C$MSG
 4024
 4025
 4026                          ;-------------------------------------------------------------------------
 4027                          .SBTTL ....ERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)
 4028                          ;-------------------------------------------------------------------------
 4029  020714                          BGNMSG  ERR12
 4030  020714                                                          ERR12::
 4031  020714                          PRINTB  #FMT21,#TXT12,MPCSR
 4032  020714  013746  002416                                                  MOV     MPCSR,-(SP)
 4033  020720  012746  016602                                                  MOV     #TXT12,-(SP)
 4034  020724  012746  013503                                                  MOV     #FMT21,-(SP)
 4035  020730  012746  000003                                                  MOV     #3,-(SP)
 4036  020734  010600                                                          MOV     SP,R0
 4037  020736  104414                                                          TRAP    C$PNTB
 4038  020740  062706  000010                                                  ADD     #10,SP
 4039  020744                          PRINTB  #FMT22
 4040  020744  012746  013513                                                  MOV     #FMT22,-(SP)
 4041  020750  012746  000001                                                  MOV     #1,-(SP)
 4042  020754  010600                                                          MOV     SP,R0
 4043  020756  104414                                                          TRAP    C$PNTB
 4044  020760  062706  000004                                                  ADD     #4,SP
 4045  020764  013701  002336          MOV     REGNUM,R1
 4046  020770  006301                  ASL     R1                      ;GET PTR TO USYRT REG ASCII
 4047  020772                          PRINTB  #FMT27,#TXTUR,TXTURT(R1)
 4048  020772  016146  020052                                                  MOV     TXTURT(R1), (SP)
 4049  020776  012746  017650                                                  MOV     #TXTUR,-(SP)
 4050  021002  012746  013712                                                  MOV     #FMT27,-(SP)
 4051  021006  012746  000003                                                  MOV     #3,-(SP)
 4052  021012  010600                                                          MOV     SP,R0
 4053  021014  104414                                                          TRAP    C$PNTB
 4054  021016  062706  000010                                                  ADD     #10,SP
 4055  021022  004737  022364          JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
 4056  021026                          ENDMSG
 4057  021026                                                          L10007:
 4058  021026  104423                                                          TRAP    C$MSG
 4059
 4060
 4061                          ;-------------------------------------------------------------------------
 4062                          .SBTTL ....ERROR HANDLER -- ERR13 -- RAM ADDRESS ERRORS
 4063                          ;-------------------------------------------------------------------------
 4064  021030                          BGNMSG  ERR13
 4065  021030                                                          ERR13::
 4066  021030                          PRINTB  #FMT21,#TXT12,MPCSR
 4067  021030  013746  002416                                                  MOV     MPCSR,-(SP)
 4068  021034  012746  016602                                                  MOV     #TXT12,-(SP)
 4069  021040  012746  013503                                                  MOV     #FMT21,-(SP)
 4070  021044  012746  000003                                                  MOV     #3,-(SP)
```

```
4071  021050  010600                                                          MOV     SP,R0
4072  021052  104414                                                          TRAP    C$PNTB
4073  021054  062706  000010                                                  ADD     #10,SP
4074  021060                        PRINTB  #FMT40,REGNUM
4075  021060  013746  002336                                                  MOV     REGNUM,-(SP)
4076  021064  012746  014150                                                  MOV     #FMT40,-(SP)
4077  021070  012746  000002                                                  MOV     #2,-(SP)
4078  021074  010600                                                          MOV     SP,R0
4079  021076  104414                                                          TRAP    C$PNTB
4080  021100  062706  000006                                                  ADD     #6,SP
4081  021104  004737  021262                JSR     PC,XORGB        ;COMPUTE XOR OF GOOD AND BAD DATA
4082  021110                        PRINTB  #FMT23,GDATA,BDATA,XDATA
4083  021110  013746  002330                                                  MOV     XDATA,-(SP)
4084  021114  013746  002326                                                  MOV     BDATA,-(SP)
4085  021120  013746  002324                                                  MOV     GDATA,-(SP)
4086  021124  012746  013535                                                  MOV     #FMT23,-(SP)
4087  021130  012746  000004                                                  MOV     #4,-(SP)
4088  021134  010600                                                          MOV     SP,R0
4089  021136  104414                                                          TRAP    C$PNTB
4090  021140  062706  000012                                                  ADD     #12,SP
4091  021144                        ENDMSG
4092  021144                                                          L10010:
4093  021144  104423                                                          TRAP    C$MSG
4094
4095
4096                          ;-------------------------------------------------------------------
4097                          .SBTTL ....ERROR HANDLER -- ERR14 -- VIA REG ERRORS (VIA PRINTOUT)
4098                          ;-------------------------------------------------------------------
4099  021146                  BGNMSG  ERR14
4100  021146                                                          ERR14::
4101  021146                  PRINTB  #FMT21,#TXT12,MPCSR
4102  021146  013746  002416                                                  MOV     MPCSR,-(SP)
4103  021152  012746  016602                                                  MOV     #TXT12,-(SP)
4104  021156  012746  013503                                                  MOV     #FMT21,-(SP)
4105  021162  012746  000003                                                  MOV     #3,-(SP)
4106  021166  010600                                                          MOV     SP,R0
4107  021170  104414                                                          TRAP    C$PNTB
4108  021172  062706  000010                                                  ADD     #10,SP
4109  021176                  PRINTB  #FMT22
4110  021176  012746  013513                                                  MOV     #FMT22,-(SP)
4111  021202  012746  000001                                                  MOV     #1,-(SP)
4112  021206  010600                                                          MOV     SP,R0
4113  021210  104414                                                          TRAP    C$PNTB
4114  021212  062706  000004                                                  ADD     #4,SP
4115  021216  013701  002336                MOV     REGNUM,R1
4116  021222  006301                        ASL     R1                      ;GET PTR TO VIA REG ASCII
4117  021224                        PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
4118  021224  016146  017766                                                  MOV     TXTVRT(R1),-(SP)
4119  021230  012746  017353                                                  MOV     #TXTVR,-(SP)
4120  021234  012746  013712                                                  MOV     #FMT27,-(SP)
4121  021240  012746  000003                                                  MOV     #3,-(SP)
4122  021244  010600                                                          MOV     SP,R0
4123  021246  104414                                                          TRAP    C$PNTB
4124  021250  062706  000010                                                  ADD     #10,SP
4125  021254  004737  022032                JSR     PC,ERR11$               ;GET & PRINT VIA REGISTERS
4126  021260                        ENDMSG
```

```
                                                                                L10011:
                                                                                        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)
```

```
4183  021426  013746  002216                                                  MOV     BSR6,-(SP)
4184  021432  013746  002214                                                  MOV     BSR5,-(SP)
4185  021436  013746  002212                                                  MOV     BSR4,-(SP)
4186  021442  012746  012345                                                  MOV     #FMT4C,-(SP)
4187  021446  012746  000005                                                  MOV     #5,-(SP)
4188  021452  010600                                                          MOV     SP,R0
4189  021454  104415                                                          TRAP    C$PNTX
4190  021456  062706  000014                                                  ADD     #14,SP
4191  021462                          PRINTX  #FMT4B,#TXT2A
4192  021462  012746  015702                                                  MOV     #TXT2A,-(SP)
4193  021466  012746  012340                                                  MOV     #FMT4B,-(SP)
4194  021472  012746  000002                                                  MOV     #2,-(SP)
4195  021476  010600                                                          MOV     SP,R0
4196  021500  104415                                                          TRAP    C$PNTX
4197  021502  062706  000006                                                  ADD     #6,SP
4198  021506                          PRINTX  #FMT4A,BSR10,BSR11,BSR12,BSR13
4199  021506  013746  002230                                                  MOV     BSR13,-(SP)
4200  021512  013746  002226                                                  MOV     BSR12,-(SP)
4201  021516  013746  002224                                                  MOV     BSR11,-(SP)
4202  021522  013746  002222                                                  MOV     BSR10,-(SP)
4203  021526  012746  012305                                                  MOV     #FMT4A,-(SP)
4204  021532  012746  000005                                                  MOV     #5,-(SP)
4205  021536  010600                                                          MOV     SP,R0
4206  021540  104415                                                          TRAP    C$PNTX
4207  021542  062706  000014                                                  ADD     #14,SP
4208  021546                          PRINTX  #FMT4B,#TXT2B
4209  021546  012746  015741                                                  MOV     #TXT2B,-(SP)
4210  021552  012746  012340                                                  MOV     #FMT4B,-(SP)
4211  021556  012746  000002                                                  MOV     #2,-(SP)
4212  021562  010600                                                          MOV     SP,R0
4213  021564  104415                                                          TRAP    C$PNTX
4214  021566  062706  000006                                                  ADD     #6,SP
4215  021572                          PRINTX  #FMT4C,BSR14,BSR15,BSR16,BSR17
4216  021572  013746  002240                                                  MOV     BSR17,-(SP)
4217  021576  013746  002236                                                  MOV     BSR16,-(SP)
4218  021602  013746  002234                                                  MOV     BSR15,-(SP)
4219  021606  013746  002232                                                  MOV     BSR14,-(SP)
4220  021612  012746  012345                                                  MOV     #FMT4C,-(SP)
4221  021616  012746  000005                                                  MOV     #5,-(SP)
4222  021622  010600                                                          MOV     SP,R0
4223  021624  104415                                                          TRAP    C$PNTX
4224  021626  062706  000014                                                  ADD     #14,SP
4225  021632  000207                  RTS     PC
4226
4227                            ;-------------------------------------------------------------------
4228                            .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR5$
4229                            ;-------------------------------------------------------------------
4230                            ;       COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS
4231  021634                    ERR5$:
4232  021634                          PRINTX  #FMT4,#TXT6,#TXT4
4233  021634  012746  016034                                                  MOV     #TXT4,-(SP)
4234  021640  012746  016137                                                  MOV     #TXT6,-(SP)
4235  021644  012746  012247                                                  MOV     #FMT4,-(SP)
4236  021650  012746  000003                                                  MOV     #3,-(SP)
4237  021654  010600                                                          MOV     SP,R0
4238  021656  104415                                                          TRAP    C$PNTX
```

```
4239  021660  062706  000010                                                    ADD     #10,SP
4240  021664                          PRINTX  #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS
4241  021664  013746  002210                                                    MOV     WSR6,-(SP)
4242  021670  013746  002206                                                    MOV     WSR4,-(SP)
4243  021674  013746  002204                                                    MOV     WSR2,-(SP)
4244  021700  013746  002202                                                    MOV     WSR0,-(SP)
4245  021704  012746  012655                                                    MOV     #FMT11,-(SP)
4246  021710  012746  000005                                                    MOV     #5,-(SP)
4247  021714  010600                                                            MOV     SP,R0
4248  021716  104415                                                            TRAP    C$PNTX
4249  021720  062706  000014                                                    ADD     #14,SP
4250  021724                          PRINTX  #FMT48,#TXT4A
4251  021724  012746  016074                                                    MOV     #TXT4A,-(SP)
4252  021730  012746  012340                                                    MOV     #FMT48,-(SP)
4253  021734  012746  000002                                                    MOV     #2,-(SP)
4254  021740  010600                                                            MOV     SP,R0
4255  021742  104415                                                            TRAP    C$PNTX
4256  021744  062706  000006                                                    ADD     #6,SP
4257  021750                          PRINTX  #FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS
4258  021750  013746  002220                                                    MOV     WSR16,-(SP)
4259  021754  013746  002216                                                    MOV     WSR14,-(SP)
4260  021760  013746  002214                                                    MOV     WSR12,-(SP)
4261  021764  013746  002212                                                    MOV     WSR10,-(SP)
4262  021770  012746  012655                                                    MOV     #FMT11,-(SP)
4263  021774  012746  000005                                                    MOV     #5,-(SP)
4264  022000  010600                                                            MOV     SP,R0
4265  022002  104415                                                            TRAP    C$PNTX
4266  022004  062706  000014                                                    ADD     #14,SP
4267  022010                          PRINTB  #ENDEMB
4268  022010  012746  012116                                                    MOV     #ENDEMB,-(SP)
4269  022014  012746  000001                                                    MOV     #1,-(SP)
4270  022020  010600                                                            MOV     SP,R0
4271  022022  104414                                                            TRAP    C$PNTB
4272  022024  062706  000004                                                    ADD     #4,SP
4273  022030  000207                  RTS     PC
4274
4275                            ;-------------------------------------------------------------------
4276                            .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR11$
4277                            ;-------------------------------------------------------------------
4278                            ;       COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS
4279
4280  022032  004737  004460  ERR11$: JSR     PC,GETVRS               ;GET VIA REGS FOR PRINTOUT
4281  022036                          PRINTX  #FMT24,#TXT16,#TXT17
4282  022036  012746  016760                                                    MOV     #TXT17,-(SP)
4283  022042  012746  016745                                                    MOV     #TXT16,-(SP)
4284  022046  012746  013614                                                    MOV     #FMT24,-(SP)
4285  022052  012746  000003                                                    MOV     #3,-(SP)
4286  022056  010600                                                            MOV     SP,R0
4287  022060  104415                                                            TRAP    C$PNTX
4288  022062  062706  000010                                                    ADD     #10,SP
4289  022066                          PRINTX  #FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6
4290  022066  013746  002270                                                    MOV     VREGS+6,-(SP)
4291  022072  013746  002266                                                    MOV     VREGS+4,-(SP)
4292  022076  013746  002264                                                    MOV     VREGS+2,-(SP)
4293  022102  013746  002262                                                    MOV     VREGS+0,-(SP)
4294  022106  012746  013627                                                    MOV     #FMT25,-(SP)
```

```
 4295  022112  012746  000005                                                  MOV     #5.-(SP)
 4296  022116  010600                                                          MOV     SP,R0
 4297  022120  104415                                                          TRAP    C$PNTX
 4298  022122  062706  000014                                                  ADD     #14.SP
 4299  022126                    PRINTX  #FMT29  #TXT18
 4300  022126  012746  017015                                                  MOV     #TXT18,-(SP)
 4301  022132  012746  013755                                                  MOV     #FMT29,-(SP)
 4302  022136  012746  000002                                                  MOV     #2.-(SP)
 4303  022142  010600                                                          MOV     SP,R0
 4304  022144  104415                                                          TRAP    C$PNTX
 4305  022146  062706  000006                                                  ADD     #6.SP
 4306  022152                    PRINTX  #FMT26,VREGS+8.,VREGS+10.,VREGS+12.,VREGS+14.
 4307  022152  013746  002300                                                  MOV     VREGS+14.,-(SP)
 4308  022156  013746  002276                                                  MOV     VREGS+12.,-(SP)
 4309  022162  013746  002274                                                  MOV     VREGS+10.,-(SP)
 4310  022166  013746  002272                                                  MOV     VREGS+8.,-(SP)
 4311  022172  012746  013657                                                  MOV     #FMT26,-(SP)
 4312  022176  012746  000005                                                  MOV     #5.-(SP)
 4313  022202  010600                                                          MOV     SP,R0
 4314  022204  104415                                                          TRAP    C$PNTX
 4315  022206  062706  000014                                                  ADD     #14.SP
 4316  022212                    PRINTX  #FMT29,#TXT19
 4317  022212  012746  017056                                                  MOV     #TXT19,-(SP)
 4318  022216  012746  013755                                                  MOV     #FMT29,-(SP)
 4319  022222  012746  000002                                                  MOV     #2.-(SP)
 4320  022226  010600                                                          MOV     SP,R0
 4321  022230  104415                                                          TRAP    C$PNTX
 4322  022232  062706  000006                                                  ADD     #6.SP
 4323  022236                    PRINTX  #FMT25,VREGS+16.,VREGS+18.,VREGS+20.,VREGS+22.
 4324  022236  013746  002310                                                  MOV     VREGS+22.,-(SP)
 4325  022242  013746  002306                                                  MOV     VREGS+20.,-(SP)
 4326  022246  013746  002304                                                  MOV     VREGS+18.,-(SP)
 4327  022252  013746  002302                                                  MOV     VREGS+16.,-(SP)
 4328  022256  012746  013627                                                  MOV     #FMT25,-(SP)
 4329  022262  012746  000005                                                  MOV     #5.-(SP)
 4330  022266  010600                                                          MOV     SP,R0
 4331  022270  104415                                                          TRAP    C$PNTX
 4332  022272  062706  000014                                                  ADD     #14.SP
 4333  022276                    PRINTX  #FMT29,#TXT20
 4334  022276  012746  017112                                                  MOV     #TXT20.-(SP)
 4335  022302  012746  013755                                                  MOV     #FMT29,-(SP)
 4336  022306  012746  000002                                                  MOV     #2.-(SP)
 4337  022312  010600                                                          MOV     SP,R0
 4338  022314  104415                                                          TRAP    C$PNTX
 4339  022316  062706  000006                                                  ADD     #6.SP
 4340  022322                    PRINTX  #FMT26,VREGS+24.,VREGS+26.,VREGS+28.,VREGS+30.
 4341  022322  013746  002320                                                  MOV     VREGS+30.,-(SP)
 4342  022326  013746  002316                                                  MOV     VREGS+28.,-(SP)
 4343  022332  013746  002314                                                  MOV     VREGS+26.,-(SP)
 4344  022336  013746  002312                                                  MOV     VREGS+24.,-(SP)
 4345  022342  012746  013657                                                  MOV     #FMT26,-(SP)
 4346  022346  012746  000005                                                  MOV     #5.-(SP)
 4347  022352  010600                                                          MOV     SP,R0
 4348  022354  104415                                                          TRAP    C$PNTX
 4349  022356  062706  000014                                                  ADD     #14.SP
 4350  022362  000207                    RTS     PC
```

```
                        .....ERROR HANDLER SUBROUTINE -- ERR11#

4351
4352                    ;-------------------------------------------------------------------------
4353                    .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR12#
4354                    ;-------------------------------------------------------------------------
4355                    ;       COMMON ERROR ROUTINE TO GET AND PRINTOUT USYRT REGISTERS
4356
4357  022364  004737  004360  ERR12#: JSR     PC,GETURS               ;GET USYRT REGS FOR PRINTOUT
4358  022370                          PRINTX  #FMT24,#TXT13,#TXT14
4359  022370  012746  016645                                                  MOV     #TXT14,-(SP)
4360  022374  012746  016630                                                  MOV     #TXT13,-(SP)
4361  022400  012746  013614                                                  MOV     #FMT24,-(SP)
4362  022404  012746  000003                                                  MOV     #3,-(SP)
4363  022410  010600                                                          MOV     SP,R0
4364  022412  104415                                                          TRAP    C$PNTX
4365  022414  062706  000010                                                  ADD     #10,SP
4366  022420                          PRINTX  #FMT25,UREGS+0,UREGS+2,UREGS+4,UREGS+6
4367  022420  013746  002250                                                  MOV     UREGS+6,-(SP)
4368  022424  013746  002246                                                  MOV     UREGS+4,-(SP)
4369  022430  013746  002244                                                  MOV     UREGS+2,-(SP)
4370  022434  013746  002242                                                  MOV     UREGS+0,-(SP)
4371  022440  012746  013627                                                  MOV     #FMT25,-(SP)
4372  022444  012746  000005                                                  MOV     #5,-(SP)
4373  022450  010600                                                          MOV     SP,R0
4374  022452  104415                                                          TRAP    C$PNTX
4375  022454  062706  000014                                                  ADD     #14,SP
4376  022460                          PRINTX  #FMT29,#TXT15
4377  022460  012746  016703                                                  MOV     #TXT15,-(SP)
4378  022464  012746  013755                                                  MOV     #FMT29,-(SP)
4379  022470  012746  000002                                                  MOV     #2,-(SP)
4380  022474  010600                                                          MOV     SP,R0
4381  022476  104415                                                          TRAP    C$PNTX
4382  022500  062706  000006                                                  ADD     #6,SP
4383  022504                          PRINTX  #FMT26,UREGS+10,UREGS+12,UREGS+14,UREGS+16
4384  022504  013746  002260                                                  MOV     UREGS+16,-(SP)
4385  022510  013746  002256                                                  MOV     UREGS+14,-(SP)
4386  022514  013746  002254                                                  MOV     UREGS+12,-(SP)
4387  022520  013746  002252                                                  MOV     UREGS+10,-(SP)
4388  022524  012746  013657                                                  MOV     #FMT26,-(SP)
4389  022530  012746  000005                                                  MOV     #5,-(SP)
4390  022534  010600                                                          MOV     SP,R0
4391  022536  104415                                                          TRAP    C$PNTX
4392  022540  062706  000014                                                  ADD     #14,SP
4393  022544  000207                  RTS     PC
4394
4395                    .EVEN
```

```
                                LOAD DEVICE PROTECTION TABLE

                        .SBTTL  LOAD DEVICE PROTECTION TABLE

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

022546                          BGNPROT
022546                                                                  L$PROT::
022546  177777                  .WORD   -1      ;DON'T CHK CSR ADRS
022550  177777                  .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
022552  177777                  .WORD   -1      ;DON'T CHK DRIVE NO.
022554                          ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

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

022554                                  BGNINIT
022554                                                                          L$INIT::

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  103473                                                                  BCS     ENDIT
022706  000414                          BR      GETPRM

022710                          STARST:
```

```
4465  022710  005037  002412                  CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
4466
4467                                  ;CLEAR DEVICE MAP
4468  022714  005037  002404                  CLR     DEVMAP
4469  022720                        NEWST:
4470  022720  012737  177777  002334          MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
4471  022726  005237  002412                  INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
4472  022732  012737  000001  002406          MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
4473
4474                                  ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
4475                                  ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
4476  022740                        GETPRM:
4477  022740  005237  002334                  INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
4478  022744                                  GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
4479  022744  013700  002334                                                          MOV     LOGDEV,R0
4480  022750  104442                                                                  TRAP    C$GPHRD
4481  022752  010001                                                                  MOV     R0,R1
4482  022754                                  BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
4483  022754  103403                                                                  BCS     10$
4484  022756  006337  002406                  ASL     DEVPTR          ;SHIFT DEVICE POINTER
4485  022762  000766                          BR      GETPRM          ;SKIP THIS DEVICE
4486  022764  053737  002406  002404  10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
4487  022772  006337  002406                  ASL     DEVPTR          ;SHIFT BIT POINTER
4488
4489  022776  012102                          MOV     (R1)+,R2        ;R2=CSR ADDR VALUE
4490  023000  012703  002416                  MOV     #MPCSR,R3       ;R3=POINTER TO CSR ADDR STORAGE AREA
4491
4492  023004  010223                  11$:    MOV     R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
4493  023006  005202                          INC     R2              ;BUMP BSEL ADDR
4494  023010  022703  002456                  CMP     #BSEL17+2,R3    ;ALL 16 ADDRESSES MOVED ?
4495  023014  001373                          BNE     11$             ;NO: DO ANOTHER ADDRESS
4496                                                                  ;YES: CONTINUE
4497
4498  023016  011137  002456                  MOV     (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
4499  023022  012137  002460                  MOV     (R1)+,MPOVEC
4500  023026  062737  000004  002460          ADD     #4,MPOVEC       ;GET DMV11 OUTPUT INTRPT VECTOR
4501  023034  012137  002462                  MOV     (R1)+,MPRIOR    ;GET DMV11 DEVICE PRIORITY
4502  023040  012137  002464                  MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
4503  023044  012137  002466                  MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
4504  023050  012137  002470                  MOV     (R1)+,BRDTYP    ;GET DMV-11 BOARD TYPE
4505  023054  012137  002472                  MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
4506  023060  011137  002474                  MOV     (R1),BDRATE     ;GET BAUD RATE FOR THIS DEVICE
4507                                  ;ISSUE LSI BUS RESET, TO INIT DMV11
4508  023064                                  BRESET
4509  023064  104433                                                                  TRAP    C$RESET
4510  023066  005000                          CLR     R0              ;# TIME DELAY TO ALLOW COMPLETION
4511  023070  000240                  15$:    NOP                     ;# OF DMV11 MICRODIAGNOSTICS.
4512  023072  077002                          SOB     R0,15$          ;#
4513  023074                        ENDIT:
4514  023074                                  ENDINIT
4515  023074                                                                  L10013:
4516  023074  104411                                                                  TRAP    C$INIT
```

```
4517                                    .SBTTL AUTO DROP UNIT SECTION
4518
4519                                    ;////////////////////////////////////////////////////////////////////////////
4520                                    ;/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
4521                                    ;/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
4522                                    ;////////////////////////////////////////////////////////////////////////////
4523
4524                                    ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
4525                                    ;
4526                                    ;       THIS ALGORITHM IS THE SAME A CVDMA TEST # 1 EXCEPT THAT TEST
4527                                    ;       WILL JUST REPORT THE FAILURE AND GO ON -- THIS ROUTINE WILL CAUSE THE
4528                                    ;       DEVICE TO BE DROPPED IF A BUS-TIMEOUT OCCURS WHEN ANY OF THE CSR'S
4529                                    ;       ARE ACCESSED WITH EITHER A "TST" OR "TSTB" INSTRUCTION.
4530                                    ;
4531                                    ;---++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
4532
4533  023076                                    BGNAUTO
4534  023076                                                                          L$AUTO::
4535
4536  023076                                    SETVEC  #4,#AD.HIT,#0   ;SETUP INVALID-ADDRESS TRAP VECTOR
4537  023076  012746  000000                                                        MOV     #0,-(SP)
4538  023102  012746  023214                                                        MOV     #AD.HIT,-(SP)
4539  023106  012746  000004                                                        MOV     #4,-(SP)
4540  023112  012746  000003                                                        MOV     #3,-(SP)
4541  023116  104437                                                                TRAP    C$SVEC
4542  023120  062706  000010                                                        ADD     #10,SP
4543  023124  005037  002546                   CLR     TMP0            ;INITIALIZE TRAP FLAG REGISTER
4544  023130  012702  000001                   MOV     #1,R2           ;FLAG BIT
4545  023134  013703  002416                   MOV     BSEL0,R3        ;INIT ADDRESS POINTER
4546
4547  023140  105723                  1$:      TSTB    (R3)+           ;ACCESS THE CSR'S BY BYTES.
4548  023142  006302                           ASL     R2
4549  023144  103375                           BCC     1$
4550
4551  023146  013703  002416                   MOV     BSEL0,R3        ;RE-INIT ADDRESS POINTER
4552  023152  012702  000001                   MOV     #1,R2           ;RE-INIT FLAG BIT
4553  023156  005723                  2$:      TST     (R3)+           ;ACCESS THE CSR'S BY WORDS.
4554  023160  006302                           ASL     R2
4555  023162  006302                           ASL     R2
4556  023164  103374                           BCC     2$
4557
4558  023166                                    CLRVEC  #4              ;RESTORE THE VECTOR TO DS
4559  023166  012700  000004                                                        MOV     #4,R0
4560  023172  104436                                                                TRAP    C$CVEC
4561  023174  005737  002546                   TST     TMP0            ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP?
4562  023200  001403                           BEQ     AD.OK           ;NO,  EXIT TEST
4563  023202                                    DODU    LOGDEV          ;YES, DROP THIS LOGICAL DEV.
4564  023202  013700  002334                                                        MOV     LOGDEV,R0
4565  023206  104451                                                                TRAP    C$DODU
4566
```

```
 4567  023210  000240                AD.OK:  NOP                     ;(FOR PATCHING IN A HALT IF NECESSARY)
 4568
 4569  023212                              ENDAUTO
 4570  023212                                                                      L10014:
 4571  023212  104461                                                                      TRAP    C$AUTO
 4572
 4573  023214  050237  002546          AD.HIT: BIS     R2,TMP0         ;FLAG THE HIT IF WE GET IT!
 4574  023220  000002                          RTI                     ;RETURN
 4575
```

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

023222                                  BGNCLN
023222                                                                  L$CLEAN::

023222                                  ENDCLN
023222                                                                  L10015:
023222  104412                                                                  TRAP    C$CLEAN
```

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

023230                          BGNAU
023230                                                          L$AU::
023230                          ENDAU
023230                                                          L10017:
023230  104452                                                          TRAP    C$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
023232                                                                  T1::

023232  004737  005376                JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

023236  004537  007234                JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
023242  063626                        DDCMP!NOCHK!SYNCH!STRIPS ;SET DDCMP,NO CHECK,SYNCH=226
023244  000000                        0                       ;USE 8 BIT CHARS
023246  103003                        BCC     .+8.            ;BR IF NO ERROR
023250                                ERROR                   ;REPORT STACKED ERROR
023250  104460                                                                TRAP    C$ERROR
023252                                ESCAPE  TST             ;SKIP TO END OF TEST
023252  104410                                                                TRAP    C$ESCAPE
023254  000254                                                                .WORD   L10020-.

023256  004537  003712                JSR     R5,WRITEI       ;SET SHIFT REGISTER TO
023262  120013                        VIAACR                  ; "SYSTEM CLOCK RATE" MODE (CB1=CLK)
023264  000210                        210                     ;(BIT 7 PREVIOUSLY SET)
023266  103003                        BCC     .+8.            ;BR IF NO ERROR
023270                                ERROR                   ; REPORT STACKED ERROR
023270  104460                                                                TRAP    C$ERROR
023272                                ESCAPE  TST             ;SKIP TO END OF TEST
023272  104410                                                                TRAP    C$ESCAPE
023274  000234                                                                .WORD   L10020-.

023276  004537  010250                JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
023302  000001                        TSOM                    ;AND KNOCK DOWN TBMT
023304  000007                        7.

023306  004537  010250                JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
023312  000001                        TSOM                    ;AND KNOCK DOWN TBMT
023314  000010                        8.

023316  004537  010250                JSR     R5,TXCTRL       ;CLEAR TSOM (GET READY TO SEND DATA)
023322  000000                        000
023324  000000                        0

023326  004537  010136                JSR     R5,TXCHAR       ;LOAD 125, TX 3RD SYNCH
023332  000125                        125
023334  000010                        8.
023336  103003                        BCC     .+8.            ;BR IF NO ERROR
```

```
4674  023340                          ERROR                   ;REPORT STACKED ERROR
4675  023340  104460                                                  TRAP    C$ERROR
4676  023342                          ESCAPE  TST             ;SKIP TO END OF TEST
4677  023342  104410                                                  TRAP    C$ESCAPE
4678  023344  000164                                                  .WORD   L10020-.
4679                          ;-----------------------------------------------------------------
4680  023346  012702  000004          MOV     #4,R2           ;** TRANSFER 4 CHARACTERS **
4681
4682  023352  004537  003712  1$:     JSR     R5,WRITEI       ;SET RTS & FULL DUPLEX (SO STEPLU WORKS)
4683  023356  120000                  VIAORB
4684  023360  000142                  TXEN!RXEN!TTLOOP
4685  023362  103003                  BCC     .+8.            ;BR IF NO ERROR
4686  023364                          ERROR                   ; REPORT STACKED ERROR
4687  023364  104460                                                  TRAP    C$ERROR
4688  023366                          ESCAPE  TST             ;SKIP TO END OF TEST
4689  023366  104410                                                  TRAP    C$ESCAPE
4690  023370  000140                                                  .WORD   L10020-.
4691
4692  023372  004537  012072          JSR     R5,STEPLU       ;FLIP TSO BIT VALUE(WILL BE SHIFTED INTO
4693  023376  000001                  1                       ; FIFO DURING FLUSHING).
4694
4695  023400  004537  003712          JSR     R5,WRITEI       ;CLEAR RTS, SET HDX (SO THAT SR CLOCK WORKS)
4696  023404  120000                  VIAORB
4697  023406  000152                  TXEN!RXEN!RTSND!TTLOOP  ;#
4698  023410  103003                  BCC     .+8.            ;BR IF NO ERROR
4699  023412                          ERROR                   ; REPORT STACKED ERROR
4700  023412  104460                                                  TRAP    C$ERROR
4701  023414                          ESCAPE  TST             ;SKIP TO END OF TEST
4702  023414  104410                                                  TRAP    C$ESCAPE
4703  023416  000112                                                  .WORD   L10020-.
4704
4705  023420  004537  003566          JSR     R5,READI        ;READ VIA SHIFT REGISTER (SHOULD CAUSE
4706  023424  120012                  VIASR                   ; 8 CLOCKS FROM CB1 LEAD => FIFO)
4707  023426  000000                  000
4708  023430  103003                  BCC     .+8.            ;BR IF NO ERROR
4709  023432                          ERROR                   ; REPORT STACKED ERROR
4710  023432  104460                                                  TRAP    C$ERROR
4711  023434                          ESCAPE  TST             ;SKIP TO END OF TEST
4712  023434  104410                                                  TRAP    C$ESCAPE
4713  023436  000072                                                  .WORD   L10020-.
4714
4715  023440  077234                  SOB     R2,1$           ;** LOOP UNTIL ALL 4 SENT (VIA CB1) **
4716                          ;-----------------------------------------------------------------
4717  023442  004537  010350          JSR     R5,RXCHAR       ;READ AND CHECK FOR 377
4718  023446  000377                  377                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
4719  023450  000000                  0                       ;* DIDN'T WORK.
4720  023452  100000                  NOCRDA
4721  023454  103003                  BCC     .+8.            ;BR IF NO ERROR
4722  023456                          ERROR                   ;REPORT STACKED ERROR
4723  023456  104460                                                  TRAP    C$ERROR
4724  023460                          ESCAPE  TST             ;SKIP TO END OF TEST
4725  023460  104410                                                  TRAP    C$ESCAPE
4726  023462  000046                                                  .WORD   L10020-.
4727
4728  023464  004537  010350          JSR     R5,RXCHAR       ;READ AND CHECK FOR 000
4729  023470  000003                  003                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
```

M9


