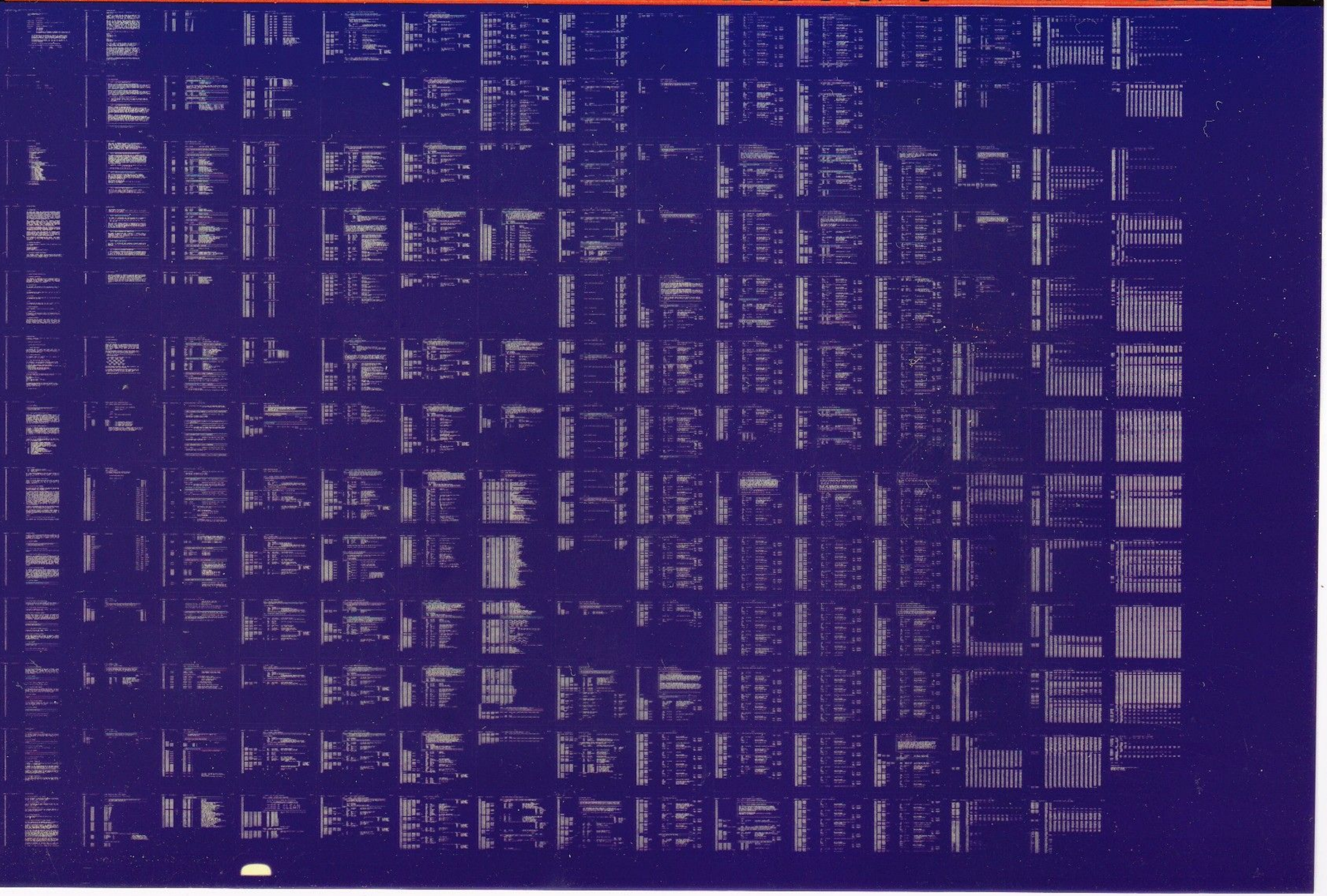
DMV11 DMV11 LINE UNIT DIAG 2 AH-F271A-MC
M8053,M8064 CVDMDA0 FICHE 1 OF 1



```
                        .TITLE CVDMDA0 DMV11 LINE UNIT DIAG2
                        .SBTTL PROGRAM DOCUMENT
                        .REM @


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

                        PRODUCT CODE:   AC-F270A-MC

                        PRODUCT NAME:   CVDMDA0 DMV-11 LINE UNIT STATIC DIAGNOSTIC PART #2

                        PRODUCT DATE:   JANUARY 1981

                        MAINTAINER:     DIAGNOSTICS MERRIMACK  CC:38P

                        AUTHORS:        CHRIS BRIENEN
                                        DAVE HOFFMAN
                                        RAY MARSHALL

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


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

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

                                COPYRIGHT (C) 1981 BY DIGITAL EQUIPMENT CORPORATION

                        THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                                DIGITAL         PDP             UNIBUS          MASSBUS
                                DEC             DECUS           DECTAPE
```

HISTORY

REV	DATE	REASON
0	14-JAN-81	INITIAL RELEASE

CONTENTS

1.0 INTRODUCTION

THE M8053 AND M8064 ARE SINGLE-LINE SYNCHRONOUS, MICRO-PROCESSOR BASED COMMUNICATIONS INTERFACES WHICH CAN SUPPORT BOTH CHARACTER-ORIENTED (DDCMP, BSC, ETC.) AND BIT-ORIENTED (SDLC, HDLC, ETC.) PROTOCOLS. THE PURPOSE OF THIS PROGRAM IS TO PERFORM STATIC DIAGNOSTIC TESTING OF THE VIA, FIFO, AND USYRT (BCP/BOP MODES) ON THESE BOARDS. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: VRC/CRC ERROR DETECTION AND ASSORTED BOP SPECIFIC FUNCTIONS (BIT STUFFING, ABORTS, FLAGS, SECONDARY STATION ADDRESSING, ETC).

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

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

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

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

2.0 HARDWARE REQUIREMENTS

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

PDP-11/03 OR PDP-11/23
16K WORDS OF MEMORY
CONSOLE TERMINAL
M8053 OR M8064 COMMUNICATIONS INTERFACE

3.0 PRELIMINARY PROGRAM REQUIREMENTS

THIS PROGRAM (CVDMD) SHOULD BE THE FOURTH OF THE FIVE DMV-11 STATIC DIAGNOSTICS TO BE RUN (CVDMA/B/C SHOULD BE RUN FIRST). ERRORS FOUND IN THIS PROGRAM SHOULD BE CORRECTED BEFORE RUNNING THE FINAL LINE UNIT DIAGNOSTIC (CVDME).

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 20 SECONDS PER PASS FOR EACH UNIT (10 SECONDS IF PDP-11/23).

4.3 XXDP+

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

4.4 ACT/SLIDE

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

4.5 APT

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

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM.

4.7 ERROR LOGGING

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

5.0 PROGRAM LOAD MEDIA

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

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

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CVDMD-A-0
DMV-11 LINE UNIT TESTS - PART 2 OF 3
UNIT IS M8053 OR M8064
DR>
```

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

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

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

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

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

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

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

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

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

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

```
325
326                    ISR   INHIBIT STATISTICAL REPORTS
327                    IDU   INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
328                    LOT   LOOP ON TEST
329
330     THE  FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0
331     ARE CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS
332     SWITCH  IS  NOT GIVEN ALL FLAGS ARE CLEARED.  SEE EXAMPLE AT
333     END OF 6.3.1.5.
334
335
336     6.3.1.4 END OF PASS SWITCH (/EOP:<INCR>)
337
338     <INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
339     PASSES)  IT  IS  DESIRED  THAT  THE  END  OF PASS MESSAGE BE
340     PRINTED.  THE DEFAULT IS AT THE  END  OF  EVERY  PASS.   SEE
341     EXAMPLE AT END OF 6.3.1.5.
342
343
344     6.3.1.5 EFFECT OF START COMMAND
345
346     THE  EFFECT OF THE START COMMAND IS TO INITIATE THE HARDWARE
347     PARAMETER DIALOGUE, THE  SOFTWARE  PARAMETER  DIALOGUE,  AND
348     THEN THE DIAGNOSTIC TESTS THEMSELVES.
349
350     THE  HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION
351     "# UNITS?" TO WHICH THE  OPERATOR  REPLIES  WITH  A  DECIMAL
352     NUMBER  N FROM 1 TO 16. THE TERM 'UNIT" REFERS TO THE DEVICE
353     TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
354     THIS  ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES WILL
355     BE BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE  CONTAINING
356     ALL  THE  HARDWARE  INFORMATION  FOR ONE UNIT.  THE OPERATOR
357     MUST SUPPLY N (NUMBER OF UNITS) VALUES  FOR  EACH  QUESTION.
358     HE  MAY  DO  THIS  BY GIVING ONE ANSWER TO EACH QUESTION (IN
359     WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
360     BY  GIVING  N  VALUES, SEPARATED BY COMMAS, TO EACH QUESTION
361     (SERIES WILL BE POSED ONCE).  EACH QUESTION IS  FOLLOWED  BY
362     THE  RESPONSE  RADIX  (D  FOR  DECIMAL,  B FOR BINARY, O FOR
363     OCTAL, L FOR YES/NO) IN PARENTHESES AND  THE  DEFAULT  VALUE
364     AFTER THE PARENTHESES.
365
366     FOLLOWING  THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS
367     TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE  MODE  (QUICK
368     VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.
369
370     WHEN  THE QUESTION "# UNITS?" IS ANSWERED, MEMORY STORAGE IS
371     ALLOCATED FOR THE P-TABLES, AND IF THERE IS  NOT  ENOUGH  TO
372     ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED.  IN
373     THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE  TO
374     TEST ALL UNITS.
375
376     EXAMPLE:
377
378     STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE
379
380     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>
**********************************************************************
```

437
438 6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)
439
440 <PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS
441 THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART.
442 IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.
443
444
445 6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)
446
447 <FLAG-LIST> IS SAME AS IN START COMMAND, BUT UNSPECIFIED
448 FLAGS RETAIN THEIR CURRENT VALUE.
449
450
451 6.3.3.3 EFFECT OF CONTINUE COMMAND
452
453 CONTINUE MUST FOLLOW A START OR RESTART, AND COMMAND MODE
454 MUST HAVE BEEN ENTERED DUE TO A HALT ON ERROR OR A
455 CONTROL/C. THE EFFECT OF THE COMMAND IS TO GO TO THE
456 BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE HALT
457 OR CONTROL/C TOOK PLACE. SOFTWARE DIALOGUE MAY OPTIONALLY
458 BE REEXECUTED. HARDWARE PARAMETERS MAY NOT BE CHANGED.
459
460
461 6.3.4 PROCEED COMMAND
462
463 **********
464 PRO(CEED)/FLAGS:<FLAG-LIST>
465 **********
466
467
468 6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
469
470 <FLAG-LIST> IS AS IN THE START COMMAND, BUT UNSPECIFIED
471 FLAGS RETAIN THEIR CURRENT VALUE.
472
473
474 6.3.4.2 EFFECT OF PROCEED COMMAND
475
476 PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE. COMMAND
477 MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR. THE EFFECT
478 OF THE COMMAND IS TO BEGIN EXECUTION AT THE LOCATION
479 FOLLOWING THE ERROR CALL. NEITHER HARDWARE NOR SOFTWARE
480 PARAMETERS MAY BE ALTERED.
481
482
483 6.3.5 ADD COMMAND
484
485 **********
486 ADD/UNITS:<UNIT-LIST>
487 **********
488
489
490 6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>
491
492 <UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 EFFECT OF ADD COMMAND

THE UNITS SPECIFIED ARE ADDED TO THE TEST SEQUENCE. EACH UNIT MUST HAVE A P-TABLE IN MEMORY DUE TO AN EARLIER HARDWARE DIALOGUE. THIS COMMAND MUST BE FOLLOWED BY A RESTART OR CONTINUE. THE UNITS SWITCH MUST BE SPECIFIED. THE ADD COMMAND IS MEANINGFUL ONLY FOR UNITS THAT WERE PREVIOUSLY DROPPED.

6.3.6 DROP COMMAND

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

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

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

6.3.6.2 EFFECT OF DROP COMMAND

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

6.3.7 PRINT COMMAND

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

6.3.7.1 EFFECT OF PRINT COMMAND

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

6.3.8 DISPLAY COMMAND

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

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

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

549
550
551 6.3.8.2 EFFECT OF DISPLAY COMMAND
552
553 THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED
554 OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS
555 THAT WERE DROPPED BY THE OPERATOR 'DROP'' COMMAND ARE SO
556 DESIGNATED.
557
558
559 6.3.9 FLAGS COMMAND
560
561 **********
562 FLA(GS)
563 **********
564
565
566 6.3.9.1 EFFECT OF FLAGS COMMAND
567
568 THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.
569
570
571 6.3.10 ZFLAGS COMMAND
572
573 **********
574 ZFL(AGS)
575 **********
576
577
578 6.3.10.1 EFFECT OF ZFLAGS COMMAND
579
580 ALL FLAGS ARE CLEARED.
581
582
583 6.3.11 CONTROL CHARACTERS
584
585 A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES
586 A RETURN TO COMMAND MODE.
587
588 A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR
589 DIALOGUES- HARD CORE QUESTIONS (SEE 6.2),HARDWARE DIALOGUE
590 (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE
591 DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.
592
593 A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC
594 CAUSES ALL TELETYPE OUTPUT TO BE SURPRESSED FOR THE
595 REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED,
596 WHICH RESTORES NORMAL TELETYPE OUTPUT.
597
598
599 6.3.12 HARDWARE PARAMETERS
600
601 THE FOLLOWING 3 QUESTIONS WILL BE ASKED ON A START COMMAND.
602 THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE
603 DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN
604 RESPONSE.

1. DEVICE CSR ADDRESS : (O) 160020?

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

2. DEVICE VECTOR ADDRESS : (O) 300 ?

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

3. DEVICE PRIORITY LEVEL · (O) 4 ?

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

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1 OF THE STATIC LOGIC TESTS.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

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

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

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

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

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

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING

VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

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

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

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

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

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

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

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

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

```
7.0 TEST DESCRIPTIONS

;*************************************************************************
;*      TEST 1  <VRC PARITY GENERATION TEST>
;*
;* SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC. AND 7-BIT CHARS SELECTED.
;* THE DATA CHARS IN PATTERN Q ARE LOADED/TRANSMITTED/READ. AS THE 8TH BIT
;* (PARITY BIT) OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TSO FOR THE PROPER
;* STATE. FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1, FOR THE
;* LAST 4 CHARACTERS IT SHOULD = 0.
;*
;* SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
;* THE DATA CHARS IN PATTERN Q ARE LOADED/TRANSMITTED/READ. AS THE 8TH BIT
;* (PARITY BIT) OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TSO FOR THE PROPER
;* STATE. FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0, FOR THE
;* LAST 4 CHARACTERS IT SHOULD = 1.
;*
;*    DATA PATTERN Q = 000,003,014,060,001,007,037,177
;*
;* NOTE: SINCE THE ROUTINE ''SERIAL'' TREATS THE FIRST BIT RECEIVED FROM THE
;*       USYRT AS THE MSB, THE ''EXPECTED BIT SEQUENCE'' WILL HAVE A REVERSED
;*       BIT ORDER.
;*************************************************************************