```
4730  023472  000000                  0                       ;* DIDN'T WORK.
4731  023474  100000                  NOCRDA
4732  023476  103003                  BCC     .+8.            ;BR IF NO ERROR
4733  023500                          ERROR                   ;REPORT STACKED ERROR
4734  023500  104460                                                  TRAP    C$ERROR
4735  023502                          ESCAPE  TST             ;SKIP TO END OF TEST
4736  023502  104410                                                  TRAP    C$ESCAPE
4737  023504  000024                                                  .WORD   L10020-.
4738
4739  023506  004537  010350          JSR     R5,RXCHAR       ;READ AND CHECK FOR 377
4740  023512  000360                  360                     ;* ERROR HERE INDICATES HI-SPEED SR CLOCK
4741  023514  000000                  0                       ;* DIDN'T WORK
4742  023516  100000                  NOCRDA
4743  023520  103003                  BCC     .+8.            ;BR IF NO ERROR
4744  023522                          ERROR                   ;REPORT STACKED ERROR
4745  023522  104460                                                  TRAP    C$ERROR
4746  023524                          ESCAPE  TST             ;SKIP TO END OF TEST
4747  023524  104410                                                  TRAP    C$ESCAPE
4748  023526  000002                                                  .WORD   L10020-.
4749  023530                  ENDTST
4750  023530                                                  L10020:
4751  023530  104401                                                  TRAP    C$ETST
```

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

                                ;++*************************************************************************
                                ;*
                                ;*      TEST 2 -- INTEGRAL MODEM INTERFACE TEST
                                ;*
                                ;*  THE INTEGRAL MODEM IS SELECTED BY THE PROGRAM AND A MESSAGE IS
                                ;*  TRANSMITTED,  RECEIVED,  AND CHECKED USING A TURNAROUND CONNECTOR ON
                                ;*  THE  BOARD  OR  AT THE END OF A CABLE. THE FOLLOWING MESSAGE WILL BE
                                ;*  SENT IN BCP MODE WITH CRC-16 SPECIFIED:
                                ;*
                                ;*      SYNC SYNC 000 125 252 377 000 CRC1 CRC2 SYNC
                                ;*
                                ;*  IF  THE  P-TABLE  FOR  THE  CURRENT  UNIT INDICATES THAT NO EXTERNAL
                                ;*  TURNAROUND IS PROVIDED, THE TEST WILL BE SKIPPED FOR THAT UNIT.
                                ;*
                                ;--*************************************************************************
                                ;
023532                          ;       BGNTST                                                  T2::
023532  012737  000002  002414          MOV     #2.,TSTNUM      ;SET TEST NO. FOR POSSIBLE PRINTOUT
023540  004737  005376                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
023544  004537  007532                  JSR     R5,CKLPBK       ;SEE IF THIS INTERFACE CAN BE RUN
023550  000001                          INTGRL
023552  103002                          BCC     2$              ;BR IF YES
023554                                  EXIT    TST             ;WRONG INTERFACE - SKIP TEST
023554  104432                                                                  TRAP    C$EXIT
023556  001630                                                                  .WORD   L10021-.
                                ;LOAD 6502 MICROCODE FOR INTEGRAL MODEM TEST INTO RAM PAGE 2
                                2$:     MOV     #MCODE,R1       ;GET STARTING ADRS OF DMV MICROCODE
                                        MOV     #RAMADR,R2      ;GET STARTING ADRS OF RAM PAGE 2
                                3$:     MOVB    (R1)+,6$        ;SET DATA BYTE TO BE WRITTEN
                                        MOV     R2,4$           ;SET RAM WRITE ADRS
                                        JSR     R5,WRITEI       ;WRITE A DATA BYTE INTO RAM
                                4$:     .WORD   0
                                6$:     .WORD   0
                                        INC     R2              ;INCR RAM ADRS
                                        CMP     R1,#ENDCOD      ;SEE IF ALL CODE LOADED YET
                                        BLO     3$              ;BR IF NOT

                                ;READ AND VERIFY 6502 MICROCODE IN RAM
                                        MOV     #MCODE,R1       ;GET STARTING ADRS OF DMV MICROCODE TO CHECK
                                        MOV     #RAMADR,R2      ;GET STARTING ADRS OF RAM PAGE 2
                                8$:     MOV     R2,10$          ;SET RAM READ ADRS
                                        JSR     R5,READI        ;READ A RAM BYTE
                                10$:    .WORD   0
                                12$:    .WORD   0
                                        CMPB    (R1)+,12$       ;SEE IF BYTE IS CORRECT
                                        BEQ     16$             ;BR IF CORRECT
                                        MOV     R2,REGNUM       ;SET RAM ADRS FOR ERROR REPORT
                                        CLR     GDATA           ;SET EXPECTED RAM DATA
                                        MOVB    -1(R1),GDATA
                                        CLR     BDATA           ;SET ACTUAL RAM DATA
                                        MOVB    12$,BDATA
                                ;REPORT RAM ERROR LOADING MICROCODE
```

```
4808  023702                                  GEDF    EM79,ERR13
4809                                                  ;       "DEVICE FATAL" ERROR # 38
4810  023702  104455                                                          TRAP    C$ERDF
4811  023704  000046                                                          .WORD   38
4812  023706  015125                                                          .WORD   EM79
4813  023710  021030                                                          .WORD   ERR13
4814  023712                                  ESCAPE  TST
4815  023712  104410                                                          TRAP    C$ESCAPE
4816  023714  001472                                                          .WORD   L10021-.
4817  023716  005202                  16$:    INC     R2              ;INCR RAM ADRS
4818  023720  020127  026235                CMP     R1,#ENDCOD      ;SEE IF ALL CODE CHECKED YET
4819  023724  103741                        BLO     8$              ;BR IF NOT
4820
4821                                  ;SET UP VIA AND USYRT FOR OPERATION
4822  023726  004537  003712                JSR     R5,WRITEI       ;RESET THE USYRT
4823  023732  120000                        VIAORB
4824  023734  000031                        RTSND!DTR!PRESET
4825  023736  004537  003712                JSR     R5,WRITEI       ;CLEAR USYRT RESET BIT
4826  023742  120000                        VIAORB
4827  023744  000030                        RTSND!DTR
4828  023746  004537  003712                JSR     R5,WRITEI       ;SET SYNCH CHAR = 226
4829  023752  120404                        PCSARL
4830  023754  000226                        226
4831  023756  012737  065400  023.          MOV     #DDCMP!STRIPS!IDLES!CRC16,18$   ;SET DDCMP,STRIP,IDLE, CRC16
4832  023764  000337  023776                SWAB    18$             ;GET DATA INTO LO BYTE
4833  023770  004537  003712                JSR     R5,WRITEI       ;PROGRAM THE PCSARH
4834  023774  120405                        PCSARH
4835  023776  000000                  18$:    .WORD   0
4836  024000  004537  005410                JSR     R5,CKUSTS       ;CHK USYRT STATUS FOR INITIALIZED STATE
4837  024004  000110                        110                     ;  TBMT=1, TSO=1
4838  024006  103003                        BCC     .+8.            ;IF ERROR, PRINT REPORT
4839  024010                                  ERROR
4840  024010  104460                                                          TRAP    C$ERROR
4841  024012                                  ESCAPE  TST
4842  024012  104410                                                          TRAP    C$ESCAPE
4843  024014  001372                                                          .WORD   L10021-.
4844  024016  004537  003712                JSR     R5,WRITEI       ;SET TSOM IN USYRT
4845  024022  120403                        TDSRH
4846  024024  000001                        TSOM
4847  024026  004537  003712                JSR     R5,WRITEI       ;LOAD 237 CHAR FOR INTGRL MODEM SYNCHRONIZATION
4848  024032  120402                        TDSRL
4849  024034  000237                        237
4850  024036  004537  006014                JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
4851  024042  000000                        0
4852  024044  103003                        BCC     .+8.            ;IF ERROR, REPORT ERROR
4853  024046                                  ERROR
4854  024046  104460                                                          TRAP    C$ERROR
4855  024050                                  ESCAPE  TST
4856  024050  104410                                                          TRAP    C$ESCAPE
4857  024052  001334                                                          .WORD   L10021-.
4858
4859                                  ;INITIATE 6502 TEST OF INTEGRAL MODEM
4860  024054  005077  156346                CLR     @SEL4           ;CLEAR SEL4
4861  024060  012777  001000  156340        MOV     #RAMADR,@SEL4   ;SET START ADRS OF RAM CODE IN SEL4
4862  024066  112777  000005  156326        MOVB    #EXECUT,@#SEL2  ;ISSUE M-LOOP CMND TO EXECUTE AT PC IN SEL4
4863                                  ;WAIT SEVERAL MILLI-SEC FOR COMPLETION OF TEST
```

```
 4864  024074  012701  023420                 MOV     #10000.,R1      ;INIT WAIT LOOP COUNTER
 4865  024100  005301                 22$:    DEC     R1              ;DECREMENT COUNTER
 4866  024102  000240                         NOP
 4867  024104  001375                         BNE     22$             ;BR IF NOT DONE COUNTING YET
 4868  024106  132777  000200  156306         BITB    #MRDY,@BSEL2    ;SEE IF M-LOOP FINISHED PROPERLY
 4869  024114  001013                         BNE     24$             ;BR IF YES
 4870  024116  004737  004166                 JSR     PC,GETWSR       ;GET CSR'S FOR PRINTOUT
 4871  024122  012737  000005  002324         MOV     #EXECUT,GDATA   ;IDENTIFY REQUESTED FUNCTION
 4872                                 ;REPORT MRDY TIMEOUT ERROR
 4873  024130                                 GEDF    EM4,ERR4
 4874                                                         ;       "DEVICE FATAL" ERROR # 39
 4875  024130  104455                                                        TRAP    C$ERDF
 4876  024132  000047                                                        .WORD   39
 4877  024134  014307                                                        .WORD   EM4
 4878  024136  020120                                                        .WORD   ERR4
 4879  024140                                 ESCAPE  TST
 4880  024140  104410                                                        TRAP    C$ESCAPE
 4881  024142  001244                                                        .WORD   L10021-.
 4882
 4883  024144  105777  156262         24$:    TSTB    @BSEL6          ;SEE IF ANY ERRORS OCCURRED IN TEST
 4884  024150  001002                         BNE     26$             ;BR IF ERROR DETECTED
 4885  024152  000137  025406                 JMP     90$             ;SUCCESSFUL COMPLETION OF TEST
 4886  024156  012737  000007  002336 26$:    MOV     #7,REGNUM       ;SET REG NO. FOR PRINTOUT
 4887  024164  005037  002324                 CLR     GDATA           ;CLEAR EXPECTED DATA AREA
 4888  024170  005037  002326                 CLR     @DATA           ;CLEAR ACTUAL DATA AREA
 4889                                 ;CHK FOR ERROR 1
 4890  024174  127727  156232  000001         CMPB    @BSEL6,#1       ;CHK FOR ERROR 1
 4891  024202  001006                         BNE     28$             ;BR IF NOT
 4892                                 ;REPORT TBMT NOT SET
 4893  024204                                 GEDF    EM73,ERR12
 4894                                                         ;       "DEVICE FATAL" ERROR # 40
 4895  024204  104455                                                        TRAP    C$ERDF
 4896  024206  000050                                                        .WORD   40
 4897  024210  014777                                                        .WORD   EM73
 4898  024212  020714                                                        .WORD   ERR12
 4899  024214                                 ESCAPE  TST
 4900  024214  104410                                                        TRAP    C$ESCAPE
 4901  024216  001170                                                        .WORD   L10021-.
 4902                                 ;CHK FOR ERROR 2
 4903  024220  127727  156206  000002 28$:    CMPB    @BSEL6,#2       ;CHK FOR ERROR 2
 4904  024226  001006                         BNE     30$             ;BR IF NOT
 4905                                 ;REPORT TBMT NOT SET
 4906  024230                                 GEDF    EM73,ERR12
 4907                                                         ;       "DEVICE FATAL" ERROR # 41
 4908  024230  104455                                                        TRAP    C$ERDF
 4909  024232  000051                                                        .WORD   41
 4910  024234  014777                                                        .WORD   EM73
 4911  024236  020714                                                        .WORD   ERR12
 4912  024240                                 ESCAPE  TST
 4913  024240  104410                                                        TRAP    C$ESCAPE
 4914  024242  001144                                                        .WORD   L10021-.
 4915                                 ;CHK FOR ERROR 3
 4916  024244  127727  156162  000003 30$:    CMPB    @BSEL6,#3       ;CHK FOR ERROR 3
 4917  024252  001006                         BNE     31$             ;BR IF NOT
 4918                                 ;REPORT TBMT NOT SET
 4919  024254                                 GEDF    EM73,ERR12
```

```
                                                        ;       "DEVICE FATAL" ERROR # 42
024254  104455                                                          TRAP    C$ERDF
024256  000052                                                          .WORD   42
024260  014777                                                          .WORD   EM73
024262  020714                                                          .WORD   ERR12
024264                                  ESCAPE  TST
024264  104410                                                          TRAP    C$ESCAPE
024266  001120                                                          .WORD   L10021-.
                                ;CHK FOR ERROR 4
024270  127727  156136  000004  31$:    CMPB    @@SEL6,#4       ;CHK FOR ERROR 4
024276  001011                        BNE     32$             ;BR IF NOT
                                ;REPORT CARRIER NOT SET
024300  012737  000001  002336        MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
024306                                  GEDF    EM80,ERR14
                                                        ;       "DEVICE FATAL" ERROR # 43
024306  104455                                                          TRAP    C$ERDF
024310  000053                                                          .WORD   43
024312  015161                                                          .WORD   EM80
024314  021146                                                          .WORD   ERR14
024316                                  ESCAPE  TST
024316  104410                                                          TRAP    C$ESCAPE
024320  001066                                                          .WORD   L10021-.
                                ;CHK FOR ERROR 5
024322  127727  156104  000005  32$:    CMPB    @@SEL6,#5       ;CHK FOR ERROR 5
024330  001006                        BNE     34$             ;BR IF NOT
                                ;REPORT TBMT NOT SET
024332                                  GEDF    EM73,ERR12
                                                        ;       "DEVICE FATAL" ERROR # 44
024332  104455                                                          TRAP    C$ERDF
024334  000054                                                          .WORD   44
024336  014777                                                          .WORD   EM73
024340  020714                                                          .WORD   ERR12
024342                                  ESCAPE  TST
024342  104410                                                          TRAP    C$ESCAPE
024344  001042                                                          .WORD   L10021-.
                                ;CHK FOR ERROR 6
024346  127727  156060  000006  34$:    CMPB    @@SEL6,#6       ;CHK FOR ERROR 6
024354  001006                        BNE     36$             ;BR IF NOT
                                ;REPORT TBMT NOT SET
024356                                  GEDF    EM73,ERR12
                                                        ;       "DEVICE FATAL" ERROR # 45
024356  104455                                                          TRAP    C$ERDF
024360  000055                                                          .WORD   45
024362  014777                                                          .WORD   EM73
024364  020714                                                          .WORD   ERR12
024366                                  ESCAPE  TST
024366  104410                                                          TRAP    C$ESCAPE
024370  001016                                                          .WORD   L10021-.
                                ;CHK FOR ERROR 7
024372  127727  156034  000007  36$:    CMPB    @@SEL6,#7       ;CHK FOR ERROR 7
024400  001006                        BNE     38$             ;BR IF NOT
                                ;REPORT TBMT NOT SET
024402                                  GEDF    EM73,ERR12
                                                        ;       "DEVICE FATAL" ERROR # 46
024402  104455                                                          TRAP    C$ERDF
024404  000056                                                          .WORD   46
```

```
 4976  024406  014777                                                          .WORD   EM73
 4977  024410  020714                                                          .WORD   ERR12
 4978  024412                          ESCAPE  TST
 4979  024412  104410                                                          TRAP    C$ESCAPE
 4980  024414  000772                                                          .WORD   L10021-.
 4981                          ;CHK FOR ERROR 8
 4982  024416  127727  156010  000010  38$:    CMPB    @@SEL6,#8.      ;CHK FOR ERROR 8
 4983  024424  001006                  BNE     40$             ;BR IF NOT
 4984                          ;REPORT TBMT NOT SET
 4985  024426                          GEDF    EM73,ERR12
 4986                                                          ;       "DEVICE FATAL" ERROR # 47
 4987  024426  104455                                                          TRAP    C$ERDF
 4988  024430  000057                                                          .WORD   47
 4989  024432  014777                                                          .WORD   EM73
 4990  024434  020714                                                          .WORD   ERR12
 4991  024436                          ESCAPE  TST
 4992  024436  104410                                                          TRAP    C$ESCAPE
 4993  024440  000746                                                          .WORD   L10021-.
 4994                          ;CHK FOR ERROR 9
 4995  024442  127727  155764  000011  40$:    CMPB    @@SEL6,#9.      ;CHK FOR ERROR 9
 4996  024450  001006                  BNE     42$             ;BR IF NOT
 4997                          ;REPORT RDA NOT SET
 4998  024452                          GEDF    EM75,ERR12
 4999                                                          ;       "DEVICE FATAL" ERROR # 48
 5000  024452  104455                                                          TRAP    C$ERDF
 5001  024454  000060                                                          .WORD   48
 5002  024456  015035                                                          .WORD   EM75
 5003  024460  020714                                                          .WORD   ERR12
 5004  024462                          ESCAPE  TST
 5005  024462  104410                                                          TRAP    C$ESCAPE
 5006  024464  000722                                                          .WORD   L10021-.
 5007                          ;CHK FOR ERROR 10
 5008  024466  127727  155740  000012  42$:    CMPB    @@SEL6,#10.     ;CHK FOR ERROR 10
 5009  024474  001017                  BNE     44$             ;BR IF NOT
 5010                          ;REPORT RCV'D DATA MISCOMPARE ERROR
 5011  024476  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
 5012  024504  112737  000000  002324          MOVB    #000,GDATA      ;SET EXPECTED DATA
 5013  024512  117737  155716  002326          MOVB    @@SEL7,BDATA    ;SET ACTUAL DATA
 5014  024520                          GEDF    EM34,ERR10
 5015                                                          ;       "DEVICE FATAL" ERROR # 49
 5016  024520  104455                                                          TRAP    C$ERDF
 5017  024522  000061                                                          .WORD   49
 5018  024524  014502                                                          .WORD   EM34
 5019  024526  020364                                                          .WORD   ERR10
 5020  024530                          ESCAPE  TST
 5021  024530  104410                                                          TRAP    C$ESCAPE
 5022  024532  000654                                                          .WORD   L10021-.
 5023                          ;CHK FOR ERROR 11
 5024  024534  127727  155672  000013  44$:    CMPB    @@SEL6,#11.     ;CHK FOR ERROR 11
 5025  024542  001006                  BNE     46$             ;BR IF NOT
 5026                          ;REPORT RDA NOT SET
 5027  024544                          GEDF    EM75,ERR12
 5028                                                          ;       "DEVICE FATAL" ERROR # 50
 5029  024544  104455                                                          TRAP    C$ERDF
 5030  024546  000062                                                          .WORD   50
 5031  024550  015035                                                          .WORD   EM75
```

```
5032  024552  020714                                                          .WORD   ERR12
5033  024554                          ESCAPE  TST
5034  024554  104410                                                          TRAP    C$ESCAPE
5035  024556  000630                                                          .WORD   L10021-.
5036                          ;CHK FOR ERROR 12
5037  024560  127727  155646  000014  46$:    CMPB    @#SEL6,#12.     ;CHK FOR ERROR 12
5038  024566  001017                  BNE     48$             ;BR IF NOT
5039                          ;REPORT RCV'D DATA MISCOMPARE ERROR
5040  024570  012737  000000  00233(          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5041  024576  112737  000125  00232+          MOVB    #125,GDATA      ;SET EXPECTED DATA
5042  024604  117737  155624  002326          MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
5043  024612                  GEDF    EM34,ERR10
5044                                          ;       "DEVICE FATAL" ERROR # 51
5045  024612  104455                                                          TRAP    C$ERDF
5046  024614  000063                                                          .WORD   51
5047  024616  014502                                                          .WORD   EM34
5048  024620  020364                                                          .WORD   ERR10
5049  024622                          ESCAPE  TST
5050  024622  104410                                                          TRAP    C$ESCAPE
5051  024624  000562                                                          .WORD   L10021-.
5052                          ;CHK FOR ERROR 13
5053  024626  127727  155600  000015  48$:    CMPB    @#SEL6,#13.     ;CHK FOR ERROR 13
5054  024634  001006                  BNE     50$             ;BR IF NOT
5055                          ;REPORT RDA NOT SET
5056  024636                  GEDF    EM75,ERR12
5057                                          ;       "DEVICE FATAL" ERROR # 52
5058  024636  104455                                                          TRAP    C$ERDF
5059  024640  000064                                                          .WORD   52
5060  024642  015035                                                          .WORD   EM75
5061  024644  020714                                                          .WORD   ERR12
5062  024646                          ESCAPE  TST
5063  024646  104410                                                          TRAP    C$ESCAPE
5064  024650  000536                                                          .WORD   L10021-.
5065                          ;CHK FOR ERROR 14
5066  024652  127727  155554  000016  50$:    CMPB    @#SEL6,#14.     ;CHK FOR ERROR 14
5067  024660  001017                  BNE     52$             ;BR IF NOT
5068                          ;REPORT RCV'D DATA MISCOMPARE ERROR
5069  024662  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5070  024670  112737  000252  002324          MOVB    #252,GDATA      ;SET EXPECTED DATA
5071  024676  117737  155532  002326          MOVB    @#SEL7,BDATA    ;SET ACTUAL DATA
5072  024704                  GEDF    EM34,ERR10
5073                                          ;       "DEVICE FATAL" ERROR # 53
5074  024704  104455                                                          TRAP    C$ERDF
5075  024706  000065                                                          .WORD   53
5076  024710  014502                                                          .WORD   EM34
5077  024712  020364                                                          .WORD   ERR10
5078  024714                          ESCAPE  TST
5079  024714  104410                                                          TRAP    C$ESCAPE
5080  024716  000470                                                          .WORD   L10021-.
5081                          ;CHK FOR ERROR 15
5082  024720  127727  155506  000017  52$:    CMPB    @#SEL6,#15.     ;CHK FOR ERROR 15
5083  024726  001006                  BNE     54$             ;BR IF NOT
5084                          ;REPORT RDA NOT SET
5085  024730                  GEDF    EM75,ERR12
5086                                          ;       "DEVICE FATAL" ERROR # 54
5087  024730  104455                                                          TRAP    C$ERDF
```

```
5088  024732  000066                                                          .WORD   54
5089  024734  015035                                                          .WORD   EM75
5090  024736  020714                                                          .WORD   ERR12
5091  024740                          ESCAPE  TST
5092  024740  104410                                                          TRAP    C$ESCAPE
5093  024742  000444                                                          .WORD   L10021-.
5094                                ;CHK FOR ERROR 16
5095  024744  127727  155462  000020 54$:    CMPB    @BSEL6,#16.     ;CHK FOR ERROR 16
5096  024752  001017                        BNE     56$             ;BR IF NOT
5097                                ;REPORT RCV'D DATA MISCOMPARE ERROR
5098  024754  012737  000000  002336        MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5099  024762  112737  000377  002324        MOVB    #377,GDATA      ;SET EXPECTED DATA
5100  024770  117737  155440  002326        MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5101  024776                                GEDF    EM34,ERR10
5102                                                  ;       "DEVICE FATAL" ERROR # 55
5103  024776  104455                                                          TRAP    C$ERDF
5104  025000  000067                                                          .WORD   55
5105  025002  014502                                                          .WORD   EM34
5106  025004  020364                                                          .WORD   ERR10
5107  025006                          ESCAPE  TST
5108  025006  104410                                                          TRAP    C$ESCAPE
5109  025010  000376                                                          .WORD   L10021-.
5110                                ;CHK FOR ERROR 17
5111  025012  127727  155414  000021 56$:    CMPB    @BSEL6,#17.     ;CHK FOR ERROR 17
5112  025020  001006                        BNE     58$             ;BR IF NOT
5113                                ;REPORT RDA NOT SET
5114  025022                                GEDF    EM75,ERR12
5115                                                  ;       "DEVICE FATAL" ERROR # 56
5116  025022  104455                                                          TRAP    C$ERDF
5117  025024  000070                                                          .WORD   56
5118  025026  015035                                                          .WORD   EM75
5119  025030  020714                                                          .WORD   ERR12
5120  025032                          ESCAPE  TST
5121  025032  104410                                                          TRAP    C$ESCAPE
5122  025034  000352                                                          .WORD   L10021-.
5123                                ;CHK FOR ERROR 18
5124  025036  127727  155370  000022 58$:    CMPB    @BSEL6,#18.     ;CHK FOR ERROR 18
5125  025044  001017                        BNE     60$             ;BR IF NOT
5126                                ;REPORT RCV'D DATA MISCOMPARE ERROR
5127  025046  012737  000000  002336        MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5128  025054  112737  000000  002324        MOVB    #000,GDATA      ;SET EXPECTED DATA
5129  025062  117737  155346  002326        MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5130  025070                                GEDF    EM34,ERR10
5131                                                  ;       "DEVICE FATAL" ERROR # 57
5132  025070  104455                                                          TRAP    C$ERDF
5133  025072  000071                                                          .WORD   57
5134  025074  014502                                                          .WORD   EM34
5135  025076  020364                                                          .WORD   ERR10
5136  025100                          ESCAPE  TST
5137  025100  104410                                                          TRAP    C$ESCAPE
5138  025102  000304                                                          .WORD   L10021-.
5139                                ;CHK FOR ERROR 19
5140  025104  127727  155322  000023 60$:    CMPB    @BSEL6,#19.     ;CHK FOR EPROR 19
5141  025112  001006                        BNE     62$             ;BR IF NOT
5142                                ;REPORT RERR NOT SET
5143  025114                                GEDF    EM36,ERR10
```

```
5144                                                    ;       "DEVICE FATAL" ERROR # 58
5145  025114  104455                                                  TRAP    C$ERDF
5146  025116  000072                                                  .WORD   58
5147  025120  014551                                                  .WORD   EM36
5148  025122  020364                                                  .WORD   ERR10
5149  025124                          ESCAPE  TST
5150  025124  104410                                                  TRAP    C$ESCAPE
5151  025126  000260                                                  .WORD   L10021-.
5152                            ;CHK FOR ERROR 20
5153  025130  127727  155276  000024  62$:    CMPB    @BSEL6,#20.     ;CHK FOR ERROR 20
5154  025136  001006                  BNE     64$             ;BR IF NOT
5155                            ;REPORT RDA NOT SET
5156  025140                          GEDF    EM75,ERR12
5157                                                    ;       "DEVICE FATAL" ERROR # 59
5158  025140  104455                                                  TRAP    C$ERDF
5159  025142  000073                                                  .WORD   59
5160  025144  015035                                                  .WORD   EM75
5161  025146  020714                                                  .WORD   ERR12
5162  025150                          ESCAPE  TST
5163  025150  104410                                                  TRAP    C$ESCAPE
5164  025152  000234                                                  .WORD   L10021-.
5165                            ;CHK FOR ERROR 21
5166  025154  127727  155252  000025  64$:    CMPB    @BSEL6,#21.     ;CHK FOR ERROR 21
5167  025162  001017                  BNE     66$             ;BR IF NOT
5168                            ;REPORT RCV'D DATA MISCOMPARE ERROR
5169  025164  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5170  025172  112737  000160  002324          MOVB    #160,GDATA      ;SET EXPECTED DATA
5171  025200  117737  155230  002326          MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
5172  025206                          GEDF    EM34,ERR10
5173                                                    ;       "DEVICE FATAL" ERROR # 60
5174  025206  104455                                                  TRAP    C$ERDF
5175  025210  000074                                                  .WORD   60
5176  025212  014502                                                  .WORD   EM34
5177  025214  020364                                                  .WORD   ERR10
5178  025216                          ESCAPE  TST
5179  025216  104410                                                  TRAP    C$ESCAPE
5180  025220  000166                                                  .WORD   L10021-.
5181                            ;CHK FOR ERROR 22
5182  025222  127727  155204  000026  66$:    CMPB    @BSEL6,#22.     ;CHK FOR ERROR 22
5183  025230  001006                  BNE     68$             ;BR IF NOT
5184                            ;REPORT RDA NOT SET
5185  025232                          GEDF    EM75,ERR12
5186                                                    ;       "DEVICE FATAL" ERROR # 61
5187  025232  104455                                                  TRAP    C$ERDF
5188  025234  000075                                                  .WORD   61
5189  025236  015035                                                  .WORD   EM75
5190  025240  020714                                                  .WORD   ERR12
5191  025242                          ESCAPE  TST
5192  025242  104410                                                  TRAP    C$ESCAPE
5193  025244  000142                                                  .WORD   L10021-.
5194                            ;CHK FOR ERROR 23
5195  025246  127727  155160  000027  68$:    CMPB    @BSEL6,#23.     ;CHK FOR ERROR 23
5196  025254  001017                  BNE     70$             ;BR IF NOT
5197                            ;REPORT RCV'D DATA MISCOMPARE ERROR
5198  025256  012737  000000  002336          MOV     #0,REGNUM       ;SET REG NO. FOR PRINTOUT
5199  025264  112737  000034  002324          MOVB    #034,GDATA      ;SET EXPECTED DATA
```

```
 5200  025272  117737  155136  002326          MOVB    @BSEL7,BDATA    ;SET ACTUAL DATA
 5201  025300                                  GEDF    EM34,ERR10
 5202                                                  ;       "DEVICE FATAL" ERROR # 62
 5203  025300  104455                                                         TRAP    C$ERDF
 5204  025302  000076                                                         .WORD   62
 5205  025304  014502                                                         .WORD   EM34
 5206  025306  020364                                                         .WORD   ERR10
 5207  025310                                  ESCAPE  TST
 5208  025310  104410                                                         TRAP    C$ESCAPE
 5209  025312  000074                                                         .WORD   L10021-.
 5210                                  ;CHK FOR ERROR 24
 5211  025314  127727  155112  000030  70$:    CMPB    @BSEL6,#24.     ;CHK FOR ERROR 24
 5212  025322  001006                          BNE     72$             ;BR IF NOT
 5213                                  ;REPORT ROA NOT SET
 5214  025324                                  GEDF    EM75,ERR12
 5215                                                  ;       "DEVICE FATAL" ERROR # 63
 5216  025324  104455                                                         TRAP    C$ERDF
 5217  025326  000077                                                         .WORD   63
 5218  025330  015035                                                         .WORD   EM75
 5219  025332  020714                                                         .WORD   ERR12
 5220  025334                                  ESCAPE  TST
 5221  025334  104410                                                         TRAP    C$ESCAPE
 5222  025336  000050                                                         .WORD   L10021-.
 5223                                  ;CHK FOR ERROR 25
 5224  025340  127727  155066  000031  72$:    CMPB    @BSEL6,#25.     ;CHK FOR ERROR 25
 5225  025346  001011                          BNE     74$             ;BR IF NOT
 5226                                  ;REPORT CARRIER NOT CLEARED
 5227  025350  012737  000001  002336          MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
 5228  025356                                  GEDF    EM81,ERR14
 5229                                                  ;       "DEVICE FATAL" ERROR # 64
 5230  025356  104455                                                         TRAP    C$ERDF
 5231  025360  000100                                                         .WORD   64
 5232  025362  015201                                                         .WORD   EM81
 5233  025364  021146                                                         .WORD   ERR14
 5234  025366                                  ESCAPE  TST
 5235  025366  104410                                                         TRAP    C$ESCAPE
 5236  025370  000016                                                         .WORD   L10021-.
 5237  025372  004737  004166          74$:    JSR     PC,GETWSR       ;GET CSR'S FOR PRINTOUT
 5238                                  ;REPORT INVALID ERROR CODE FROM 6502
 5239  025376                                  GEDF    EM82,ERR3
 5240                                                  ;       "DEVICE FATAL" ERROR # 65
 5241  025376  104455                                                         TRAP    C$ERDF
 5242  025400  000101                                                         .WORD   65
 5243  025402  015225                                                         .WORD   EM82
 5244  025404  020072                                                         .WORD   ERR3
 5245
 5246  025406                          90$:
 5247  025406                          ENDTST
 5248  025406                                                          L10021:
 5249  025406  104401                                                         TRAP    C$ETST
```

```
 5250  025410                          MCODE:
 5251                                  ;
 5252                                  ;LINE# LOC   CODE        LINE
 5253                                  ;
 5254                                  ;0001  0000                    *=^O200                  ;START OF MICROCODE FOR INTEGRAL
 5255                                  ;0002  0200
 5256                                  ;0003  C200
 5257                                  ;0005  0200
 5258                                  ;0006  0200  ;*****************************************************************************
 5259                                  ;0007  0200  ; THIS IS THE 6502 MICROCODE WHICH IS LOADED INTO RAM AND EXECUTED FOR THE
 5260                                  ;0008  0200  ; PURPOSE OF TESTING THE INTEGRAL MODEM ON THE M8064, AT 56K BAUD. AFTER TH
 5261                                  ;0009  0200  ; LSI-11 PROGRAM DOES SOME INITIAL SETUP, IT TRANSFERS CONTROL TO THIS CODE
 5262                                  ;0010  0200  ; IN RAM, AND WAITS FOR COMPLETION OF THE TEST, AS INDICATED BY MRDY SET.
 5263                                  ;0011  0200  ; THIS CODE TRANSMITS, RECEIVES, AND CHECKS THE FOLLOWING CHARACTERS :
 5264                                  ;0012  0200  ; 2 SYNCH CHARACTERS, 5 DATA CHARACTERS 000, 125, 252, 377, 000, 2 CRC-16
 5265                                  ;0013  0200  ; CHARACTERS 160 AND 034, AND 2 TERMINATING SYNCHS. THE MESSAGE IS SENT USI
 5266                                  ;0014  0200  ; CHARACTER (DDCMP) MODE, THE SYNCH CHARACTER USED IS 226, STRIP SYNCH AND
 5267                                  ;0015  0200  ; IDLE MODES ARE SET, AND THE DATA CLOCK IS PROVIDED BY THE INTEGRAL MODEM.
 5268                                  ;0016  0200  ; ALL DATA AND CRC CHARACTERS ARE CHECKED AS THEY ARE RECEIVED, AND THE CRC
 5269                                  ;0017  0200  ; ERROR CHECK BIT IS CHECKED TO BE SET WITH RECEPTION OF THE LAST DATA
 5270                                  ;0018  0200  ; CHARACTER (000).
 5271                                  ;0019  0200  ;*****************************************************************************
 5272                                  ;0020  0200
 5273                                  ;0021  0200            ;EQUATES FOR BIT DEFINITIONS
 5274                                  ;0022  0200          BIT0    =^B1
 5275                                  ;0023  0200          BIT1    =^B2
 5276                                  ;0024  0200          BIT2    =^B4
 5277                                  ;0025  0200          BIT3    =^B10
 5278                                  ;0026  0200          BIT4    =^B20
 5279                                  ;0027  0200          BIT5    =^B40
 5280                                  ;0028  0200          BIT6    =^B100
 5281                                  ;0029  0200          BIT7    =^B200
 5282                                  ;0030  0200          BIT8    =^B400
 5283                                  ;0031  0200          BIT9    =^B1000
 5284                                  ;0032  0200          BIT10   =^B2000
 5285                                  ;0033  0200          BIT11   =^B4000
 5286                                  ;0034  0200          BIT12   =^B10000
 5287                                  ;0035  0200          BIT13   =^B20000
 5288                                  ;0036  0200          BIT14   =^B40000
 5289                                  ;0037  0200          BIT15   =^B100000
 5290                                  ;0038  0200
 5291                                  ;0039  0200
 5292                                  ;0040  0200
 5293                                  ;0041  0200            ;ADDRESS EQUATES FOR CSR REGISTERS
 5294                                  ;0042  0200          SEL0    =^O10
 5295                                  ;0043  0200          BSEL0   =SEL0
 5296                                  ;0044  0200          BSEL1   =SEL0+1
 5297                                  ;0045  0200          SEL2    =SEL0+2
 5298                                  ;0046  0200          BSEL2   =SEL0+2
 5299                                  ;0047  0200          BSEL3   =SEL0+3
 5300                                  ;0048  0200          SEL4    =SEL0+4
 5301                                  ;0049  0200          BSEL4   =SEL0+4
 5302                                  ;0050  0200          BSEL5   =SEL0+5
 5303                                  ;0051  0200          SEL6    =SEL0+6
 5304                                  ;0052  0200          BSEL6   =SEL0+6
 5305                                  ;0053  0200          BSEL7   =SEL0+7
```

```
5306                                    ;0054  0200
5307                                    ;0055  0200
5308                                    ;0056  0200
5309                                    ;0057  0200          ;VERSATILE INTERFACE ADAPTER REGISTER EQUATES
5310                                    ;0058  0200     OREGB  = $A000                    ;OUTPUT REGISTER B
5311                                    ;0059  0200     OREGA  = OREGB+1                  ;OUTPUT REGISTER A
5312                                    ;0060  0200     DDRB   = OREGB+2                  ;DATA DIRECTION REGISTER B
5313                                    ;0061  0200     DDRA   = OREGB+3                  ;DATA DIRECTION REGISTER A
5314                                    ;0062  0200     T1LL   = OREGB+6                  ;TIMER 1 LATCH LOW BITS
5315                                    ;0063  0200     T1LH   = OREGB+7                  ;TIMER 1 LATCH HIGH BITS
5316                                    ;0064  0200     ACR    = OREGB+$B                 ;AUXILIARY CONTROL REGISTER
5317                                    ;0065  0200     PCR    = OREGB+$C                 ;PERIPHERAL CONTROL REGISTER
5318                                    ;0066  0200
5319                                    ;0067  0200
5320                                    ;0068  0200
5321                                    ;0069  0200          ;VIA OUTPUT REGISTER B BIT EQUATES
5322                                    ;0070  0200     NULCLK = BIT7
5323                                    ;0071  0200     RXEN   = BIT6
5324                                    ;0072  0200     TXEN   = BIT5
5325                                    ;0073  0200     DTR    = BIT4
5326                                    ;0074  0200     RTSND  = BIT3
5327                                    ;0075  0200     HDX    = BIT2
5328                                    ;0076  0200     TTLOOP = BIT1
5329                                    ;0077  0200     PRESET = BIT0
5330                                    ;0078  0200
5331                                    ;0079  0200
5332                                    ;0080  0200
5333                                    ;0081  0200          ;VIA OUTPUT REGISTER A BIT EQUATES
5334                                    ;0082  0200     RING   = BIT7
5335                                    ;0083  0200     CARIER = BIT6
5336                                    ;0084  0200     MDMRDY = BIT5
5337                                    ;0085  0200     BDRATE = BIT4
5338                                    ;0086  0200     CTS    = BIT3
5339                                    ;0087  0200     TM     = BIT2
5340                                    ;0088  0200     RCVDAT = BIT1
5341                                    ;0089  0200     UMAINT = BIT0
5342                                    ;0090  0200
5343                                    ;0091  0200
5344                                    ;0092  0200          ;USYRT REGISTER ADDRESS EQUATES
5345                                    ;0093  0200     RXDB   = $A100
5346                                    ;0094  0200     RDSR   = RXDB+1
5347                                    ;0095  0200     TXDB   = RXDB+2
5348                                    ;0096  0200     TDSR   = RXDB+3
5349                                    ;0097  0200     SAR    = RXDB+4
5350                                    ;0098  0200     PCSAR  = RXDB+5
5351                                    ;0099  0200     PCTLR  = RXDB+7
5352                                    ;0100  0200
5353                                    ;0101  0200
5354                                    ;0102  0200
5355                                    ;0103  0200          ;USYRT TDSR REGISTER BIT EQUATES
5356                                    ;0104  0200     TEOM   = BIT1
5357                                    ;0105  0200     TSOM   = BIT0
5358                                    ;0106  0200
5359                                    ;0107  0200
5360                                    ;0108  0200
5361                                    ;0109  0200          ;USYRT RDSR BIT EQUATES
```

```
5362                                    ;0110  0200              RERR   = BIT7
5363                                    ;0111  0200
5364                                    ;0112  0200
5365                                    ;0113  0200
5366                                    ;0114  0200               ;USYRT STATUS REGISTER EQUATES
5367                                    ;0115  0200              USTATR = #A400
5368                                    ;0116  0200              RDA    = BIT7
5369                                    ;0117  0200              TBMT   = BIT6
5370                                    ;0118  0200              RXACT  = BIT5
5371                                    ;0119  0200              RSA    = BIT4
5372                                    ;0120  0200              TSO    = BIT3
5373                                    ;0121  0200              TXACT  = BIT2
5374                                    ;0122  020C              TXUERR = BIT1
5375                                    ;0123  0200              SYNFLG = BIT0
5376                                    ;0124  0200
5377                                    ;0125  0200
5378                                    ;0126  0200
5379                                    ;0127  0200               ;MISCELLANEOUS EQUATES
5380                                    ;0128  0200              SYNCH  = 8226
5381                                    ;0129  02??
5382                                    ;0130  0200
5383                                    ;0131  0200  A0 00              LDY     #0
5384  025410    240    000                      .BYTE   240,000
5385                                    ;0132  0202  84 16              STY     BSEL6           ;CLEAR BSEL6
5386  025412    204    026                      .BYTE   204,026
5387                                    ;0133  0204  84 17              STY     BSEL7           ;CLEAR BSEL7
5388  025414    204    027                      .BYTE   204,027
5389                                    ;0134  0206                ;TURN ON THE USYRT, CLOCK
5390                                    ;0135  0206  A2 60              LDX     #TXEN!RXEN      ;ASSERT TXEN,RXEN,RTS,DTR
5391  025416    242    140                      .BYTE   242,140
5392                                    ;0136  0208  8E 00 A0           STX     OREGB           ;  AND RELEASE INT MODEM RESET
5393  025420    216    000    240               .BYTE   216,000,240
5394                                    ;0137  0208  A2 00              LDX     #0              ;INIT TBMT TIME-OUT COUNTER
5395  025423    242    000                      .BYTE   242,000
5396                                    ;0138  0200  2C 00 A4           BIT     USTATR          ;SEE IF TBMT SET
5397  025425    054    000    244               .BYTE   054,000,244
5398                                    ;0139  0210  70 08              BVS     *+10            ;BR IF TBMT SET
5399  025430    160    010                      .BYTE   160,010
5400                                    ;0140  0212  E8                 INX                     ;INCREMENT TIME-OUT COUNTER
5401  025432    350                              .BYTE   350
5402                                    ;0141  0213  D0 F8              BNE     *-6             ;BR IF NO TIME-OUT
5403  025433    320    370                      .BYTE   320,370
5404                                    ;0142  0215                ; *** ERROR 1 ***
5405                                    ;0143  0215  A0 01              LDY     #1              ;SET CODE FOR TBMT TIME-OUT ERRO
5406  025435    240    001                      .BYTE   240,001
5407                                    ;0144  0217  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5408  025437    114    220    003               .BYTE   114,220,003
5409                                    ;0145  021A                ;LOAD FIRST SYNCH CHAR INTO TRANSMITTER
5410                                    ;0146  021A  A2 96              LDX     #SYNCH          ;LOAD FIRST SYNCH CHAR
5411  025442    242    226                      .BYTE   242,226
5412                                    ;0147  021C  8E 02 A1           STX     TXDB
5413  025444    216    002    241               .BYTE   216,002,241
5414                                    ;0148  021F  A2 00              LDX     #0              ;INIT TBMT TIME-OUT COUNTER
5415  025447    242    000                      .BYTE   242,000
5416                                    ;0149  0221  2C 00 A4           BIT     USTATR          ;SEE IF TBMT SET
5417  025451    054    000    244               .BYTE   054,000,244
```

```
5418                                    ;0150  0224  70 08          BVS     .+10            ;BR IF TBMT SET
5419  025454    160    010                      .BYTE   160,010
5420                                    ;0151  0226  E8              INX                     ;INCREMENT TIME-OUT COUNTER
5421  025456    350                      .BYTE   350
5422                                    ;0152  0227  D0 F8          BNE     .-6             ;BR IF NO TIME-OUT
5423  025457    320    370               .BYTE   320,370
5424                                    ;0153  0229                  ; *** ERROR 2 ***
5425                                    ;0154  0229  A0 02          LDY     #2              ;SET CODE FOR TBMT TIME-OUT ERRO
5426  025461    240    002               .BYTE   240,002
5427                                    ;0155  022B  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5428  025463    114    220    003        .BYTE   114,220,003
5429                                    ;0156  022E                  ;LOAD SECOND SYNCH CHAR INTO TRANSMITTER
5430                                    ;0157  022E  A2 96          LDX     #SYNCH          ;LOAD SECOND SYNCH CHAR
5431  025466    242    226               .BYTE   242,226
5432                                    ;0158  0230  8E 02 A1       STX     TXDB
5433  025470    216    002    241        .BYTE   216,002,241
5434                                    ;0159  0233  A2 00          LDX     #0              ;INIT TBMT TIME-OUT COUNTER
5435  025473    242    000               .BYTE   242,000
5436                                    ;0160  0235  2C 00 A4       BIT     USTATR          ;SEE IF TBMT SET
5437  025475    054    000    244        .BYTE   054,000,244
5438                                    ;0161  0238  70 08          BVS     .+10            ;BR IF TBMT SET
5439  025500    160    010               .BYTE   160,010
5440                                    ;0162  023A  E8              INX                     ;INCREMENT TIME-OUT COUNTER
5441  025502    350                      .BYTE   350
5442                                    ;0163  023B  D0 F8          BNE     .-6             ;BR IF NO TIME-OUT
5443  025503    320    370               .BYTE   320,370
5444                                    ;0164  023D                  ; *** ERROR 3 ***
5445                                    ;0165  023D  A0 03          LDY     #3              ;SET CODE FOR TBMT TIME-OUT ERRO
5446  025505    240    003               .BYTE   240,003
5447                                    ;0166  023F  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5448  025507    114    220    003        .BYTE   114,220,003
5449                                    ;0167  0242                  ;CHECK FOR CARRIER SET
5450                                    ;0168  0242  2C 01 A0       BIT     OREGA           ;SEE IF CARRIER SET YET
5451  025512    054    001    240        .BYTE   054,001,240
5452                                    ;0169  0245  70 05          BVS     .+7             ;BR IF CARRIER SET
5453  025515    160    005               .BYTE   160,005
5454                                    ;0170  0247                  ; *** ERROR 4 ***
5455                                    ;0171  0247  A0 04          LDY     #4              ;SET CODE FOR CARRIER NOT SET ER
5456  025517    240    004               .BYTE   240,004
5457                                    ;0172  0249  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5458  025521    114    220    003        .BYTE   114,220,003
5459                                    ;0173  024C                  ;LOAD TRANSMITTER WITH 000 CHAR
5460                                    ;0174  024C  A2 00          LDX     #000            ;CLEAR TSOM
5461  025524    242    000               .BYTE   242,000
5462                                    ;0175  024E  8E 03 A1       STX     TDSR
5463  025526    216    003    241        .BYTE   216,003,241
5464                                    ;0176  0251  8E 02 A1       STX     TXDB            ;LOAD 000 CHAR
5465  025531    216    002    241        .BYTE   216,002,241
5466                                    ;0177  0254  A2 00          LDX     #0              ;INIT TBMT TIME-OUT COUNTER
5467  025534    242    000               .BYTE   242,000
5468                                    ;0178  0256  2C 00 A4       BIT     USTATR          ;SEE IF TBMT SET
5469  025536    054    000    244        .BYTE   054,000,244
5470                                    ;0179  0259  70 08          BVS     .+10            ;BR IF TBMT SET
5471  025541    160    010               .BYTE   160,010
5472                                    ;0180  025B  E8              INX                     ;INCREMENT TIME-OUT COUNTER
5473  025543    350                      .BYTE   350
```

```
5474
5475  025544  320  370                ;0181  025C  D0 F8          BNE     *-6             ;BR IF NO TIME-OUT
5476                                          .BYTE   320,370
5477                                   ;0182  025E                ; *** ERROR 5 ***
5478  025546  240  005                ;0183  025E  A0 05          LDY     #5              ;SET CODE FOR TBMT TIME-OUT ERRO
5479                                          .BYTE   240,005
5480  025550  114  220  003           ;0184  0260  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5481                                          .BYTE   114,220,003
5482                                   ;0185  0263                ;LOAD TRANSMITTER WITH 125 CHAR
5483  025553  242  125                ;0186  0263  A2 55          LDX     #0125           ;LOAD 125 CHAR
5484                                          .BYTE   242,125
5485  025555  216  002  241           ;0187  0265  8E 02 A1       STX     TXDB
5486                                          .BYTE   216,002,241
5487  025560  242  000                ;0188  0268  A2 00          LDX     #0              ;INIT TBMT TIME-OUT COUNTER
5488                                          .BYTE   242,000
5489  025562  054  000  244           ;0189  026A  2C 00 A4       BIT     USTATR          ;SEE IF TBMT SET
5490                                          .BYTE   054,000,244
5491  025565  160  010                ;0190  026D  70 08          BVS     *+10            ;BR IF TBMT SET
5492                                          .BYTE   160,010
5493  025567  350                      ;0191  026F  E8             INX                     ;INCREMENT TIME-OUT COUNTER
5494                                          .BYTE   350
5495  025570  320  370                ;0192  0270  D0 F8          BNE     *-6             ;BR IF NO TIME-OUT
5496                                          .BYTE   320,370
5497                                   ;0193  0272                ; *** ERROR 6 ***
5498  025572  240  006                ;0194  0272  A0 06          LDY     #6              ;SET CODE FOR TBMT TIME-OUT ERRO
5499                                          .BYTE   240,006
5500  025574  114  220  003           ;0195  0274  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5501                                          .BYTE   114,220,003
5502                                   ;0196  0277                ;LOAD TRANSMITTER WITH 252 CHAR
5503  025577  242  252                ;0197  0277  A2 AA          LDX     #0252           ;LOAD 252 CHAR
5504                                          .BYTE   242,252
5505  025601  216  002  241           ;0198  0279  8E 02 A1       STX     TXDB
5506                                          .BYTE   216,002,241
5507  025604  242  000                ;0199  027C  A2 00          LDX     #0              ;INIT TBMT TIME-OUT COUNTER
5508                                          .BYTE   242,000
5509  025606  054  000  244           ;0200  027E  2C 00 A4       BIT     USTATR          ;SEE IF TBMT SET
5510                                          .BYTE   054,000,244
5511  025611  160  010                ;0201  0281  70 08          BVS     *+10            ;BR IF TBMT SET
5512                                          .BYTE   160,010
5513  025613  350                      ;0202  0283  E8             INX                     ;INCREMENT TIME-OUT COUNTER
5514                                          .BYTE   350
5515  025614  320  370                ;0203  0284  D0 F8          BNE     *-6             ;BR IF NO TIME-OUT
5516                                          .BYTE   320,370
5517                                   ;0204  0286                ; *** ERROR 7 ***
5518  025616  240  007                ;0205  0286  A0 07          LDY     #7              ;SET CODE FOR TBMT TIME-OUT ERRO
5519                                          .BYTE   240,007
5520  025620  114  220  003           ;0206  0288  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5521                                          .BYTE   114,220,003
5522                                   ;0207  0288                ;LOAD TRANSMITTER WITH 377 CHAR AND END OF MESSAGE
5523  025623  242  377                ;0208  0288  A2 FF          LDX     #0377           ;LOAD 377 CHAR
5524                                          .BYTE   242,377
5525  025625  216  002  241           ;0209  028D  8E 02 A1       STX     TXDB
5526                                          .BYTE   216,002,241
5527  025630  242  000                ;0210  0290  A2 00          LDX     #0              ;INIT TBMT TIME-OUT COUNTER
5528                                          .BYTE   242,000
5529  025632  054  000  244           ;0211  0292  2C 00 A4       BIT     USTATR          ;SEE IF TBMT SET
                                              .BYTE   054,000,244
```

```
5530                                    ;0212  0295  70 08          BVS     .+10            ;BR IF TBMT SET
5531  025635    160     010                     .BYTE   160,010
5532                                    ;0213  0297  E8              INX                     ;INCREMENT TIME-OUT COUNTER
5533  025637    350                             .BYTE   350
5534                                    ;0214  0298  D0 F8          BNE     .-6             ;BR IF NO TIME-OUT
5535  025640    320     370                     .BYTE   320,370
5536                                    ;0215  029A                 ; *** ERROR 8 ***
5537                                    ;0216  029A  A0 08          LDY     #8              ;SET CODE FOR TBMT TIME-OUT ERRO
5538  025642    240     010                     .BYTE   240,010
5539                                    ;0217  029C  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5540  025644    114     220     003             .BYTE   114,220,003
5541                                    ;0218  029F                 ;LOAD TRANSMITTER WITH 000 CHAR
5542                                    ;0219  029F  A2 00          LDX     #000            ;LOAD 000 CHAR
5543  025647    242     000                     .BYTE   242,000
5544                                    ;0220  02A1  8E 02 A1       STX     TXDB
5545  025651    216     002     241             .BYTE   216,002,241
5546                                    ;0221  02A4  A2 00          LDX     #0              ;INIT RDA TIME-OUT COUNTER
5547  025654    242     000                     .BYTE   242,000
5548                                    ;0222  02A6  2C 00 A4       BIT     USTATR          ;SEE IF RDA SET
5549  025656    054     000     244             .BYTE   054,000,244
5550                                    ;0223  02A9  30 08          BMI     .+10            ;BR IF RDA SET
5551  025661    060     010                     .BYTE   060,010
5552                                    ;0224  02AB  E8              INX                     ;INCREMENT TIME-OUT COUNTER
5553  025663    350                             .BYTE   350
5554                                    ;0225  02AC  D0 F8          BNE     .-6             ;BR IF NO TIME-OUT
5555  025664    320     370                     .BYTE   320,370
5556                                    ;0226  02AE                 ; *** ERROR 9 ***
5557                                    ;0227  02AE  A0 09          LDY     #9              ;SET CODE FOR RDA TIME-OUT ERROR
5558  025666    240     011                     .BYTE   240,011
5559                                    ;0228  02B0  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5560  025670    114     220     003             .BYTE   114,220,003
5561                                    ;0229  02B3                 ;READ AND CHECK 000 CHAR
5562                                    ;0230  02B3  AD 00 A1       LDA     RXDB            ;READ RECEIVER BUFFER
5563  025673    255     000     241             .BYTE   255,000,241
5564                                    ;0231  02B6  C9 00          CMP     #8000           ;CHK FOR 000
5565  025676    311     000                     .BYTE   311,000
5566                                    ;0232  02B8  F0 05          BEQ     .+7             ;BR IF 000
5567  025700    360     005                     .BYTE   360,005
5568                                    ;0233  02BA                 ; *** ERROR 10 ***
5569                                    ;0234  02BA  A0 0A          LDY     #10             ;SET CODE FOR DATA MISCOMPARE ER
5570  025702    240     012                     .BYTE   240,012
5571                                    ;0235  02BC  4C 90 03       JMP     A100            ;GO TAKE ERROR EXIT
5572  025704    114     220     003             .BYTE   114,220,003
5573                                    ;0236  02BF  A2 00          LDX     #0              ;INIT RDA TIME-OUT COUNTER
5574  025707    242     000                     .BYTE   242,000
5575                                    ;0237  02C1  2C 00 A4       BIT     USTATR          ;SEE IF RDA SET
5576  025711    054     000     244             .BYTE   054,000,244
5577                                    ;0238  02C4  30 08          BMI     .+10            ;BR IF RDA SET
5578  025714    060     010                     .BYTE   060,010
5579                                    ;0239  02C6  E8              INX                     ;INCREMENT TIME-OUT COUNTER
5580  025716    350                             .BYTE   350
5581                                    ;0240  02C7  D0 F8          BNE     .-6             ;BR IF NO TIME-OUT
5582  025717    320     370                     .BYTE   320,370
5583                                    ;0241  02C9                 ; *** ERROR 11 ***
5584                                    ;0242  02C9  A0 0B          LDY     #11             ;SET CODE FOR RDA TIME-OUT ERROR
5585  025721    240     013                     .BYTE   240,013
```

```
5586                                    ;0243  02CB  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5587  025723    114     220     003             .BYTE   114,220,003
5588                                    ;0244  02CE  A2 02             LDX     #TEOM           ;SET TEOM TO TERMINATE MSG
5589  025726    242     002                     .BYTE   242,002
5590                                    ;0245  02D0  8E 03 A1          STX     TDSR
5591  025730    216     003     241             .BYTE   216,003,241
5592                                    ;0246  02D3                    ;READ AND CHECK 125 CHAR
5593                                    ;0247  02D3  AD 00 A1          LDA     RXDB            ;READ RECEIVER BUFFER
5594  025733    255     000     241             .BYTE   255,000,241
5595                                    ;0248  02D6  C9 55             CMP     #0125           ;CHK FOR 125
5596  025736    311     125                     .BYTE   311,125
5597                                    ;0249  02D8  F0 05             BEQ     .+7             ;BR IF 125
5598  025740    360     005                     .BYTE   360,005
5599                                    ;0250  02DA                    ; *** ERROR 12 ***
5600                                    ;0251  02DA  A0 0C             LDY     #12             ;SET CODE FOR DATA MISCOMPARE ER
5601  025742    240     014                     .BYTE   240,014
5602                                    ;0252  02DC  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5603  025744    114     220     003             .BYTE   114,220,003
5604                                    ;0253  02DF  A2 00             LDX     #0              ;INIT RDA TIME-OUT COUNTER
5605  025747    242     000                     .BYTE   242,000
5606                                    ;0254  02E1  2C 00 A4          BIT     USTATR          ;SEE IF RDA SET
5607  025751    054     000     244             .BYTE   054,000,244
5608                                    ;0255  02E4  30 08             BMI     .+10            ;BR IF RDA SET
5609  025754    060     010                     .BYTE   060,010
5610                                    ;0256  02E6  E8                INX                     ;INCREMENT TIME-OUT COUNTER
5611  025756    350                             .BYTE   350
5612                                    ;0257  02E7  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
5613  025757    320     370                     .BYTE   320,370
5614                                    ;0258  02E9                    ; *** ERROR 13 ***
5615                                    ;0259  02E9  A0 00             LDY     #13             ;SET CODE FOR RDA TIME-OUT ERROR
5616  025761    240     015                     .BYTE   240,015
5617                                    ;0260  02EB  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5618  025763    114     220     003             .BYTE   114,220,003
5619                                    ;0261  02EE                    ;READ AND CHECK 252 CHAR
5620                                    ;0262  02EE  AD 00 A1          LDA     RXDB            ;READ RECEIVER BUFFER
5621  025766    255     000     241             .BYTE   255,000,241
5622                                    ;0263  02F1  C9 AA             CMP     #0252           ;CHK FOR 252
5623  025771    311     252                     .BYTE   311,252
5624                                    ;0264  02F3  F0 05             BEQ     .+7             ;BR IF 252
5625  025773    360     005                     .BYTE   360,005
5626                                    ;0265  02F5                    ; *** ERROR 14 ***
5627                                    ;0266  02F5  A0 0E             LDY     #14             ;SET CODE FOR DATA MISCOMPARE ER
5628  025775    240     016                     .BYTE   240,016
5629                                    ;0267  02F7  4C 90 03          JMP     A100            ;GO TAKE ERROR EXIT
5630  025777    114     220     003             .BYTE   114,220,003
5631                                    ;0268  02FA  A2 00             LDX     #0              ;INIT RDA TIME-OUT COUNTER
5632  026002    242     000                     .BYTE   242,000
5633                                    ;0269  02FC  2C 00 A4          BIT     USTATR          ;SEE IF RDA SET
5634  026004    054     000     244             .BYTE   054,000,244
5635                                    ;0270  02FF  30 08             BMI     .+10            ;BR IF RDA SET
5636  026007    060     010                     .BYTE   060,010
5637                                    ;0271  0301  E8                INX                     ;INCREMENT TIME-OUT COUNTER
5638  026011    350                             .BYTE   350
5639                                    ;0272  0302  D0 F8             BNE     .-6             ;BR IF NO TIME-OUT
5640  026012    320     370                     .BYTE   320,370
5641                                    ;0273  0304                    ; *** ERROR 15 ***
```

```
5642                                    ;0274  0304  A0 0F              LDY     #15             ;SET CODE FOR RDA TIME-OUT ERROR
5643  026014    240     017                     .BYTE   240,017
5644                                    ;0275  0306  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5645  026016    114     220     003             .BYTE   114,220,003
5646                                    ;0276  0309                    ;READ AND CHECK 377 CHAR
5647                                    ;0277  0309  AD 00 A1           LDA     RXDB            ;READ RECEIVER BUFFER
5648  026021    255     000     241             .BYTE   255,000,241
5649                                    ;0278  030C  C9 FF              CMP     #0377           ;CHK FOR 377
5650  026024    311     377                     .BYTE   311,377
5651                                    ;0279  030E  F0 05              BEQ     .+7             ;BR IF 377
5652  026026    360     005                     .BYTE   360,005
5653                                    ;0280  0310                    ; *** ERROR 16 ***
5654                                    ;0281  0310  A0 10              LDY     #16             ;SET CODE FOR DATA MISCOMPARE ER
5655  026030    240     020                     .BYTE   240,020
5656                                    ;0282  0312  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5657  026032    114     220     003             .BYTE   114,220,003
5658                                    ;0283  0315  A2 00              LDX     #0              ;INIT RDA TIME-OUT COUNTER
5659  026035    242     000                     .BYTE   242,000
5660                                    ;0284  0317  2C 00 A4           BIT     USTATR          ;SEE IF RDA SET
5661  026037    054     000     244             .BYTE   054,000,244
5662                                    ;0285  031A  30 08              BMI     .+10            ;BR IF RDA SET
5663  026042    060     010                     .BYTE   060,010
5664                                    ;0286  031C  E8                 INX                     ;INCREMENT TIME-OUT COUNTER
5665  026044    350                             .BYTE   350
5666                                    ;0287  031D  D0 F8              BNE     .-6             ;BR IF NO TIME-OUT
5667  026045    320     370                     .BYTE   320,370
5668                                    ;0288  031F                    ; *** ERROR 17 ***
5669                                    ;0289  031F  A0 11              LDY     #17             ;SET CODE FOR RDA TIME-OUT ERROR
5670  026047    240     021                     .BYTE   240,021
5671                                    ;0290  0321  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5672  026051    114     220     003             .BYTE   114,220,003
5673                                    ;0291  0324                    ;READ AND CHECK 000 CHAR
5674                                    ;0292  0324  AD 00 A1           LDA     RXDB            ;READ RECEIVER BUFFER
5675  026054    255     000     241             .BYTE   255,000,241
5676                                    ;0293  0327  C9 00              CMP     #0000           ;CHK FOR 000
5677  026057    311     000                     .BYTE   311,000
5678                                    ;0294  0329  F0 05              BEQ     .+7             ;BR IF 000
5679  026061    360     005                     .BYTE   360,005
5680                                    ;0295  032B                    ; *** ERROR 18 ***
5681                                    ;0296  032B  A0 12              LDY     #18             ;SET CODE FOR DATA MISCOMPARE ER
5682  026063    240     022                     .BYTE   240,022
5683                                    ;0297  032D  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5684  026065    114     220     003             .BYTE   114,220,003
5685                                    ;0298  0330  AE 01 A1           LDX     RDSR            ;CHECK FOR RERR BIT SET
5686  026070    256     001     241             .BYTE   256,001,241
5687                                    ;0299  0333  30 05              BMI     .+7             ;BR IF RERR BIT SET (NO CRC ERRO
5688  026073    060     005                     .BYTE   060,005
5689                                    ;0300  0335                    ; *** ERROR 19 ***
5690                                    ;0301  0335  A0 13              LDY     #19
5691  026075    240     023                     .BYTE   240,023
5692                                    ;0302  0337  4C 90 03           JMP     A100            ;GO TAKE ERROR EXIT
5693  026077    114     220     003             .BYTE   114,220,003
5694                                    ;0303  033A  A2 00              LDX     #0              ;INIT RDA TIME-OUT COUNTER
5695  026102    242     000                     .BYTE   242,000
5696                                    ;0304  033C  2C 00 A4           BIT     USTATR          ;SEE IF RDA SET
5697  026104    054     000     244             .BYTE   054,000,244
```

```
5698                                    ;0305  033F  30 08            BMI     .+10            ;BR IF RDA SET
5699  026107    060     010                     .BYTE   060,010
5700                                    ;0306  0341  E8               INX                     ;INCREMENT TIME-OUT COUNTER
5701  026111    350                             .BYTE   350
5702                                    ;0307  0342  D0 F8            BNE     .-6             ;BR IF NO TIME-OUT
5703  026112    320     370                     .BYTE   320,370
5704                                    ;0308  0344                   ; *** ERROR 20 ***
5705                                    ;0309  0344  A0 14            LDY     #20             ;SET CODE FOR RDA TIME-OUT ERROR
5706  026114    240     024                     .BYTE   240,024
5707                                    ;0310  0346  4C 90 03         JMP     A100            ;GO TAKE ERROR EXIT
5708  026116    114     220     003             .BYTE   114,220,003
5709                                    ;0311  0349                   ;READ AND CHECK FIRST CRC CHAR (160)
5710                                    ;0312  0349  AD 00 A1         LDA     RXDB            ;READ RECEIVER BUFFER
5711  026121    255     000     241             .BYTE   255,000,241
5712                                    ;0313  034C  C9 70            CMP     #0160           ;CHK FOR 160
5713  026124    311     160                     .BYTE   311,160
5714                                    ;0314  034E  F0 05            BEQ     .+7             ;BR IF 160
5715  026126    360     005                     .BYTE   360,005
5716                                    ;0315  0350                   ; *** ERROR 21 ***
5717                                    ;0316  0350  A0 15            LDY     #21             ;SET CODE FOR DATA MISCOMPARE ER
5718  026130    240     025                     .BYTE   240,025
5719                                    ;0317  0352  4C 90 03         JMP     A100            ;GO TAKE ERROR EXIT
5720  026132    114     220     003             .BYTE   114,220,003
5721                                    ;0318  0355  A2 00            LDX     #0              ;INIT RDA TIME-OUT COUNTER
5722  026135    242     000                     .BYTE   242,000
5723                                    ;0319  0357  2C 00 A4         BIT     USTATR          ;SEE IF RDA SET
5724  026137    054     000     244             .BYTE   054,000,244
5725                                    ;0320  035A  30 08            BMI     .+10            ;BR IF RDA SET
5726  026142    060     010                     .BYTE   060,010
5727                                    ;0321  035C  E8               INX                     ;INCREMENT TIME-OUT COUNTER
5728  026144    350                             .BYTE   350
5729                                    ;0322  035D  D0 F8            BNE     .-6             ;BR IF NO TIME-OUT
5730  026145    320     370                     .BYTE   320,370
5731                                    ;0323  035F                   ; *** ERROR 22 ***
5732                                    ;0324  035F  A0 16            LDY     #22             ;SET CODE FOR RDA TIME-OUT ERROR
5733  026147    240     026                     .BYTE   240,026
5734                                    ;0325  0361  4C 90 03         JMP     A100            ;GO TAKE ERROR EXIT
5735  026151    114     220     003             .BYTE   114,220,003
57'6                                    ;0326  0364                   ;READ AND CHECK 2ND CRC CHAR (034)
5737                                    ;0327  0364  AD 00 A1         LDA     RXDB            ;READ RECEIVER BUFFER
5738  026154    255     000     241             .BYTE   255,000,241
5739                                    ;0328  0367  C9 1C            CMP     #0034           ;CHK FOR 034
5740  026157    311     034                     .BYTE   311,034
5741                                    ;0329  0369  F0 05            BEQ     .+7             ;BR IF 034
5742  026161    360     005                     .BYTE   360,005
5743                                    ;0330  0368                   ; *** ERROR 23 ***
5744                                    ;0331  0368  A0 17            LDY     #23             ;SET CODE FOR DATA MISCOMPARE ER
5745  026163    240     027                     .BYTE   240,027
5746                                    ;0332  036D  4C 90 03         JMP     A100            ;GO TAKE ERROR EXIT
5747  026165    114     220     003             .BYTE   114,220,003
5748                                    ;0333  0370  A2 00            LDX     #0              ;INIT RDA TIME-OUT COUNTER
5749  026170    242     000                     .BYTE   242,000
5750                                    ;0334  0372  2C 00 A4         BIT     USTATR          ;SEE IF RDA SET
5751  026172    054     000     244             .BYTE   054,000,244
5752                                    ;0335  0375  30 08            BMI     .+10            ;BR IF RDA SET
5753  026175    060     010                     .BYTE   060,010
```

```
5754                                    ;0336  0377  E8              INX             ;INCREMENT TIME-OUT COUNTER
5755  026177    350                     .BYTE   350
5756                                    ;0337  0378  D0 F8            BNE     .-6     ;BR IF NO TIME-OUT
5757  026200    320   370               .BYTE   320,370
5758                                    ;0338  037A                  ; *** ERROR 24 ***
5759                                    ;0339  037A  A0 18            LDY     #24     ;SET CODE FOR RDA TIME-OUT ERROR
5760  026202    240   030               .BYTE   240,030
5761                                    ;0340  037C  4C 90 03         JMP     A100    ;GO TAKE ERROR EXIT
5762  026204    114   220   003         .BYTE   114,220,003
5763                                    ;0341  037F                  ;DROP RTS, CHECK FOR CARRIER TO DROP
5764                                    ;0342  037F  A2 68            LDX     #TXEN!RXEN!RTSND  ;DE-ASSERT RTS
5765  026207    242   150               .BYTE   242,150
5766                                    ;0343  0381  8E 00 A0         STX     OREGB
5767  026211    216   000   240         .BYTE   216,000,240
5768                                    ;0344  0384  A2 00            LDX     #0      ;INIT CARRIER DROP TIME-OUT COUN
5769  026214    242   000               .BYTE   242,000
5770                                    ;0345  0386  2C 01 A0         BIT     OREGA   ;SEE IF CARRIER CLEARED
5771  026216    054   001   240         .BYTE   054,001,240
5772                                    ;0346  0389  50 05            BVC     .+7     ;BR IF CARRIER CLEARED
5773  026221    120   005               .BYTE   120,005
5774                                    ;0347  038B  E8               INX             ;INCREMENT TIME-OUT COUNTER
5775  026223    350                     .BYTE   350
5776                                    ;0348  038C  D0 F8            BNE     .-6     ;BR IF NO TIME-OUT
5777  026224    320   370               .BYTE   320,370
5778                                    ;0349  038E                  ; *** ERROR 25 ***
5779                                    ;0350  038E  A0 19            LDY     #25     ;SET CODE FOR CARRIER DROP TIME-
5780  026226    240   031               .BYTE   240,031
5781                                    ;0351  0390                  ;COME HERE FOR EXIT
5782                                    ;0352  0390  84 16      A100  STY     BSEL6   ;PUT ERROR NO. (IF ANY) INTO BSE
5783  026230    204   026               .BYTE   204,026
5784                                    ;0353  0392  85 17            STA     BSEL7   ;PUT BAD DATA (IF ANY) INTO BSEL
5785  026232    205   027               .BYTE   205,027
5786                                    ;0354  0394  60               RTS             ;RETURN CONTROL TO LSI-11 PROGRA
5787  026234    140                     .BYTE   140
5788                                    ;0355  0395
5789                                    ;0356  0395
5790                                    ;
5791                                    ;
5792                                    ;ERRORS = 0000
5793                                    ;
5794  026235                            ENDCOD:
5795        026236                      .EVEN
```

```
                                .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::
026236                                  MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
026236  012737  000003  002526          JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
026244  004737  005376                  BIC     #NOLOOP,1$      ;
026250  042737  001000  026326          MOV     #9.,3$          ;*INIT ENTRAN COUNT/TTLOOP STATUS
026256  012737  000011  027100          CMP     BRDTYP,#0       ;IS THIS AN M8064?
026264  023727  002470  000000          BEQ     2$              ; YES: USE TTLOOP (NOT XLB).
026272  001412                          CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
026274  023727  002472  000004          BEQ     2$              ;BR IF NO
026302  001406                          BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
026304  052737  001000  026326          BIS     #BIT15,3$       ;      AND SET MSB OF ENTRAN STATUS (NOLOOP)
026312  052737  100000  027100  ;-----------------------------------------------------------------
026320  004537  007234          2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026324  065626                          DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
026326  000000                  1$:     0                       ;USE 8 BIT CHARS
026330  103003                          BCC     .+8.            ;BR IF NO ERROR
026332                                  ERROR                   ;REPORT STACKED ERROR
026332  104460                                                                  TRAP    C$ERROR
026334                                  ESCAPE  TST             ;SKIP TO END OF TEST
026334  104410                                                                  TRAP    C$ESCAPE
026336  000554                                                                  .WORD   L10022-.
                                ;
026340  004537  010136                  JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
026344  000226                          SYNCH
026346  000007                          7.
026350  103003                          BCC     .+8.            ;BR IF NO ERROR
026352                                  ERROR                   ;REPORT STACKED ERROR
026352  104460                                                                  TRAP    C$ERROR
026354                                  ESCAPE  TST             ;SKIP TO END OF TEST
026354  104410                                                                  TRAP    C$ESCAPE
026356  000534                                                                  .WORD   L10022-.
```

```
5852  026360  004537  010136          JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
5853  026364  000226                  SYNCH
5854  026366  000010                  8.
5855  026370  103003                  BCC     .+8.            ;BR IF NO ERROR
5856  026372                          ERROR                   ;REPORT STACKED ERROR
5857  026372  104460                                                  TRAP    C$ERROR
5858  026374                          ESCAPE  TST             ;SKIP TO END OF TEST
5859  026374  104410                                                  TRAP    C$ESCAPE
5860  026376  000514                                                  .WORD   L10022-.
5861
5862  026400  004537  010250          JSR     R5,TXCTRL       ;CLEAR TSOM
5863  026404  000000                  000
5864  026406  000000                  0
5865
5866  026410  004537  010136          JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
5867  026414  000125                  125
5868  026416  000010                  8.
5869  026420  103003                  BCC     .+8.            ;BR IF NO ERROR
5870  026422                          ERROR                   ;REPORT STACKED ERROR
5871  026422  104460                                                  TRAP    C$ERROR
5872  026424                          ESCAPE  TST             ;SKIP TO END OF TEST
5873  026424  104410                                                  TRAP    C$ESCAPE
5874  026426  000464                                                  .WORD   L10022-.
5875
5876  026430  004537  010136          JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
5877  026434  000252                  252
5878  026436  000010                  8.
5879  026440  103003                  BCC     .+8.            ;BR IF NO ERROR
5880  026442                          ERROR                   ;REPORT STACKED ERROR
5881  026442  104460                                                  TRAP    C$ERROR
5882  026444                          ESCAPE  TST             ;SKIP TO END OF TEST
5883  026444  104410                                                  TRAP    C$ESCAPE
5884  026446  000444                                                  .WORD   L10022-.
5885
5886  026450  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 252(DATA2)
5887  026454  000000                  000
5888  026456  000010                  8.
5889  026460  103003                  BCC     .+8.            ;BR IF NO ERROR
5890  026462                          ERROR                   ;REPORT STACKED ERROR
5891  026462  104460                                                  TRAP    C$ERROR
5892  026464                          ESCAPE  TST             ;SKIP TO END OF TEST
5893  026464  104410                                                  TRAP    C$ESCAPE
5894  026466  000424                                                  .WORD   L10022-.
5895
5896  026470  004537  010136          JSR     R5,TXCHAR       ;LOAD 377(DATA4), TX 000(DATA3)
5897  026474  000377                  377
5898  026476  000010                  8.
5899  026500  103003                  BCC     .+8.            ;BR IF NO ERROR
5900  026502                          ERROR                   ;REPORT STACKED ERROR
5901  026502  104460                                                  TRAP    C$ERROR
5902  026504                          ESCAPE  TST             ;SKIP TO END OF TEST
5903  026504  104410                                                  TRAP    C$ESCAPE
5904  026506  000404                                                  .WORD   L10022-.
5905
5906  026510  004537  010136          JSR     R5,TXCHAR       ;LOAD 001(DATA5)
5907  026514  000001                  001
```

```
 5908  026516  000000                          0
 5909  026520  103003                          BCC     .+8.            ;BR IF NO ERROR
 5910  026522                                  ERROR                   ;REPORT STACKED ERROR
 5911  026522  104460                                                                  TRAP    C$ERROR
 5912  026524                                  ESCAPE  TST             ;SKIP TO END OF TEST
 5913  026524  104410                                                                  TRAP    C$ESCAPE
 5914  026526  000364                                                                  .WORD   L10022-.
 5915
 5916  026530  004537  011624                  JSR     R5,RCV1ST       ;CLOCK AND RCV 125
 5917  026534  000000                          0
 5918  026536  103003                          BCC     .+8.            ;BR IF NO ERROR
 5919  026540                                  ERROR                   ;REPORT STACKED ERROR
 5920  026540  104460                                                                  TRAP    C$ERROR
 5921  026542                                  ESCAPE  TST             ;SKIP TO END OF TEST
 5922  026542  104410                                                                  TRAP    C$ESCAPE
 5923  026544  000346                                                                  .WORD   L10022-.
 5924
 5925  026546  004537  010350                  JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
 5926  026552  000125                          125
 5927  026554  000000                          0
 5928  026556  000010                          8.
 5929  026560  103003                          BCC     .+8.            ;BR IF NO ERROR
 5930  026562                                  ERROR                   ;REPORT STACKED ERROR
 5931  026562  104460                                                                  TRAP    C$ERROR
 5932  026564                                  ESCAPE  TST             ;SKIP TO END OF TEST
 5933  026564  104410                                                                  TRAP    C$ESCAPE
 5934  026566  000324                                                                  .WORD   L10022-.
 5935                          ;-----------------------------------------------------------------
 5936                          ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATX"
 5937                          ;-----------------------------------------------------------------
 5938  026570  012702  002646          MOV     #PATX+1,R2      ;SET UP TABLE POINTER
 5939  026574  112237  026632  5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
 5940  026600  116237  000003  026612  MOVB    3(R2),10$       ;SET UP TRANSMIT CHARACTER
 5941
 5942  026606  004537  010136          JSR     R5,TXCHAR       ;LOAD A CHARACTER
 5943  026612  000000                  10$:    000             ;** HOLE FOR NEXT TX CHARACTER
 5944  026614  000000                          0
 5945  026616  103003                          BCC     .+8.            ;BR IF NO ERROR
 5946  026620                                  ERROR                   ;REPORT STACKED ERROR
 5947  026620  104460                                                                  TRAP    C$ERROR
 5948  026622                                  ESCAPE  TST             ;SKIP TO END OF TEST
 5949  026622  104410                                                                  TRAP    C$ESCAPE
 5950  026624  000266                                                                  .WORD   L10022-.
 5951
 5952  026626  004537  010350                  JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
 5953  026632  000000                  20$:    000             ;** HOLE FOR EXPECTED CHARACTER
 5954  026634  000000                          0
 5955  026636  000010                          8.
 5956  026640  103003                          BCC     .+8.            ;BR IF NO ERROR
 5957  026642                                  ERROR                   ;REPORT STACKED ERROR
 5958  026642  104460                                                                  TRAP    C$ERROR
 5959  026644                                  ESCAPE  TST             ;SKIP TO END OF TEST
 5960  026644  104410                                                                  TRAP    C$ESCAPE
 5961  026646  000244                                                                  .WORD   L10022-.
 5962
 5963  026650  022702  002665          CMP     #PATX+16.,R2    ;CHECK FOR 20TH CHARACTER OF TABLE
```

```
 5964  026654  001347                    BNE     5$              ;BR IF NOT DONE
 5965                                    ;-----------------------------------------------------------------
 5966  026656  004537  010250            JSR     R5,TXCTRL       ;LOAD 1ST TEOM
 5967  026662  000002                    TEOM
 5968  026664  000000                    0
 5969  026666  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 357(DATA17), RCV 337(DATA18)
 5970  026672  000357                    357
 5971  026674  000000                    0
 5972  026676  000010                    8.
 5973  026700  103003                    BCC     .+8.            ;BR IF NO ERROR
 5974  026702                            ERROR                   ;REPORT STACKED ERROR
 5975  026702  104460                                                    TRAP    C$ERROR
 5976  026704                            ESCAPE  TST             ;SKIP TO END OF TEST
 5977  026704  104410                                                    TRAP    C$ESCAPE
 5978  026706  000204                                                    .WORD   L10022-.
 5979
 5980  026710  004537  010250            JSR     R5,TXCTRL       ;LOAD 2ND TEOM
 5981  026714  000002                    TEOM
 5982  026716  000000                    0
 5983  026720  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 337(DATA18), RCV 277(DATA19)
 5984  026724  000337                    337
 5985  026726  000000                    0
 5986  026730  000010                    8.
 5987  026732  103003                    BCC     .+8.            ;BR IF NO ERROR
 5988  026734                            ERROR                   ;REPORT STACKED ERROR
 5989  026734  104460                                                    TRAP    C$ERROR
 5990  026736                            ESCAPE  TST             ;SKIP TO END OF TEST
 5991  026736  104410                                                    TRAP    C$ESCAPE
 5992  026740  000152                                                    .WORD   L10022-.
 5993
 5994  026742  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 277(DATA19), RCV 177(DATA20)
 5995  026746  000277                    277
 5996  026750  000000                    0
 5997  026752  000010                    8.
 5998  026754  103003                    BCC     .+8.            ;BR IF NO ERROR
 5999  026756                            ERROR                   ;REPORT STACKED ERROR
 6000  026756  104460                                                    TRAP    C$ERROR
 6001  026760                            ESCAPE  TST             ;SKIP TO END OF TEST
 6002  026760  104410                                                    TRAP    C$ESCAPE
 6003  026762  000130                                                    .WORD   L10022-.
 6004
 6005  026764  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 177(DATA20), RCV FIRST CRC BYTE
 6006  026770  100177                    RXERR!177
 6007  026772  000001                    RERCHK
 6008  026774  000010                    8.
 6009  026776  103003                    BCC     .+8.            ;BR IF NO ERROR
 6010  027000                            ERROR                   ;REPORT STACKED ERROR
 6011  027000  104460                                                    TRAP    C$ERROR
 6012  027002                            ESCAPE  TST             ;SKIP TO END OF TEST
 6013  027002  104410                                                    TRAP    C$ESCAPE
 6014  027004  000106                                                    .WORD   L10022-.
 6015
 6016  027006  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
 6017  027012  000156                    156
 6018  027014  000000                    0
 6019  027016  000010                    8.
```

```
6020  027020  103003                  BCC     .+8.            ;BR IF NO ERROR
6021  027022                          ERROR                   ;REPORT STACKED ERROR
6022  027022  104460                                                  TRAP    C$ERROR
6023  027024                          ESCAPE  TST             ;SKIP TO END OF TEST
6024  027024  104410                                                  TRAP    C$ESCAPE
6025  027026  000064                                                  .WORD   L10022-.
6026
6027  027030  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
6028  027034  000236                  236
6029  027036  000000                  0
6030  027040  000010                  8.
6031  027042  103003                  BCC     .+8.            ;BR IF NO ERROR
6032  027044                          ERROR                   ;REPORT STACKED ERROR
6033  027044  104460                                                  TRAP    C$ERROR
6034  027046                          ESCAPE  TST             ;SKIP TO END OF TEST
6035  027046  104410                                                  TRAP    C$ESCAPE
6036  027050  000042                                                  .WORD   L10022-.
6037                          ;-----------------------------------------------------------------
6038  027052  005337  002526          DEC     REGO            ;DECREMENT COUNT
6039  027056  001406                  BEQ     40$             ;BR IF TRIPLE LOOP IS COMPLETED
6040
6041  027060  004537  010250          JSR     R5,TXCTRL       ;CLEAR TEOM, SET TSOM
6042  027064  000001                  ^SOM
6043  027066  000001                  1
6044  027070  000137  026320          JMP     2$              ;AND RUN TX/RX AGAIN
6045
6046  027074  004537  011772  40$:    JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
6047  027100  000011  3$:     9.
6048  027102  103003                  BCC     .+8.            ;BR IF NO ERROR
6049  027104                          ERROR                   ;REPORT STACKED ERROR
6050  027104  104460                                                  TRAP    C$ERROR
6051  027106                          ESCAPE  TST             ;SKIP TO END OF TEST
6052  027106  104410                                                  TRAP    C$ESCAPE
6053  027110  000002                                                  .WORD   L10022-.
6054  027112          ENDTST
6055  027112                                                  L10022:
6056  027112  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
027114                                                                          T4::
027114  004737  005376          JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
027120  042737  001000  027176  BIC     #NOLOOP,1$      ;
027126  012737  000011  027642  MOV     #9.,3$          ;INIT ENTRAN COUNT/TTLOOP STATUS
027134  023727  002470  000000  CMP     BRDTYP,#0       ;IS THIS AN M8064?
027142  001412                  BEQ     2$              ; YES: USE TTLOOP (NOT XLB).
027144  023727  002472  000004  CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
027152  001406                  BEQ     2$              ;BR IF NO
027154  052737  001000  027176  BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
027162  052737  100000  027642  BIS     #BIT15,3$       ;      AND SPECIFY NOLOOP IN ENTRAN
                        ;-----------------------------------------------------------------
027170  004537  007234  2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
027174  062226                  DDCMP!STRIPS!OVRC!SYNCH ;SET DDCMP,STRIP SYNC,ODD VRC,SYNCH=226
027176  000347          1$:     TXDL!RXDL               ;USE 7 BIT CHARS FOR RX & TX
027200  103003                  BCC     .+8.            ;BR IF NO ERROR
027202                          ERROR                   ;REPORT STACKED ERROR
027202  104460                                                          TRAP    C$ERROR
027204                          ESCAPE  TST             ;SKIP TO END OF TEST
027204  104410                                                          TRAP    C$ESCAPE
027206  000446                                                          .WORD   L10023-.

027210  004537  010136          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
027214  000226                  SYNCH
027216  000007                  7.
027220  103003                  BCC     .+8.            ;BR IF NO ERROR
027222                          ERROR                   ;REPORT STACKED ERROR
027222  104460                                                          TRAP    C$ERROR
027224                          ESCAPE  TST             ;SKIP TO END OF TEST
027224  104410                                                          TRAP    C$ESCAPE
027226  000426                                                          .WORD   L10023-.
```

```
 6113  027230  004537  010136          JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
 6114  027234  000226                  SYNCH
 6115  027236  000010                  8.
 6116  027240  103003                  BCC     .+8.            ;BR IF NO ERROR
 6117  027242                          ERROR                   ;REPORT STACKED ERROR
 6118  027242  104460                                                  TRAP    C$ERROR
 6119  027244                          ESCAPE  TST             ;SKIP TO END OF TEST
 6120  027244  104410                                                  TRAP    C$ESCAPE
 6121  027246  000406                                                  .WORD   L10023-.
 6122
 6123  027250  004537  010250          JSR     R5,TXCTRL       ;CLEAR TSOM
 6124  027254  000000                  000
 6125  027256  000000                  0
 6126
 6127  027260  004537  010136          JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
 6128  027264  000125                  125
 6129  027266  000010                  8.
 6130  027270  103003                  BCC     .+8.            ;BR IF NO ERROR
 6131  027272                          ERROR                   ;REPORT STACKED ERROR
 6132  027272  104460                                                  TRAP    C$ERROR
 6133  027274                          ESCAPE  TST             ;SKIP TO END OF TEST
 6134  027274  104410                                                  TRAP    C$ESCAPE
 6135  027276  000356                                                  .WORD   L10023-.
 6136
 6137  027300  004537  010136          JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
 6138  027304  000252                  252
 6139  027306  000010                  8.
 6140  027310  103003                  BCC     .+8.            ;BR IF NO ERROR
 6141  027312                          ERROP                   ;REPORT STACKED ERROR
 6142  027312  104460                                                  TRAP    C$ERROR
 6143  027314                          ESCAPE  TST             ;SKIP TO END OF TEST
 6144  027314  104410                                                  TRAP    C$ESCAPE
 6145  027316  000336                                                  .WORD   L10023-.
 6146
 6147  027320  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(DATA3)
 6148  027324  000000                  000
 6149  027326  000000                  0
 6150  027330  103003                  BCC     .+8.            ;BR IF NO ERROR
 6151  027332                          ERROR                   ;REPORT STACKED ERROR
 6152  027332  104460                                                  TRAP    C$ERROR
 6153  027334                          ESCAPE  TST             ;SKIP TO END OF TEST
 6154  027334  104410                                                  TRAP    C$ESCAPE
 6155  027336  000316                                                  .WORD   L10023-.
 6156
 6157  027340  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV 1.5(DATA1)
 6158  027344  000000                  0
 6159  027346  103003                  BCC     .+8.            ;BR IF NO ERROR
 6160  027350                          ERROR                   ;REPORT STACKED ERROR
 6161  027350  104460                                                  TRAP    C$ERROR
 6162  027352                          ESCAPE  TST             ;SKIP TO END OF TEST
 6163  027352  104410                                                  TRAP    C$ESCAPE
 6164  027354  000300                                                  .WORD   L10023-.
 6165
 6166  027356  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
 6167  027362  000125                  125
 6168  027364  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
```

```
6169  027366  000010                        8.
6170  027370  103003                        BCC     .+8.            ;BR IF NO ERROR
6171  027372                                ERROR                   ;REPORT STACKED ERROR
6172  027372  104460                                                                TRAP    C$ERROR
6173  027374                                ESCAPE  TST             ;SKIP TO END OF TEST
6174  027374  104410                                                                TRAP    C$ESCAPE
6175  027376  000256                                                                .WORD   L10023-.
6176                                  ;-----------------------------------------------------------------
6177                                  ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATX"
6178                                  ;-----------------------------------------------------------------
6179  027400  012702  002646                MOV     #PATX+1,R2      ;SET UP TABLE POINTER
6180  027404  112237  027442          5$:   MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
6181  027410  116237  000001  027422        MOVB    1(R2),10$       ;SET UP TRANSMIT CHARACTER
6182
6183  027416  004537  010136                JSR     R5,TXCHAR       ;LOAD A CHARACTER
6184  027422  000000                  10$:   000                     ;** HOLE FOR NEXT TX CHARACTER
6185  027424  000000                        0
6186  027426  103003                        BCC     .+8.            ;BR IF NO ERROR
6187  027430                                ERROR                   ;REPORT STACKED ERROR
6188  027430  104460                                                                TRAP    C$ERROR
6189  027432                                ESCAPE  TST             ;SKIP TO END OF TEST
6190  027432  104410                                                                TRAP    C$ESCAPE
6191  027434  000220                                                                .WORD   L10023-.
6192
6193  027436  004537  010350                JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
6194  027442  000000                  20$:   000                     ;** HOLE FOR EXPECTED CHARACTER
6195  027444  000001                        RF%CHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6196  027446  000010                        8.
6197  027450  103003                        BCC     .+8.            ;BR IF NO ERROR
6198  027452                                ERROR                   ;REPORT STACKED ERROR
6199  027452  104460                                                                TRAP    C$ERROR
6200  027454                                ESCAPE  TST             ;SKIP TO END OF TEST
6201  027454  104410                                                                TRAP    C$ESCAPE
6202  027456  000176                                                                .WORD   L10023-.
6203
6204  027460  022702  002737                CMP     #EPATX-1,R2
6205  027464  001347                        BNE     5$
6206                                  ;-----------------------------------------------------------------
6207  027466  004537  010250                JSR     R5,TXCTRL       ;LOAD TSOM
6208  027472  000001                        TSOM
6209  027474  000000                        0
6210  027476  103003                        BCC     .+8.            ;BR IF NO ERROR
6211  027500                                ERROR                   ;REPORT STACKED ERROR
6212  027500  104460                                                                TRAP    C$ERROR
6213  027502                                ESCAPE  TST             ;SKIP TO END OF TEST
6214  027502  104410                                                                TRAP    C$ESCAPE
6215  027504  000150                                                                .WORD   L10023-.
6216
6217  027506  004537  010350                JSR     R5,RXCHAR       ;READ & CHK 277, RCV 177
6218  027512  000277                        277
6219  027514  000001                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6220  027516  000010                        8.
6221  027520  103003                        BCC     .+8.            ;BR IF NO ERROR
6222  027522                                ERROR                   ;REPORT STACKED ERROR
6223  027522  104460                                                                TRAP    C$ERROR
6224  027524                                ESCAPE  TST             ;SKIP TO END OF TEST
```

```
6225  027524  104410                                                          TRAP    C$ESCAPE
6226  027526  000126                                                          .WORD   L10023-.
6227
6228  027530  004537  010250          JSR     R5,TXCTRL       ;LOAD 2ND TSOM
6229  027534  000001                  TSOM
6230  027536  000000                  0
6231  027540  103003                  BCC     .+8.            ;BR IF NO ERROR
6232  027542                          ERROR                   ;REPORT STACKED ERROR
6233  027542  104460                                                          TRAP    C$ERROR
6234  027544                          ESCAPE  TST             ;SKIP TO END OF TEST
6235  027544  104410                                                          TRAP    C$ESCAPE
6236  027546  000106                                                          .WORD   L10023-.
6237
6238  027550  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 177, RCV FIRST SYNC
6239  027554  000177                  177
6240  027556  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6241  027560  000010                  8.
6242  027562  103003                  BCC     .+8.            ;BR IF NO ERROR
6243  027564                          ERROR                   ;REPORT STACKED ERROR
6244  027564  104460                                                          TRAP    C$ERROR
6245  027566                          ESCAPE  TST             ;SKIP TO END OF TEST
6246  027566  104410                                                          TRAP    C$ESCAPE
6247  027570  000064                                                          .WORD   L10023-.
6248
6249  027572  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 1ST SYNC, RCV SECOND SYNC
6250  027576  000226                  SYNCH
6251  027600  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6252  027602  000010                  8.
6253  027604  103003                  BCC     .+8.            ;BR IF NO ERROR
6254  027606                          ERROR                   ;REPORT STACKED ERROR
6255  027606  104460                                                          TRAP    C$ERROR
6256  027610                          ESCAPE  TST             ;SKIP TO END OF TEST
6257  027610  104410                                                          TRAP    C$ESCAPE
6258  027612  000042                                                          .WORD   L10023-.
6259
6260  027614  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND SYNC, RCV NEXT ONE
6261  027620  000226                  SYNCH
6262  027622  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6263  027624  000010                  8.
6264  027626  103003                  BCC     .+8.            ;BR IF NO ERROR
6265  027630                          ERROR                   ;REPORT STACKED ERROR
6266  027630  104460                                                          TRAP    C$ERROR
6267  027632                          ESCAPE  TST             ;SKIP TO END OF TEST
6268  027632  104410                                                          TRAP    C$ESCAPE
6269  027634  000020                                                          .WORD   L10023-.
6270
```

```
 6271  027636  004537  ^11772                JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
 6272  027642  000011                  3$:   9.
 6273  027644  103003                        BCC     .+8.            ;BR IF NO ERROR
 6274  027646                                ERROR                   ;REPORT STACKED ERROR
 6275  027646  104460                                                                TRAP    C$ERROR
 6276  027650                                ESCAPE  TST             ;SKIP TO END OF TEST
 6277  027650  104410                                                                TRAP    C$ESCAPE
 6278  027652  000002                                                                .WORD   L10023-.
 6279  027654                        ENDTST
 6280  027654                                                        L10023:
 6281  027654  104401                                                                TRAP    C$ETST
```

```
                        .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
                                                                                        T5::
027656                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
027662  004737  005376  BIC     @#NOLOOP,1$     ;
027670  042737  001000  027740  MOV     #9.,3$          ;INIT ENTRAN COUNT/STATUS
027676  012737  000011  030404  CMP     BRDTYP,#60      ;IS BOARD TYPE M8064?
027704  023727  002470  000000  BEQ     2$              ; YES: SPECIFY TTLOOP (NOT XLB)
027706  001412                  CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
027714  023727  002472  000004  BEQ     2$              ;BR IF NO
027716  001406                  BIS     @#NOLOOP,1$     ; YES: SPECIFY NO TTLOOP (INITRN)
027724  052737  001000  027740  BIS     BIT15,3$        ;       AND SPECIFY NOLOOP IN ENTRAN
030404  053737  100000  030404  ;-----------------------------------------------------------------
                        2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
027732  004537  007234          DDCMP!STRIPS!EVRC!SYNCH  ;SET DDCMP,STRIP SYNCH,EVEN VRC,SYNCH=226
027736  062626                  TXDL!RXDL       ;USE 7 BIT CHARS FOR TX & RX
027740  000347          1$:     BCC     .+8.            ;BR IF NO ERROR
027742  103003                  ERROR                   ;REPORT STACKED ERROR
027744                                                                  TRAP    C$ERROR
027744  104460                  ESCAPE  TST             ;SKIP TO END OF TEST
027746                                                                  TRAP    C$ESCAPE
027746  104410                                                          .WORD   L10024-.
027750  000446
                        ;
027752  004537  010136          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
027756  000226                  SYNCH
027760  000007                  7.
027762  103003                  BCC     .+8.            ;BR IF NO ERROR
027764                          ERROR                   ;REPORT STACKED ERROR
027764  104460                                                          TRAP    C$ERROR
027766                          ESCAPE  TST             ;SKIP TO END OF TEST
027766  104410                                                          TRAP    C$ESCAPE
027770  000426                                                          .WORD   L10024-.
                        ;
027772  004537  010136          JSR     R5,TXCHAR       ;LOAD 3RD SYNCH, TX 2ND SYNCH
```

```
 6338  027776  000226                    SYNCH
 6339  030000  000010                    8.
 6340  030002  103003                    BCC     .+8.            ;BR IF NO ERROR
 6341  030004                            ERROR                   ;REPORT STACKED ERROR
 6342  030004  104460                                                    TRAP    C$ERROR
 6343  030006                            ESCAPE  TST             ;SKIP TO END OF TEST
 6344  030006  104410                                                    TRAP    C$ESCAPE
 6345  030010  000406                                                    .WORD   L10024-.
 6346
 6347  030012  004537  010250            JSR     R5,TXCTRL       ;CLEAR TSOM
 6348  030016  000000                    000
 6349  030020  000000                    0
 6350
 6351  030022  004537  010136            JSR     R5,TXCHAR       ;LOAD 125(DATA1), TX 3RD SYNCH
 6352  030026  000125                    125
 6353  030030  000010                    8.
 6354  030032  103003                    BCC     .+8.            ;BR IF NO ERROR
 6355  030034                            ERROR                   ;REPORT STACKED ERROR
 6356  030034  104460                                                    TRAP    C$ERROR
 6357  030036                            ESCAPE  TST             ;SKIP TO END OF TEST
 6358  030036  104410                                                    TRAP    C$ESCAPE
 6359  030040  000356                                                    .WORD   L10024-.
 6360
 6361  030042  004537  010136            JSR     R5,TXCHAR       ;LOAD 252(DATA2), TX 125(DATA1)
 6362  030046  000252                    252
 6363  030050  000010                    8.
 6364  030052  103003                    BCC     .+8.            ;BR IF NO ERROR
 6365  030054                            ERROR                   ;REPORT STACKED ERROR
 6366  030054  104460                                                    TRAP    C$ERROR
 6367  030056                            ESCAPE  TST             ;SKIP TO END OF TEST
 6368  030056  104410                                                    TRAP    C$ESCAPE
 6369  030060  000336                                                    .WORD   L10024-.
 6370
 6371  030062  004537  010136            JSR     R5,TXCHAR       ;LOAD 000(DATA3)
 6372  030066  000000                    000
 6373  030070  000000                    0
 6374  030072  103003                    BCC     .+8.            ;BR IF NO ERROR
 6375  030074                            ERROR                   ;REPORT STACKED ERROR
 6376  030074  104460                                                    TRAP    C$ERROR
 6377  030076                            ESCAPE  TST             ;SKIP TO END OF TEST
 6378  030076  104410                                                    TRAP    C$ESCAPE
 6379  030100  000316                                                    .WORD   L10024-.
 6380
 6381  030102  004537  011624            JSR     R5,RCV1ST       ;CLOCK AND RCV 125(DATA1)
 6382  030106  000000                    0
 6383  030110  103003                    BCC     .+8.            ;BR IF NO ERROR
 6384  030112                            ERROR                   ;REPORT STACKED ERROR
 6385  030112  104460                                                    TRAP    C$ERROR
 6386  030114                            ESCAPE  TST             ;SKIP TO END OF TEST
 6387  030114  104410                                                    TRAP    C$ESCAPE
 6388  030116  000300                                                    .WORD   L10024-.
 6389
 6390  030120  004537  010350            JSR     R5,RXCHAR       ;READ & CHK 125(DATA1), RCV 252(DATA2)
 6391  030124  000125                    125
 6392  030126  000001                    RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
 6393  030130  000010                    8.
```

```
6394  030132  103003                        BCC     .+8.            ;BR IF NO ERROR
6395  030134                                ERROR                   ;REPORT STACKED ERROR
6396  030134  104460                                                                TRAP    C$ERROR
6397  030136                                ESCAPE  TST             ;SKIP TO END OF TEST
6398  030136  104410                                                                TRAP    C$ESCAPE
6399  030140  000256                                                                .WORD   L10024-.
6400                                  ;-----------------------------------------------------------------
6401                                  ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATX"
6402                                  ;-----------------------------------------------------------------
6403  030142  012702  002646                5$:     MOV     #PATX+1,R2      ;SET UP TABLE POINTER
6404  030146  112237  030204                        MOVB    (R2)+,20$       ;SET UP EXPECTED CHARACTER
6405  030152  116237  000001  030164              MOVB    1(R2),10$       ;SET UP TRANSMIT CHARACTER
6406
6407  030160  004537  010136                        JSR     R5,TXCHAR       ;LOAD A CHARACTER
6408  030164  000000                        10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
6409  030166  000000                                0
6410  030170  103003                        BCC     .+8.            ;BR IF NO ERROR
6411  030172                                ERROR                   ;REPORT STACKED ERROR
6412  030172  104460                                                                TRAP    C$ERROR
6413  030174                                ESCAPE  TST             ;SKIP TO END OF TEST
6414  030174  104410                                                                TRAP    C$ESCAPE
6415  030176  000220                                                                .WORD   L10024-.
6416
6417  030200  004537  010350                        JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTEP
6418  030204  000000                        20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
6419  030206  000001                                RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6420  030210  000010                                8.
6421  030212  103003                        BCC     .+8.            ;BR IF NO ERROR
6422  030214                                ERROR                   ;REPORT STACKED ERROR
6423  030214  104460                                                                TRAP    C$ERROR
6424  030216                                ESCAPE  TST             ;SKIP TO END OF TEST
6425  030216  104410                                                                TRAP    C$ESCAPE
6426  030220  000176                                                                .WORD   L10024-.
6427
6428  030222  022702  002737                        CMP     #EPATX-1,R2
6429  030226  001347                        BNE     5$
6430                                  ;-----------------------------------------------------------------
6431  030230  004537  010250                        JSR     R5,TXCTRL       ;LOAD 1ST TSOM
6432  030234  000001                                TSOM
6433  030236  000000                                0
6434  030240  103003                        BCC     .+8.            ;BR IF NO ERROR
6435  030242                                ERROR                   ;REPORT STACKED ERROR
6436  030242  104460                                                                TRAP    C$ERROR
6437  030244                                ESCAPE  TST             ;SKIP TO END OF TEST
6438  030244  104410                                                                TRAP    C$ESCAPE
6439  030246  000150                                                                .WORD   L10024-.
6440
6441  030250  004537  010350                        JSR     R5,RXCHAR       ;READ & CHK 277, RCV 177
6442  030254  000277                                277
6443  030256  000001                                RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6444  030260  000010                                8.
6445  030262  103003                        BCC     .+8.            ;BR IF NO ERROR
6446  030264                                ERROR                   ;REPORT STACKED ERROR
6447  030264  104460                                                                TRAP    C$ERROR
6448  030266                                ESCAPE  TST             ;SKIP TO END OF TEST
6449  030266  104410                                                                TRAP    C$ESCAPE
```

```
6450  030270  000126                                                          .WORD   L10024-.
6451
6452  030272  004537  010250          JSR     R5,TXCTRL       ;LOAD 2ND TSOM
6453  030276  000001                  TSOM
6454  030300  000000                  0
6455  030302  103003                  BCC     .+8.            ;BR IF NO ERROR
6456  030304                          ERROR                   ;REPORT STACKED ERROR
6457  030304  104460                                                          TRAP    C$ERROR
6458  030306                          ESCAPE  TST             ;SKIP TO END OF TEST
6459  030306  104410                                                          TRAP    C$ESCAPE
6460  030310  000106                                                          .WORD   L10024-.
6461
6462  030312  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 177, RCV FIRST SYNC
6463  030316  000177                  177
6464  030320  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6465  030322  000010                  8.
6466  030324  103003                  BCC     .+8.            ;BR IF NO ERROR
6467  030326                          ERROR                   ;REPORT STACKED ERROR
6468  030326  104460                                                          TRAP    C$ERROR
6469  030330                          ESCAPE  TST             ;SKIP TO END OF TEST
6470  030330  104410                                                          TRAP    C$ESCAPE
6471  030332  000064                                                          .WORD   L10024-.
6472
6473  030334  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 1ST SYNC, RCV SECOND SYNC
6474  030340  000226                  SYNCH
6475  030342  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6476  030344  000010                  8.
6477  030346  103003                  BCC     .+8.            ;BR IF NO ERROR
6478  030350                          ERROR                   ;REPORT STACKED ERROR
6479  030350  104460                                                          TRAP    C$ERROR
6480  030352                          ESCAPE  TST             ;SKIP TO END OF TEST
6481  030352  104410                                                          TRAP    C$ESCAPE
6482  030354  000042                                                          .WORD   L10024-.
6483
6484  030356  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 2ND SYNC, RCV NEXT ONE
6485  030362  000226                  SYNCH
6486  030364  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD VRC)
6487  030366  000010                  8.
6488  030370  103003                  BCC     .+8.            ;BR IF NO ERROR
6489  030372                          ERROR                   ;REPORT STACKED ERROR
6490  030372  104460                                                          TRAP    C$ERROR
6491  030374                          ESCAPE  TST             ;SKIP TO END OF TEST
6492  030374  104410                                                          TRAP    C$ESCAPE
6493  030376  000020                                                          .WORD   L10024-.
6494
6495  030400  004537  011772          JSR     R5,ENTRAN       ;SHUT DOWN TRANSMITTER, RECEIVER
6496  030404  000011          3$:     9.                      ;
6497  030406  103003                  BCC     .+8.            ;BR IF NO ERROR
6498  030410                          ERROR                   ;REPORT STACKED ERROR
6499  030410  104460                                                          TRAP    C$ERROR
6500  030412                          ESCAPE  TST             ;SKIP TO END OF TEST
6501  030412  104410                                                          TRAP    C$ESCAPE
6502  030414  000002                                                          .WORD   L10024-.
6503  030416                  ENDTST
6504  030416                                                          L10024:
6505  030416  104401                                                          TRAP    C$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).
                                ;*
                                ;---*********************************************************************
                                ;
                                ;       BGNTST
030420                                                                          T6::
030420  012737  000003  002526          MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
030426  004737  005376                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
030432  042737  001000  030474          BIC     #NOLOOP,1$      ;
030440  023727  002470  000000          CMP     BRDTYP,#0       ;IS BOARD TYPE = M8064 ?
030446  001407                          BEQ     2$              ; YES: SPECIFY TTLOOP (NO XLB)
030450  023727  002472  000004          CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
030456  001403                          BEQ     2$              ;BR IF NO
030460  052737  001000  030474          BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
                                ;-------------------------------------------------------------------
030466  004537  007234          2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030472  000000                          0                       ;SET BOP MODE,CRC-CCITT=>1'S CHECK
030474  000000                  1$:     0                       ;USE 8 BIT CHARS
030476  103003                          BCC     .+8.            ;BR IF NO ERROR
030500                                  ERROR                   ;REPORT STACKED ERROR
030500  104460                                                                  TRAP    C$ERROR
030502                                  ESCAPE  TST             ;SKIP TO END OF TEST
030502  104410                                                                  TRAP    C$ESCAPE