;*************************************************************************
;*      TEST 2  <VRC ERROR DETECTION TEST>
;*
;* SUBTEST 1 - FORCING OF RERR USING ODD VRC
;* THE USYRT IS PLACED IN CHAR MODE WITH ODD VRC AND BOTH TX AND RX CHAR
;* LENGTH=7 BITS. THE RECEIVER AND TRANSMITTER ARE THEN SYNC'D. WHEN THE FIRST
;* DATA CHARACTER IS LOADED INTO TXDB, THE RX CHAR LENGTH IS CHANGED TO 6 BITS.
;* TWO 7 BIT CHARACTERS (+PARITY) ARE THEN TRANSMITTED, RESULTING IN A 16 BIT
;* STREAM WHICH THE RECEIVER WILL READ AS TWO 6 BIT CHARS (+PARITY + 2 LEFT).
;* THE FIRST ''CHARACTER'' READ WILL HAVE THE CORRECT PARITY; THE SECOND WILL
;* NOT.
;*
;* SUBTEST 2 - FORCING OF RERR USING EVEN VRC
;* THE USYRT IS PLACED IN CHAR MODE WITH EVEN VRC AND BOTH TX AND RX CHAR
;* LENGTH=7 BITS. THE RECEIVER AND TRANSMITTER ARE THEN SYNC'D. WHEN THE FIRST
;* DATA CHARACTER IS LOADED INTO TXDB, THE RX CHAR LENGTH IS CHANGED TO 6 BITS.
;* TWO 7 BIT CHARACTERS (+PARITY) ARE THEN TRANSMITTED, RESULTING IN A 16 BIT
;* STREAM WHICH THE RECEIVER WILL READ AS TWO 6 BIT CHARS (+PARITY + 2 LEFT).
;* THE FIRST ''CHARACTER'' READ WILL HAVE THE CORRECT PARITY; THE SECOND WILL
;* NOT.
;*
;*************************************************************************


;*************************************************************************
;*      TEST 3  <BCP CRC GENERATION/DETECTION TEST>
;*
```

```
772                     ;*  THIS  TEST  IS COMPOSED OF 2 SUBTESTS -- #1 EXPECTS GOOD CRC
773                     ;*  GENERATION  AND  REPORT  ERRORS  -- #2 FORCES AN ERROR AND ONLY
774                     ;*  REPORT  WHEN THE CRC IS ACCEPTED AS GOOD. EACH IS
775                     ;*  RUN  AT  THE CHARACTER LENGTHS OF 8 BITS FOR THE ENTIRITY
776                     ;*  OF  EACH  MESSAGE.  BOTH THE TRANSMITTER AND RECEIVER WILL BE SET TO
777                     ;*  THE  SAME  CHARACTER  LENGTH.  ERROR  LOOPING WILL BE ON THE FAILING
778                     ;*  SUBTEST. TEXT STRINGS WILL BE LIMITED TO 5 CHARACTERS.
779                     ;*****************************************************************************
780
781
782                     ;*****************************************************************************
783                     ;*      TEST 4  <BOP RX BASIC RECEIVE/FLAG RECOGNITION TEST>
784                     ;*
785                     ;*  THE  USYRT IS INITIALIZED FOR BOP MODE WITH TTL LOOPBACK SELECTED.
786                     ;*  'SECONDARY STATION ADDRESS'' IS NOT USED AND NO CRC/VRC IS CALCULATED.
787                     ;*  A PATTERN  IS TRANSMITTED AND TERMINATED FOLLOWED BY A SECOND MESSAGE.
788                     ;*  TERMINATION  OF  THE  FIRST  MESSAGE  IS  ACCOMPLISHED  WITH  A FLAG
789                     ;*  CHARACTER  BUT  RXE IS NOT DROPPED SO THAT THE SECOND MESSAGE CAN BE
790                     ;*  SENT WITHOUT RE-SYNCRONIZITION. SEVERAL FLAG'S ARE IDLED BETWEEN THE
791                     ;*  TWO MESSAGES. DURING THE SECOND MESSAGE A RECEIVER OVERRUN CONDITION
792                     ;*  IS FORCED. THROUGHOUT THIS TEST, BASIC RECEIVER OPERATION AND TIMING
793                     ;*  IS CHECKED. TRANSMITTED INFORMATION IS VERIFIED BY CHECKING THE DATA
794                     ;*  MADE AVAILABLE AT RXDB.
795                     ;*
796                     ;*     TRANSMITTED  PATTERN:  FLAG FLAG 123 321 000 377 101 FLAG... FLAG
797                     ;*                                  321 123 377 000 276.
798                     ;*
799                     ;*     RECEIVED PATTERN: 123 321 000 377 101 ..... 321 123.
800                     ;*****************************************************************************
801
802
803                     ;*****************************************************************************
804                     ;*      TEST 5  <BOP RX SECONDARY STATION ADDRESSING>
805                     ;*
806                     ;*  THE  USYRT IS INITIALIZED FOR BOP MODE WITH TTL LEVEL LOOPBACK,
807                     ;*  SAM = 1, APA=0, AND ECM = 7. USING SHORT MESSAGES, THE ADDRESSES
808                     ;*  000, 125, 252, 176, AND 177 ARE CHECKED TO SEE THAT THE RECEIVER
809                     ;*  RECOGNIZES THEM CORRECTLY. IN EACH CASE (AT EACH ADDRESS), A SERIES OF
810                     ;*  20  DIFFERENT  MESSAGES ARE SENT TO VERIFY THAT THE USYRT WILL ONLY
811                     ;*  RESPOND TO THE SPECIFIED VALUE.
812                     ;*
813                     ;*  TEST PATTERN: ADR 000 OCR ADR
814                     ;*  WHERE ADR IS THE ADDRESS BEING TESTED AND OCA IS THE ONE'S
815                     ;*  COMPLEMENT OF THAT ADDRESS.
816                     ;*
817                     ;*****************************************************************************
818
819
820                     ;*****************************************************************************
821                     ;*      TEST 6  <BOP RX ALL PARTIES ADDRESS TEST>
822                     ;*
823                     ;*   INITIALIZE  THE  USYRT  FOR BOP MODE WITH TTL LEVEL LOOPBACK
824                     ;*   SAM = 1, S/AR = 123(OCT.), APA = 1, AND ECM = 7.
825                     ;*   A SERIES OF 256 DIFFERENT SHORT MESSAGES ARE SENT TO VERIFY THAT
826                     ;*   THE USYRT WILL ONLY RESPOND TO THE SPECIFIED VALUE AND ALSO 377 (FF
827                     ;*   HEX.).
```

```
828                     ;*
829                     ;*   TEST PATTERN: ADR 000 OCA ADR
830                     ;*   WHERE  ADR  IS  THE  ADDRESS  BEING  TESTED  AND  OCA  IS THE ONE'S
831                     ;*   COMPLEMENT OF THAT ADDRESS.
832                     ;*********************************************************************************
833
834
835                     ;*********************************************************************************
836                     ;*      TEST 7  <BOP RX BIT STUFFING TEST>
837                     ;*
838                     ;*   THE  USYRT  IS  INITIALIZED  AND  THE FOLLOWING TEXT IS TRANSMITTED
839                     ;*   (DELIMITED BY THE APPROPRIATE CONTROL CHARACTERS -- OF COURSE):
840                     ;*
841                     ;*     000, 017, 036, 074, 170, 360, 037, 076, 174, 370, 077, 176, 374,
842                     ;*     177, 376, 377.
843                     ;*
844                     ;*  NOTE  THAT  THIS  PATTERN  CONSISTS  OF CHARACTERS WHICH REQUIRE BIT
845                     ;*  STUFFING   BOTH   INDIVIDUALLY  AND  IN  COMBINATION  WITH  ADJACENT
846                     ;*  CHARACTERS.  THERE ARE ALSO CHARACTERS WHICH REQUIRE NO BIT STUFFING
847                     ;*  AT  ALL.  ALL 16 CHARACTERS ARE READ BY THE RECEIVER AND COMPARED AS
848                     ;*  THEY ARE MADE AVAILABLE AT RXDB.
849                     ;*********************************************************************************
850
851
852                     ;*********************************************************************************
853                     ;*      TEST 8  <BOP RX UNDERRUN IDLE ABORTS/FLAGS>
854                     ;*
855                     ;*  THE  USYRT  IS  INITIALIZED  AND  A  MESSAGE  IS  STARTED.  THEN,  A
856                     ;*  TRANSMITTER  UNDERRUN  IS  FORCED  WITH  IDLE  =  0 -- CAUSING ABORT
857                     ;*  CHARACTERS  TO  BE  IDLED. THE RECEIVER SHOULD BE RESET BY THE ABORT
858                     ;*  CHARACTER(S). VERIFY THAT RAB/GA BIT=1.
859                     ;*  REPEAT THE ABOVE WITH IDLE=1.
860                     ;*********************************************************************************
861
862
863                     ;*********************************************************************************
864                     ;*      TEST 9  <BOP RX LOST RXE TEST>
865                     ;*
866                     ;*  THE  USYRT  IS  INITIALIZED  AND  A MESSAGE IS STARTED. WHILE IN THE
867                     ;*  MIDDLE  OF  TEXT, RXE IS DROPPED AND THE REACTION OF THE RECEIVER IS
868                     ;*  MONITORED.
869                     ;*********************************************************************************
870
871
872                     ;*********************************************************************************
873                     ;*      TEST 10 <BOP RX GA (GO-AHEAD) RECOGNITION>
874                     ;*
875                     ;*  A  SHORT  MESSAGE IS TRANSMITTED FOLLOWED BY A GA CHARACTER (INSTEAD
876                     ;*  OF  A  FLAG CHARACTER). THE RECEIVER IS OBSERVED FOR PROPER HANDLING
877                     ;*  OF  BOTH  THE  MESSAGE  AND  THE GA CHARACTER. THE RAB/GA STATUS BIT
878                     ;*  SHOULD BE SET BY THE RECEIVER UPON RECOGNITION OF THE GA CHARACTER.
879                     ;*********************************************************************************
880
881
882                     ;*********************************************************************************
883                     ;*      TEST 11 <BOP RX "ABC" TEST>
```