030504  000374                                                                  .WORD   L10025-.
                                ;
030506  004537  010250                  JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
030512  000001                          TSOM
030514  000007                          7.
030516  004537  010250                  JSR     R5,TXCTRL       ;LOAD 3RD FLAG, TX 2ND FLAG
030522  000001                          TSOM
030524  000010                          8.
030526  004537  010250                  JSR     R5,TXCTRL       ;LOAD 4TH FLAG, TX 3RD FLAG
030532  000001                          TSOM
030534  000010                          8.
030536  004537  010250                  JSR     R5,TXCTRL       ;CLEAR TSOM
030542  000000                          000
030544  000000                          0
030546  004537  010136                  JSR     R5,TXCHAR       ;LOAD DATA1(125), TX 4TH FLAG
030552  000125                          125
030554  000010                          8.
030556  103003                          BCC     .+8.            ;BR IF NO ERROR
```

```
6562  030560                        ERROR                   ;REPORT STACKED ERROR
6563  030560  104460                                                  TRAP    C$ERROR
6564  030562                        ESCAPE  TST             ;SKIP TO END OF TEST
6565  030562  104410                                                  TRAP    C$ESCAPE
6566  030564  000314                                                  .WORD   L10025-.
6567
6568  030566  004537  010136        JSR     R5,TXCHAR       ;LOAD DATA2(252), TX DATA1(125)
6569  030572  000252                252
6570  030574  000010                8.
6571  030576  103003                BCC     .+8.            ;BR IF NO ERROR
6572  030600                        ERROR                   ;REPORT STACKED ERROR
6573  030600  104460                                                  TRAP    C$ERROR
6574  030602                        ESCAPE  TST             ;SKIP TO END OF TEST
6575  030602  104410                                                  TRAP    C$ESCAPE
6576  030604  000274                                                  .WORD   L10025-.
6577
6578  030606  004537  010136        JSR     R5,TXCHAR       ;LOAD DATA3(000), TX DATA2(252)
6579  030612  000000                000
6580  030614  000010                8.
6581  030616  103003                BCC     .+8.            ;BR IF NO ERROR
6582  030620                        ERROR                   ;REPORT STACKED ERROR
6583  030620  104460                                                  TRAP    C$ERROR
6584  030622                        ESCAPE  TST             ;SKIP TO END OF TEST
6585  030622  104410                                                  TRAP    C$ESCAPE
6586  030624  000254                                                  .WORD   L10025-.
6587
6588  030626  004537  010136        JSR     R5,TXCHAR       ;LOAD DATA4(377), TX DATA3(000)
6589  030632  000377                377
6590  030634  000010                8.
6591  030636  103003                BCC     .+8.            ;BR IF NO ERROR
6592  030640                        ERROR                   ;REPORT STACKED ERROR
6593  030640  104460                                                  TRAP    C$ERROR
6594  030642                        ESCAPE  TST             ;SKIP TO END OF TEST
6595  030642  104410                                                  TRAP    C$ESCAPE
6596  030644  000234                                                  .WORD   L10025-.
6597
6598  030646  004537  010136        JSR     R5,TXCHAR       ;LOAD DATA5(001), TX DATA4(377)
6599  030652  000001                001
6600  030654  000011                9.
6601  030656  103003                BCC     .+8.            ;BR IF NO ERROR
6602  030660                        ERROR                   ;REPORT STACKED ERROR
6603  030660  104460                                                  TRAP    C$ERROR
6604  030662                        ESCAPE  TST             ;SKIP TO END OF TEST
6605  030662  104410                                                  TRAP    C$ESCAPE
6606  030664  000214                                                  .WORD   L10025-.
6607
6608  030666  004537  010250        JSR     R5,TXCTRL       ;SET TEOM
6609  030672  000002                TEOM
6610  030674  000000                0
6611
6612  030676  004537  011624        JSR     R5,RCV1ST       ;CLOCK AND RCV DATA1(125)
6613  030702  000000                0
6614  030704  103003                BCC     .+8.            ;BR IF NO ERROR
6615  030706                        ERROR                   ;REPORT STACKED ERROR
6616  030706  104460                                                  TRAP    C$ERROR
6617  030710                        ESCAPE  TST             ;SKIP TO END OF TEST
```

```
 6618  030710  104410                                                          TRAP    C$ESCAPE
 6619  030712  000166                                                          .WORD   L10025-.
 6620
 6621  030714  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA1(125), RCV DATA2(252)
 6622  030720  000525                  RXSOM!125               ; & CHECK RSOM=1
 6623  030722  000000                  0
 6624  030724  000010                  8.
 6625  030726  103003                  BCC     .+8.            ;BR IF NO ERROR
 6626  030730                          ERROR                   ;REPORT STACKED ERROR
 6627  030730  104460                                                          TRAP    C$ERROR
 6628  030732                          ESCAPE  TST             ;SKIP TO END OF TEST
 6629  030732  104410                                                          TRAP    C$ESCAPE
 6630  030734  000144                                                          .WORD   L10025-.
 6631
 6632  030736  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
 6633  030742  000002                  TEOM
 6634  030744  000000                  0
 6635
 6636  030746  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA2(252), RCV DATA3(000)
 6637  030752  000252                  252
 6638  030754  000000                  0
 6639  030756  000010                  8.
 6640  030760  103003                  BCC     .+8.            ;BR IF NO ERROR
 6641  030762                          ERROR                   ;REPORT STACKED ERROR
 6642  030762  104460                                                          TRAP    C$ERROR
 6643  030764                          ESCAPE  TST             ;SKIP TO END OF TEST
 6644  030764  104410                                                          TRAP    C$ESCAPE
 6645  030766  000112                                                          .WORD   L10025-.
 6646
 6647  030770  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA3(000), RCV DATA4(377)
 6648  030774  000000                  000
 6649  030776  000000                  0
 6650  031000  000010                  8.
 6651  031002  103003                  BCC     .+8.            ;BR IF NO ERROR
 6652  031004                          ERROR                   ;REPORT STACKED ERROR
 6653  031004  104460                                                          TRAP    C$ERROR
 6654  031006                          ESCAPE  TST             ;SKIP TO END OF TEST
 6655  031006  104410                                                          TRAP    C$ESCAPE
 6656  031010  000070                                                          .WORD   L10025-.
 6657
 6658  031012  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA4(377), RCV DATA5(001)
 6659  031016  000377                  377
 6660  031020  000000                  0
 6661  031022  020010                  NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
 6662  031024  103003                  BCC     .+8.            ;BR IF NO ERROR
 6663  031026                          ERROR                   ;REPORT STACKED ERROR
 6664  031026  104460                                                          TRAP    C$ERROR
 6665  031030                          ESCAPE  TST             ;SKIP TO END OF TEST
 6666  031030  104410                                                          TRAP    C$ESCAPE
 6667  031032  000046                                                          .WORD   L10025-.
 6668
 6669  031034  004537  010350          JSR     R5,RXCHAR       ;READ/CHK DATA5(001), RCV FIRST FLAG
 6670  031040  001001                  RXEOM!001               ; & CHECK REOM
 6671  031042  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
 6672  031044  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
 6673  031046  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
 6674  031050                                 ERROR                    ;REPORT STACKED ERROR
 6675  031050  104460                                                                TRAP    C$ERROR
 6676  031052                                 ESCAPE  TST              ;SKIP TO END OF TEST
 6677  031052  104410                                                                TRAP    C$ESCAPE
 6678  031054  000024                                                                .WORD   L10025-.
 6679                                 ;-----------------------------------------------------------------
 6680  031056  005337  002526                 DEC     REG0             ;DECREMENT COUNT
 6681  031062  001406                         BEQ     40$              ;BR IF TRIPLE LOOP IS COMPLETED
 6682
 6683  031064  004537  010250                 JSR     R5,TXCTRL        ;CLEAR TEOM, SET TSOM
 6684  031070  000001                         TSOM
 6685  031072  000001                         1
 6686  031074  000137  030466                 JMP     2$               ;AND RUN TX/RX AGAIN
 6687  031100                         40$:
 6688  031100                         ENDTST
 6689  031100                                                      L10025:
 6690  031100  104401                                                                TRAP    C$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 DETECTI " 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
                                                                                        T7::
031102                                  MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
031102  012737  000003  002526          JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
031110  004737  005376                  BIC     #NOLOOP,1$      ;
031114  042737  001000  031156          CMP     BRDTYP,#0       ;BOARD TYPE = M8064 ?
031122  023727  002470  000000          BEQ     2$              ; YES: SPECIFY TTLOOP (NOT XLB)
031130  001407                          CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
031132  023727  002472  000004          BEQ     2$              ;BR IF NO
031140  001403                          BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
031142  052737  001000  031156          ;-------------------------------------------------------------------
                                2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
031150  004537  007234                  CRCOS                   ;SET BOP MODE,CRC-CCITT=>0'S CHECK
031154  000400                  1$:     0                       ;USE 8 BIT CHARS
031156  000000                          BCC     .+8.            ;BR IF NO ERROR
031160  103003                          ERROR                   ;REPORT STACKED ERROR
031162                                                                  TRAP    C$ERROR
031162  104460                          ESCAPE  TST             ;SKIP TO END OF TEST
031164                                                                  TRAP    C$ESCAPE
031164  104410                                                          .WORD   L10026-.
031166  000374
                                        JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
031170  004537  010250                  TSOM
031174  000001                          7.
031176  000007                          JSR     R5,TXCTRL       ;LOAD 3RD FLAG, TX 2ND FLAG
031200  004537  010250                  TSOM
031204  000001                          8.
031206  000010                          JSR     R5,TXCTRL       ;LOAD 4TH FLAG, TX 3RD FLAG
031210  004537  010250                  TSOM
031214  000001                          8.
031216  000010                          JSR     R5,TXCTRL       ;CLEAR TSOM
031220  004537  010250                  000
031224  000000                          0
031226  000000                          JSR     R5,TXCHAR       ;LOAD DATA1(125), TX 4TH FLAG
031230  004537  010136                  125
031234  000125                          8.
031236  000010                          BCC     .+8.            ;BR IF NO ERROR
031240  103003
```

```
6747  031242                                ERROR                   ;REPORT STACKED ERROR
6748  031242  104460                                                                  TRAP    C$ERROR
6749  031244                                ESCAPE  TST             ;SKIP TO END OF TEST
6750  031244  104410                                                                  TRAP    C$ESCAPE
6751  031246  000314                                                                  .WORD   L10026-.
6752
6753  031250  004537  010136               JSR     R5,TXCHAR       ;LOAD DATA2(252), TX DATA1(125)
6754  031254  000252                       252
6755  031256  000010                       8.
6756  031260  103003                       BCC     .+8.            ;BR IF NO ERROR
6757  031262                               ERROR                   ;REPORT STACKED ERROR
6758  031262  104460                                                                  TRAP    C$ERROR
6759  031264                               ESCAPE  TST             ;SKIP TO END OF TEST
6760  031264  104410                                                                  TRAP    C$ESCAPE
6761  031266  000274                                                                  .WORD   L10026-.
6762
6763  031270  004537  010136               JSR     R5,TXCHAR       ;LOAD DATA3(000), TX DATA2(252)
6764  031274  000000                       000
6765  031276  000010                       8.
6766  031300  103003                       BCC     .+8.            ;BR IF NO ERROR
6767  031302                               ERROR                   ;REPORT STACKED ERROR
6768  031302  104460                                                                  TRAP    C$ERROR
6769  031304                               ESCAPE  TST             ;SKIP TO END OF TEST
6770  031304  104410                                                                  TRAP    C$ESCAPE
6771  031306  000254                                                                  .WORD   L10026-.
6772
6773  031310  004537  010136               JSR     R5,TXCHAR       ;LOAD DATA4(377), TX DATA3(000)
6774  031314  000377                       377
6775  031316  000010                       8.
6776  031320  103003                       BCC     .+8.            ;BR IF NO ERROR
6777  031322                               ERROR                   ;REPORT STACKED ERROR
6778  031322  104460                                                                  TRAP    C$ERROR
6779  031324                               ESCAPE  TST             ;SKIP TO END OF TEST
6780  031324  104410                                                                  TRAP    C$ESCAPE
6781  031326  000234                                                                  .WORD   L10026-.
6782
6783  031330  004537  010136               JSR     R5,TXCHAR       ;LOAD DATA5(001), TX DATA4(377)
6784  031334  000001                       001
6785  031336  000011                       9.
6786  031340  103003                       BCC     .+8.            ;BR IF NO ERROR
6787  031342                               ERROR                   ;REPORT STACKED ERROR
6788  031342  104460                                                                  TRAP    C$ERROR
6789  031344                               ESCAPE  TST             ;SKIP TO END OF TEST
6790  031344  104410                                                                  TRAP    C$ESCAPE
6791  031346  000214                                                                  .WORD   L10026-.
6792
6793  031350  004537  010250               JSR     R5,TXCTRL       ;SET TEOM
6794  031354  000002                       TEOM
6795  031356  000000                       0
6796  031360  004537  011624               JSR     R5,RCV1ST       ;CLOCK AND RCV DATA1(125)
6797  031364  000000                       0
6798  031366  103003                       BCC     .+8.            ;BR IF NO ERROR
6799  031370                               ERROR                   ;REPORT STACKED ERROR
6800  031370  104460                                                                  TRAP    C$ERROR
6801  031372                               ESCAPE  TST             ;SKIP TO END OF TEST
6802  031372  104410                                                                  TRAP    C$ESCAPE
```

```
                                                                                .WORD   L10026-.
                
                        JSR     R5,RXCHAR       ;READ/CHK DATA1(125), RCV DATA2(252)
                        RXSOM!125               ; & CHECK RSOM=1
                        0
                        8.
                        BCC     .+8.            ;BR IF NO ERROR
                        ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10026-.
                
                        JSR     R5,TXCTRL       ;SET TEOM
                        TEOM
                        0
                        JSR     R5,RXCHAR       ;READ/CHK DATA2(252), RCV DATA3(000)
                        252
                        0
                        8.
                        BCC     .+8.            ;BR IF NO ERROR
                        ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10026-.
                
                        JSR     R5,RXCHAR       ;READ/CHK DATA3(000), RCV DATA4(377)
                        000
                        0
                        8.
                        BCC     .+8.            ;BR IF NO ERROR
                        ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10026-.
                
                        JSR     R5,RXCHAR       ;READ/CHK DATA4(377), RCV DATA5(001)
                        377
                        0
                        NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
                        BCC     .+8.            ;BR IF NO ERROR
                        ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10026-.
                
                        JSR     R5,RXCHAR       ;READ/CHK DATA5(001), RCV FIRST FLAG
                        RXEOM!001               ; & CHECK REOM
                        RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
                        NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
                        BCC     .+8.            ;BR IF NO ERROR
                        ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
```

```
 6859  031534                                 ESCAPE  TST             ;SKIP TO END OF TEST
 6860  031534  104410                                                                 TRAP    C$ESCAPE
 6861  031536  000024                                                                 .WORD   L10026-.
 6862                                 ;-------------------------------------------------------------------
 6863  031540  005337  002526                 DEC     REG0            ;DECREMENT COUNT
 6864  031544  001406                         BEQ     40$             ;BR IF TRIPLE LOOP IS COMPLETED
 6865
 6866  031546  004537  010250                 JSR     R5,TXCTRL       ;CLEAR TEOM, SET TSOM
 6867  031552  000001                         TSOM
 6868  031554  000001                         1
 6869  031556  000137  031150                 JMP     2$              ;AND RUN TX/RX AGAIN
 6870  031562                                 40$:
 6871  031562                                 ENDTST
 6872  031562                                                                 L10026:
 6873  031562  104401                                                                 TRAP    C$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
031564                                                                          T8::
031564  012737  000010  002414          MOV     #8.,TSTNUM      ;SET TEST NO. FOR POSSIBLE PRINTOUT
031572  004537  007532                  JSR     R5,CKLPBK       ;CHK FOR H3254/5 INSTALLED
031576  100000                          TCCHEK
031600  103002                          BCC     2$              ;BR IF YES, TO RUN TEST
031602                                  EXIT    TST             ;NO TEST CONNECTOR, SKIP TEST
031602  104432                                                                  TRAP    C$EXIT
031604  000722                                                                  .WORD   L10027-.
031606                          2$:
031606                                  BGNSUB
031606                                                                  T8.1:
031606  104402                                                                  TRAP    C$BSUB
                                ;INIT DMV, SET TTL LOOPBACK, CHK MODEM STATUS
031610  004737  003352                  JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR TO INIT DMV11
031614  004537  003712                  JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
031620  120002                          VIADPB
031622  000377                          377
031624  004537  003712                  JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
031630  120003                          VIADPA
031632  000000                          000
031634  004537  003712                  JSR     R5,WRITEI       ;SET TTL LOOPBACK
031640  120000                          VIAORB
031642  000002                          TTLOOP
031644  012737  000001  002336          MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
                                ;CHK FOR RING, CARRIER, MODEM RDY, CTS, TEST MODE, = 1
031652  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
031656  120017                          VIAORA
031660  000000                  4$:     .WORD   0
031662  142737  000023  031660          BICB    #SPEED!RCVDAT!UMAINT,4$ ;CLEAR UNNEEDED BITS
031670  123727  031660  000354          CMPB    4$,#RING!CARIER!MDMRDY!CTS!TM ;CHK FOR BITS SET
```

```
6930  031676  001414                          BEQ     8$                ;BR IF ALL BITS SET
6931  031700  012737  000354  002324          MOV     #RING!CARIER!MDMRDY!CTS!TM,GDATA ;SET EXPECTED DATA
6932  031706  013737  031660  002326          MOV     4$,BDATA          ;SET ACTUAL DATA
6933                                ;REPORT "MODEM STATUS INCORRECT"
6934  031714                                  GEDF    EM83,ERR11
6935                                                          ;       "DEVICE FATAL" ERROR # 66
6936  031714  104455                                                                  TRAP    C$ERDF
6937  031716  000102                                                                  .WORD   66
6938  031720  015262                                                                  .WORD   EM83
6939  031722  020540                                                                  .WORD   ERR11
6940  031724                                  ESCAPE  SUB
6941  031724  104410                                                                  TRAP    C$ESCAPE
6942  031726  000106                                                                  .WORD   L10030-.
6943                                ;DE-ASSERT RTS, CHK FOR CTS = 0
6944  031730  004537  003712          8$:     JSR     R5,WRITEI         ;DE-ASSERT RTS
6945  031734  120000                          VIAORB
6946  031736  000012                          RTSND!TTLOOP
6947  031740  004537  003566                  JSR     R5,READI          ;READ MODEM STATUS
6948  031744  120017                          VIAORA
6949  031746  000000                  10$:    .WORD   0
6950  031750  132737  000010  031746          BITB    #CTS,10$          ;CHK FOR CTS = 0
6951  031756  001406                          BEQ     12$               ;BR IF YES
6952                                ;REPORT CTS NOT CLEARED
6953  031760                                  GEDF    EM84,ERR14
6954                                                          ;       "DEVICE FATAL" ERROR # 67
6955  031760  104455                                                                  TRAP    C$ERDF
6956  031762  000103                                                                  .WORD   67
6957  031764  015311                                                                  .WORD   EM84
6958  031766  021146                                                                  .WORD   ERR14
6959  031770                                  ESCAPE  SUB
6960  031770  104410                                                                  TRAP    C$ESCAPE
6961  031772  000042                                                                  .WORD   L10030-.
6962                                ;ASSERT RTS, CHK FOR CTS = 1
6963  031774  004537  003712          12$:    JSR     R5,WRITEI         ;ASSERT RTS
6964  032000  120000                          VIAORB
6965  032002  000002                          TTLOOP
6966  032004  004537  003566                  JSR     R5,READI          ;READ MODEM STATUS
6967  032010  120017                          VIAORA
6968  032012  000000                  14$:    .WORD   0
6969  032014  132737  000010  032012          BITB    #CTS,14$          ;CHK FOR CTS = 1
6970  032022  001004                          BNE     15$               ;BR IF YES
6971                                ;REPORT CTS NOT SET
6972  032024                                  GEDF    EM85,ERR14
6973                                                          ;       "DEVICE FATAL" ERROR # 68
6974  032024  104455                                                                  TRAP    C$ERDF
6975  032026  000104                                                                  .WORD   68
6976  032030  015326                                                                  .WORD   EM85
6977  032032  021146                                                                  .WORD   ERR14
6978  032034                          15$:
6979  032034                                  ENDSUB
6980  032034                                                                  L10030:
6981  032034  104403                                                                  TRAP    C$ESUB
6982                                ;SEE IF BOARD IS M8053 WITH H3254 INSTALLED
6983  032036  005737  002470                  TST     BRDTYP            ;SEE IF M8053
6984  032042  001002                          BNE     17$               ;BR IF YES
6985  032044  000137  032526          16$:    JMP     A1                ;SKIP THIS SECTION OF CODE
```

```
 6986  032050  023727  002472  000000  17$:    CMP     TSTCON,#0       ;SEE IF M3254 INSTALLED
 6987  032056  001372                          BNE     16$             ;BR IF NOT, TO SKIP CODE
 6988  032060                                  BGNSUB
 6989  032060                                                                          T8.2:
 6990  032060  104402                                                                  TRAP    C$BSUB
 6991                                  ;INIT DMV, (TTL LOOPBACK IS CLEARED), CHK MODEM STATUS
 6992  032062  004737  003352                  JSR     PC,MSTCLR       ;PERFORM MASTER CLEAR TO INIT DMV11
 6993  032066  004537  003712                  JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
 6994  032072  120002                          VIAOPB
 6995  032074  000377                          377
 6996  032076  004537  003712                  JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
 6997  032102  120003                          VIAOPA
 6998  032104  000000                          000
 6999  032106  004537  003712                  JSR     R5,WRITEI       ;DISABLE TTLOOP
 7000  032112  120000                          VIAORB
 7001  032114  000000                          000
 7002  032116  012737  000001  002336          MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
 7003                                  ;-----------------------------------------------------------------
 7004                                  ;CHK FOR RING (IF EIA), CARRIER, MODEM RDY, CTS = 1, TEST MODE, = 0
 7005                                  ;-----------------------------------------------------------------
 7006  032124  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
 7007  032130  120017                          VIAORA
 7008  032132  000000                  18$:    .WORD   0
 7009
 7010  032134  023727  002470  000001          CMP     BRDTYP,#1       ;IS V.35 THE SELECTED I/F ?
 7011  032142  001013                          BNE     21$             ; NO: BR TO DO CHECK WITH RING
 7012                                                                  ;YES: REMOVE RING BEFORE CHECKING
 7013  032144  142737  000223  032132          BICB    #RING!SPEED!RCVDAT!UMAINT,18$ ;CLEAR UNNEEDED BITS
 7014  032152  123727  032132  000150          CMPB    18$,#CARIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
 7015  032160  001427                          BEQ     20$             ;BR IF STATUS CORRECT
 7016  032162  012737  000150  002324          MOV     #CARIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
 7017  032170  000412                          BR      19$
 7018                                                                  ; DO CHECK WITH RING....
 7019  032172  142737  000023  032132  21$:    BICB    #SPEED!RCVDAT!UMAINT,18$ ;CLEAR UNNEEDED BITS
 7020  032200  123727  032132  000350          CMPB    18$,#RING!CARIER!MDMRDY!CTS ;CHK FOR CORRECT STATUS
 7021  032206  001414                          BEQ     20$             ;BR IF STATUS CORRECT
 7022  032210  012737  000350  002324          MOV     #RING!CARIER!MDMRDY!CTS,GDATA ;SET EXPECTED DATA
 7023
 7024                                  ;REPORT "MODEM STATUS INCORRECT"
 7025  032216  013737  032132  002326  19$:    MOV     18$,BDATA       ;SET ACTUAL DATA
 7026  032224                                  GEDF    EM83,ERR11
 7027                                                                  ;       "DEVICE FATAL" ERROR # 69
 7028  032224  104455                                                                  TRAP    C$ERDF
 7029  032226  000105                                                                  .WORD   69
 7030  032230  015262                                                                  .WORD   EM83
 7031  032232  020540                                                                  .WORD   ERR11
 7032  032234                                  ESCAPE  SUB
 7033  032234  104410                                                                  TRAP    C$ESCAPE
 7034  032236  000266                                                                  .WORD   L10031-.
 7035                                  ;DE-ASSERT RTS, CHK FOR CTS,CARRIER = 0
 7036  032240  004537  003712          20$:    JSR     R5,WRITEI       ;DE-ASSERT RTS
 7037  032244  120000                          VIAORB
 7038  032246  000010                          RTSMD
 7039  032250  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
 7040  032254  120017                          VIAORA
 7041  032256  000000                  22$:    .WORD   0
```

```
7042  032260  132737  000010  032256          BITB    #CTS,22$        ;CHK FOR CTS = 0
7043  032266  001406                          BEQ     24$             ;BR IF YES
7044                                  ;REPORT CTS NOT CLEARED
7045  032270                                  GEDF    EM84,ERR14
7046                                                  ;       "DEVICE FATAL" ERROR # 70
7047  032270  104455                                                          TRAP    C$ERDF
7048  032272  000106                                                          .WORD   70
7049  032274  015311                                                          .WORD   EM84
7050  032276  021146                                                          .WORD   ERR14
7051  032300                                  ESCAPE  SUB
7052  032300  104410                                                          TRAP    C$ESCAPE
7053  032302  000222                                                          .WORD   L10031-.
7054  032304  132737  000100  032256  24$:    BITB    #CARIER,22$     ;CHK FOR CARRIER = 0
7055  032312  001406                          BEQ     26$             ;BR IF YES
7056                                  ;REPORT CARRIER NOT CLEARED
7057  032314                                  GEDF    EM86,ERR14
7058                                                  ;       "DEVICE FATAL" ERROR # 71
7059  032314  104455                                                          TRAP    C$ERDF
7060  032316  000107                                                          .WORD   71
7061  032320  015342                                                          .WORD   EM86
7062  032322  021146                                                          .WORD   ERR14
7063  032324                                  ESCAPE  SUB
7064  032324  104410                                                          TRAP    C$ESCAPE
7065  032326  000176                                                          .WORD   L10031-.
7066                                  ;ASSERT RTS, CHK FOR CTS,CARRIER = 1
7067  032330  004537  003712          26$:    JSR     R5,WRITEI       ;ASSERT RTS
7068  032334  120000                          VIAORB
7069  032336  000000                          000
7070  032340  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
7071  032344  120017                          VIAORA
7072  032346  000000                  28$:    .WORD   0
7073  032350  132737  000010  032346          BITB    #CTS,28$        ;CHK FOR CTS = 1
7074  032356  001006                          BNE     30$             ;BR IF YES
7075                                  ;REPORT CTS NOT SET
7076  032360                                  GEDF    EM85,ERR14
7077                                                  ;       "DEVICE FATAL" ERROR # 72
7078  032360  104455                                                          TRAP    C$ERDF
7079  032362  000110                                                          .WORD   72
7080  032364  015326                                                          .WORD   EM85
7081  032366  021146                                                          .WORD   ERR14
7082  032370                                  ESCAPE  SUB
7083  032370  104410                                                          TRAP    C$ESCAPE
7084  032372  000132                                                          .WORD   L10031-.
7085  032374  132737  000100  032346  30$:    BITB    #CARIER,28$     ;CHK FOR CARRIER = 1
7086  032402  001006                          BNE     32$             ;BR IF YES
7087                                  ;REPORT CARRIER NOT SET
7088  032404                                  GEDF    EM87,ERR14
7089                                                  ;       "DEVICE FATAL" ERROR # 73
7090  032404  104455                                                          TRAP    C$ERDF
7091  032406  000111                                                          .WORD   73
7092  032410  015363                                                          .WORD   EM87
7093  032412  021146                                                          .WORD   ERR14
7094  032414                                  ESCAPE  SUB
7095  032414  104410                                                          TRAP    C$ESCAPE
7096  032416  000106                                                          .WORD   L10031-.
7097                                  ;DE-ASSERT DTR, CHK FOR MODEM READY = 0
```

```
 7098  032420  004537  003712          32$:    JSR     R5,WRITEI       ;DE-ASSERT DTR
 7099  032424  120000                          VIAORB
 7100  032426  000020                          DTR
 7101  032430  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
 7102  032434  120017                          VIAORA
 7103  032436  000000                  34$:    .WORD   0
 7104  032440  132737  000040  032436          BITB    #MDMRDY,34$     ;CHK FOR MODEM READY = 0
 7105  032446  001406                          BEQ     36$             ;BR IF YES
 7106                                  ;REPORT MODEM READY NOT CLEARED
 7107  032450                                  GEDF    EM88,ERR14
 7108                                                                  ;       "DEVICE FATAL" ERROR # 74
 7109  032450  104455                                                                  TRAP    C$ERDF
 7110  032452  000112                                                                  .WORD   74
 7111  032454  015403                                                                  .WORD   EM88
 7112  032456  021146                                                                  .WORD   ERR14
 7113  032460                          ESCAPE  SUB
 7114  032460  104410                                                                  TRAP    C$ESCAPE
 7115  032462  000042                                                                  .WORD   L10031-.
 7116                                  ;ASSERT DTR, CHK FOR MODEM READY = 1
 7117  032464  004537  003712          36$:    JSR     R5,WRITEI       ;ASSERT DTR
 7118  032470  120000                          VIAORB
 7119  032472  000000                          000
 7120  032474  004537  003566                  JSR     R5,READI        ;READ MODEM STATUS
 7121  032500  120017                          VIAORA
 7122  032502  000000                  38$:    .WORD   0
 7123  032504  132737  000040  032502          BITB    #MDMRDY,38$     ;CHK FOR MODEM READY = 1
 7124  032512  001004                          BNE     40$             ;BR IF YES
 7125                                  ;REPORT MODEM READY NOT SET
 7126  032514                                  GEDF    EM89,ERR14
 7127                                                                  ;       "DEVICE FATAL" ERROR # 75
 7128  032514  104455                                                                  TRAP    C$ERDF
 7129  032516  000113                                                                  .WORD   75
 7130  032520  015426                                                                  .WORD   EM89
 7131  032522  021146                                                                  .WORD   ERR14
 7132  032524                          40$:
 7133  032524                          ENDSUB
 7134  032524                                                                  L10031.
 7135  032524  104403                                                                  TRAP    C$ESUB
 7136  032526                          A1:
 7137  032526                          ENDTST
 7138  032526                                                                  L10027:
 7139  032526  104401                                                                  TRAP    C$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
                                                                                        T9::
032530                          
032530  012737  000003  002526        MOV     #3,REG0         ;INIT COUNT (TEXT TRANSMITTED 3 TIMES)
032536  004737  005376                JSR     PC,INIDMV       ;INIT DMV-11, ENTER MAINT LOOP
032542  042737  001000  032604        BIC     #NOLOOP,1$      ;
032550  023727  002470  000000        CMP     BRDTYP,#0       ;BOARD TYPE = M8^64 ?
032556  001407                        BEQ     2$              ; YES: SPECIFY ;TLOOP (NOT XLB)
032560  023727  002472  000004        CMP     TSTCON,#4       ;IS A LOOPBACK CONNECTOR/CABLE SPECIFIED ?
032566  001403                        BEQ     2$              ;BR IF NO
032570  052737  001000  032604        BIS     #NOLOOP,1$      ; YES: SPECIFY NO TTLOOP (INITRN)
                                ;-----------------------------------------------------------------------
032576  004537  007234        2$:     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
032602  065626                        DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
032604  000000                1$:     0                       ;USE 8 BIT CHARS
032606  103003                        BCC     .+8.            ;BR IF NO ERROR
032610                                ERROR                   ;REPORT STACKED ERROR
032610  104460                                                                  TRAP    C$ERROR
032612                                ESCAPE  TST             ;SKIP TO END OF TEST
032612  104410                                                                  TRAP    C$ESCAPE
032614  001032                                                                  .WORD   L10032-.
                                
032616  004537  010250                JSR     R5,TXCTRL       ;SET TSOM, TX 1ST SYNCH
032622  000001                        TSOM
032624  000007                        7.
032626  004537  010250                JSR     R5,TXCTRL       ;TX 2ND SYNCH
032632  000001                        TSOM
032634  000010                        8.
032636  004537  010250                JSR     R5,TXCTRL       ;CLEAR TSOM
032642  000000                        000
032644  000000                        0
```

```
7196  032646  004537  010136          JSR     R5,TXCHAR       ;LOAD 201(HEADR1), TX 3RD SYNCH
7197  032652  000201                  201
7198  032654  000010                  8.
7199  032656  103003                  BCC     .+8.            ;BR IF NO ERROR
7200  032660                          ERROR                   ;REPORT STACKED ERROR
7201  032660  104460                                                  TRAP    C$ERROR
7202  032662                          ESCAPE  TST             ;SKIP TO END OF TEST
7203  032662  104410                                                  TRAP    C$ESCAPE
7204  032664  000762                                                  .WORD   L10032-.
7205
7206  032666  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(HEADR2), TX HEADR1
7207  032672  000000                  000
7208  032674  000010                  8.
7209  032676  103003                  BCC     .+8.            ;BR IF NO ERROR
7210  032700                          ERROR                   ;REPORT STACKED ERROR
7211  032700  104460                                                  TRAP    C$ERROR
7212  032702                          ESCAPE  TST             ;SKIP TO END OF TEST
7213  032702  104410                                                  TRAP    C$ESCAPE
7214  032704  000742                                                  .WORD   L10032-.
7215
7216  032706  004537  010136          JSR     R5,TXCHAR       ;LOAD 075(HEADR3), TX HEADR2
7217  032712  000075                  075
7218  032714  000010                  8.
7219  032716  103003                  BCC     .+8.            ;BR IF NO ERROR
7220  032720                          ERROR                   ;REPORT STACKED ERROR
7221  032720  104460                                                  TRAP    C$ERROR
7222  032722                          ESCAPE  TST             ;SKIP TO END OF TEST
7223  032722  104410                                                  TRAP    C$ESCAPE
7224  032724  000722                                                  .WORD   L10032-.
7225
7226  032726  004537  010136          JSR     R5,TXCHAR       ;LOAD 003(HEADR4), TX HEADR3
7227  032732  000003                  003
7228  032734  000010                  8.
7229  032736  103003                  BCC     .+8.            ;BR IF NO ERROR
7230  032740                          ERROR                   ;REPORT STACKED ERROR
7231  032740  104460                                                  TRAP    C$ERROR
7232  032742                          ESCAPE  TST             ;SKIP TO END OF TEST
7233  032742  104410                                                  TRAP    C$ESCAPE
7234  032744  000702                                                  .WORD   L10032-.
7235
7236  032746  004537  010136          JSR     R5,TXCHAR       ;LOAD 002(HEADR5)
7237  032752  000002                  002
7238  032754  000000                  0
7239  032756  103003                  BCC     .+8.            ;BR IF NO ERROR
7240  032760                          ERROR                   ;REPORT STACKED ERROR
7241  032760  104460                                                  TRAP    C$ERROR
7242  032762                          ESCAPE  TST             ;SKIP TO END OF TEST
7243  032762  104410                                                  TRAP    C$ESCAPE
7244  032764  000662                                                  .WORD   L10032-.
7245
7246  032766  004537  011624          JSR     R5,RCV1ST       ;CLOCK AND RCV 201
7247  032772  000000                  0
7248  032774  103003                  BCC     .+8.            ;BR IF NO ERROR
7249  032776                          ERROR                   ;REPORT STACKED ERROR
7250  032776  104460                                                  TRAP    C$ERROR
7251  033000                          ESCAPE  TST             ;SKIP TO END OF TEST
```

```
7252  033000  104410                                                          TRAP    C$ESCAPE
7253  033002  000644                                                          .WORD   L10032-.
7254
7255  033004  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 201(HEADR1), RCV HEADR2
7256  033010  000201                  201
7257  033012  000000                  0
7258  033014  000010                  8.
7259  033016  103003                  BCC     .+8.            ;BR IF NO ERROR
7260  033020                          ERROR                   ;REPORT STACKED ERROR
7261  033020  104460                                                          TRAP    C$ERROR
7262  033022                          ESCAPE  TST             ;SKIP TO END OF TEST
7263  033022  104410                                                          TRAP    C$ESCAPE
7264  033024  000622                                                          .WORD   L10032-.
7265
7266  033026  004537  010136          JSR     R5,TXCHAR       ;LOAD 001(HEADR6)
7267  033032  000001                  001
7268  033034  000000                  0
7269  033036  103003                  BCC     .+8.            ;BR IF NO ERROR
7270  033040                          ERROR                   ;REPORT STACKED ERROR
7271  033040  104460                                                          TRAP    C$ERROR
7272  033042                          ESCAPE  TST             ;SKIP TO END OF TEST
7273  033042  104410                                                          TRAP    C$ESCAPE
7274  033044  000602                                                          .WORD   L10032-.
7275
7276  033046  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 000(HEADR2), RCV HEADR3
7277  033052  000000                  000
7278  033054  000000                  0
7279  033056  000010                  8.
7280  033060  103003                  BCC     .+8.            ;BR IF NO ERROR
7281  033062                          ERROR                   ;REPORT STACKED ERROR
7282  033062  104460                                                          TRAP    C$ERROR
7283  033064                          ESCAPE  TST             ;SKIP TO END OF TEST
7284  033064  104410                                                          TRAP    C$ESCAPE
7285  033066  000560                                                          .WORD   L10032-.
7286
7287  033070  004537  010250          JSR     R5,TXCTRL       ;SET TEOM
7288  033074  000002                  TEOM                    ;(STARTS CRC-16 CHARACTER)
7289  033076  000000                  0
7290  033100  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 075(HEADR3), RCV HEADR4
7291  033104  000075                  075
7292  033106  000000                  0
7293  033110  000010                  8.
7294  033112  103003                  BCC     .+8.            ;BR IF NO ERROR
7295  033114                          ERROR                   ;REPORT STACKED ERROR
7296  033114  104460                                                          TRAP    C$ERROR
7297  033116                          ESCAPE  TST             ;SKIP TO END OF TEST
7298  033116  104410                                                          TRAP    C$ESCAPE
7299  033120  000526                                                          .WORD   L10032-.
7300
7301  033122  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 003(HEADR4), RCV HEADR5
7302  033126  000003                  003
7303  033130  000000                  0
7304  033132  000010                  8.
7305  033134  103003                  BCC     .+8.            ;BR IF NO ERROR
7306  033136                          ERROR                   ;REPORT STACKED ERROR
7307  033136  104460                                                          TRAP    C$ERROR
```

```
7308  033140                          ESCAPE  TST             ;SKIP TO END OF TEST
7309  033140  104410                                                  TRAP    C$ESCAPE
7310  033142  000504                                                  .WORD   L10032-.
7311
7312  033144  004537  010250          JSR     R5,TXCTRL       ;CLEAR TEOM
7313  033150  000000                  000
7314  033152  000000                  0
7315                                ;-----------------------------------------------------------------
7316  033154  004537  010136          JSR     R5,TXCHAR       ;LOAD 000(DATA1)
7317  033160  000000                  000
7318  033162  000000                  0
7319  033164  103003                  BCC     .+8.            ;BR IF NO ERROR
7320  033166                          ERROR                   ;REPORT STACKED ERROR
7321  033166  104460                                                  TRAP    C$ERROR
7322  033170                          ESCAPE  TST             ;SKIP TO END OF TEST
7323  033170  104410                                                  TRAP    C$ESCAPE
7324  033172  000454                                                  .WORD   L10032-.
7325
7326  033174  004537  010350          JSR     R5,RXCHAR       ;READ & CHK 002(HEADR5), RCV HEADR6
7327  033200  000002                  002
7328  033202  000000                  0
7329  033204  000010                  8.
7330  033206  103003                  BCC     .+8.            ;BR IF NO ERROR
7331  033210                          ERROR                   ;REPORT STACKED ERROR
7332  033210  104460                                                  TRAP    C$ERROR
7333  033212                          ESCAPE  TST             ;SKIP TO END OF TEST
7334  033212  104410                                                  TRAP    C$ESCAPE
7335  033214  000432                                                  .WORD   L10032-.
7336
7337  033216  004537  010136          JSR     R5,TXCHAR       ;LOAD 377(DATA2)
7338  033222  000377                  377
7339  033224  000000                  0
7340  033226  103003                  BCC     .+8.            ;BR IF NO ERROR
7341  033230                          ERROR                   ;REPORT STACKED ERROR
7342  033230  104460                                                  TRAP    C$ERROR
7343  033232                          ESCAPE  TST             ;SKIP TO END OF TEST
7344  033232  104410                                                  TRAP    C$ESCAPE
7345  033234  000412                                                  .WORD   L10032-.
7346
7347  033236  004537  010350          JSR     R5,RXCHAR       ;READ/CHK 001(HEADR6), "RCV" FIRST CRC BYTE
7348  033242  100001                  RXERR!001
7349  033244  000001                  RERCHK
7350  033246  000010                  8.
7351  033250  103003                  BCC     .+8.            ;BR IF NO ERROR
7352  033252                          ERROR                   ;REPORT STACKED ERROR
7353  033252  104460                                                  TRAP    C$ERROR
7354  033254                          ESCAPE  TST             ;SKIP TO END OF TEST
7355  033254  104410                                                  TRAP    C$ESCAPE
7356  033256  000370                                                  .WORD   L10032-.
7357
7358  033260  004537  010136          JSR     R5,TXCHAR       ;LOAD 376(DATA3)
7359  033264  000376                  376
7360  033266  000000                  0
7361  033270  103003                  BCC     .+8.            ;BR IF NO ERROR
7362  033272                          ERROR                   ;REPORT STACKED ERROR
7363  033272  104460                                                  TRAP    C$ERROR
```

```
7364  033274                                ESCAPE  TST             ;SKIP TO END OF TEST
7365  033274  104410                                                        TRAP    C$ESCAPE
7366  033276  000350                                                        .WORD   L10032-.
7367
7368  033300  004537  010350               JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
7369  033304  000043                       043
7370  033306  000000                       0
7371  033310  000010                       8.
7372  033312  103003                       BCC     .+8.            ;BR IF NO ERROR
7373  033314                               ERROR                   ;REPORT STACKED ERROR
7374  033314  104460                                                        TRAP    C$ERROR
7375  033316                               ESCAPE  TST             ;SKIP TO END OF TEST
7376  033316  104410                                                        TRAP    C$ESCAPE
7377  033320  000326                                                        .WORD   L.10032-.
7378
7379  033322  004537  010136               JSR     R5,TXCHAR       ;LOAD 375(DATA4)
7380  033326  000375                       375
7381  033330  000000                       0
7382  033332  103003                       BCC     .+8.            ;BR IF NO ERROR
7383  033334                               ERROR                   ;REPORT STACKED ERROR
7384  033334  104460                                                        TRAP    C$ERROR
7385  033336                               ESCAPE  TST             ;SKIP TO END OF TEST
7386  033336  104410                                                        TRAP    C$ESCAPE
7387  033340  000306                                                        .WORD   L10032-.
7388
7389  033342  004537  010350               JSR     R5,RXCHAR       ;READ & CHK SECOND CRC BYTE; RCV DATA1
7390  033346  000035                       035
7391  033350  000000                       0
7392  033352  000010                       8.
7393  033354  103003                       BCC     .+8.            ;BR IF NO ERROR
7394  033356                               ERROR                   ;REPORT STACKED ERROR
7395  033356  104460                                                        TRAP    C$ERROR
7396  033360                               ESCAPE  TST             ;SKIP TO END OF TEST
7397  033360  104410                                                        TRAP    C$ESCAPE
7398  033362  000264                                                        .WORD   L10032-.
7399
7400                                ;-----------------------------------------------------------------
7401                                ; TRANSMIT THE BULK OF DATA OUT OF TABLE "PATK"
7402                                ;-----------------------------------------------------------------
7403  033364  012702  003012               MOV     #PATK,R2        ;SET UP TABLE POINTER
7404  033370  112237  033426       5$:     MOVB    (R2)+,20$       ;SET UP EXPECTED RX CHARACTER
7405  033374  116237  000003  033406       MOVB    3(R2),10$       ;SET UP TRANSMIT CHARACTER
7406
7407  033402  004537  010136               JSR     R5,TXCHAR       ;LOAD A CHARACTER
7408  033406  000000                10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
7409  033410  000000                        0
7410  033412  103003                        BCC     .+8.            ;BR IF NO ERROR
7411  033414                                ERROR                   ;REPORT STACKED ERROR
7412  033414  104460                                                        TRAP    C$ERROR
7413  033416                                ESCAPE  TST             ;SKIP TO END OF TEST
7414  033416  104410                                                        TRAP    C$ESCAPE
7415  033420  000226                                                        .WORD   L10032-.
7416
7417  033422  004537  010350               JSR     R5,RXCHAR       ;CLK/RECEIVE/CHECK PREVIOUS CHARACTER
7418  033426  000000                20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
7419  033430  000000                        0
```

```
 7420  033432  000010                    8.
 7421  033434  103003                    BCC     .+8.            ;BR IF NO ERROR
 7422  033436                            ERROR                   ;REPORT STACKED ERROR
 7423  033436  104460                                                    TRAP    C$ERROR
 7424  033440                            ESCAPE  TST             ;SKIP TO END OF TEST
 7425  033440  104410                                                    TRAP    C$ESCAPE
 7426  033442  000204                                                    .WORD   L10032-.
 7427
 7428  033444  022702  003027            CMP     @PATK+13.,R2    ;CHECK FOR 14TH CHARACTER OF TABLE
 7429  033450  001347                    BNE     5$              ;BR IF NOT DONE
 7430                                  ;-----------------------------------------------------------------
 7431  033452  004537  010250            JSR     R5,TXCTRL       ;SET TEOM
 7432  033456  000002                    TEOM
 7433  033460  000000                    0
 7434  033462  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 200(DATA13), RCV 125(DATA14)
 7435  033466  000200                    200
 7436  033470  000000                    0
 7437  033472  000010                    8.
 7438  033474  103003                    BCC     .+8.            ;BR IF NO ERROR
 7439  033476                            ERROR                   ;REPORT STACKED ERROR
 7440  033476  104460                                                    TRAP    C$ERROR
 7441  033500                            ESCAPE  TST             ;SKIP TO END OF TEST
 7442  033500  104410                                                    TRAP    C$ESCAPE
 7443  033502  000144                                                    .WORD   L10032-.
 7444
 7445  033504  004537  010250            JSR     R5,TXCTRL       ;SET TEOM
 7446  033510  000002                    TEOM
 7447  033512  000000                    0
 7448  033514  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 125(DATA14), RCV 252(DATA15)
 7449  033520  000125                    125
 7450  033522  000000                    0
 7451  033524  000010                    8.
 7452  033526  103003                    BCC     .+8.            ;BR IF NO ERROR
 7453  033530                            ERROR                   ;REPORT STACKED ERROR
 7454  033530  104460                                                    TRAP    C$ERROR
 7455  033532                            ESCAPE  TST             ;SKIP TO END OF TEST
 7456  033532  104410                                                    TRAP    C$ESCAPE
 7457  033534  000112                                                    .WORD   L10032-.
 7458
 7459  033536  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 252(DATA15), RCV 000(DATA16)
 7460  033542  000252                    252
 7461  033544  000000                    0
 7462  033546  000010                    8.
 7463  033550  103003                    BCC     .+8.            ;BR IF NO ERROR
 7464  033552                            ERROR                   ;REPORT STACKED ERROR
 7465  033552  104460                                                    TRAP    C$ERROR
 7466  033554                            ESCAPE  TST             ;SKIP TO END OF TEST
 7467  033554  104410                                                    TRAP    C$ESCAPE
 7468  033556  000070                                                    .WORD   L10032-.
 7469
 7470  033560  004537  010350            JSR     R5,RXCHAR       ;READ/CHK 000(DATA16), RCV FIRST CRC BYTE
 7471  033564  100000                    RXERR!000
 7472  033566  000001                    RERCHK
 7473  033570  000010                    8.
 7474  033572  103003                    BCC     .+8.            ;BR IF NO ERROR
 7475  033574                            ERROR                   ;REPORT STACKED ERROR
```

```
                                                                                TRAP    C$ERROR
                        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10032-.

                        JSR     R5,RXCHAR       ;READ & CHK 1ST CRC BYTE, RCV SECOND CRC BYTE
                        231
                        0
                        8.
                        BCC     .+8.            ;BR IF NO ERROR
                        ERROR                   ;REPORT STACKED ERROR
                                                                                TRAP    C$ERROR
                        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10032-.

                        JSR     R5,RXCHAR       ;READ & CHK 2ND CRC BYTE, RCV 1ST SYNCH
                        176
                        0
                        8.
                        BCC     .+8.            ;BR IF NO ERROR
                        ERROR                   ;REPORT STACKED ERROR
                                                                                TRAP    C$ERROR
                        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
                                                                                .WORD   L10032-.
                ENDTST
                                                                        L10032:
                                                                                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.
                                ;////////////////////////////////////////////////////////////////////////

033650                                  BGNHRD
033650  000027                                                        .WORD L10033-L$HARD/2
033652                                                          L$HARD::

033652                                  GPRMA   ADDRES,0,0,160020,177776,YES
033652  000031                                                          .WORD   T$CODE
033654  033730                                                          .WORD   ADDRES
033656  160020                                                          .WORD   T$LOLIM
033660  177776                                                          .WORD   T$HILIM
033662                                  GPRMA   VECTOR,2,0,0,674,YES
033662  001031                                                          .WORD   T$CODE
033664  033756                                                          .WORD   VECTOR
033666  000000                                                          .WORD   T$LOLIM
033670  000674                                                          .WORD   T$HILIM
033672                                  GPRMD   PRIRTY,4,0,7000,4,7,YES
033672  002032                                                          .WORD   T$CODE
033674  034007                                                          .WORD   PRIRTY
033676  007000                                                          .WORD   7000
033700  000004                                                          .WORD   T$LOLIM
033702  000007                                                          .WORD   T$HILIM
033704                                          GPRMD   BDTY.M,12,0,7,0,2,YES
033704  005032                                                          .WORD   T$CODE
033706  034040                                                          .WORD   BDTY.M
033710  000007                                                          .WORD   7
033712  000000                                                          .WORD   T$LOLIM
033714  000002                                                          .WORD   T$HILIM
033716                                          GPRMD   TCON.M,14,0,7,0,4,YES
033716  006032                                                          .WORD   T$CODE
033720  034123                                                          .WORD   TCON.M
033722  000007                                                          .WORD   7
033724  000000                                                          .WORD   T$LOLIM
033726  000004                                                          .WORD   T$HILIM

033730                                  ENDHRD
                                                                .EVEN
033730                                                          L10033:

                                .NLIST  BEX
```

```
        033730  042504  044526  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
        033756  042504  044526  042503  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
        034007     104  053105  041511  PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
        034040  047502  051101  020104  BDTY.M: .ASCIZ  /BOARD TYPE (0=M8064, 1=M8053-V.35, 2=M8053-EIA) : /
        034123     124  051125  040516  TCON.M: .ASCII  /TURNAROUND CONNECTOR TYPE -/<15><12>
        034160  03005  044075  031063          .ASCII  /(0=H3254&H3255, 1=INTEGRAL MODEM CABLE, 2=EIA CABLE,/<15><12>
        034246  03144  053075  031456          .ASCIZ  / 3=V.35 CABLE, 4=NONE) : /
                                        .LIST   BEX
  7556                                  .EVEN
```

```
                                SOFTWARE PARAMETER CODING SECTION

 7557                           .SBTTL  SOFTWARE PARAMETER CODING SECTION
 7558
 7559
 7560                           ;////////////////////////////////////////////////////////////////////////
 7561                           ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 7562                           ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 7563                           ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 7564                           ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 7565                           ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 7566                           ;/ WITH THE OPERATOR.
 7567                           ;////////////////////////////////////////////////////////////////////////
 7568
 7569  034300                          BGNSFT
 7570  034300  000000                                                          .WORD L10034-L$SOFT/2
 7571  034302                                                          L$SOFT::
 7572
 7573  034302                          ENDSFT
 7574                                                                   .EVEN
 7575  034302                                                          L10034:
```