```
884                     ;*
885                     ;*  THIS TEST IS COMPOSED OF 7 SUBTESTS -- EACH ONE CHECKING A DIFFERENT
886                     ;*  EXPECTED VALUE IN ABC (THE 3 BIT "ASSEMBLED BIT COUNT" FIELD WITHIN
887                     ;*  RDSR). IN EACH SUBTEST THE USYRT IS INITIALIZED AND A SMALL MESSAGE
888                     ;*  IS STARTED. THE LAST CHARACTER IS SENT WITH ITS LENGTH BEING
889                     ;*  SPECIFIED FIRST AS 1 BIT, THEN AS 2 BITS, THEN AS 3 BITS, ETC. IN THE
890                     ;*  TRANSMITTER SIDE OF THE USYRT. IN ALL CASES THE RECEIVER IS LEFT SET
891                     ;*  TO 8 BITS IN LENGTH AND WHEN THE FLAG CHARACTER IS DETECTED, ABC IS
892                     ;*  CHECKED AND SHOULD MATCH TXCL. ERROR LOOPING WILL BE ON THE FAILING
893                     ;*  SUBTEST.
894                     ;*
895                     ;*****************************************************************************
```

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

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

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

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

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

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

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

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

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


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

002000                          .=2000

                                .MCALL  SVC
002000                          SVC     ; INITIALIZE SUPERVISOR MACROS


002000                          BGNMOD  LU1MOD


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

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

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

                                POINTER BGNAU,BGNDU,ERRTBL


                                HEADER  CVDMD,A,0,30.,0

                        LSNAME::
002000  103                             .ASCII /C/
002001  126                             .ASCII /V/
002002  104                             .ASCII /D/
002003  115                             .ASCII /M/
002004  104                             .ASCII /D/
002005  000                             .BYTE   0
002006  000                             .BYTE   0
002007  000                             .BYTE   0
002010                  LSREV::
002010  101                             .ASCII  /A/
002011                  LSDEPO::
002011  060                             .ASCII  /0/
002012                  LSUNIT::
002012  000000                          .WORD   0
002014                  LSTIML::
002014  000036                          .WORD   30.
002016                  LSHPCP::
002016  034044                          .WORD   LSHARD
002020                  LSSPCP::
002020  000000                          .WORD   0
002022                  LSHPTP::
002022  002154                          .WORD   LSHW
002024                  LSSPTP::
002024  000000                          .WORD   0
002026                  LSLADP::
002026  034322                          .WORD   LSLAST
002030                  LSSTA::
002030  000000                          .WORD   0
002032                  LSCO::
002032  000000                          .WORD   0
002034                  LSDTYP::
002034  000000                          .WORD   0
002036                  LSAPT::
002036  000000                          .WORD   0
002040                  LSDTP::
002040  002124                          .WORD   LSDISPATCH
002042                  LSPRIO::
002042  000000                          .WORD   0
002044                  LSENVI::
002044  000000                          .WORD   0
002046                  LSEXP1::
002046  000000                          .WORD   0
002050                  LSMREV::
002050  003                             .BYTE   CSREVISION
002051  003                             .BYTE   CSEDIT
```

```
1022  002052
1023  002052  000000
1024  002054  000000
1025  002056
1026  002056  000000
1027  002060
1028  002060  003232
1029  002062
1030  002062  000000
1031  002064
1032  002064  000000
1033  002066
1034  002066  000000
1035  002070
1036  002070  024312
1037  002072
1038  002072  024306
1039  002074
1040  002074  000000
1041  002076
1042  002076  003252
1043  002100
1044  002100  104035
1045  002102
1046  002102  002176
1047  002104
1048  002104  023644
1049  002106
1050  002106  024304
1051  002110
1052  002110  024160
1053  002112
1054  002112  023636
1055  002114
1056  002114  000000
1057  002116
1058  002116  000000
1059  002120
1060  002120  000000
1061
1062
1063                    .EVEN

LSEF::
        .WORD   0
        .WORD   0
LSSPC::
        .WORD   0
LSDEVP::
        .WORD   LSDVTYP
LSREPP::
        .WORD   0
LSEXP4::
        .WORD   0
LSEXP5::
        .WORD   0
LSAUT::
        .WORD   LSAU
LSDUT::
        .WORD   LSDU
LSLUN::
        .WORD   0
LSDESP::
        .WORD   LSDESC
LSLOAD::
        EMT     ESLOAD
LSETP::
        .WORD   LSERRTBL
LSICP::
        .WORD   LSINIT
LSCCP::
        .WORD   LSCLEAN
LSACP::
        .WORD   LSAUTO
LSPRT::
        .WORD   LSPROT
LSTEST::
        .WORD   0
LSDLY::
        .WORD   0
LSHIME::
        .WORD   0
```

```
                                .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 11.

002122  000013                                                  .WORD   11
002124                                                  LSDISPATCH::
002124  024314                                                  .WORD   T1
002126  025222                                                  .WORD   T2
002130  026034                                                  .WORD   T3
002132  026636                                                  .WORD   T4
002134  027464                                                  .WORD   T5
002136  030176                                                  .WORD   T6
002140  030674                                                  .WORD   T7
002142  032040                                                  .WORD   T8
002144  032660                                                  .WORD   T9
002146  033122                                                  .WORD   T10
002150  033364                                                  .WORD   T11
```

```
1088                                    .SBTTL  DEFAULT HARDWARE P-TABLE
1089
1090                                    ;////////////////////////////////////////////////////////////////////////////
1091                                    ;/ THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1092                                    ;/ THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1093                                    ;/ IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
1094                                    ;////////////////////////////////////////////////////////////////////////////
1095
1096  002152                                    BGNHW   DFPTBL
1097  002152  000010                                                                   .WORD   L10000-L$HW/2
1098  002154                                                                    L$HW::
1099  002154                                                                    DFPTBL::
1100
1101  002154  160020                            .WORD   160020          ;DMV11 CSR UNIBUS ADDRESS
1102  002156  000300                            .WORD      300          ;DMV11 INTERRUPT VECTOR
1103  002160  004000                            .WORD     4000          ;DMV11 INTERRUPT PRIORITY LEVEL = 4
1104  002162  000000                            .WORD      000          ;SWITCH REG. #1 (BOOT ADDRESS)
1105  002164  000000                            .WORD      000          ;SWITCH REG. #2 (DDCMP ADDRESS)
1106  002166  000000                            .WORD        0          ;MODULE IS M8064
1107  002170  000000                            .WORD        0          ;M3254&M3255 USED
1108  002172  000001                            .WORD        1          ;BAUD RATE = 56 K
1109                                                                    ;          0 = 19.2 K
1110                                                                    ;          1 = 56 K
1111
1112  002174                                    ENDHW
1113  002174                                                                    L10000:
```

```
1114                                    .SBTTL  SOFTWARE P-TABLE
1115
1116                                    ;////////////////////////////////////////////////////////////////////////
1117                                    ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1118                                    ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1119                                    ;////////////////////////////////////////////////////////////////////////
1120
1121  002174                                    BGNSW   SFPTBL
1122  002174  000000                                                                   .WORD   L10001-L$SW/2
1123  002176                                                                    L$SW::
1124  002176                                                                    SFPTBL::
1125
1126  002176                                    ENDSW
1127  002176                                                                    L10001:
```

```
1128                                    .SBTTL  GLOBAL EQUATES SECTION -- BASIC EQUATES
1129
1130
1131                                    ;////////////////////////////////////////////////////////////////////////
1132                                    ;/      THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1133                                    ;/      ARE USED IN MORE THAN ONE TEST.
1134                                    ;////////////////////////////////////////////////////////////////////////
1135
1136  002176                                    EQUALS
1137                                    ;
1138                                    ; BIT DIFINITIONS
1139                                    ;
1140            100000                  BIT15== 100000
1141            040000                  BIT14== 40000
1142            020000                  BIT13== 20000
1143            010000                  BIT12== 10000
1144            004000                  BIT11== 4000
1145            002000                  BIT10== 2000
1146            001000                  BIT09== 1000
1147            000400                  BIT08== 400
1148            000200                  BIT07== 200
1149            000100                  BIT06== 100
1150            000040                  BIT05== 40
1151            000020                  BIT04== 20
1152            000010                  BIT03== 10
1153            000004                  BIT02== 4
1154            000002                  BIT01== 2
1155            000001                  BIT00== 1
1156                                    ;
1157            001000                  BIT9==  BIT09
1158            000400                  BIT8==  BIT08
1159            000200                  BIT7==  BIT07
1160            000100                  BIT6==  BIT06
1161            000040                  BIT5==  BIT05
1162            000020                  BIT4==  BIT04
1163            000010                  BIT3==  BIT03
1164            000004                  BIT2==  BIT02
1165            000002                  BIT1==  BIT01
1166            000001                  BIT0==  BIT00
1167                                    ;
1168                                    ; EVENT FLAG DEFINITIONS
1169                                    ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
1170                                    ;
1171            000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
1172            000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
1173            000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
1174            000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
1175            000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
1176                                    ;
1177                                    ;
1178                                    ; PRIORITY LEVEL DEFINITIONS
1179                                    ;
1180            000340                  PRI07== 340
1181            000300                  PRI06== 300
1182            000240                  PRI05== 240
1183            000200                  PRI04== 200
```

```
1184    000140          PRI03== 140
1185    000100          PRI02== 100
1186    000040          PRI01== 40
1187    000000          PRI00== 0
1188                    ;
1189                    ;OPERATOR FLAG BITS
1190                    ;
1191    000004          EVL==         4
1192    000010          LOT==        10
1193    000020          ADR==        20
1194    000040          IDU==        40
1195    000100          ISR==       100
1196    000200          UAM==       200
1197    000400          BOE==       400
1198    001000          PNT==      1000
1199    002000          PRI==      2000
1200    004000          IXE==      4000
1201    010000          IBE==     10000
1202    020000          IER==     20000
1203    040000          LOE==     40000
1204    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         ;RECEIVE  ABORT/GO AHEAD CHARACTER
001000                  RXEOM   = BIT9          ;RECEIVE  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
```

```
1298    002000          TXAB    = BIT10         ;TRANSMIT ABORT
1299    001000          TXEOM   = BIT9          ;TRANSMIT END-OF-MESSAGE
1300    000400          TXSOM   = BIT8          ;TRANSMIT START-OF-MESSAGE
1301
1302                    ;;*************************************************************************
1303                    ;* USYRT "SYNC/SECONDARY ADDRESS" REGISTER
1304                    ;;*************************************************************************
1305
1306    120404          PCSARL  = 120404        ;ADDRESS OF THIS REG
1307    000226          SYNCH   = 226           ;STANDARD SYNCH CHARACTER
1308
1309                    ;;*************************************************************************
1310                    ;* USYRT "MODE CONTROL"
1311                    ;;*************************************************************************
1312
1313    120405          PCSARH  = 120405        ;ADDRESS OF THIS REG
1314
1315                    ;BIT DEFINITIONS ON BYTE BASIS:
1316
1317    000200          APA     = BIT7          ;"ALL PARTIES ADDRESS" ENABLE
1318    000100          PROTO   = BIT6          ;SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
1319    000040          STRIP   = BIT5          ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1320    000020          SECAD   = BIT4          ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1321    000010          IDLE    = BIT3          ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1322    000007          XYZ     = BIT2!BIT1!BIT0 ;CRC/PARITY SELECTION CONTROL
1323
1324                    ;BIT DEFINITIONS ON WORD BASIS:
1325
1326    100000          APAD    = BIT15         ;"ALL PARTIES ADDRESS" ENABLE
1327    040000          DDCMP   = BIT14         ;CODE FOR DDCMP MODE
1328    020000          STRIPS  = BIT13         ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1329    010000          SECADR  = BIT12         ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1330    004000          IDLES   = BIT11         ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1331    000400          CRCOS   = BIT8          ;CODE FOR CRC-CCITT-0 SELECTION
1332    001400          CRC16   = BIT9!BIT8     ;CODE FOR CRC-16 SELECTION
1333    003400          NOCHK   = BIT10!BIT9!BIT8 ;CODE FOR NO ERROR CHECKING
1334    002400          EVRC    = BIT10!BIT8    ;CODE FOR VRC EVEN CHECK
1335    002000          OVRC    = BIT10         ;CODE FOR VRC ODD CHECK
1336
1337                    ;;*************************************************************************
1338                    ;* USYRT "DATA LENGTH SELECT" REGISTER
1339                    ;;*************************************************************************
1340
1341    120407          PCR     = 120407        ;ADDRESS OF THIS REG
1342
1343                    ;BIT DEFINITIONS:
1344
1345    000340          TXDL    = BIT7!BIT6!BIT5 ;TRANSMIT DATA LENGTH SELECTION
1346    000020          EXADD   = BIT4          ;EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
1347    000010          EXCON   = BIT3          ;EXTENDED CONTROL FIELD -- NOT USED OR TESTED
1348    000007          RXDL    = BIT2!BIT1!BIT0 ;RECEIVER DATA LENGTH SELECTION
1349
1350                    ;;*************************************************************************
1351                    ;* USYRT STATUS REGISTER (ADDR. A400)
1352                    ;;*************************************************************************
1353    122000          USTATR = 122000         ;USYRT STATUS REGISTER ADDRESS = A400 (HEX)
```

```
1354
1355                                    ;BIT DEFINITIONS:
1356
1357        000200      RDA     = BIT7      ;RECEIVER DATA AVAILABLE
1358        000100      TBMT    = BIT6      ;TRANSMITTER BUFFER EMPTY
1359        000040      RXACT   = BIT5      ;RECEIVER ACTIVE
1360        000020      RSA     = BIT4      ;RECEIVER STATUS AVAILABLE
1361        000010      TSO     = BIT3      ;TRANSMITTER SERIAL OUTPUT
1362        000004      TXACT   = BIT2      ;TRANSMITTER ACTIVE
1363        000002      TXU     = BIT1      ;TRANSMITTER UNDERRUN
1364        000001      SFR     = BIT0      ;SYNC/FLAG RECEIVED
```

```
CVDMDA.P11    10-DEC-80 09:15        REGISTER DEFINITIONS -- 6522 VIA CHIP

                                     .SBTTL  REGISTER DEFINITIONS -- 6522 VIA CHIP

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

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

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

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

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

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

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


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

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

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

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


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

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

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

```
1421
1422                            ;       INITIALIZED TO 001 (HEX = 01) -- PORT A IS READ ONLY (EXCEPT FOR
1423                            ;                                       BIT0 WHICH ENABLES USYRT INTERNAL LOOPBACK).
1424
1425
1426
1427                            ;;*****************************************************************************
1428                            ;* TIMER 1 LOW ORDER (LATCH & COUNTER) -- 'T1L-L'' & 'T1C-L'' -- WRITE & READ
1429                            ;;*****************************************************************************
1430
1431     120004                 VIAT1A  = 120004                ;ADDRESS OF THIS REGISTER -- HEX = A0X4
1432
1433                            ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1434                            ;  WHEN READING, LOW ORDER COUNTER IS READ.
1435
1436
1437
1438                            ;;*****************************************************************************
1439                            ;* TIMER 1 HIGH ORDER COUNTER & TRIGGER -- 'T1L-H AND TRIGGER'' & 'T1C-H''
1440                            ;*       -- WRITE & READ
1441                            ;;*****************************************************************************
1442
1443     120005                 VIAT1B  = 120005                ;ADDRESS OF THIS REGISTER -- HEX = A0X5
1444
1445                            ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1446                            ;                 ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1447
1448                            ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1449
1450
1451
1452                            ;;*****************************************************************************
1453                            ;* TIMER 1 LOW ORDER LATCH -- 'T1L-L'' -- READ/WRITE
1454                            ;;*****************************************************************************
1455
1456     120006                 VIAT1C  = 120006                ;ADDRESS OF THIS REGISTER -- HEX = A0X6
1457
1458                            ;  THE LOW ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1459                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1460
1461
1462
1463                            ;;*****************************************************************************
1464                            ;* TIMER 1 HIGH ORDER LATCH -- 'T1L-H'' -- READ/WRITE
1465                            ;;*****************************************************************************
1466
1467     120007                 VIAT1D  = 120007                ;ADDRESS OF THIS REGISTER -- HEX = A0X7
1468
1469                            ;  THE HIGH ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1470                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1471
1472
1473
1474                            ;;*****************************************************************************
1475                            ;* TIMER 2 LOW ORDER (LATCH & COUNTER) -- 'T2L-L'' & 'T2C-L'' -- WRITE & READ
1476                            ;;*****************************************************************************
```

```
1477
1478          120010          VIAT2A  = 120010                ;ADDRESS OF THIS REGISTER -- HEX = A0X8
1479
1480                          ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1481                          ;  WHEN READING, LOW ORDER COUNTER IS READ.
1482
1483
1484
1485                          ;;*****************************************************************************
1486                          ;* TIMER 2 HIGH ORDER COUNTER & TRIGGER -- 'T2L-H AND TRIGGER' & 'T2C-H'
1487                          ;*      -- WRITE & READ
1488                          ;;*****************************************************************************
1489
1490          120011          VIAT2B  = 120011                ;ADDRESS OF THIS REGISTER -- HEX = A0X9
1491
1492                          ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1493                          ;                ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1494
1495                          ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1496
1497                          ;;*****************************************************************************
1498                          ;* SHIFT REGISTER -- 'SR' -- READ/WRITE
1499                          ;;*****************************************************************************
1500
1501          120012          VIASR   = 120012                ;ADDRESS OF THIS REGISTER -- HEX = A0XA
1502
1503                          ;  SHIFTING IS CONTROLLED BY THE SETTING OF VIASRC (ACR2 ---> ACR4) IN VIAACR
1504
1505
1506
1507                          ;;*****************************************************************************
1508                          ;* AUXILIARY CONTROL REGISTER -- 'ACR' -- READ/WRITE
1509                          ;;*****************************************************************************
1510
1511          120013          VIAACR  = 120013                ;ADDRESS OF THIS REGISTER -- HEX = A0XB
1512
1513          000300          T1MODE  = BIT7!BIT6             ;CONTROL THE MODE OF TIMER # 1
1514
1515                                          ;BIT 7:
1516                                          ; 0     PB7 DISABLED -- ONLY T1TO IN VIAIFR REFLECTS TIMEOUT
1517                                          ; 1     PB7 & T1TO REFLECT TIMEOUT
1518
1519                                          ;BIT 6:
1520                                          ; 0     TIMER 1 IN ONE-SHOT MODE
1521                                          ; 1     TIMER 1 IN CONTINUOUS SQUARE WAVE MODE
1522
1523          000040          T2MODE  = BIT5                  ;CONTROLS THE MODE OF TIMER # 1
1524
1525                                          ; 0     PULSE COUNTING MODE
1526                                          ; 1     INTERVAL TIMER MODE
1527
1528          000034          SRMODE  = BIT4!BIT3!BIT2        ;CONTROLS THE MODE OF THE SHIFT REGISTER
1529
1530                                          ; 0     SR DISABLED
1531                                          ; 1     SHIFT IN UNDER CONTROL OF T2, SHFT PULSES GEN'D ON CB1
1532                                          ; 2     SHIFT IN AT SYS. CLOCK RATE, SHFT PULSES GEN'D ON CB1
```

```
1533                                    ; 3     SHIFT IN UNDER CONTROL OF EXTERNAL INPUT PULSES
1534                                    ; 4     SHIFT OUT -- FREE RUNNING -- RATE CONTROLLED BY T2
1535                                    ; 5     SHIFT OUT -- RATE CONTROLLED BY T2 -- PULSES ON CB1
1536                                    ; 6     SHIFT OUT -- SYS. CLOCK RATE -- PULSES ON CB1
1537                                    ; 7     SHIFT OUT -- UNDER CONTROL OF PULSES APPLIED TO CB1
1538
1539        000002      PBLENB  = BIT1              ;PB LATCH CONTROL -- 1 ENABLES LATCH
1540        000001      PALENB  = BIT0              ;PA LATCH CONTROL -- 1 ENABLES LATCH
1541
1542
1543
1544
1545                    ;;*************************************************************************
1546                    ;* PERIPHERAL CONTROL REGISTER -- "PCR" -- READ/WRITE
1547                    ;;*************************************************************************
1548
1549        120014      VIAPCR  = 120014            ;ADDRESS OF THIS REGISTER -- HEX = A0XC
1550
1551        000340      CB2CTL  = BIT7!BIT6!BIT5    ;CB2 MODE SELECT
1552        000020      CB1CTL  = BIT4              ;CB1 MODE SELECT
1553        000016      CA2CTL  = BIT3!BIT2!BIT1    ;CA2 MODE SELECT
1554        000001      CA1CTL  = BIT0              ;CA1 MODE SELECT
1555
1556
1557
1558                    ;;*************************************************************************
1559                    ;* INTERRUPT FLAG REGISTER -- "IFR" -- READ ONLY
1560                    ;;*************************************************************************
1561
1562        120015      VIAIFR  = 120015            ;ADDRESS OF THIS REGISTER -- HEX = A0XD
1563
1564        000200      FLGIRQ  = BIT7              ;SET WHEN A FLAG IN THIS REG. GOES HIGH AND
1565                                                ;ITS CORRESPONDING BIT IN VIAIER IS SET.
1566                                                ;(I.E. VIAIER IS THE ENABLE REGISTER FOR THE
1567                                                ;FOR THE SETTING OF IRQ AND THE ISSUEANCE OF
1568                                                ;AN INTERRUPT TO THE 6502 WHEN IRQ IS SET.)
1569
1570        000100      FLGT1   = BIT6              ;TIMEOUT OF TIMER 1
1571        000040      FLGT2   = BIT5              ;TIMEOUT OF TIMER 2
1572        000020      FLGCB1  = BIT4              ;ACTIVE TRANSITION OF PIN 18 (CB1)
1573        000010      FLGCB2  = BIT3              ;ACTIVE TRANSITION OF PIN 19 (CB2)
1574        000004      FLGSR   = BIT2              ;COMPLETION OF 8 SHIFTS
1575        000002      FLGCA1  = BIT1              ;ACTIVE TRANSITION OF PIN 40 (CA1)
1576        000001      FLGCA2  = BIT0              ;ACTIVE TRANSITION OF PIN 39 (CA2)
1577
1578
1579
1580                    ;;*************************************************************************
1581                    ;* INTERRUPT ENABLE REGISTER -- "IER" -- READ/WRITE
1582                    ;;*************************************************************************
1583
1584        120016      VIAIER  = 120016            ;ADDRESS OF THIS REGISTER -- HEX = A0XE
1585
1586        000200      INTSC   = BIT7              ;CONTROLS THE SETTING OR CLEARING OF BITS IN
1587                                                ;THE REST OF IER.  IF = 0  THE OTHER BITS IN
1588                                                ;THIS REG., IF SET, WILL CLEAR THEIR RESPECTIVE
```