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

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

034302                                  PATCH:
        034502                                  .=.+200
034502  000240                                  NOP
034504  000240                                  NOP
034506  000240                                  NOP
                                        ;**********************************

                                        .SBTTL  "ENDMOD" STATEMENT

                                                ENDMOD

034510                                  .SBTTL  "LASTAD" STATEMENT & END OF PROGRAM
034510                                          LASTAD
                                                                                .EVEN
034510  000000                                                                  .WORD   0
034512  000000                                                                  .WORD   0
034514                                  L$LAST::

        000001                          .END
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
ABC    = 000160           1269#
ADDRES  033730            7524    7555#
ADR    = 000020 G         1201#
AD.HIT  023214            4538    4573#
AD.OK   023210            4562    4567#
APA    = 000200           1325#
APAD   = 100000           1334#
ASSEMB= 000010             959
A1      032526            6985    7136#
BADDAT  002366            1720#
BDATA   002326            1703#   2356*   2359    2362    2636*   2739*   2740*   3507*   3508*   3525*   3526*   3919    3968
                          4013    4084    4146    4805*   4806*   4888*   5013*   5042*   5071*   5100*   5129*   5171*   5200*
                          6932*   7025*
BDRATE  002474            1767#   4506*
BDTY.M  034040            7540    7555#
BIT0   = 000001 G         1174#   1219    1231    1273    1282    1301    1330    1356    1372    1390    1406    1548    1562
                          1584    1637    2770    2815    2860    2905    2950    2993    3040    3107    4472
BIT00  = 000001 G         1163#   1174
BIT01  = 000002 G         1162#   1173
BIT02  = 000004 G         1161#   1172
BIT03  = 000010 G         1160#   1171
BIT04  = 000020 G         1159#   1170
BIT05  = 000040 G         1158#   1169
BIT06  = 000100 G         1157#   1168
BIT07  = 000200 G         1156#   1167
BIT08  = 000400 G         1155#   1166
BIT09  = 001000 G         1154#   1165
BIT1   = 000002 G         1173#   1272    1300    1330    1356    1371    1389    1405    1547    1561    1583    1636    3064
BIT10  = 002000 G         1153#   1278    1306    1341    1342    1343
BIT11  = 004000 G         1152#   1277    1305    1338
BIT12  = 010000 G         1151#   1337
BIT13  = 020000 G         1150#   1336    1646
BIT14  = 040000 G         1149#   1335    1639    1645
BIT15  = 100000 G         1148#   1276    1304    1334    1633    1644    5830    6090    6315
BIT2   = 000004 G         1172#   1271    1299    1330    1356    1370    1388    1404    1536    1561    1582
BIT3   = 000010 G         1171#   1270    1298    1329    1355    1369    1387    1403    1536    1561    1581
BIT4   = 000020 G         1170#   1218    1269    1328    1354    1368    1386    1402    1536    1560    1580
BIT5   = 000040 G         1169#   1269    1327    1353    1367    1385    1401    1531    1559    1579    3649
BIT6   = 000100 G         1168#   1230    1269    1326    1353    1366    1384    1400    1521    1559    1578    3669
BIT7   = 000200 G         1167#   1229    1237    1268    1297    1325    1353    1365    1383    1399    1521    1559    1572
                          1594    1642    3486    3487    3658
BIT8   = 000400 G         1166#   1280    1308    1339    1340    1341    1342
BIT9   = 001000 G         1165#   1279    1307    1340    1341    1640
BOE    = 000400 G         1205#
BRDTYP  002470            1765#   3267    3278    3286    3296    4504*   5825    6085    6310    6530    6715    6983    7010
                          7171
BSEL0   002416            1734#   2298    4545    4551
BSEL1   002420            1737#   2094*   2299
BSEL10  002436            1747#   2306
BSEL11  002440            1749#   2307
BSEL12  002442            1750#   2308
BSEL13  002444            1752#   2309
BSEL14  002446            1753#   2310
BSEL15  002450            1755#   2311
BSEL16  002452            1756#   2312
BSEL17  002454            1758#   2313    4494
```