```
1589                                        ;BITS IN THE INT. ENAB. REG..  IF = 1, THE
1590                                        ;RESPECTIVE BITS WILL BE SET.
1591
1592                    ;  WHEN WRITING THIS REG., THE COMMENT ABOVE HOLDS.
1593
1594                    ;  WHEN READING THIS REG., THE CURRENT STATE OF THE INT. ENABLE REG. IS RETURNED.
1595
1596                    ;  THE BIT ASSIGNMENTS ARE THE SAME AS FOR VIAIFR AS DEFINED ABOVE.
1597
1598
1599
1600                    ;;*************************************************************************
1601                    ;* OUTPUT REGISTER A -- 'ORA'' -- READ ONLY (OR READ/WRITE UNDER CONTROL OF 'DDPA')
1602                    ;;*************************************************************************
1603
1604      120017        VIAORA  = 120017            ;ADDRESS OF THIS REGISTER -- HEX = A0XF
1605
1606                    ;  THIS ADDRESS ACCESSES THE SAME DATA AS 'VIAMS'' EXCEPT THAT NO 'HANDSHAKING''
1607                    ;  WILL TAKE PLACE (I.E. THERE IS NO CHANGE IN IRQ OR CA2 AS A RESULT OF
1608                    ;  READING ORA THROUGH THIS ADDRESS)
1609
1610                    ;THE BIT ASSIGNMENTS ARE THE SAME AS FOR 'VIAMS'' ABOVE.
1611
1612
1613
```

```
1614                            .SBTTL  REGISTER DEFINITIONS -- MISC
1615
1616                            ;;*****************************************************************************
1617                            ;* SWITCH PACKS
1618                            ;;*****************************************************************************
1619
1620    121000                  SWPBOT  = 121000                ;'BOOT ADDRESS'' SWITCH PACK [A200]
1621    121400                  SWPDDCMP = 121400               ;'DDCMP ADDRESS'' SWITCH PACK [A300]
1622
1623                            ;MISCELLANEOUS EQUATES
1624
1625    100000                  TCCHEK  = BIT15         ;FLAG TO REQUEST H3254,5 CHECK
1626    001000                  RAMADR  = 001000        ;STARTING ADRS OF RAM PAGE 2 (ADRS 0200 HEX)
1627
1628    000002                  EIAV35  = BIT1                  ;SELECT V.35 OR EIA 423/232C
1629    000001                  INTGRL  = BIT0                  ;SELECT INTEGRAL MODEM
1630
1631    040000                  NORXEN  = BIT14         ;KILL RXEN DURING ''INITRN''
1632    001000                  NOLOOP  = BIT9          ;KILL TTLOOP DURING ''INITRN''
1633
1634    000200                  NCTBMT  = BIT7          ;DISABLE INITIAL TBMT=0 CHECK IN TXCHAR
1635
1636    100000                  NOCRDA  = BIT15         ;DISABLE INITIAL RDA=0 CHECK IN RXCHAR
1637    040000                  NFCRDA  = BIT14         ;DISABLE FINAL RDA=1 CHECK IN RXCHAR
1638    020000                  NCRACT  = BIT13         ;DISABLE RXACT=1 CHECK AFTER CLOCKING (RXCHAR)
1639
```

```
1640                            .SBTTL  GLOBAL DATA SECTION
1641
1642                            ;////////////////////////////////////////////////////////////////////////////
1643                            ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1644                            ;/      IN MORE THAN ONE TEST.
1645                            ;////////////////////////////////////////////////////////////////////////////
1646
1647                            ;++*************************************************************************
1648                            ; CONTROL BLOCK FOR STACKED ERROR MESSAGES
1649                            ;--*************************************************************************
1650
1651  002176                            ERRTBL
1652  002176                                                                  L$ERRTBL::
1653  002176  000000            ERRTYP::        .WORD   0
1654  002200  000000            ERRNBR::        .WORD   0
1655  002202  000000            ERRMSG::        .WORD   0
1656  002204  000000            ERRBLK::        .WORD   0
1657
1658                            ;;*************************************************************************
1659                            ;* STORAGE FOR DEVICE REGISTERS
1660                            ;;*************************************************************************
1661  002206                    WSR0:                           ;STORAGE FOR DEVICE CSR REGISTERS
1662  002206  000000            BSR0:   .WORD   0
1663  002210                    WSR2:
1664  002210  000000            BSR1:   .WORD   0
1665  002212                    WSR4:
1666  002212  000000            BSR2:   .WORD   0
1667  002214                    WSR6:
1668  002214  000000            BSR3:   .WORD   0
1669  002216                    WSR10:
1670  002216  000000            BSR4:   .WORD   0
1671  002220                    WSR12:
1672  002220  000000            BSR5:   .WORD   0
1673  002222                    WSR14:
1674  002222  000000            BSR6:   .WORD   0
1675  002224                    WSR16:
1676  002224  000000            BSR7:   .WORD   0
1677  002226  000000            BSR10:  .WORD   0
1678  002230  000000            BSR11:  .WORD   0
1679  002232  000000            BSR12:  .WORD   0
1680  002234  000000            BSR13:  .WORD   0
1681  002236  000000            BSR14:  .WORD   0
1682  002240  000000            BSR15:  .WORD   0
1683  002242  000000            BSR16:  .WORD   0
1684  002244  000000            BSR17:  .WORD   0
1685
1686  002246  00001C            UREGS:  .BLKW   8.              ;THE FIRST 7 ARE FOR THE USYRT'S ACTUAL
1687                                                            ;REGISTERS.  THE LAST ONE IS FOR THE STATUS
1688                                                            ;REG. (USTATR).
1689  002266  000020            VREGS:  .BLKW   16.             ;STORAGE FOR VIA REGISTERS FOR PRINTOUT
```