```
BSEL2   002422          1738#  2101   2137*  2144   2184*  2191   2256*  2263   2300   4862*  4868
BSEL3   002424          1740#  2301
BSEL4   002426          1741#  2302
BSEL5   002430          1743#  2303
BSEL6   002432          1744#  2159   2304   4883   4890   4903   4916   4929   4943   4956   4969   4982   4995
                        5008   5024   5037   5053   5066   5082   5095   5111   5124   5140   5153   5166   5182
                        5195   5211   5224
BSEL7   002434          1746#  2305   5013   5042   5071   5100   5129   5171   5200
BSR0    002202          1670#  2298*  4168
BSR1    002204          1672#  2299*  4167
BSR10   002222          1685#  2306*  4202
BSR11   002224          1686#  2307*  4201
BSR12   002226          1687#  2308*  4200
BSR13   002230          1688#  2309*  4199
BSR14   002232          1689#  2310*  4219
BSR15   002234          1690#  2311*  4218
BSR16   002236          1691#  2312*  4217
BSR17   002240          1692#  2313*  4216
BSR2    002206          1674#  2300*  4166
BSR3    002210          1676#  2301*  4165
BSR4    002212          1678#  2302*  4185
BSR5    002214          1680#  2303*  4184
BSR6    002216          1682#  2304*  4183
BSR7    002220          1684#  2305*  4182
CARIER= 000100          1400#  6929   6931   7014   7016   7020   7022   7054   7085
CA1CTL= 000001          1562#
CA2CTL= 000016          1561#
CB1CTL= 000020          1560#
CB2CTL= 000340          1559#
CHKTSO  007074          3102#
CHPTYP  002400          1725#  4421*
CXLPBK  007532          3246#  4774   6902
CXRACT  005654          2810#  3651   3705   3732   3764   3775
CXRDA   006154          2900#  3660   3671   3708   3724   3729
CXROR   006454          2988#
CXRSA   006314          2945#
CXSEOM  006614          3035#  3711
CXTACT  005514          2765#  3217   3383   3435   3761   3772
CXTBMT  006014          2855#  3200   3214   3392   3399   3440   3447   4850
CXUSTS  005410          2730#  3183   4836
CRCOS = 000400          1339#  6722
CRC16 = 001400          1340#  4831   5833   7178
CTS   = 000010          1403#  6929   6931   6950   6969   7014   7016   7020   7022   7042   7073
C$AU  = 000052           959#  4617
C$AUTO= 000061           959#  4571
C$BRK = 000022           959#
C$BSEG= 000004           959#
C$BSUB= 000002           959#  6911   6990
C$CEFG= 000045           959#
C$CLCK= 000062           959#
C$CLEA= 000012           959#  4589
C$CLOS= 000035           959#
C$CLP1= 000006           959#
C$CVEC= 000036           959#  4560
C$DCLN= 000044           959#
C$DODU= 000051           959#  4565
```

```
C$DRPT= 000024   9590#
C$DU  = 000053   9590#  4604#
C$EDIT= 000003   9590#  1031#
C$ERDF= 000055   9590#  4810#  4875#  4895#  4908#  4921#  4935#  4948#  4961#  4974#  4987#  5000#  5016#
                 5029#  5045#  5058#  5074#  5087#  5103#  5116#  5132#  5145#  5158#  5174#  5187#  5203#
                 5216#  5230#  5241#  6936#  6955#  6974#  7028#  7047#  7059#  7078#  7090#  7109#  7128#
C$ERHR= 000056   9590#
C$ERRO= 000060   9590#  4643#  4653#  4675#  4687#  4700#  4710#  4723#  4734#  4745#  4840#  4854#  5837#
                 5847#  5857#  5871#  5881#  5891#  5901#  5911#  5920#  5931#  5947#  5958#  5975#  5989#
                 6000#  6011#  6022#  6033#  6050#  6098#  6108#  6118#  6132#  6142#  6152#  6161#  6172#
                 6188#  6199#  6212#  6223#  6233#  6244#  6255#  6266#  6275#  6322#  6332#  6342#  6356#
                 6366#  6376#  6385#  6396#  6412#  6423#  6436#  6447#  6457#  6468#  6479#  6490#  6499#
                 6541#  6563#  6573#  6583#  6593#  6603#  6616#  6627#  6642#  6653#  6664#  6675#  6726#
                 6748#  6758#  6768#  6778#  6788#  6800#  6811#  6825#  6836#  6847#  6858#  7182#  7201#
                 7211#  7221#  7231#  7241#  7250#  7261#  7271#  7282#  7296#  7307#  7321#  7332#  7342#
                 7353#  7363#  7374#  7384#  7395#  7412#  7423#  7440#  7454#  7465#  7476#  7487#  7498#
C$ERSF= 000054   9590#
C$ERSO= 000057   9590#
C$ESCA= 000010   9590#  4645#  4655#  4677#  4689#  4702#  4712#  4725#  4736#  4747#  4815#  4842#  4856#
                 4880#  4900#  4913#  4926#  4940#  4953#  4966#  4979#  4992#  5005#  5021#  5034#  5050#
                 5063#  5079#  5092#  5108#  5121#  5137#  5150#  5163#  5179#  5192#  5208#  5221#  5235#
                 5839#  5849#  5859#  5873#  5883#  5893#  5903#  5913#  5922#  5933#  5949#  5960#  5977#
                 5991#  6002#  6013#  6024#  6035#  6052#  6100#  6110#  6120#  6134#  6144#  6154#  6163#
                 6174#  6190#  6201#  6214#  6225#  6235#  6246#  6257#  6268#  6277#  6324#  6334#  6344#
                 6358#  6368#  6378#  6387#  6398#  6414#  6425#  6438#  6449#  6459#  6470#  6481#  6492#
                 6501#  6543#  6565#  6575#  6585#  6595#  6605#  6618#  6629#  6644#  6655#  6666#  6677#
                 6728#  6750#  6760#  6770#  6780#  6790#  6802#  6813#  6827#  6838#  6849#  6860#  6941#
                 6960#  7033#  7052#  7064#  7083#  7095#  7114#  7184#  7203#  7213#  7223#  7233#  7243#
                 7252#  7263#  7273#  7284#  7298#  7309#  7323#  7334#  7344#  7355#  7365#  7376#  7386#
                 7397#  7414#  7425#  7442#  7456#  7467#  7478#  7489#  7500#
C$ESEG= 000005   9590#
C$ESUB= 000003   9590#  6981#  7135#
C$ETST= 000001   9590#  4751#  5249#  6056#  6281#  6505#  6690#  6873#  7139#  7504#
C$EXIT= 000032   9590#  4778#  6906#
C$GETB= 000026   9590#
C$GETW= 000027   9590#
C$GMAN= 000043   9590#
C$GPHR= 000042   9590#  4480#
C$GPLO= 000030   9590#
C$GPRI= 000040   9590#
C$INIT= 000011   9590#  4516#
C$INLP= 000020   9590#
C$MANI= 000050   9590#
C$MEM = 000031   9590#
C$MSG = 000023   9590#  3861#  3900#  3934#  3978#  4023#  4058#  4093#  4128#
C$OPEN= 000034   9590#
C$PNTB= 000014   9590#  3857#  3914#  3924#  3930#  3947#  3953#  3963#  3973#  3992#  3998#  4008#  4018#
                 4037#  4043#  4053#  4072#  4079#  4089#  4107#  4113#  4123#  4271#
C$PNTF= 000017   9590#  3308#  3319#  3330#  3348#
C$PNTS= 000016   9590#
C$PNTX= 000015   9590#  3877#  3893#  4162#  4172#  4179#  4189#  4196#  4206#  4213#  4223#  4238#  4248#
                 4255#  4265#  4287#  4297#  4304#  4314#  4321#  4331#  4338#  4348#  4364#  4374#  4381#
                 4391#
C$QIO = 000377   9590#
C$RDBU= 000007   9590#
C$REFG= 000047   9590#  4438#  4445#  4452#  4459#
```

```
C$RESE= 000033            9590#  4509    4601
C$REVI= 000003            9590#  1030
C$RFLA= 000021            9590#
C$RPT = 000025            9590#
C$SEFG= 000046            9590#
C$SPRI= 000041            9590#
C$SVEC= 000037            9590#  4541
C$TPRI= 000013            9590#
DDCMP = 040000           13350#  4639    4831    5833    6094    6318    7178
DEVMAP  002404           17270#  4468*   4486*
DEVPTR  002406           17280#  4472*   4484*   4486    4487*
DFPTBL  002150 G         11070#
DIAGMC= 000000             959
DOTBMT= 000007           12470#
DTR   = 000020           13860#  2618    2621    2694    3156    3159    4824    4827    7100
DTRL  = 000000           13910#
D.BUG = 000000           16480#
EF.CON= 000036 G         11610#  4458
EF.NEW= 000035 G         11820#  4451
EF.PWR= 000034 G         11830#
EF.RES= 000037 G         11800#  4444
EF.STA= 000040 G         11790#  4437
EIAV35= 000002           16360#  3262
EM100   015536            3116   38160#
EM101   015556            3128   38160#
EM14    014324            3573   38160#
EM16    014340            3563   38160#
EM2     014214            2642   38160#
EM25    014360            2370   38160#
EM28    014406            3060   3632    38160#
EM29    014427            3049   3642    38160#
EM3     014263            2109   38160#
EM30    014444            3084   3609    38160#
EM31    014465            3073   3619    38160#
EM34    014502            3514   38160#  5018    5047    5076    5105    5134    5176    5205
EM35    014530            3540   38160#
EM36    014551            3550   38160#  5147
EM39    014566            3586   38160#
EM4     014307            2153   2200    2271    38160#  4877
EM40    014610            3596   38160#
EM54    014626            3501   38160#
EM68    014650            2776   38160#
EM69    014677            2779   38160#
EM70    014715            2790   38160#
EM71    014737            2824   38160#
EM72    014755            2835   38160#
EM73    014777            2869   38160#  4897    4910    4923    4950    4963    4976    4989
EM74    015014            2880   38160#
EM75    015035            2914   38160#  5002    5031    5060    5089    5118    5160    5189    5218
EM76    015051            2925   38160#
EM77    015071            2959   38160#
EM78    015105            2970   38160#
EM79    015125           38160#  4812
EM80    015161           38160#  4937
EM81    015201           38160#  5232
EM82    015225           38160#  5243
```

```
EM83    015262            3816#   6938    7030
EM84    015311            3816#   6957    7049
EM85    015326            3816#   6976    7080
EM86    015342            3816#   7061
EM87    015363            3816#   7092
EM88    015403            3816#   7111
EM89    015426            3816#   7130
EM90    015450            3002    3816#
EM91    015501            3013    3816#
ENDCOD  026235            4790    4818    5794#
ENDEMB  012116            3816#   3854    3927    4268
ENDIT   023074            4461    4513#
ENDPAT  003064            2026#
ENTRAN  011772            3755#   6046    6271    6495
EPATX   002740            1926#   6204    6428
ERRBLK  002200 G          1664#   2110*   2154*   2201*   2272*   2371*   2643*   2747*   2780*   2791*   2825*   2836*   2870*
                          2881*   2915*   2926*   2960*   2971*   3003*   3014*   3050*   3061*   3074*   3085*   3117*   3129*
                          3502*   3515*   3541*   3551*   3564*   3574*   3587*   3597*   3610*   3620*   3633*   3643*
ERRFLG  002350            1713#
ERRMSG  002176 G          1663#   2109*   2153*   2200*   2271*   2370*   2642*   2746*   2779*   2790*   2824*   2835*   2869*
                          2880*   2914*   2925*   2959*   2970*   3002*   3013*   3049*   3060*   3073*   3084*   3116*   3128*
                          3501*   3514*   3540*   3550*   3563*   3573*   3586*   3596*   3609*   3619*   3632*   3642*
ERRNBR  002174 G          1662#   2108*   2152*   2199*   2270*   2369*   2641*   2745*   2778*   2789*   2823*   2834*   2868*
                          2879*   2913*   2924*   2958*   2969*   3001*   3012*   3048*   3059*   3072*   3083*   3115*   3127*
                          3500*   3513*   3539*   3549*   3562*   3572*   3585*   3595*   3608*   3618*   3631*   3641*
ERROR1  002376            1724#   4422*
ERRTYP  002172 G          1661#   2107*   2151*   2198*   2269*   2368*   2640*   2744*   2777*   2788*   2822*   2833*   2867*
                          2878*   2912*   2923*   2957*   2968*   3000*   3011*   3047*   3058*   3071*   3082*   3114*   3126*
                          3499*   3512*   3538*   3548*   3561*   3571*   3584*   3594*   3607*   3617*   3630*   3640*
ERR10   020364 G          2643    2747    3515    3940#   5019    5048    5077    5106    5135    5148    5177    5206
ERR11   020540 G          3985#   6939    7031
ERR11$  022032            4020    4125    4280#
ERR12   020714 G          2780    2791    2825    2836    2870    2881    2915    2926    2960    2971    3003    3014    3050
                          3061    3074    3085    3117    3129    3502    3541    3551    3564    3574    3587    3597    3610
                          3620    3633    3643    4030#   4898    4911    4924    4951    4964    4977    4990    5003    5032
                          5061    5090    5119    5161    5190    5219
ERR12$  022364            3975    4055    4557#
ERR13   021030 G          4065#   4813
ERR14   021146 G          4100#   4938    5233    6958    6977    7050    7062    7081    7093    7112    7131
ERR3    020072 G          3851#   5244
ERR4    020120 G          2110    2154    2201    2272    3866#   4878
ERR4$   021306            3852    4156#
ERR5$   021634            3897    4231#
ERR7A   020244 G          2371    3905#
EVL     = 000004 G        1199#
EVRC    = 002400          1342#   6318
EXADD   = 000020          1354#
EXCON   = 000010          1355#
EXFCUT  = 000005          1246#   4862    4871
E$END   = 002100           959#
E$LOAD  = 000035           959#   1054
FLGCA1  = 000002          1583#
FLGCA2  = 000001          1584#
FLGCB1  = 000020          1580#
FLGCB2  = 000010          1581#
FLGIRQ  = 000200          1572#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
FLGSR = 000004           1582#
FLGT1 = 000100           1578#
FLGT2 = 000040           1579#
FMT10   012560           3816#
FMT10A  012634           3816#
FMT11   012655           3816#  4245    4262
FMT12   012674           3816#
FMT13   012703           3816#
FMT14   012751           3816#
FMT15   012766           3816#  3911
FMT15A  013020           3816#
FMT16   013072           3816#
FMT16A  013115           3816#
FMT17   013157           3816#
FMT17A  013236           3816#
FMT17B  013317           3816#
FMT17C  013372           3816#
FMT19   013452           3345   3816#
FMT2    012126           3816#
FMT21   013503           3816#  3944    3989    4034    4069    4104
FMT22   013513           3816#  3950    3995    4040    4110
FMT23   013535           3816#  3970    4015    4086
FMT24   013614           3816#  4284    4361
FMT25   013627           3816#  4294    4328    4371
FMT26   013657           3816#  4311    4345    4388
FMT27   013712           3816#  3960    4005    4050    4120
FMT28   013721           3816#
FMT29   013755           3816#  4301    4318    4335    4378
FMT3    012163           3816#  3921
FMT30   013762           3305   3816#
FMT31   014017           3316   3816#
FMT32   014065           3327   3816#
FMT39   014117           3816#
FMT4    012247           3816#  4159    4235
FMT4A   012305           3816#  4169    4203
FMT4B   012340           3816#  4176    4193    4210    4252
FMT4C   012345           3816#  4186    4220
FMT40   014150           3816#  4076
FMT5    012400           3816#  3874
FMT5A   012443           3816#  3890
FMT7    012530           3816#
FRSTIM  002370           1721#  4424    4433*
F$AU  = 000015            959#  4614    4616
F$AUTO= 000020            959#  4534    4570
F$BGN = 000040            959#   962    3851    3866    3905    3940    3985    4030    4065    4100    4404    4417    4534
                         4504   4598    4614    4634    4645    4655    4677    4689    4702    4712    4725    4736    4747
                         4750   4771    4778    4815    4842    4856    4880    4900    4913    4926    4940    4953    4966
                         4979   4992    5005    5021    5034    5050    5063    5079    5092    5108    5121    5137    5150
                         5163   5179    5192    5208    5221    5235    5248    5820    5839    5849    5859    5873    5883
                         5893   5903    5913    5922    5933    5949    5960    5977    5991    6002    6013    6024    6035
                         6052   6055    6081    6100    6110    6120    6134    6144    6154    6163    6174    6190    6201
                         6214   6225    6235    6246    6257    6268    6277    6280    6306    6324    6334    6344    6358
                         6368   6378    6387    6398    6414    6425    6438    6449    6459    6470    6481    6492    6501
                         6504   6526    6543    6565    6575    6585    6595    6605    6618    6629    6644    6655    6666
                         6677   6689    6711    6728    6750    6760    6770    6780    6790    6802    6813    6827    6838
                         6849   6860    6872    6900    6906    6910    6941    6960    6980    6989    7033    7052    7064
```

```
                    7083  7095  7114  7134  7138  7167  7184  7203  7213  7223  7233  7243  7252
                    7263  7273  7284  7298  7309  7323  7334  7344  7355  7365  7376  7386  7397
                    7414  7425  7442  7456  7467  7478  7489  7500  7503  7519  7570  7588
F$CLEA= 000007        9590  4584  4588
F$DU  = 000016        9590  4598  4603
F$END = 000041        9590   962  3862  3901  3935  3979  4024  4059  4094  4129  4517  4572  4590
                    4605  4618  4634  4645  4655  4677  4689  4702  4712  4725  4736  4747  4750
                    4752  4771  4778  4815  4842  4856  4880  4900  4913  4926  4940  4953  4966
                    4979  4992  5005  5021  5034  5050  5063  5079  5092  5108  5121  5137  5150
                    5163  5179  5192  5208  3221  5235  5248  5250  5820  5839  5849  5859  5873
                    5883  5893  5903  5913  5922  5933  5949  5960  5977  5991  6002  6013  6024
                    6035  6052  6055  6057  6081  6100  6110  6120  6134  6144  6154  6163  6174
                    6190  6201  6214  6225  6235  6246  6257  6268  6277  6280  6282  6306  6324
                    6334  6344  6358  6368  6378  6387  6398  6414  6425  6438  6449  6459  6470
                    6481  6492  6501  6504  6506  6526  6543  6565  6575  6585  6595  6605  6618
                    6629  6644  6655  6666  6677  6689  6691  6711  6728  6750  6760  6770  6780
                    6790  6802  6813  6827  6838  6849  6860  6872  6874  6900  6906  6910  6941
                    6960  6980  6982  6989  7033  7052  7064  7083  7095  7114  7134  7136  7138
                    7140  7167  7184  7203  7213  7223  7233  7243  7252  7263  7273  7284  7298
                    7309  7323  7334  7344  7355  7365  7376  7386  7397  7414  7425  7442  7456
                    7467  7478  7489  7500  7503  7505  7554  7576  7588
F$HARD= 000004        9590  7519  7552
F$HW  = 000013        9590  1105  1121
F$INIT= 000006        9590  4417  4515
F$JMP = 000050        9590  4778  6906
F$MOD = 000000        9590   962  7588
F$MSG = 000011        9590  3851  3860  3866  3899  3905  3933  3940  3977  3985  4022  4030  4057
                    4065  4092  4100  4127
F$PROT= 000021        9590  4404  4409
F$PWR = 000017        9590
F$RPT = 000012        9590
F$SEG = 000003        9590
F$SOFT= 000005        9590  7570  7574
F$SRV = 000010        9590
F$SUB = 000002        9590  6911  6980  6990  7134
F$SW  = 000014        9590  1130  1135
F$TEST= 000001        9590  4635  4750  4772  5248  5821  6055  6082  6280  6307  6504  6527  6689
                    6712  6872  6901  7138  7168  7503
GDATA   002324      17020  2104* 2148* 2195* 2266* 2362  2634* 2635* 2738* 3505* 3506* 3523* 3524*
                    3867* 3869  3889  3920  3969  4014  4085  4145  4803* 4804* 4871* 4887* 5012*
                    5041* 5070* 5099* 5128* 5170* 5199* 6931* 7016* 7022*
GETBSR  004024      22980
GETPRM  022?40      4462  44760 4485
GETURS  004360      23940 4357
GETVRS  004460      24250 4280
GETWSR  004166      2103  2147  2194  2265  23190 4870  5237
GOODAT  002364      17190
G$CNTO= 000200        9590
G$DELM= 000372        9590
G$DISP= 000003        9590
G$EXCP= 000400        9590
G$HILI= 000002        9590
G$LOLI= 000001        9590
G$NO  = 000000        9590
G$OFFS= 000400        9590  7523  7528  7533  7539  7545
G$OFSI= 000376        9590  7523  7528  7533  7539  7545
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
G$PRMA= 000001            9594#  7523   7528
G$PRMD= 000002            9594#  7533   7539   7545
G$PRML= 000000            9594#
G$RADA= 000140            9594#
G$RADB= 000000            9594#
G$RADD= 000040            9594#
G$RADL= 000120            9594#
G$RADO= 000020            9594#  7523   7528   7533   7539   7545
G$XFER= 000004            9594#
G$YES = 000010            9594#  7523   7528   7533   7539   7545
HDX   = 000004           13884#
HELP  = 000000            9504#   984   1072   1096   2073
HOE   = 100000 G         12124#
IBE   = 010000 G         12094#
IDLE  = 000010           13294#
IDLES = 004000           13384#  4831   5833   7178
IDU   = 000040 G         12024#
IEI   = 000001           12194#
IEO   = 000020           12184#
IER   = 020000 G         12104#
INIDMV  005376           27164#  4636   4773   5822   6082   6307   6528   6713   7169
INITRN  007234           31524#  4638   5832   6093   6317   6536   6721   7177
INITT1  004534           24634#
INITT2  004734           25504#
INTFLG  002346           17114#
INTGRL= 000001           16374#  3260   4775
INTSC = 000200           15944#
ISR   = 000100 G         12034#
IXE   = 004000 G         12084#
I$AU  = 000041            9594#  4614#  4618#
I$AUTO= 000041            9594#  4534#  4572#
I$CLN = 000041            9594#  4584#  4590#
I$DU  = 000041            9594#  4598#  4605#
I$HRD = 000041           7519#  7554#
I$INIT= 000041            9594#  4417#  4517#
I$MOD = 000041            9594#   962#  7588#
I$MSG = 000041            9594#  3851#  3862#  3866#  3901#  3905#  3935#  3940#  3979#  3985#  4024#  4030#  4059#
                         4065#  4094#  4100#  4129#
I$PROT= 000040            9594#  4404#
I$PTAB= 000041            9594#
I$PWR = 000041            9594#
I$RPT = 000041            9594#
I$SEG = 000041            9594#  4634   4771   5820   6081   6306   6526   6711   6900   6910   6989   7167
I$SETU= 000041            9594#
I$SFT = 000041           7570#  7576#
I$SRV = 000041            9594#
I$SUB = 000041            9594#  4634   4771   5820   6081   6306   6526   6711   6900   6910#  6941   6960   6980#
                         6982#  6989#  7033   7052   7064   7083   7095   7114   7134#  7136#  7167
I$TST = 000041            9594#  4634#  4645   4655   4677   4689   4702   4712   4725   4736   4747   4750#  4752#
                         4771#  4778   4815   4842   4856   4880   4900   4913   4926   4940   4953   4966   4979
                         4992   5005   5021   5034   5050   5063   5079   5092   5108   5121   5137   5150   5163
                         5179   5192   5208   5221   5235   5248#  5250#  5820#  5839   5849   5859   5873   5883
                         5893   5903   5913   5922   5933   5949   5960   5977   5991   6002   6013   6024   6035
                         6052   6055#  6057#  6081#  6100   6110   6120   6134   6144   6154   6163   6174   6190
                         6201   6214   6225   6235   6246   6257   6268   6277   6280#  6282#  6306#  6324   6334
                         6344   6358   6368   6378   6387   6398   6414   6425   6438   6449   6459   6470   6481
```

```
                    6492    6501    6504#   6506#   6526#   6543    6565    6575    6585    6595    6605    6618    6629
                    6644    6655    6666    6677    6689#   6691#   6711#   6728    6750    6760    6770    6780    6790
                    6802    6813    6827    6838    6849    6860    6872#   6874#   6900#   6906    6910    6989    7138#
                    7140#   7167#   7184    7203    7213    7223    7233    7243    7252    7263    7273    7284    7298
                    7309    7323    7334    7344    7355    7365    7376    7386    7397    7414    7425    7442    7456
                    7467    7478    7489    7500    7503#   7505#
J$JMP = 000167        959#
LOADAT  002362       1718#
LOE   = 040000 G     1211#
LOGDEV  002334       1706#   4470*   4477*   4479    4564
LOT   = 000010 G     1200#
LUSWI1  002464       1763#   4502*
LUSWI2  002466       1764#   4503*
LU1MOD  002000 G      962#
L$ACP   002110 G     1061#
L$APT   002036 G     1019#
L$AU    023230 G     1046    4614#
L$AUT   002070 G     1045#
L$AUTO  023076 G     1062    4534#
L$CCP   002106 G     1059#
L$CLEA  023222 G     1060    4584#
L$CO    002032 G     1015#
L$DEPO  002011 G      997#
L$DESC  003304 G     1052    2061#
L$DESP  002076 G     1051#
L$DEVP  002060 G     1037#
L$DISP  002124 G     1022    1085#
L$DLY   002116 G     1067#
L$DTP   002040 G     1021#
L$DTYP  002034 G     1017#
L$DU    023224 G     1048    4598#
L$DUT   002072 G     1047#
L$DVTY  003264 G     1038    2049#
L$EF    002052 G     1032#
L$ENVI  002044 G     1025#
L$ERRT  002172 G     1056    1660#
L$ETP   002102 G     1055#
L$EXP1  002046 G     1027#
L$EXP4  002064 G     1041#
L$EXP5  002066 G     1043#
L$HARD  033652 G     1004    7519    7520#
L$HIME  002120 G     1069#
L$HPCP  002016 G     1003#
L$HPTP  002022 G     1007#
L$HW    002150 G     1008    1105    1106#
L$ICP   002104 G     1057#
L$INIT  022554 G     1058    4417#
L$LADP  002026 G     1011#
L$LAST  034514 G     1012    7594#
L$LOAD  002100 G     1053#
L$LUN   002074 G     1049#
L$MREV  002050 G     1029#
L$NAME  002000 G      986#
L$PRIO  002042 G     1023#
L$PROT  022546 G     1064    4404*
L$PRT   002112 G     1063#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
L$REPP  002062 G        10390
L$REV   002010 G         9950
L$SOFT  034302 G        7570    75710
L$SPC   002056 G        10350
L$SPCP  002020 G        10050
L$SPTP  002024 G        10090
L$STA   002030 G        10130
L$SW    002172 G        1130    11310
L$TEST  002114 G        10650
L$TIML  002014 G        10010
L$UNIT  002012 G         9990
L10000  002170          1105    11210
L10001  002172          1130    11350
L10002  020116          38600
L10003  020242          38990
L10004  020362          39330
L10005  020536          39770
L10006  020712          40220
L10007  021026          40570
L10010  021144          40920
L10011  021260          41270
L10013  023074          45150
L10014  023212          45700
L10015  023222          45880
L10016  023226          46030
L10017  023230          46160
L10020  023530          4646    4656    4678    4690    4703    4713    4726    4737    4748    47500
L10021  025406          4779    4816    4843    4857    4881    4901    4914    4927    4941    4954    4967    4980    4993
                        5006    5022    5035    5051    5064    5080    5093    5109    5122    5138    5151    5164    5180
                        5193    5209    5222    5236    52480
L10022  027112          5840    5850    5860    5874    5884    5894    5904    5914    5923    5934    5950    5961    5978
                        5992    6003    6014    6025    6036    6053    60550
L10023  027654          6101    6111    6121    6135    6145    6155    6164    6175    6191    6202    6215    6226    6236
                        6247    6258    6269    6278    62800
L10024  030416          6325    6335    6345    6359    6369    6379    6388    6399    6415    6426    6439    6450    6460
                        6471    6482    6493    6502    65040
L10025  031100          6544    6566    6576    6586    6596    6606    6619    6630    6645    6656    6667    6678    66890
L10026  031562          6729    6751    6761    6771    6781    6791    6803    6814    6828    6839    6850    6861    68720
L10027  032526          6907    71380
L10030  032034          6942    6961    69800
L10031  032524          7034    7053    7065    7084    7096    7115    71340
L10032  033646          7185    7204    7214    7224    7234    7244    7253    7264    7274    7285    7299    7310    7324
                        7335    7345    7356    7366    7377    7387    7398    7415    7426    7443    7457    7468    7479
                        7490    7501    75030
L10053  033730          7519    75530
L10034  034302          7570    75750
MCLR  - 000100          12300   1232    2094    2104
MCODE   025410          4782    4794    52500
MDMRDY- 000040          14010   6929    6931    7014    7016    7020    7022    7104    7123
MLWRI   003722          2231    22560
MPCSR   002416          17360   3942    3987    4032    4067    4102    4490
MPIVEC  002456          17600   4498*
MPOVEC  002460          17610   4499*   4500*
MPRIOR  002462          17620   4501*
MRDY  - 000200          12370   2101    2144    2191    2263    4868
MREQ  - 000001          12310   1232    2094    2104
```

```
MSTCLR  003352          20940#  2716    6913    6992
NCRACT= 020000          16460#  6661    6672    6844    6855
NCTBMT= 000200          16420#  3389
NEWLIN  012123          38160#
NEWST   022720          4454    44690#
NFCRDA= 040000          16450#  6672    6855
NOCHK = 003400          13410#  4639
NOCRDA= 100000          16440#  4720    4731    4742
NOLOOP= 001000          16400#  5823    5829    6083    6089    6308    6314    6529    6534    6714    6719    7170    7175
NORXEN= 040000          16390#
NULCLK= 000200          13830#
OVRC  = 002000          13430#  6094
O$APTS= 000000           9590#  1013
O$AU  = 000001           9590#   9830#  1045
O$BGNR= 000000           9590#  1039
O$BGNS= 000000           9590#  1005
O$DU  = 000001           9590#   9830#  1047
O$ERRT= 000001           9590#   9830#  1055
O$GNSW= 000000           9590#  1009
O$POIN= 000001           9590#   9830#  1071
O$SETU= 000000           9590#   999    7592
PALENB= 000001          15480#
PATCH   034302          75780#
PATE    002576          18160#
PATF    002606          18270#  2624
PATG    002616          18380#
PATI    002741          19290#
PATJ    003002          19650#
PATK    003012          19760#  7403    7428
PATL    003033          19960#
PATM    003054          20160#
PATX    002645          18640#  5938    5963    6179    6403
PBLENB= 000002          15470#
PCR   = 120407          13490#  3172
PCSARH= 120405          13210#  3166    4834
PCSARL= 120404          13140#  3162    4829
PNT   = 001000 G        12060#
PRESET= 000001          13900#  2618    3156    4824
PRI   = 002000 G        12070#
PRIOR   002342          17090#
PRIRTY  034007           7534    75550#
PRI00 = 000000 G        11950#
PRI01 = 000040 G        11940#
PRI02 = 000100 G        11930#
PRI03 = 000140 G        11920#
PRI04 = 000200 G        11910#
PRI05 = 000240 G        11900#
PRI06 = 000300 G        11890#
PRI07 = 000340 G        11880#
PROTO = 000100          13260#
PSTACK  002340          17080#  4419*
RABGA = 000004          12710#  3577    3579    3589
RAMADR= 001000          16340#  4783    4795    4861
RCVBUF  003064          20320#
RCVDAT= 000002          14050#  6928    7013    7019
RCV1ST  011624          36990#  5916    6157    6381    6612    6796    7246
```

```
RDA    = 000200          13650  2907   2918   3720
RDSRH  = 120401          12650  2991   3038   3477
RDSRL  = 120400          12590  3480
READ     003454          21360  2357   2398   2403   2428
READI    003566          21820  2485   2572   2626   2731   2767   2812   2857   2902   2947   2990   3037   3104
                         3206   3476   3479   3490   3717   4705   4797   6925   6947   6966   7006   7039   7070
                         7101   7120
REDBYT   002356          17160
REDDAT   002516          17810
REDLOC=  000001          12420  2137   2148   2184   2195
REDPAG=  000003          12440
REGNUM   002336          17070  2632*  2633*  2737*  2766*  2811*  2856*  2901*  2946*  2989*  3036*  3103*  3495*
                         3504*  3522*  3527*  3906   3955   4000   4045   4075   4115   4802*  4886*  4932*  5011*
                         5040*  5069*  5098*  5127*  5169*  5198*  5227*  6923*  7002*
REG0     002526          17840  5821*  6038*  6527*  6680*  6712*  6863*  7168*
REG1     002530          17850
REG2     002532          17860
REG3     002534          17870
REG4     002536          17880
REG5     002540          17890
REG6     002542          17900
REG7     002544          17910
REOM   = 000002          12720  3066   3077   3600   3602   3612
RERCHK=  000001          12820  3528   6007   6168   6195   6219   6240   6251   6262   6392   6413   6443   6464
                         6475   6486   6671   6854   7349   7472
RERR   = 000200          12680  3531   3533   3543
RETADR   002354          17150
RING   = 000200          13990  6929   6931   7013   7020   7022
ROR    = 000010          12700  2995   3006   3554   3556   3566
RSA    = 000020          13680  2952   2963
RSOM   = 000001          12730  3042   3053   3623   3625   3635
RSTCHK   005104          26120
RTSND  = 000010          13870  2618   2621   2694   3156   3159   4697   4824   4827   6946   7038
RUN    = 000200          12290  1232   2094   2104
RXABGA=  002000          12780
RXACT  = 000040          13670  2817   2828
RXCHAR   010350          34730  4717   4728   4739   5925   5952   5969   5983   5994   6005   6016   6027   6166
                         6193   6217   6238   6249   6260   6390   6417   6441   6462   6473   6484   6621   6636
                         6647   6658   6669   6805   6819   6830   6841   6852   7255   7276   7290   7301   7326
                         7347   7368   7389   7417   7434   7448   7459   7470   7481   7492
RXDL   = 000007          13560  3484   6095   6319
RXEN   = 000100          13840  3192   3222   4684   4697
RXEOM  = 001000          12790  6670   6853
RXERR  = 100000          12760  6006   7348   7471
RXOR   = 004000          12770
RXSOM  = 000400          12800  6622   6806
SAVE4    002372          17220  4426*  4430
SAVE6    002374          17230  4427*  4431
SAVLEN   002402          17260  3169*  3170*  3484   4423*
SCRACH   002332          17050
SECAD  = 000020          13280
SECADR=  010000          13370
SEL0     002416          17350  2319
SEL10    002436          17480  2323
SEL12    002442          17510  2324
SEL14    002446          17540  2325
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
SEL16   002452           1757#  2326
SEL2    002422           1739#  2320
SEL4    002426           1742#  2136*  2183*  2229*  2254*  2321   4860*  4861*
SEL6    002432           1745#  2206   2230*  2255*  2322
SETVIA  005304           2682#  2717
SFPTBL  002172 G         1132#
SFR   = 000001           1372#
SPEED = 000020           1402#  6928   7013   7019
SRMODE= 000034           1536#
STALL   004356           2381#
STARES  002412           1730#  3302   3313   3324   3341   4465*  4471*
STARST  022710           4440   4447   4464#
STEPLU  012072           3204   3395   3443   3664   3714   3770   3793#  4692
STRIP = 000040           1327#
STRIPS= 020000           1336#  4639   4831   5833   6094   6318   7178
STRTML= 000301           1232#
STUREG  004250           2348#
SUBRPC  002344           1710#  4420*
SVCGBL= 000000            959#   962    969#   986    995    997    999    1001   1003   1005   1007   1009   1011
                         1013   1015   1017   1019   1021   1023   1025   1027   1029   1032   1035   1037   1039
                         1041   1043   1045   1047   1049   1051   1053   1055   1057   1059   1061   1063   1065
                         1067   1069   1085   1106   1107   1131   1132   1660   2049   2061   3851   3866   3905
                         3940   3985   4030   4065   4100   4404   4417   4534   4584   4598   4614   7520   7571
                         7594#  7595
SVCINS= 000001            959#   966#   987    988    989    990    991    992    993    994    996    998    1000
                         1002   1004   1006   1008   1010   1012   1014   1016   1018   1020   1022   1024   1026
                         1028   1030   1031   1033   1034   1036   1038   1040   1042   1044   1046   1048   1050
                         1052   1054   1056   1058   1060   1062   1064   1066   1068   1070   1084   1086   1087
                         1088   1089   1090   1091   1092   1093   1094   1105   1130   2050   2053   2062   2069
                         2107   2108   2109   2110   2151   2152   2153   2154   2198   2199   2200   2201   2269
                         2270   2271   2272   2368   2369   2370   2371   2640   2641   2642   2643   2744   2745
                         2746   2747   2777   2778   2779   2780   2788   2789   2790   2791   2822   2823   2824
                         2825   2833   2834   2835   2836   2867   2868   2869   2870   2878   2879   2880   2881
                         2912   2913   2914   2915   2923   2924   2925   2926   2957   2958   2959   2960   2968
                         2969   2970   2971   3000   3001   3002   3003   3011   3012   3013   3014   3047   3048
                         3049   3057   3058   3059   3060   3061   3071   3072   3073   3074   3082   3083   3084
                         3085   3114   3115   3116   3117   3126   3127   3128   3129   3305   3306   3307   3308
                         3309   3316   3317   3318   3319   3320   3327   3328   3329   3330   3331   3344   3345
                         3346   3347   3348   3349   3499   3500   3501   3502   3512   3513   3514   3515   3538
                         3539   3540   3541   3548   3549   3550   3551   3561   3562   3563   3564   3571   3572
                         3573   3574   3584   3585   3586   3587   3594   3595   3596   3597   3607   3608   3609
                         3610   3617   3618   3619   3620   3630   3631   3632   3633   3640   3641   3642   3643
                         3854   3855   3856   3857   3858   3861   3873   3874   3875   3876   3877   3878   3888
                         3889   3890   3891   3892   3893   3894   3900   3909   3910   3911   3912   3913   3914
                         3915   3918   3919   3920   3921   3922   3923   3924   3925   3927   3928   3929   3930
                         3931   3934   3942   3943   3944   3945   3946   3947   3948   3950   3951   3952   3953
                         3954   3958   3959   3960   3961   3962   3963   3964   3967   3968   3969   3970   3971
                         3972   3973   3974   3978   3987   3988   3989   3990   3991   3992   3993   3995   3996
                         3997   3998   3999   4003   4004   4005   4006   4007   4008   4009   4012   4013   4014
                         4015   4016   4017   4018   4019   4023   4032   4033   4034   4035   4036   4037   4038
                         4040   4041   4042   4043   4044   4048   4049   4050   4051   4052   4053   4054   4058
                         4067   4068   4069   4070   4071   4072   4073   4075   4076   4077   4078   4079   4080
                         4083   4084   4085   4086   4087   4088   4089   4090   4093   4102   4103   4104   4105
                         4106   4107   4108   4110   4111   4112   4113   4114   4118   4119   4120   4121   4122
                         4123   4124   4128   4157   4158   4159   4160   4161   4162   4163   4165   4166   4167
                         4168   4169   4170   4171   4172   4173   4175   4176   4177   4178   4179   4180   4182
```

4183	4184	4185	4186	4187	4188	4189	4190	4192	4193	4194	4195	4196
4197	4199	4200	4201	4202	4203	4204	4205	4206	4207	4209	4210	4211
4212	4213	4214	4216	4217	4218	4219	4220	4221	4222	4223	4224	4233
4234	4235	4236	4237	4238	4239	4241	4242	4243	4244	4245	4246	4247
4248	4249	4251	4252	4253	4254	4255	4256	4258	4259	4260	4261	4262
4263	4264	4265	4266	4268	4269	4270	4271	4272	4282	4283	4284	4285
4286	4287	4288	4290	4291	4292	4293	4294	4295	4296	4297	4298	4300
4301	4302	4303	4304	4305	4307	4308	4309	4310	4311	4312	4313	4314
4315	4317	4318	4319	4320	4321	4322	4324	4325	4326	4327	4328	4329
4330	4331	4332	4334	4335	4336	4337	4338	4339	4341	4342	4343	4344
4345	4346	4347	4348	4349	4359	4360	4361	4362	4363	4364	4365	4367
4368	4369	4370	4371	4372	4373	4374	4375	4377	4378	4379	4380	4381
4382	4384	4385	4386	4387	4388	4389	4390	4391	4392	4437	4438	4440
4444	4445	4447	4451	4452	4454	4458	4459	4461	4479	4480	4481	4483
4509	4516	4537	4538	4539	4540	4541	4542	4559	4560	4564	4565	4571
4589	4601	4604	4617	4643	4645	4646	4653	4655	4656	4675	4677	4678
4687	4689	4690	4700	4702	4703	4710	4712	4713	4723	4725	4726	4734
4736	4737	4745	4747	4748	4751	4778	4779	4810	4811	4812	4813	4815
4816	4840	4842	4843	4854	4856	4857	4875	4876	4877	4878	4880	4881
4895	4896	4897	4898	4900	4901	4908	4909	4910	4911	4913	4914	4921
4922	4923	4924	4926	4927	4935	4936	4937	4938	4940	4941	4948	4949
4950	4951	4953	4954	4961	4962	4963	4964	4966	4967	4974	4975	4976
4977	4979	4980	4987	4988	4989	4990	4992	4993	5000	5001	5002	5003
5005	5006	5016	5017	5018	5019	5021	5022	5029	5030	5031	5032	5034
5035	5045	5046	5047	5048	5050	5051	5058	5059	5060	5061	5063	5064
5074	5075	5076	5077	5079	5080	5087	5088	5089	5090	5092	5093	5103
5104	5105	5106	5108	5109	5116	5117	5118	5119	5121	5122	5132	5133
5134	5135	5137	5138	5145	5146	5147	5148	5150	5151	5158	5159	5160
5161	5163	5164	5174	5175	5176	5177	5179	5180	5187	5188	5189	5190
5192	5193	5203	5204	5205	5206	5208	5209	5216	5217	5218	5219	5221
5222	5230	5231	5232	5233	5235	5236	5241	5242	5243	5244	5249	5837
5839	5840	5847	5849	5850	5857	5859	5860	5871	5873	5874	5881	5883
5884	5891	5893	5894	5901	5903	5904	5911	5913	5914	5920	5922	5923
5931	5933	5934	5947	5949	5950	5958	5960	5961	5975	5977	5978	5989
5991	5992	6000	6002	6003	6011	6013	6014	6022	6024	6025	6033	6035
6036	6050	6052	6053	6056	6098	6100	6101	6108	6110	6111	6118	6120
6121	6132	6134	6135	6142	6144	6145	6152	6154	6155	6161	6163	6164
6172	6174	6175	6188	6190	6191	6199	6201	6202	6212	6214	6215	6223
6225	6226	6233	6235	6236	6244	6246	6247	6255	6257	6258	6266	6268
6269	6275	6277	6278	6281	6322	6324	6325	6332	6334	6335	6342	6344
6345	6356	6358	6359	6366	6368	6369	6376	6378	6379	6385	6387	6388
6396	6398	6399	6412	6414	6415	6423	6425	6426	6436	6438	6439	6447
6449	6450	6457	6459	6460	6468	6470	6471	6479	6481	6482	6490	6492
6493	6499	6501	6502	6505	6541	6543	6544	6563	6565	6566	6573	6575
6576	6583	6585	6586	6593	6595	6596	6603	6605	6606	6616	6618	6619
6627	6629	6630	6642	6644	6645	6653	6655	6656	6664	6666	6667	6675
6677	6678	6690	6726	6728	6729	6748	6750	6751	6758	6760	6761	6768
6770	6771	6778	6780	6781	6788	6790	6791	6800	6802	6803	6811	6813
6814	6825	6827	6828	6836	6838	6839	6847	6849	6850	6858	6860	6861
6873	6906	6907	6911	6936	6937	6938	6939	6941	6942	6955	6956	6957
6958	6960	6961	6974	6975	6976	6977	6981	6990	7028	7029	7030	7031
7033	7034	7047	7048	7049	7050	7052	7053	7059	7060	7061	7062	7064
7065	7078	7079	7080	7081	7083	7084	7090	7091	7092	7093	7095	7096
7109	7110	7111	7112	7114	7115	7128	7129	7130	7131	7135	7139	7182
7184	7185	7201	7203	7204	7211	7213	7214	7221	7223	7224	7231	7233
7234	7241	7243	7244	7250	7252	7253	7261	7263	7264	7271	7273	7274

```
                    7282   7284   7285   7296   7298   7299   7307   7309   7310   7321   7323   7324   7332
                    7334   7335   7342   7344   7345   7353   7355   7356   7363   7365   7366   7374   7376
                    7377   7384   7386   7387   7395   7397   7398   7412   7414   7415   7423   7425   7426
                    7440   7442   7443   7454   7456   7457   7465   7467   7468   7476   7478   7479   7487
                    7489   7490   7498   7500   7501   7504   7519   7523   7524   7525   7526   7528   7529
                    7530   7531   7533   7534   7535   7536   7537   7539   7540   7541   7542   7543   7545
                    7546   7547   7548   7549   7552   7570   7574   7591   7592   7593
SVCSUB= 000001       9590   9680   6910   6989
SVCTAG= 000001       9590   9700   1077   1081   1121   1135   3860   3899   3933   3977   4022   4057   4092
                    4127   4515   4570   4588   4603   4616   4750   5248   6055   6280   6504   6689   6872
                    6980   7134   7138   7503   7553   7575
SVCTST= 000001       9590   9670   4634   4771   5820   6081   6306   6526   6711   6900   7167
SWPBOT= 1^1000      16280
SWPDOC= 121400      16290
SYNCH = 000226      13150   3199   4639   5833   5843   5853   6094   6104   6114   6250   6261   6318   6328
                    6338   6474   6485   7178
S$LSYM= 010000       9590  11220  11360  38610  39000  39340  39780  40230  40580  40930  41280  45160  45710
                    45890  46040  46170  47510  52490  60560  62810  65050  66900  68730  69810  71350  71390
                    75040  75540  75760
TAB   = 000004      12990
TBMT  = 000100      13660   2862   2873   3209
TCCHEK= 100000      16330   3247   6903
TCON.M  034123       7546   7555#
TDATA   002322      17010   2353
TDSRH = 120403      12940   3195   3430   4845
TDSRL = 120402      12880   3198   3377   4848
TDSRNR  002575      18130
TEOM  = 000002      13000   5967   5981   6609   6633   6794   6817   7288   7432   7446
TERR  = 000200      12970
TGA   = 000010      12980
TIMFLG  002352      17140
TM    = 000004      14040   6929   6931
TMP0    002546      17940   4543*  4561   4573*
TMP1    002550      17950
TMP2    002552      17960
TMP3    002554      17970
TMP4    002556      17980
TMP5    002560      17990
TMP6    002562      18000
TMP7    002564      18010
TSO   = 000010      13690   3109   3121
TSOM  = 000001      13010   3196   4659   4663   4846   6042   6208   6229   6432   6453   6547   6550   6553
                    6684   6732   6735   6738   6867   7188   7191
TSICON  002472      17660   3249   3254   3270   3272   3281   3284   3290   4505*  5827   6087   6312   6532
                    6717   6986   7173
TSTNUM  002414      17310   3344   4772*  6901*
TTLOOP= 000002      13890   3192   3222   3756   3769   4684   4697   6922   6946   696^
TXAB  = 002000      13060
TXACT = 000004      13700   2772   2783
TXCHAR  010136      33720   4670   5842   5852   5866   5876   5886   5896   5906   5942   6103   6113   6127
                    6137   6147   6183   6327   6337   6351   6361   6371   6407   6558   6568   6578   6588
                    6598   6743   6753   6763   6773   6783   7196   7206   7216   7226   7236   7266   7316
                    7337   7358   7379   7407
TXCTRL  010250      34250   4658   4662   4666   5862   5966   5980   6041   6123   6207   6228   6347   6431
                    6452   6546   6549   6552   6555   6608   6632   6683   6731   6734   6737   6740   6793
                    6816   6866   7187   7190   7193   7287   7312   7431   7445
```

```
TXDL  = 000340          13530   3484    6095    6319
TXEN  = 000040          13850   3192    3222    4684    4697
TXEOM = 001000          13070
TXERR = 100000          13040
TXGA  = 004000          13050
TXSOM = 000400          13080
TXTMLT  017744          38250   3888
TXTML0  017154          38160   3825
TXTML1  017160          38160   3825
TXTML2  017174          38160   3825
TXTML3  017211          38160   3825
TXTML4  017233          38160   3825
TXTML5  017254          38160   3825
TXTML6  017304          38160   3825
TXTML7  017316          38160   3825
TXTNP   017500          38160   3837
TXTNPT  020030          38380
TXTNP0  017505          38160   3838
TXTNP1  017515          38160   3838
TXTNP2  017525          38160   3838
TXTNP3  017535          38160   3838
TXTNP4  017552          38160   3838
TXTNP5  017567          38160   3838
TXTNP6  017604          38160   3838
TXTNP7  017620          38160   3838
TXTNP8  017634          38160   3838
TXTNUL  017152          38160
TXTUR   017650          38160   3910    3959    4049
TXTURT  020052          38410   3909    3958    4048
TXTUR0  017663          38160   3841
TXTUR1  017671          38160   3841
TXTUR2  017677          38160   3841
TXTUR3  017705          38160   3841
TXTUR4  017713          38160   3841
TXTUR5  017722          38160   3841
TXTUR6  017731          38160   3841
TXTUR7  017735          38160   3841
TXTVR   017353          38160   3829    4004    4119
TXTVRA  017451          38160   3833
TXTVRB  017454          38160   3833
TXTVRC  017460          38160   3833
TXTVRD  017464          38160   3833
TXTVRE  017470          38160   3833
TXTVRF  017474          38160   3833
TXTVRT  017766          38300   4003    4118
TXTVR0  017371          38160   3830
TXTVR1  017375          38160   3830
TXTVR2  017401          38160   3830
TXTVR3  017406          38160   3830
TXTVR4  017413          38160   3830
TXTVR5  017420          38160   3830
TXTVR6  017425          38160   3830
TXTVR7  017432          38160   3830
TXTVR8  017437          38160   3833
TXTVR9  017444          38160   3833
TXT1    015602          38160   4157
```

```
TXT10   016402            3816#
TXT11   016422            3816#
TXT11A  016474            3816#
TXT11B  016532            3816#
TXT12   016602            3816#  3943   3988   4033   4068   4103
TXT13   016630            3816#  4360
TXT14   016645            3816#  4359
TXT15   016703            3816#  4377
TXT16   016745            3816#  4283
TXT17   016760            3816#  4282
TXT18   017015            3816#  4300
TXT19   017056            3816#  4317
TXT2    015640            3816#  4175
TXT2A   015702            3816#  4192
TXT2B   015741            3816#  4209
TXT20   017112            3816#  4334
TXT3    016004            3816#  4158
TXT4    016034            3816#  4233
TXT4A   016074            3816#  4251
TXT5    016135            3816#
TXT6    016137            3816#  4234
TXT7    016162            3816#
TXT7A   016252            3816#
TXT8    016342            3816#
TXT9    016362            3816#
TXU   = 000002            1371#  3493
T$ARGC= 000005             987#   988#   989#   990#   991#   992#  3305#  3309   3316#  3320   3327#  3331   3344#
                          3349   3854#  3859   3873#  3878   3888#  3894   3909#  3915   3918#  3925   3927#  3931
                          3942#  3948   3950#  3954   3958#  3964   3967#  3974   3987#  3993   3995#  3999   4003#
                          4009   4012#  4019   4032#  4038   4040#  4044   4048#  4054   4067#  4073   4075#  4080
                          4083#  4090   4102#  4108   4110#  4114   4118#  4124   4157#  4163   4165#  4173   4175#
                          4180   4182#  4190   4192#  4197   4199#  4207   4209#  4214   4216#  4224   4233#  4239
                          4241#  4249   4251#  4256   4258#  4266   4268#  4272   4282#  4288   4290#  4298   4300#
                          4305   4307#  4315   4317#  4322   4324#  4332   4334#  4339   4341#  4349   4359#  4365
                          4367#  4375   4377#  4382   4384#  4392
T$CODE= 006032            7523#  7528#  7533#  7539#  7545#
T$ERRN= 000113             959#  4811#  4876#  4896#  4909#  4922#  4936#  4949#  4962#  4975#  4988#  5001#  5017#
                          5030#  5046#  5059#  5075#  5088#  5104#  5117#  5133#  5146#  5159#  5175#  5188#  5204#
                          5217#  5231#  5242#  6937#  6956#  6975#  7029#  7048#  7060#  7079#  7091#  7110#  7129#
T$EXCP= 000000            7523#  7527   7528#  7532   7533#  7538   7539#  7544   7545#  7550
T$FLAG= 000040            4645#  4655#  4677#  4689#  4702#  4712#  4725#  4736#  4747#  4778#  4815#  4842#  4856#
                          4880#  4900#  4913#  4926#  4940#  4953#  4966#  4979#  4992#  5005#  5021#  5034#  5050#
                          5063#  5079#  5092#  5108#  5121#  5137#  5150#  5163#  5179#  5192#  5208#  5221#  5235#
                          5839#  5849#  5859#  5873#  5883#  5893#  5903#  5913#  5922#  5933#  5949#  5960#  5977#
                          5991#  6002#  6013#  6024#  6035#  6052#  6100#  6110#  6120#  6134#  6144#  6154#  6163#
                          6174#  6190#  6201#  6214#  6225#  6235#  6246#  6257#  6268#  6277#  6324#  6334#  6344#
                          6358#  6368#  6378#  6387#  6398#  6414#  6425#  6438#  6449#  6459#  6470#  6481#  6492#
                          6501#  6543#  6565#  6575#  6585#  6595#  6605#  6618#  6629#  6644#  6655#  6666#  6677#
                          6728#  6750#  6760#  6770#  6780#  6790#  6802#  6813#  6827#  6838#  6849#  6860#  6906#
                          6941#  6960#  7033#  7052#  7064#  7083#  7095#  7114#  7184#  7203#  7213#  7223#  7233#
                          7243#  7252#  7263#  7273#  7284#  7298#  7309#  7323#  7334#  7344#  7355#  7365#  7376#
                          7386#  7397#  7414#  7425#  7442#  7456#  7467#  7478#  7489#  7500#
T$GMAN= 000000             959#
T$HILI= 000004            7523#  7526   7528#  7531   7533#  7537   7539#  7543   7545#  7549
T$LAST= 000001             959#  7592#
T$LOLI= 000000            7523#  7525   7528#  7530   7533#  7536   7539#  7542   7545#  7548
```

```
T$LSYM= 010000          9590  1122  1136  3861  3900  3934  3978  4023  4058  4093  4128  4216  4571
                        4589  4604  4617  4751  5249  6056  6281  6505  6690  6873  6981  7135  7139
                        7504  7554  7576
T$LTNO= 000011          7595#
T$NEST= 177777          9590   962#  1105#  1121#  1130#  1135#  3851#  3860#  3866#  3899#  3905#  3933#  3940#
                        3977#  3985#  4022#  4030#  4057#  4065#  4092#  4100#  4127#  4404#  4409#  4417#  4515#
                        4534#  4570#  4584#  4588#  4598#  4603#  4614#  4616#  4635#  4750#  4772#  5248#  5821#
                        6055#  6082#  6280#  6307#  6504#  6527#  6689#  6712#  6872#  6901#  6911#  6980#  6990#
                        7134#  7138#  7168#  7503#  7519#  7552#  7570#  7574#  7588#
T$NS0 = 000000           962#  7588
T$NS1 = 000005         1105#  1121  1130#  1135  3851#  3860  3866#  3899  3905#  3933  3940#  3977  3985#
                        4022  4030#  4057  4065#  4092  4100#  4127  4404#  4409  4417#  4515  4534#  4570
                        4584#  4588  4598#  4603  4614#  4616  4635#  4750  4772#  5248  5821#  6055  6082#
                        6280  6307#  6504  6527#  6689  6712#  6872  6901#  7138  7168#  7503  7519#  7552
                        7570#  7574
T$NS2 = 000002          6911#  6980  6990#  7134
T$PTNU= 000000          9590#
T$SAVL= 177777          9590#
T$SEGL= 177777          9590#
T$SUBN= 000000          9590#  4634#  4771#  5820#  6081#  6306#  6526#  6711#  6900#  6910#  6989#  7167#
T$TAGL= 177777          9590#
T$TAGN= 010035          9590#  1105#  1130#  3851#  3866#  3905#  3940#  3985#  4030#  4065#  4100#  4404#  4417#
                        4534#  4584#  4598#  4614#  4635#  4772#  5821#  6082#  6307#  6527#  6712#  6901#  6911#
                        6990#  7168#  7519#  7570#
T$TEMP= 000000          1077#  1081#  1086#  1087#  1088#  1089#  1090#  1091#  1092#  1093#  1094#  1095#  1121#
                        1135#  3860#  3899#  3933#  3977#  4022#  4057#  4092#  4127#  4409#  4515#  4570#  4588#
                        4603#  4616#  4645#  4646  4655#  4656  4677#  4678  4689#  4690  4702#  4703  4712#
                        4713  4725#  4726  4736#  4737  4747#  4748  4750#  4778#  4779  4815#  4816  4842#
                        4843  4856#  4857  4880#  4881  4900#  4901  4913#  4914  4926#  4927  4940#  4941
                        4953#  4954  4966#  4967  4979#  4980  4992#  4993  5005#  5006  5021#  5022  5034#
                        5035  5050#  5051  5063#  5064  5079#  5080  5092#  5093  5108#  5109  5121#  5122
                        5137#  5138  5150#  5151  5163#  5164  5179#  5180  5192#  5193  5208#  5209  5221#
                        5222  5235#  3236  5248#  5839#  5840  5849#  5850  5859#  5860  5873#  5874  5883#
                        5884  5893#  5894  5903#  5904  5913#  5914  5922#  5923  5933#  5934  5949#  5950
                        5960#  5961  5977#  5978  5991#  5992  6002#  6003  6013#  6014  6024#  6025  6035#
                        6036  6052#  6053  6055#  6100#  6101  6110#  6111  6120#  6121  6134#  6135  6144#
                        6145  6154#  6155  6163#  6164  6174#  6175  6190#  6191  6201#  6202  6214#  6215
                        6225#  6226  6235#  6236  6246#  6247  6257#  6258  6268#  6269  6277#  6278  6280#
                        6324#  6325  6334#  6335  6344#  6345  6358#  6359  6368#  6369  6378#  6379  6387#
                        6388  6398#  6399  6414#  6415  6425#  6426  6438#  6439  6449#  6450  6459#  6460
                        6470#  6471  6481#  6482  6492#  6493  6501#  6502  6504#  6543#  6544  6565#  6566
                        6575#  6576  6585#  6586  6595#  6596  6605#  6606  6618#  6619  6629#  6630  6644#
                        6645  6655#  6656  6666#  6667  6677#  6678  6689#  6728#  6729  6750#  6751  6760#
                        6761  6770#  6771  6780#  6781  6790#  6791  6802#  6803  6813#  6814  6827#  6828
                        6838#  6839  6849#  6850  6860#  6861  6872#  6906#  6907  6941#  6942  6960#  6961
                        6980#  7033#  7034  7052#  7053  7064#  7065  7083#  7084  7095#  7096  7114#  7115
                        7134#  7138#  7184#  7185  7203#  7204  7213#  7214  7223#  7224  7233#  7234  7243#
                        7244  7252#  7253  7263#  7264  7273#  7274  7284#  7285  7298#  7299  7309#  7310
                        7323#  7324  7334#  7335  7344#  7345  7355#  7356  7365#  7366  7376#  7377  7386#
                        7387  7397#  7398  7414#  7415  7425#  7426  7442#  7443  7456#  7457  7467#  7468
                        7478#  7479  7489#  7490  7500#  7501  7503#  7523#  7528#  7533#  7539#  7545#  7552#
                        7574#  7588#
T$TEST= 000011          9590#  4634#  4771#  5820#  6081#  6306#  6526#  6711#  6900#  6910  6989  7167#  7595
T$TSTM= 177777          9590#  3308  3319  3330  3348  3857  3861  3877  3893  3900  3914  3924  3930
                        3934  3947  3953  3963  3973  3978  3992  3998  4008  4018  4023  4037  4043
                        4053  4058  4072  4079  4089  4093  4107  4113  4123  4128  4162  4172  4179
```

```
                    4189  4196  4206  4213  4223  4238  4248  4255  4265  4271  4287  4297  4304
                    4314  4321  4331  4338  4348  4364  4374  4381  4391  4438  4445  4452  4459
                    4480  4509  4516  4541  4560  4565  4571  4589  4601  4604  4617  4643  4645
                    4653  4655  4675  4677  4687  4689  4700  4702  4710  4712  4723  4725  4734
                    4736  4745  4747  4751  4778  4810  4815  4840  4842  4854  4856  4875  4880
                    4895  4900  4908  4913  4921  4926  4935  4940  4948  4953  4961  4966  4974
                    4979  4987  4992  5000  5005  5016  5021  5029  5034  5045  5050  5058  5063
                    5074  5079  5087  5092  5103  5108  5116  5121  5132  5137  5145  5150  5158
                    5163  5174  5179  5187  5192  5203  5208  5216  5221  5230  5235  5241  5249
                    5837  5839  5847  5849  5857  5859  5871  5873  5881  5883  5891  5893  5901
                    5903  5911  5913  5920  5922  5931  5933  5947  5949  5958  5960  5975  5977
                    5989  5991  6000  6002  6011  6013  6022  6024  6033  6035  6050  6052  6056
                    6098  6100  6108  6110  6118  6120  6132  6134  6142  6144  6152  6154  6161
                    6163  6172  6174  6188  6190  6199  6201  6212  6214  6223  6225  6233  6235
                    6244  6246  6255  6257  6266  6268  6275  6277  6281  6322  6324  6332  6334
                    6342  6344  6356  6358  6366  6368  6376  6378  6385  6387  6396  6398  6412
                    6414  6423  6425  6436  6438  6447  6449  6457  6459  6468  6470  6479  6481
                    6490  6492  6499  6501  6505  6541  6543  6563  6565  6573  6575  6583  6585
                    6593  6595  6603  6605  6616  6618  6627  6629  6642  6644  6653  6655  6664
                    6666  6675  6677  6690  6726  6728  6748  6750  6758  6760  6768  6770  6778
                    6780  6788  6790  6800  6802  6811  6813  6825  6827  6836  6838  6847  6849
                    6858  6860  6873  6906  6911  6936  6941  6955  6960  6974  6981  6990  7028
                    7033  7047  7052  7059  7064  7078  7083  7090  7095  7109  7114  7128  7135
                    7139  7182  7184  7201  7203  7211  7213  7221  7223  7231  7233  7241  7243
                    7250  7252  7261  7263  7271  7273  7282  7284  7296  7298  7307  7309  7321
                    7323  7332  7334  7342  7344  7353  7355  7363  7365  7374  7376  7384  7386
                    7395  7397  7412  7414  7423  7425  7440  7442  7454  7456  7465  7467  7476
                    7478  7487  7489  7498  7500  7504
T$TSTS= 000001        959#  4635#  4772#  5821#  6082#  6307#  6527#  6712#  6901#  7168#
T$$AU = 010017       4614#  4616
T$$AUT= 010014       4534#  4570
T$$CLE= 010015       4584#  4588
T$$DU = 010016       4598#  4603
T$$HAR= 010033       7519#  7553
T$$HW = 010000       1105#  1121
T$$INI= 010013       4417#  4515
T$$MSG= 010011       3851#  3860  3866#  3899  3925#  3933  3940#  3977  3985#  4022  4030#  4057  4065#
                     4092  4100#  4127
T$$PRO= 010012       4404#
T$$SOF= 010034       7570#  7575
T$$SUB= 010031       6911#  6941  6960  6980  6990#  7033  7052  7064  7083  7095  7114  7134
T$$SW = 010001       1130#  1135
T$$TES= 010032       4635#  4645  4655  4677  4689  4702  4712  4725  4736  4747  4750  4772#  4778
                     4815  4842  4856  4880  4900  4913  4926  4940  4953  4966  4979  4992  5005
                     5021  5034  5050  5063  5079  5092  5108  5121  5137  5150  5163  5179  5192
                     5208  5221  5235  5248  5821#  5839  5849  5859  5873  5883  5893  5903  5913
                     5922  5933  5949  5960  5977  5991  6002  6013  6024  6035  6052  6055  6082#
                     6100  6110  6120  6134  6144  6154  6163  6174  6190  6201  6214  6225  6235
                     6246  6257  6268  6277  6280  6307#  6324  6334  6344  6358  6368  6378  6387
                     6398  6414  6425  6438  6449  6459  6470  6481  6492  6501  6504  6527#  6543
                     6565  6575  6585  6595  6605  6618  6629  6644  6655  6666  6677  6689  6712#
                     6728  6750  6760  6770  6780  6790  6802  6813  6827  6838  6849  6860  6872
                     6901#  6906  7138  7168#  7184  7203  7213  7223  7233  7243  7252  7263  7273
                     7284  7298  7309  7323  7334  7344  7355  7365  7376  7386  7397  7414  7425
                     7442  7456  7467  7478  7489  7500  7503
T.EDF = 000001       1648#  2107  2151  2198  2269  2368  2640  2744  2777  2788  2822  2833  2867
```

```
                    2878    2912    2923    2957    2968    3000    3011    3047    3058    3071    3082    3114    3126
                    3499    3512    3538    3548    3561    3571    3584    3594    3607    3617    3630    3640
T.EHRD= 000002      1648#
T.ESF = 000000      1648#
T.ESFT= 000003      1648#
T1      023232 G    1086    4634#
T1MODE= 000300      1521#
T2      023532 G    1087    4771#
T2MODE= 000040      1531#
T3      026236 G    1088    5820#
T4      027114 G    1089    6081#
T5      027656 G    1090    6306#
T6      030420 G    1091    6526#
T7      031102 G    1092    6711#
T8      031564 G    1093    6900#
T8.1    031606      6910#
T8.2    032060      6989#
T9      032530 G    1094    7167#
UAM   = 000200 G    1204#
UMAINT= 000001      1406#   6928    7013    7019
UNIT    002410      1729#
UPBITS  002566      1804#
UREGS   002242      1694#   2394    2397*   2400    4367    4368    4369    4370    4384    4385    4386    4387
USTATR= 122000      1361#   2399    2732    2768    2813    2858    2903    2948    3105    3207    3491    3718
USYREG  002476      1771#   2625
USYRT = 120400      1253#   2395    2408    2412
VECTOR  033756      7529    7555#
VIA   = 120000      1375#   2426    2433
VIAACR= 120013      1519#   2486    2495    2573    2582    2696    3175    4649
VIADPA= 120003      1425#   2687    6918    6997
VIADPB= 120002      1413#   2684    6915    6994
VIAIER= 120016      1592#   2482    2569    2702
VIAIFR= 120015      1570#
VIAMS = 120001      1397#
VIAORA= 120017      1612#   2690    6926    6948    6967    7007    7040    7071    7102    7121
VIAORB= 120000      1381#   2617    2620    2693    3155    3158    3191    3768    4683    4696    4823    4826    6921
                    6945    6964    7000    7037    7068    7099    7118
VIAPCR= 120014      1557#   2699
VIASR = 120012      1509#   4706
VIAT1A= 120004      1439#   2508
VIAT1B= 120005      1451#   2513    3797
VIAT1C= 120006      1464#   2499    3178
VIAT1D= 120007      1475#   2504    3181
VIAT2A= 120010      1486#   2586
VIAT2B= 120011      1498#   2591
VREGS   002262      1697#   2425    4290    4291    4292    4293    4307    4308    4309    4310    4324    4325    4326
                    4327    4341    4342    4343    4344
WAITSO  005270      2659#
WRIBYT  002360      1717#
WRILOC= 000002      1243#   2256    2266
WRIPAG= 000004      1245#
WRITE   003700      2229#   2351
WRITEI  003712      2253#   2481    2494    2498    2503    2507    2512    2568    2581    2585    2590    2616    2619
                    2683    2686    2689    2692    2695    2698    2701    3154    3157    3161    3165    3171    3174
                    3177    3180    3190    3194    3197    3376    3429    3767    3796    4648    4682    4695    4786
                    4822    4825    4828    4833    4844    4847    6914    6917    6920    6944    6963    6993    6996
```

```
                        6999    7036    7067    7098    7117
WSR0    002202          1669#   2319*   4244
WSR10   002212          1677#   2323*   4261
WSR12   002214          1679#   2324*   4260
WSR14   002216          1681#   2325*   4259
WSR16   002220          1683#   2326*   4258
WSR2    002204          1671#   2320*   4243
WSR4    002206          1673#   2321*   4242
WSR6    002210          1675#   2322*   4241
XDATA   002330          1704#   3918    3967    4012    4083    4146*   4147*
XORGB   021262          3916    3965    4010    4081    4144#
XYZ   = 000007          1330#
X$ALWA= 000000           959#
X$FALS= 000040           959#
X$OFFS= 000400           959#
X$TRUE= 000020           959#
$E    = 000113          1648#   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#   4809#
                        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#
                        5240#   6935#   6954#   6973#   7027#   7046#   7058#   7077#   7089#   7108#   7127#
$LSTIN= 000001           964#
$LSTTA= 000001           965#
$T    = 000011          1648#   4618#   4752#   5796#   6057#   6282#   6506#   6691#   6874#   7140#
.     = 034514           955#   1694#   1697#   1781#   2032#   2053#   2069#   3817#   4641    4646    4651    4656    4673
                        4678    4685    4690    4698    4703    4708    4713    4721    4726    4732    4737    4743    4748
                        4779    4816    4838    4843    4852    4857    4881    4901    4914    4927    4941    4954    4967
                        4980    4993    5006    5022    5035    5051    5064    5080    5093    5109    5122    5138    5151
                        5164    5180    5193    5209    5222    5236    5795#   5835    5840    5845    5850    5855    5860
                        5869    5874    5879    5884    5889    5894    5899    5904    5909    5914    5918    5923    5929
                        5934    5945    5950    5956    5961    5973    5978    5987    5992    5998    6003    6009    6014
                        6020    6025    6031    6036    6048    6053    6096    6101    6106    6111    6116    6121    6130
                        6135    6140    6145    6150    6155    6159    6164    6170    6175    6186    6191    6197    6202
                        6210    6215    6221    6226    6231    6236    6242    6247    6253    6258    6264    6269    6273
                        6278    6320    6325    6330    6335    6340    6345    6354    6359    6364    6369    6374    6379
                        6383    6388    6394    6399    6410    6415    6421    6426    6434    6439    6445    6450    6455
                        6460    6466    6471    6477    6482    6488    6493    6497    6502    6539    6544    6561    6566
                        6571    6576    6581    6586    6591    6596    6601    6606    6614    6619    6625    6630    6640
                        6645    6651    6656    6662    6667    6673    6678    6724    6729    6746    6751    6756    6761
                        6766    6771    6776    6781    6786    6791    6798    6803    6809    6814    6823    6828    6834
                        6859    6845    6850    6856    6861    6907    6942    6961    7034    7053    7065    7084    7096
                        7115    7180    7185    7199    7204    7209    7214    7219    7224    7229    7234    7239    7244
                        7248    7253    7259    7264    7269    7274    7280    7285    7294    7299    7305    7310    7319
                        7324    7330    7335    7340    7345    7351    7356    7361    7366    7372    7377    7382    7387
                        7393    7398    7410    7415    7421    7426    7438    7443    7452    7457    7463    7468    7474
                        7479    7485    7490    7496    7501    7579#
```

BCOMPL	1#	9590	4439	4446	4453	4460	4482				
BERROR	1#	9590									
BGNAU	1#	9590	4613								
BGNAUT	1#	9590	4533								
BGNCLN	1#	9590	4583								
BGNDU	1#	9590	4597								
BGNHRD	1#	9590	7518								
BGNHW	1#	9590	1104								
BGNINI	1#	9590	4416								
BGNMOD	1#	9590	961								
BGNMSG	1#	9590	3850	3865	3904	3939	3984	4029	4064	4099	
BGNPRO	1#	9590	4403								
BGNPTA	1#	9590									
BGNRPT	1#	9590									
BGNSEG	1#	9590									
BGNSET	1#	9590									
BGNSFT	1#	9590	7569								
BGNSRV	1#	9590									
BGNSUB	1#	9590	6909	6988							
BGNSW	1#	9590	1129								
BGNTST	1#	9590	4634	4771	5820	6081	6306	6526	6711	6900	7167
BNCUMP	1#	9590									
BNERRO	1#	9590									
BREAK	1#	9590									
BRESET	1#	9590	4508	4600							
CKLOOP	1#	9590									
CLOCK	1#	9590									
CLOSE	1#	9590									
CLRVEC	1#	9590	4558								
COMMEN	1#	9590									
DELAY	1#	9590									
DESCRI	1#	9590	2060								
DEVTYP	1#	9590	2048								
DISPAT	1#	9590	1083								
DISPLA	1#	9590									
DOCLN	1#	9590									
DODU	1#	9590	4563								
DORPT	1#	9590									
ENDAU	1#	9590	4615								
ENDAUT	1#	9590	4569								
ENDCLN	1#	9590	4587								
ENDCOM	1#	9590									
ENDDU	1#	9590	4602								
ENDHRD	1#	9590	7551								
ENDHW	1#	9590	1120								
ENDINI	1#	9590	4514								
ENDMOD	1#	9590	7587								
ENDMSG	1#	9590	3859	3898	3932	3976	4021	4056	4091	4126	
ENDPRO	1#	9590	4408								
ENDPTA	1#	9590									
ENDRPT	1#	9590									
ENDSEG	1#	9590									
ENDSET	1#	9590									
ENDSFT	1#	9590	7573								
ENDSRV	1#	9590									
ENDSUB	1#	9590	6979	7133							

```
ENDSW       1#    9590    1134
ENDTST      1#    9590    4749    5247    6054    6279    6503    6688    6871    7137    7502
EQUALS      1#    9590    1144
ERRDF       1#    9590    4810    4875    4895    4908    4921    4935    4948    4961    4974    4987    5000    5016    5029
          5045    5058    5074    5087    5103    5116    5132    5145    5158    5174    5187    5203    5216    5230    5241
          6936    6955    6974    7028    7047    7059    7078    7090    7109    7128
ERRMRD      1#    9590
ERROR       1#    9590    4642    4652    4674    4686    4699    4709    4722    4733    4744    4839    4853    5836    5846
          5856    5870    5880    5890    5900    5910    5919    5930    5946    5957    5974    5988    5999    6010    6021
          6032    6049    6097    6107    6117    6131    6141    6151    6160    6171    6187    6198    6211    6222    6232
          6243    6254    6265    6274    6321    6331    6341    6355    6365    6375    6384    6395    6411    6422    6435
          6446    6456    6467    6478    6489    6498    6540    6562    6572    6582    6592    6602    6615    6626    6641
          6652    6663    6674    6725    6747    6757    6767    6777    6787    6799    6810    6824    6835    6846    6857
          7181    7200    7210    7220    7230    7240    7249    7260    7270    7281    7295    7306    7320    7331    7341
          7352    7362    7373    7383    7394    7411    7422    7439    7453    7464    7475    7486    7497
ERRSF       1#    9590
ERRSOF      1#    9590
ERRTBL      1#    9590    1659
ESCAPE      1#    9590    4644    4654    4676    4688    4701    4711    4724    4735    4746    4814    4841    4855    4879
          4899    4912    4925    4939    4952    4965    4978    4991    5004    5020    5033    5049    5062    5078    5091
          5107    5120    5136    5149    5162    5178    5191    5207    5220    5234    5838    5848    5858    5872    5882
          5892    5902    5912    5921    5932    5948    5959    5976    5990    6001    6012    6023    6034    6051    6099
          6109    6119    6133    6143    6153    6162    6173    6189    6200    6213    6224    6234    6245    6256    6267
          6276    6323    6333    6343    6357    6367    6377    6386    6397    6413    6424    6437    6448    6458    6469
          6480    6491    6500    6542    6564    6574    6584    6594    6604    6617    6628    6643    6654    6665    6676
          6727    6749    6759    6769    6779    6789    6801    6812    6826    6837    6848    6859    6940    6959    7032
          7051    7063    7082    7094    7113    7183    7202    7212    7222    7232    7242    7251    7262    7272    7283
          7297    7308    7322    7333    7343    7354    7364    7375    7385    7396    7413    7424    7441    7455    7466
          7477    7488    7499
EXIT        1#    9590    4777    6905
FEQUAL      1#    9590
GEDF      1648#    4808    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    5239    6934
          6953    6972    7026    7045    7057    7076    7088    7107    7126
GEHRD     1648#
GESF      1648#
GESFT     1648#
GETBYT      1#    9590
GETPRI      1#    9590
GETWOR      1#    9590
GMANIA      1#    9590
GMANID      1#    9590
GMANIL      1#    9590
GPHARD      1#    9590    4478
GPRMA       1#    9590    7522    7527
GPRMD       1#    9590    7532    7538    7544
GPRML       1#    9590
GTDF      1648#    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
GTHRD     1648#
GTSF      1648#
GTSFT     1648#
HEADER      1#    9590     985
INLOOP      1#    9590
IOSETU      1#    9590
```

```
                7052#  7064#  7083#  7095#  7114#  7184#  7203#  7213#  7223#  7233#  7243#  7252#  7263#  7273#  7284#
                7298#  7309#  7323#  7334#  7344#  7355#  7365#  7376#  7386#  7397#  7414#  7425#  7442#  7456#  7467#
                7478#  7489#  7500#
M$EXCP             1#  9559#  7523#  7528#  7533#  7539#  7545#
M$EXIT             1#  9559#  4778#  4779   6906#  6907
M$EXSE             1#  9559#  4778#  6906#
M$EXTJ             1#  9559#  4778#  6906#
M$GEN              1#  9559#  9629#  9869#  9959#  9979#  9999#  1001#  1003#  1005#  1007#  1009#  1011#  1013#  1015#
               1017#  1019#  1021#  1023#  1025#  1027#  1029#  1032#  1035#  1037#  1039#  1041#  1043#  1045#  1047#
               1049#  1051#  1053#  1055#  1057#  1059#  1061#  1063#  1065#  1067#  1069#  1085#  1106#  1107#  1121#
               1131#  1132#  1135#  1660#  2049#  2061#  3851#  3860#  3866#  3899#  3905#  3933#  3940#  3977#  3985#
               4022#  4030#  4057#  4065#  4092#  4100#  4127#  4404#  4417#  4515#  4534#  4570#  4584#  4588#  4598#
               4603#  4614#  4616#  4634#  4750#  4771#  5248#  5820#  6055#  6081#  6280#  6306#  6504#  6526#  6689#
               6711#  6872#  6900#  6910#  6980#  6989#  7134#  7138#  7167#  7503#  7520#  7553#  7571#  7575#  7594#
M$GENB             1#  9559#
M$GETS             1#  9559#  1121#  1135#  3860#  3899#  3933#  3977#  4022#  4057#  4092#  4127#  4409#  4515#  4570#
               4588#  4603#  4616#  4750#  5248#  6055#  6280#  6504#  6689#  6872#  6980#  7134#  7138#  7503#  7552#
               7574#  7588#
M$GETT             1#  9559#  4645#  4655#  4677#  4689#  4702#  4712#  4725#  4736#  4747#  4778#  4815#  4842#  4856#
               4880#  4900#  4913#  4926#  4940#  4953#  4966#  4979#  4992#  5005#  5021#  5034#  5050#  5063#  5079#
               5092#  5108#  5121#  5137#  5150#  5163#  5179#  5192#  5208#  5221#  5235#  5839#  5849#  5859#  5873#
               5883#  5893#  5903#  5913#  5922#  5933#  5949#  5960#  5977#  5991#  6002#  6013#  6024#  6035#  6052#
               6100#  6110#  6120#  6134#  6144#  6154#  6163#  6174#  6190#  6201#  6214#  6225#  6235#  6246#  6257#
               6268#  6277#  6324#  6334#  6344#  6358#  6368#  6378#  6387#  6398#  6414#  6425#  6438#  6449#  6459#
               6470#  6481#  6492#  6501#  6543#  6565#  6575#  6585#  6595#  6605#  6618#  6629#  6644#  6655#  6666#
               6677#  6728#  6750#  6760#  6770#  6780#  6790#  6802#  6813#  6827#  6838#  6849#  6860#  6906#  6941#
               6960#  7033#  7052#  7064#  7083#  7095#  7114#  7184#  7203#  7213#  7223#  7233#  7243#  7252#  7263#
               7273#  7284#  7298#  7309#  7323#  7334#  7344#  7355#  7365#  7376#  7386#  7397#  7414#  7425#  7442#
               7456#  7467#  7478#  7489#  7500#
M$GNGB             1#  9559#  9629#  9869#  9959#  9979#  9999#  1001#  1003#  1005#  1007#  1009#  1011#  1013#  1015#
               1017#  1019#  1021#  1023#  1025#  1027#  1029#  1032#  1035#  1037#  1039#  1041#  1043#  1045#  1047#
               1049#  1051#  1053#  1055#  1057#  1059#  1061#  1063#  1065#  1067#  1069#  1084#  1085   1105#  1106
               1107   1130#  1131   1132   1660#  2049#  2061#  3851#  3866#  3905#  3940#  3985#  4030#  4065#  4100#
               4404#  4417#  4534#  4584#  4598#  4614#  7519#  7520   7570#  7571   7591#  7594
M$GNIN             1#  9559#  9869#  987    988    989    990    991    9929#  9939#  9949#  9959#  996    997#  998
               9999#  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   1067#  1068   1069#  1070   1084#  1086#  1087#
               1088#  1089#  1090#  1091#  1092#  1093#  1094#  1105#  1130#  2049#  2050   2053   2061#  2062   2069
               2107   2108   2109   2110   2151   2152   2153   2154   2198   2199   2200   2201   2269   2270   2271
               2272   2368   2369   2370   2371   2640   2641   2642   2643   2744   2745   2746   2747   2777   2778
               2779   2780   2788   2789   2790   2791   2822   2823   2824   2825   2833   2834   2835   2836   2867
               2868   2869   2870   2878   2879   2880   2881   2912   2913   2914   2915   2923   2924   2925   2926
               2957   2958   2959   2960   2968   2969   2970   2971   3000   3001   3002   3003   3011   3012   3013
               3014   3047   3048   3049   3050   3058   3059   3060   3061   3071   3072   3073   3074   3082   3083
               3084   3085   3114   3115   3116   3117   3126   3127   3128   3129   3305#  3306#  3307   3308#  3309
               3316#  3317#  3318   3319#  3320   3327#  3328#  3329   3330#  3331   3344#  3345#  3346#  3347   3348#
               3349   3499   3500   3501   3502   3512   3513   3514   3515   3538   3539   3540   3541   3548   3549
               3550   3551   3561   3562   3563   3564   3571   3572   3573   3574   3584   3585   3586   3587   3594
               3595   3596   3597   3607   3608   3609   3610   3617   3618   3619   3620   3630   3631   3632   3633
               3640   3641   3642   3643   3854#  3855#  3856   3857#  3858   3861#  3873#  3874#  3875#  3876   3877#
               3878   3888#  3889#  3890#  3891#  3892   3893#  3894   3900#  3909#  3910#  3911#  3912#  3913   3914#
               3915   3918#  3919#  3920#  3921#  3922#  3923   3924#  3925   3927#  3928#  3929   3930#  3931   3934#
               3942#  3943#  3944#  3945#  3946   3947#  3948   3950#  3951#  3952   3953#  3954   3958#  3959#  3960#
```

3961#	3962	3963#	3964	3967#	3968#	3969#	3970#	3971#	3972	3973#	3974	3978#	3987#	3988#
3989#	3990#	3991	3992#	3993	3995#	3996#	3997	3998#	3999	4003#	4004#	4005#	4006#	4007
4008#	4009	4012#	4013#	4014#	4015#	4016#	4017	4018#	4019	4023#	4032#	4033#	4034#	4035#
4036	4037#	4038	4040#	4041#	4042	4043#	4044	4048#	4049#	4050#	4051#	4052	4053#	4054
4058#	4067#	4068#	4069#	4070#	4071	4072#	4073	4075#	4076#	4077#	4078	4079#	4080	4083#
4084#	4085#	4086#	4087#	4088	4089#	4090	4093#	4102#	4103#	4104#	4105#	4106	4107#	4108
4110#	4111#	4112	4113#	4114	4118#	4119#	4120#	4121#	4122	4123#	4124	4128#	4157#	4158#
4159#	4160#	4161	4162#	4163	4165#	4166#	4167#	4168#	4169#	4170#	4171	4172#	4173	4175#
4176#	4177#	4178	4179#	4180	4182#	4183#	4184#	4185#	4186#	4187#	4188	4189#	4190	4192#
4193#	4194#	4195	4196#	4197	4199#	4200#	4201#	4202#	4203#	4204#	4205	4206#	4207	4209#
4210#	4211#	4212	4213#	4214	4216#	4217#	4218#	4219#	4220#	4221#	4222	4223#	4224	4233#
4234#	4235#	4236#	4237	4238#	4239	4241#	4242#	4243#	4244#	4245#	4246#	4247	4248#	4249
4251#	4252#	4253#	4254	4255#	4256	4258#	4259#	4260#	4261#	4262#	4263#	4264	4265#	4266
4268#	4269#	4270	4271#	4272	4282#	4283#	4284#	4285#	4286	4287#	4288	4290#	4291#	4292#
4293#	4294#	4295#	4296	4297#	4298	4300#	4301#	4302#	4303	4304#	4305	4307#	4308#	4309#
4310#	4311#	4312#	4313	4314#	4315	4317#	4318#	4319#	4320	4321#	4322	4324#	4325#	4326#
4327#	4328#	4329#	4330	4331#	4332	4334#	4335#	4336#	4337	4338#	4339	4341#	4342#	4343#
4344#	4345#	4346#	4347	4348#	4349	4359#	4360#	4361#	4362#	4363	4364#	4365	4367#	4368#
4369#	4370#	4371#	4372#	4373	4374#	4375	4377#	4378#	4379#	4380	4381#	4382	4384#	4385#
4386#	4387#	4388#	4389#	4390	4391#	4392	4437#	4438#	4440#	4444#	4445#	4447#	4451#	4452#
4454#	4458#	4459#	4461#	4479#	4480#	4481#	4483#	4509#	4516#	4537#	4538#	4539#	4540#	4541#
4542	4559#	4560#	4564#	4565#	4571#	4589#	4601#	4604#	4617#	4643#	4645#	4646#	4653#	4655#
4656#	4675#	4677#	4678#	4687#	4689#	4690#	4700#	4702#	4703#	4710#	4712#	4713#	4723#	4725#
4726#	4734#	4736#	4737#	4745#	4747#	4748#	4751#	4778#	4779#	4810#	4811#	4812#	4813#	4815#
4816#	4840#	4842#	4843#	4854#	4856#	4857#	4875#	4876#	4877#	4878#	4880#	4881#	4895#	4896#
4897#	4898#	4900#	4901#	4908#	4909#	4910#	4911#	4913#	4914#	4921#	4922#	4923#	4924#	4926#
4927#	4935#	4936#	4937#	4938#	4940#	4941#	4948#	4949#	4950#	4951#	4953#	4954#	4961#	4962#
4963#	4964#	4966#	4967#	4974#	4975#	4976#	4977#	4979#	4980#	4987#	4988#	4989#	4990#	4992#
4993#	5000#	5001#	5002#	5003#	5005#	5006#	5016#	5017#	5018#	5019#	5021#	5022#	5029#	5030#
5031#	5032#	5034#	5035#	5045#	5046#	5047#	5048#	5050#	5051#	5058#	5059#	5060#	5061#	5063#
5064#	5074#	5075#	5076#	5077#	5079#	5080#	5087#	5088#	5089#	5090#	5092#	5093#	5103#	5104#
5105#	5106#	5108#	5109#	5116#	5117#	5118#	5119#	5121#	5122#	5132#	5133#	5134#	5135#	5137#
5138#	5145#	5146#	5147#	5148#	5150#	5151#	5158#	5159#	5160#	5161#	5163#	5164#	5174#	5175#
5176#	5177#	5179#	5180#	5187#	5188#	5189#	5190#	5192#	5193#	5203#	5204#	5205#	5206#	5208#
5209#	5216#	5217#	5218#	5219#	5221#	5222#	5230#	5231#	5232#	5233#	5235#	5236#	5241#	5242#
5243#	5244#	5249#	5837#	5839#	5840#	5847#	5849#	5850#	5857#	5859#	5860#	5871#	5873#	5874#
5881#	5883#	5884#	5891#	5893#	5894#	5901#	5903#	5904#	5911#	5913#	5914#	5920#	5922#	5923#
5931#	5933#	5934#	5947#	5949#	5950#	5958#	5960#	5961#	5975#	5977#	5978#	5989#	5991#	5992#
6000#	6002#	6003#	6011#	6013#	6014#	6022#	6024#	6025#	6033#	6035#	6036#	6050#	6052#	6053#
6056#	6098#	6100#	6101#	6108#	6110#	6111#	6118#	6120#	6121#	6132#	6134#	6135#	6142#	6144#
6145#	6152#	6154#	6155#	6161#	6163#	6164#	6172#	6174#	6175#	6188#	6190#	6191#	6199#	6201#
6202#	6212#	6214#	6215#	6223#	6225#	6226#	6233#	6235#	6236#	6244#	6246#	6247#	6255#	6257#
6259#	6266#	6268#	6269#	6275#	6277#	6278#	6281#	6322#	6324#	6325#	6332#	6334#	6335#	6342#
6344#	6345#	6356#	6358#	6359#	63 6#	6368#	6369#	6376#	6378#	6379#	6385#	6387#	6388#	6396#
6398#	6399#	6412#	6414#	6415#	6423#	6425#	6426#	6436#	6438#	6439#	6447#	6449#	6450#	6457#
6459#	6460#	6468#	6470#	6471#	6479#	6481#	6482#	6490#	6492#	6493#	6499#	6501#	6502#	6505#
6541#	6543#	6544#	6563#	6565#	6566#	6573#	6575#	6576#	6583#	6585#	6586#	6593#	6595#	6596#
6603#	6605#	6606#	6616#	6618#	6619#	6627#	6629#	6630#	6642#	6644#	6645#	6653#	6655#	6656#
6664#	6666#	6667#	6675#	6677#	6678#	6690#	6726#	6728#	6729#	6748#	6750#	6751#	6758#	6760#
6761#	6768#	6770#	6771#	6778#	6780#	6781#	6788#	6790#	6791#	6800#	6802#	6803#	6811#	6813#
6814#	6825#	6827#	6828#	6836#	6838#	6839#	6847#	6849#	6850#	6858#	6860#	6861#	6873#	6906#
6907#	6911#	6936#	6937#	6938#	6939#	6941#	6942#	6955#	6956#	6957#	6958#	6960#	6961#	6974#
6975#	6976#	6977#	6981#	6990#	7028#	7029#	7030#	7031#	7033#	7034#	7047#	7048#	7049#	7050#
7052#	7053#	7059#	7060#	7061#	7062#	7064#	7065#	7078#	7079#	7080#	7081#	7083#	7084#	7090#
7091#	7092#	7093#	7095#	7096#	7109#	7110#	7111#	7112#	7114#	7115#	7128#	7 #9#	7130#	7131#
7135#	7139#	7182#	7184#	7185#	7201#	7203#	7204#	7211#	7213#	7214#	7221#	7223#	7224#	7231#

```
        7233#  7234#  7241#  7243#  7244#  7250#  7252#  7253#  7261#  7263#  7264#  7271#  7273#  7274#  7282#
        7284#  7285#  7296#  7298#  7299#  7307#  7309#  7310#  7321#  7323#  7324#  7332#  7334#  7335#  7342#
        7344#  7345#  7353#  7355#  7356#  7363#  7365#  7366#  7374#  7376#  7377#  7384#  7386#  7387#  7395#
        7397#  7398#  7412#  7414#  7415#  7423#  7425#  7426#  7440#  7442#  7443#  7454#  7456#  7457#  7465#
        7467#  7468#  7476#  7478#  7479#  7487#  7489#  7490#  7498#  7500#  7501#  7504#  7519#  7523#  7524
        7525   7526   7528#  7529   7530   7531   7533#  7534   7535   7536   7537   7539#  7540   7541   7542
        7543   7545#  7546   7547   7548   7549   7552#  7570#  7574#  7591#  7592#  7593#
M$GNLS     1#   959#
M$GNSU     1#   959#  6910#  6989#
M$GNTA     1#   959#  1121#  1135#  3860#  3899#  3933#  3977#  4022#  4057#  4092#  4127#  4515#  4570#  4588#
        4603#  4616#  4750#  5248#  6055#  6280#  6504#  6689#  6872#  6980#  7134#  7138#  7503#  7552#  7553
        7574#  7575
M$GNTE     1#   959#  4634#  4771#  5820#  6081#  6306#  6526#  6711#  6900#  7167#
M$HAPT     1#   959#   986#
M$HNAP     1#   959#   986#  1025
M$INCR     1#   959#   962#  1105#  1130#  3308#  3319#  3330#  3348#  3851#  3857#  3861#  3866#  3877#  3893#
        3900#  3905#  3914#  3924#  3930#  3934#  3940#  3947#  3953#  3963#  3973#  3978#  3985#  3992#  3998#
        4008#  4018#  4023#  4030#  4037#  4043#  4053#  4058#  4065#  4072#  4079#  4089#  4093#  4100#  4107#
        4113#  4123#  4128#  4162#  4172#  4179#  4189#  4196#  4206#  4213#  4223#  4238#  4248#  4255#  4265#
        4271#  4287#  4297#  4304#  4314#  4321#  4331#  4338#  4348#  4364#  4374#  4381#  4391#  4404#  4417#
        4438#  4445#  4452#  4459#  4480#  4509#  4516#  4534#  4541#  4560#  4565#  4571#  4584#  4589#  4598#
        4601#  4604#  4614#  4617#  4634#  4635#  4643#  4645#  4653#  4655#  4675#  4677#  4687#  4689#  4700#
        4702#  4710#  4712#  4723#  4725#  4734#  4736#  4745#  4747#  4751#  4771#  4772#  4778#  4810#  4815#
        4840#  4842#  4854#  4856#  4875#  4880#  4895#  4900#  4908#  4913#  4921#  4926#  4935#  4940#  4948#
        4953#  4961#  4966#  4974#  4979#  4987#  4992#  5000#  5005#  5016#  5021#  5029#  5034#  5045#  5050#
        5058#  5063#  5074#  5079#  5087#  5092#  5103#  5108#  5116#  5121#  5132#  5137#  5145#  5150#  5158#
        5163#  5174#  5179#  5187#  5192#  5203#  5208#  5216#  5221#  5230#  5235#  5241#  5249#  5820#  5821#
        5837#  5839#  5847#  5849#  5857#  5859#  5871#  5873#  5881#  5883#  5891#  5893#  5901#  5903#  5911#
        5913#  5920#  5922#  5931#  5933#  5947#  5949#  5958#  5960#  5975#  5977#  5989#  5991#  6000#  6002#
        6011#  6013#  6022#  6024#  6033#  6035#  6050#  6052#  6056#  6081#  6082#  6098#  6100#  6108#  6110#
        6118#  6120#  6132#  6134#  6142#  6144#  6152#  6154#  6161#  6163#  6172#  6174#  6188#  6190#  6199#
        6201#  6212#  6214#  6223#  6225#  6233#  6235#  6244#  6246#  6255#  6257#  6266#  6268#  6275#  6277#
        6281#  6306#  6307#  6322#  6324#  6332#  6334#  6342#  6344#  6356#  6358#  6366#  6368#  6376#  6378#
        6385#  6387#  6396#  6398#  6412#  6414#  6423#  6425#  6436#  6438#  6447#  6449#  6457#  6459#  6468#
        6470#  6479#  6481#  6490#  6492#  6499#  6501#  6505#  6526#  6527#  6541#  6543#  6563#  6565#  6573#
        6575#  6583#  6585#  6593#  6595#  6603#  6605#  6616#  6618#  6627#  6629#  6642#  6644#  6653#  6655#
        6664#  6666#  6675#  6677#  6690#  6711#  6712#  6726#  6728#  6748#  6750#  6758#  6760#  6768#  6770#
        6778#  6780#  6788#  6790#  6800#  6802#  6811#  6813#  6825#  6827#  6836#  6838#  6847#  6849#  6858#
        6860#  6873#  6900#  6901#  6906#  6910#  6911#  6936#  6941#  6955#  6960#  6974#  6981#  6989#  6990#
        7028#  7033#  7047#  7052#  7059#  7064#  7078#  7083#  7090#  7095#  7109#  7114#  7128#  7135#  7139#
        7167#  7168#  7182#  7184#  7201#  7203#  7211#  7213#  7221#  7223#  7231#  7233#  7241#  7243#  7250#
        7252#  7261#  7263#  7271#  7273#  7282#  7284#  7296#  7298#  7307#  7309#  7321#  7323#  7332#  7334#
        7342#  7344#  7353#  7355#  7363#  7365#  7374#  7376#  7384#  7386#  7395#  7397#  7412#  7414#  7423#
        7425#  7440#  7442#  7454#  7456#  7465#  7467#  7476#  7478#  7487#  7489#  7498#  7500#  7504#  7519#
        7570#
M$IOSE     1#   959#
M$LDRO     1#   959#  4437#  4444#  4451#  4458#  4479#  4559#  4564#
M$MASK     1#   959#
M$MCHI     1#   959#
M$MCLO     1#   959#
M$MSK1     1#   959#
M$POP      1#   959#  1121#  1135#  3860#  3899#  3933#  3977#  4022#  4057#  4092#  4127#  4409#  4515#  4570#
        4588#  4603#  4616#  4750#  5248#  6055#  6280#  6504#  6689#  6872#  6980#  7134#  7138#  7503#  7552#
        7574#  7588#
M$PRIN     1#   959#  3305#  3316#  3327#  3344#  3854#  3873#  3888#  3909#  3918#  3927#  3942#  3950#  3958#
        3967#  3987#  3995#  4003#  4012#  4032#  4040#  4048#  4067#  4075#  4083#  4102#  4110#  4118#  4157#
```

```
        4165#  4175#  4182#  4192#  4199#  4209#  4216#  4233#  4241#  4251#  4258#  4268#  4282#  4290#  4300#
        4307#  4317#  4324#  4334#  4341#  4359#  4367#  4377#  4384#
M$PUSH     1#   959#   962#  1105#  1130#  3851#  3866#  3905#  3940#  3985#  4030#  4065#  4100#  4404#  4417#
        4534#  4584#  4598#  4614#  4634#  4635   4771#  4772   5820#  5821   6081#  6082   6306#  6307   6526#
        6527   6711#  6712   6900#  6901   6910#  6911   6989#  6990   7167#  7168   7519#  7570#
M$PUT      1#   959#  3305#  3316#  3327#  3344#  3854#  3873#  3888#  3909#  3918#  3927#  3942#  3950#  3958#
        3967#  3987#  3995#  4003#  4012#  4032#  4040#  4048#  4067#  4075#  4083#  4102#  4110#  4118#  4157#
        4165#  4175#  4182#  4192#  4199#  4209#  4216#  4233#  4241#  4251#  4258#  4268#  4282#  4290#  4300#
        4307#  4317#  4324#  4334#  4341#  4359#  4367#  4377#  4384#  4537#
M$PUT1     1#   959#  3305#  3306   3316#  3317   3327#  3328   3344#  3345   3346   3854#  3855   3873#  3874
        3875   3888#  3889   3890   3891   3909#  3910   3911   3912   3918#  3919   3920   3921   3922   3927#
        3928   3942#  3943   3944   3945   3950#  3951   3958#  3959   3960   3961   3967#  3968   3969   3970
        3971   3987#  3988   3989   3990   3995#  3996   4003#  4004   4005   4006   4012#  4013   4014   4015
        4016   4032#  4033   4034   4035   4040#  4041   4048#  4049   4050   4051   4067#  4068   4069   4070
        4075#  4076   4077   4083#  4084   4085   4086   4087   4102#  4103   4104   4105   4110#  4111   4118#
        4119   4120   4121   4157#  4158   4159   4160   4165#  4166   4167   4168   4169   4170   4175#  4176
        4177   4182#  4183   4184   4185   4186   4187   4192#  4193   4194   4199#  4200   4201   4202   4203
        4204   4209#  4210   4211   4216#  4217   4218   4219   4220   4221   4233#  4234   4235   4236   4241#
        4242   4243   4244   4245   4246   4251#  4252   4253   4258#  4259   4260   4261   4262   4263   4268#
        4269   4282#  4283   4284   4285   4290#  4291   4292   4293   4294   4295   4300#  4301   4302   4307#
        4308   4309   4310   4311   4312   4317#  4318   4319   4324#  4325   4326   4327   4328   4329   4334#
        4335   4336   4341#  4342   4343   4344   4345   4346   4359#  4360   4361   4362   4367#  4368   4369
        4370   4371   4372   4377#  4378   4379   4384#  4385   4386   4387   4388   4389   4537#  4538   4539
        4540
M$RADI     1#   959#  7523#  7528#  7533#  7539#  7545#
M$RBRO     1#   959#
M$RNRO     1#   959#  4479#  4481
M$SETS     1#   959#   962#  1105#  1130#  3851#  3866#  3905#  3940#  3985#  4030#  4065#  4100#  4404#  4417#
        4534#  4584#  4598#  4614#  4635#  4772#  5821#  6082#  6307#  6527#  6712#  6901#  6911#  6990#  7168#
        7519#  7570#
M$STAR     1#   959#
M$SVC      1#   959#  3305#  3306   3316#  3319   3327#  3330   3344#  3348   3854#  3857   3860#  3861   3873#
        3877   3888#  3893   3899#  3900   3909#  3914   3918#  3924   3927#  3930   3933#  3934   3942#  3947
        3950#  3953   3958#  3963   3967#  3973   3977#  3978   3987#  3992   3995#  3998   4003#  4008   4012#
        4018   4022#  4023   4032#  4037   4040#  4043   4048#  4053   4057#  4058   4067#  4072   4075#  4079
        4083#  4089   4092#  4093   4102#  4107   4110#  4113   4118#  4123   4127#  4128   4157#  4162   4165#
        4172   4175#  4179   4182#  4189   4192#  4196   4199#  4206   4209#  4213   4216#  4223   4233#  4238
        4241#  4248   4251#  4255   4258#  4265   4268#  4271   4282#  4287   4290#  4297   4300#  4304   4307#
        4314   4317#  4321   4324#  4331   4334#  4338   4341#  4348   4359#  4364   4367#  4374   4377#  4381
        4384#  4391   4437#  4438   4444#  4445   4451#  4452   4458#  4459   4479#  4480   4509#  4515#  4516
        4537#  4541   4559#  4560   4564#  4565   4570#  4571   4588#  4589   4601#  4603#  4604   4616#  4617
        4643#  4645#  4653#  4655#  4675#  4677#  4687#  4689#  4700#  4702#  4710#  4712#  4723#  4725#  4734#
        4736#  4745#  4747#  4750#  4751   4778#  4810   4815#  4840#  4842#  4854#  4856#  4875   4880#  4895
        4900#  4908   4913#  4921   4926#  4935   4940#  4948   4953#  4961   4966#  4974   4979#  4987   4992#
        5000   5005#  5016   5021#  5029   5034#  5045   5050#  5058   5063#  5074   5079#  5087   5092#  5103
        5108#  5116   5121#  5132   5137#  5145   5150#  5158   5163#  5174   5179#  5187   5192#  5203   5208#
        5216   5221#  5230   5235#  5241   5248#  5249   5837#  5839#  5847#  5849#  5857#  5859#  5871#  5873#
        5881#  5883#  5891#  5893#  5901#  5903#  5911#  5913#  5920#  5922#  5931#  5933#  5947#  5949#  5958#
        5960#  5975#  5977#  5989#  5991#  6000#  6002#  6011#  6015#  6022#  6024#  6033#  6035#  6050#  6052#
        6055#  6056   6098#  6100#  6108#  6110#  6118#  6120#  6132#  6134#  6142#  6144#  6152#  6154#  6161#
        6163#  6172#  6174#  6188#  6190#  6199#  6201#  6212#  6214#  6223#  6225#  6233#  6235#  6244#  6246#
        6255#  6257#  6266#  6268#  6275#  6277#  6280#  6281   6322#  6324#  6332#  6334#  6342#  6344#  6356#
        6358#  6366#  6368#  6376#  6378#  6385#  6387#  6396#  6398#  6412#  6414#  6423#  6425#  6436#  6438#
        6447#  6449#  6457#  6459#  6468#  6470#  6479#  6481#  6490#  6492#  6499#  6501#  6504#  6505   6541#
        6543#  6563#  6565#  6573#  6575#  6583#  6585#  6593#  6595#  6603#  6605#  6616#  6618#  6627#  6629#
        6642#  6644#  6653#  6655#  6664#  6666#  6675#  6677#  6689#  6690   6726#  6728#  6748#  6750#  6758#
```

	6760#	6768#	6770#	6778#	6780#	6788#	6790#	6800#	6802#	6811#	6813#	6825#	6827#	6836#	6838#
	6847#	6849#	6858#	6860#	6872#	6873	6906#	6910#	6911	6936	6941#	6955	6960#	6974	6980#
	6981	6989#	6990	7028	7033#	7047	7052#	7059	7064#	7078	7083#	7090	7095#	7109	7114#
	7128	7134#	7135	7138#	7139	7182#	7184#	7201#	7203#	7211#	7213#	7221#	7223#	7231#	7233#
	7241#	7243#	7250#	7252#	7261#	7263#	7271#	7273#	7282#	7284#	7296#	7298#	7307#	7309#	7321#
	7323#	7332#	7334#	7342#	7344#	7353#	7355#	7363#	7365#	7374#	7376#	7384#	7386#	7395#	7397#
	7412#	7414#	7423#	7425#	7440#	7442#	7454#	7456#	7465#	7467#	7476#	7478#	7487#	7489#	7498#
	7500#	7503#	7504												
M$TLAB	1#	959#	3308#	3319#	3330#	3348#	3857#	3861#	3877#	3893#	3900#	3914#	3924#	3930#	3934#
	3947#	3953#	3963#	3973#	3978#	3992#	3998#	4008#	4018#	4023#	4037#	4043#	4053#	4058#	4072#
	4079#	4089#	4093#	4107#	4113#	4123#	4128#	4162#	4172#	4179#	4189#	4196#	4206#	4213#	4223#
	4238#	4248#	4255#	4265#	4271#	4287#	4297#	4304#	4314#	4321#	4331#	4338#	4348#	4364#	4374#
	4381#	4391#	4438#	4445#	4452#	4459#	4480#	4509#	4516#	4541#	4560#	4565#	4571#	4589#	4601#
	4604#	4617#	4643#	4645#	4653#	4655#	4675#	4677#	4687#	4689#	4700#	4702#	4710#	4712#	4723#
	4725#	4734#	4736#	4745#	4747#	4751#	4778#	4810#	4815#	4840#	4842#	4854#	4856#	4875#	4880#
	4895#	4900#	4908#	4913#	4921#	4926#	4935#	4940#	4948#	4953#	4961#	4966#	4974#	4979#	4987#
	4992#	5000#	5005#	5016#	5021#	5029#	5034#	5045#	5050#	5058#	5063#	5074#	5079#	5087#	5092#
	5103#	5108#	5116#	5121#	5132#	5137#	5145#	5150#	5158#	5163#	5174#	5179#	5187#	5192#	5203#
	5208#	5216#	5221#	5230#	5235#	5241#	5249#	5837#	5839#	5847#	5849#	5857#	5859#	5871#	5873#
	5881#	5883#	5891#	5893#	5901#	5903#	5911#	5913#	5920#	5922#	5931#	5933#	5947#	5949#	5958#
	5960#	5975#	5977#	5989#	5991#	6000#	6002#	6011#	6013#	6022#	6024#	6033#	6035#	6050#	6052#
	6056#	6098#	6100#	6108#	6110#	6118#	6120#	6132#	6134#	6142#	6144#	6152#	6154#	6161#	6163#
	6172#	6174#	6188#	6190#	6199#	6201#	6212#	6214#	6223#	6225#	6233#	6235#	6244#	6246#	6255#
	6257#	6266#	6268#	6275#	6277#	6281#	6322#	6324#	6332#	6334#	6342#	6344#	6356#	6358#	6366#
	6368#	6376#	6378#	6385#	6387#	6396#	6398#	6412#	6414#	6423#	6425#	6436#	6438#	6447#	6449#
	6457#	6459#	6468#	6470#	6479#	6481#	6490#	6492#	6499#	6501#	6505#	6541#	6543#	6563#	6565#
	6573#	6575#	6583#	6585#	6593#	6595#	6603#	6605#	6616#	6618#	6627#	6629#	6642#	6644#	6653#
	6655#	6664#	6666#	6675#	6677#	6690#	6726#	6728#	6748#	6750#	6758#	6760#	6768#	6770#	6778#
	6780#	6788#	6790#	6800#	6802#	6811#	6813#	6825#	6827#	6836#	6838#	6847#	6849#	6858#	6860#
	6873#	6906#	6911#	6936#	6941#	6955#	6960#	6974#	6981#	6990#	7028#	7033#	7047#	7052#	7059#
	7064#	7078#	7083#	7090#	7095#	7109#	7114#	7128#	7135#	7139#	7182#	7184#	7201#	7203#	7211#
	7213#	7221#	7223#	7231#	7233#	7241#	7243#	7250#	7252#	7261#	7263#	7271#	7273#	7282#	7284#
	7296#	7298#	7307#	7309#	7321#	7323#	7332#	7334#	7342#	7344#	7353#	7355#	7363#	7365#	7374#
	7376#	7384#	7386#	7395#	7397#	7412#	7414#	7423#	7425#	7440#	7442#	7454#	7456#	7465#	7467#
	7476#	7478#	7487#	7489#	7498#	7500#	7504#								
M$TSTL	1#	959#	3308#	3319#	3330#	3348#	3857#	3861#	3877#	3893#	3900#	3914#	3924#	3930#	3934#
	3947#	3953#	3963#	3973#	3978#	3992#	3998#	4008#	4018#	4023#	4037#	4043#	4053#	4058#	4072#
	4079#	4089#	4093#	4107#	4113#	4123#	4128#	4162#	4172#	4179#	4189#	4196#	4206#	4213#	4223#
	4238#	4248#	4255#	4265#	4271#	4287#	4297#	4304#	4314#	4321#	4331#	4338#	4348#	4364#	4374#
	4381#	4391#	4438#	4445#	4452#	4459#	4480#	4509#	4516#	4541#	4560#	4565#	4571#	4589#	4601#
	4604#	4617#	4643#	4645#	4653#	4655#	4675#	4677#	4687#	4689#	4700#	4702#	4710#	4712#	4723#
	4725#	4734#	4736#	4745#	4747#	4751#	4778#	4810#	4815#	4840#	4842#	4854#	4856#	4875#	4880#
	4895#	4900#	4908#	4913#	4921#	4926#	4935#	4940#	4948#	4953#	4961#	4966#	4974#	4979#	4987#
	4992#	5000#	5005#	5016#	5021#	5029#	5034#	5045#	5050#	5058#	5063#	5074#	5079#	5087#	5092#
	5103#	5108#	5116#	5121#	5132#	5137#	5145#	5150#	5158#	5163#	5174#	5179#	5187#	5192#	5203#
	5208#	5216#	5221#	5230#	5235#	5241#	5249#	5837#	5839#	5847#	5849#	5857#	5859#	5871#	5873#
	5881#	5883#	5891#	5893#	5901#	5903#	5911#	5913#	5920#	5922#	5931#	5933#	5947#	5949#	5958#
	5960#	5975#	5977#	5989#	5991#	6000#	6002#	6011#	6013#	6022#	6024#	6033#	6035#	6050#	6052#
	6056#	6098#	6100#	6108#	6110#	6118#	6120#	6132#	6134#	6142#	6144#	6152#	6154#	6161#	6163#
	6172#	6174#	6188#	6190#	6199#	6201#	6212#	6214#	6223#	6225#	6233#	6235#	6244#	6246#	6255#
	6257#	6266#	6268#	6275#	6277#	6281#	6322#	6324#	6332#	6334#	6342#	6344#	6356#	6358#	6366#
	6368#	6376#	6378#	6385#	6387#	6396#	6398#	6412#	6414#	6423#	6425#	6436#	6438#	6447#	6449#
	6457#	6459#	6468#	6470#	6479#	6481#	6490#	6492#	6499#	6501#	6505#	6541#	6543#	6563#	6565#
	6573#	6575#	6583#	6585#	6593#	6595#	6603#	6605#	6616#	6618#	6627#	6629#	6642#	6644#	6653#
	6655#	6664#	6666#	6675#	6677#	6690#	6726#	6728#	6748#	6750#	6758#	6760#	6768#	6770#	6778#
	6780#	6788#	6790#	6800#	6802#	6811#	6813#	6825#	6827#	6836#	6838#	6847#	6849#	6858#	6860#

```
                6873#  6906#  6911#  6936#  6941#  6955#  6960#  6974#  6981#  6990#  7028#  7033#  7047#  7052#  7059#
                7064#  7078#  7083#  7090#  7095#  7109#  7114#  7128#  7135#  7139#  7182#  7184#  7201#  7203#  7211#
                7213#  7221#  7223#  7231#  7233#  7241#  7243#  7250#  7252#  7261#  7263#  7271#  7273#  7282#  7284#
                7296#  7298#  7307#  7309#  7321#  7323#  7332#  7334#  7342#  7344#  7353#  7355#  7363#  7365#  7374#
                7376#  7384#  7386#  7395#  7397#  7412#  7414#  7423#  7425#  7440#  7442#  7454#  7456#  7465#  7467#
                7476#  7478#  7487#  7489#  7498#  7500#  7504#
M$WORD             1#   959#  1025#  1034  1084#  1086  1087  1088  1089  1090  1091  1092  1093  1094  4778#
                4810#  4811  4812  4813  4875#  4876  4877  4878  4895#  4896  4897  4898  4908#  4909  4910
                4911  4921#  4922  4923  4924  4935#  4936  4937  4938  4948#  4949  4950  4951  4961#  4962
                4963  4964  4974#  4975  4976  4977  4987#  4988  4989  4990  5000#  5001  5002  5003  5016#
                5017  5018  5019  5029#  5030  5031  5032  5045#  5046  5047  5048  5058#  5059  5060  5061
                5074#  5075  5076  5077  5087#  5088  5089  5090  5103#  5104  5105  5106  5116#  5117  5118
                5119  5132#  5133  5134  5135  5145#  5146  5147  5148  5158#  5159  5160  5161  5174#  5175
                5176  5177  5187#  5188  5189  5190  5203#  5204  5205  5206  5216#  5217  5218  5219  5230#
                5231  5232  5233  5241#  5242  5243  5244  6906#  6936#  6937  6938  6939  6955#  6956  6957
                6958  6974#  6975  6976  6977  7028#  7029  7030  7031  7047#  7048  7049  7050  7059#  7060
                7061  7062  7078#  7079  7080  7081  7090#  7091  7092  7093  7109#  7110  7111  7112  7128#
                7129  7130  7131  7523#  7528#  7533#  7539#  7545#  7592  7593
M$XFER             1#   959#
NEWTST          1648#  4618  4752  5796  6057  6282  6506  6691  6874  7140
NTST            1648#  4618  4752  5796  6057  6282  6506  6691  6874  7140
OPEN               1#   959#
POINTE             1#   959#   982
PRINTB             1#   959#  3853  3908  3917  3926  3941  3949  3957  3966  3986  3994  4002  4011  4031
                4039  4047  4066  4074  4082  4101  4109  4117  4267
PRINTF             1#   959#  3304  3315  3326  3343
PRINTS             1#   959#
PRINTX             1#   959#  3872  3887  4156  4164  4174  4181  4191  4198  4208  4215  4232  4240  4250
                4257  4281  4289  4299  4306  4316  4323  4333  4340  4358  4366  4376  4383
READBU             1#   959#
READEF             1#   959#  4436  4443  4450  4457
RFLAGS             1#   959#
SETDF           1648#  2107  2151  2198  2269  2368  2640  2744  2777  2788  2822  2833  2867  2878  2912
                2923  2957  2968  3000  3011  3047  3058  3071  3082  3114  3126  3499  3512  3538  3548
                3561  3571  3584  3594  3607  3617  3630  3640
SETHRD          1648#
SETPRI             1#   959#
SETSF           1648#
SETSFT          1648#
SETVEC             1#   959#  4536
SLASH              1#   959#  1076  1080
STARS              1#   959#
SVC                1#   957#   958
T$GEN           1648#  2107  2151  2198  2269  2368  2640  2744  2777  2788  2822  2833  2867  2878  2912
                2923  2957  2968  3000  3011  3047  3058  3071  3082  3114  3126  3499  3512  3538  3548
                3561  3571  3584  3594  3607  3617  3630  3640
XFER               1#   959#  4778#  6906#
XFERF              1#   959#
XFERT              1#   959#
$GEDF           1648#  4809  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  5240  6935
                6954  6973  7027  7046  7058  7077  7089  7108  7127
$GEHRD          1648#
$GESF           1648#
$GESFT          1648#
$GTDF           1648#  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
$GTHRD    1648#
$GTSF     1648#
$GTSFT    1648#


. ABS.  034514     000

 ERRORS DETECTED:  0

 CVDMEC,CVDMEC/SOL/CRF=SVC34R.MLB,CVDMEC.P11
 RUN-TIME: 33 41 4 SECONDS
 RUN-TIME RATIO: 104/79=1.3
 CORE USED:  20K  (39 PAGES)

.
```