```
                          ;*************************************************************************
                          ;* MISCELLANEOUS STORAGE
                          ;*************************************************************************
002326  000000            TDATA:   .WORD   0              ;TEST DATA
002330  000000            GDATA:   .WORD   0              ;GOOD DATA
002332  000000            BDATA:   .WORD   0              ;BAD DATA
002334  000000            XDATA:   .WORD   0              ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
002336  000000            SCRACH:  .WORD   0              ;GEN'L PURPOSE SCRATCH WORD
002340  000000            LOGDEV:  .WORD   0              ;LOGICAL DEVICE NUMBER
002342  000000            REGNUM:  .WORD   0              ;CONTAINS A DEVICE REGISTER NUMBER
002344  000000            PSTACK:  .WORD   0              ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
002346  000000            PRIOR:   .WORD   0              ;CPU PRIORITY FOR PRINTOUT
002350  000000            SUBRPC:  .WORD   0              ;PC OF SUBR CALL FOR ERROR REPORTS
002352  000000            INTFLG:  .WORD   0              ;INTERRUPT RECEIVED FLAGS
                                                          ;  BIT 0 FOR TX, BIT 1 FOR RCV
002354  000000            ERRFLG:  .WORD   0              ;SUBROUTINE ERROR FLAG
002356  000000            TIMFLG:  .WORD   0              ;EVENT TIME-OUT FLAG
002360  000000            RETADR:  .WORD   0              ;SUBR ERROR RETURN ADDRESS
002362  000000            REDBYT:  .WORD   0              ;LO BYTE CONTAINS BYTE READ FROM LU REG
002364  000000            WRIBYT:  .WORD   0              ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
002366  000000            LOADAT:  .WORD   0              ;CONTAINS TEST DATA LOADED INTO REG
002370  000000            GOODAT:  .WORD   0              ;STORAGE FOR EXPECTED DATA
002372  000000            BADDAT:  .WORD   0              ;STORAGE FOR ACTUAL DATA
002374  000000            FRSTIM:  .WORD   0              ;FLAG=0 IF PROGRAM JUST LOADED
002376  000000            SAVE4:   .WORD   0              ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
002400  000000            SAVE6:   .WORD   0              ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
002402  000000            ERROR1:  .WORD   0              ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
002404  000000            CHPTYP:  .WORD   0              ;USYRT CHIP TYPE, =0 FOR SMC, ELSE =1
002406  000000            SAVLEN:  .WORD   0              ;SAVED TX AND RCV CHAR LENGTHS
002410  000000            DEVMAP:  .WORD   0              ;BIT MAP OF ACTIVE DEVICES
002412  000000            DEVPTR:  .WORD   0              ;DEVICE MAP BIT POINTER
002414  000000            UNIT:    .WORD   0              ;CONTAINS UNIT NO. (1 TO N)
002416  000000            STARES:  .WORD   0              ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
002420  000000            TSTNUM:  .WORD   0              ;NO. OF CURRENT TEST (FOR SOME TESTS)
```

```
1725
1726  002422
1727  002422
1728  002422  160020          ;;********* CURRENT DEVICE PARAMETERS *************
                              BSEL0:
                              SEL0:
                              MPCSR:   .WORD   160020          ;POINTER TO DMV11 CSR'S
1729  002424  160021          BSEL1:   .WORD   160021          ;POINTER TO BSEL1
1730  002426
                              BSEL2:
1731  002426  160022          SEL2:    .WORD   160022          ;POINTER TO SEL2
1732  002430  160023          BSEL3:   .WORD   160023          ;POINTER TO BSEL3
1733  002432
                              BSEL4:
1734  002432  160024          SEL4:    .WORD   160024          ;POINTER TO SEL4
1735  002434  160025          BSEL5:   .WORD   160025          ;POINTER TO BSEL5
1736  002436
                              BSEL6:
1737  002436  160026          SEL6:    .WORD   160026          ;POINTER TO SEL6
1738  002440  160027          BSEL7:   .WORD   160027          ;POINTER TO BSEL7
1739  002442
                              BSEL10:
1740  002442  160030          SEL10:   .WORD   160030          ;POINTER TO SEL10
1741  002444  160031          BSEL11:  .WORD   160031          ;POINTER TO BSEL11
1742  002446
                              BSEL12:
1743  002446  160032          SEL12:   .WORD   160032          ;POINTER TO SEL12
1744  002450  160033          BSEL13:  .WORD   160033          ;POINTER TO BSEL13
1745  002452
                              BSEL14:
1746  002452  160034          SEL14:   .WORD   160034          ;POINTER TO SEL14
1747  002454  160035          BSEL15:  .WORD   160035          ;POINTER TO BSEL15
1748  002456
                              BSEL16:
1749  002456  160036          SEL16:   .WORD   160036          ;POINTER TO SEL16
1750  002460  160037          BSEL17:  .WORD   160037          ;POINTER TO BSEL17
1751
1752  002462  000300          MPIVEC:  .WORD   300             ;DMV11 INPUT INTERRUPT VECTOR
1753  002464  000304          MPOVEC:  .WORD   304             ;DMV11 OUTPUT INTERRUPT VECTOR
1754  002466  000240          MPRIOR:  .WORD   240             ;DMV11 DEVICE PRIORITY
1755  002470  000000          LUSWI1:  .WORD     0             ;LINE UNIT SWITCH PACK #1
1756  002472  000000          LUSWI2:  .WORD     0             ;LINE UNIT SWITCH PACK #2
1757  002474  000000          BRDTYP:  .WORD     0             ;0=M8064, 1=M8053/V.35,2=M8053/EIA
1758  002476  000000          TSTCON:  .WORD     0             ;TEST CONNECTOR INDICATOR
1759  002500  000001          BDRATE:  .WORD     1             ;BAUD RATE = 56 K
1760                                                           ;        0 = 19.2 K
1761                                                           ;        1 = 56 K
```

```
1762
1763  002502  120400          ;TABLE OF USYRT REGISTER ADDRESSES
1764  002504  120401  USYREG: .WORD   120400          ;ADDRESS OF RDSRL
1765  002506  120402          .WORD   120401          ;ADDRESS OF RDSRH
1766  002510  120403          .WORD   120402          ;ADDRESS OF TDSRL
1767  002512  120404          .WORD   120403          ;ADDRESS OF TDSRH
1768  002514  120405          .WORD   120404          ;ADDRESS OF PCSARL
1769  002516  120407          .WORD   120405          ;ADDRESS OF PCSARH
1770  002520  122000          .WORD   120407          ;ADDRESS OF PCR
1771                          .WORD   122000          ;ADDRESS OF USYRT STATUS REG
1772
1773  002522  000010  ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
1774          REDDAT: .BLKB   8.
1775
1776  002532  000000  ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
1777  002534  000000  REG0:   .WORD   0
1778  002536  000000  REG1:   .WORD   0
1779  002540  000000  REG2:   .WORD   0
1780  002542  000000  REG3:   .WORD   0
1781  002544  000000  REG4:   .WORD   0
1782  002546  000000  REG5:   .WORD   0
1783  002550  000000  REG6:   .WORD   0
1784                  REG7:   .WORD   0
1785
1786  002552  000000  ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
1787  002554  000000  TMP0:   .WORD   0
1788  002556  000000  TMP1:   .WORD   0
1789  002560  000000  TMP2:   .WORD   0
1790  002562  000000  TMP3:   .WORD   0
1791  002564  000000  TMP4:   .WORD   0
1792  002566  000000  TMP5:   .WORD   0
1793  002570  000000  TMP6:   .WORD   0
1794                  TMP7:   .WORD   0
1795
1796  002572          ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
1797  002572      377 UPBITS:
1798  002573      000         .BYTE   377             ;MASK FOR RDBR
1799  002574      000         .BYTE   000             ;MASK FOR RDSR
1800  002575      360         .BYTE   000             ;MASK FOR TDBR
1801  002576      000         .BYTE   360             ;MASK FOR TDSR
1802  002577      000         .BYTE   000             ;MASK FOR SSAR
1803  002600      347         .BYTE   000             ;MASK FOR PCSAR
1804                          .BYTE   347             ;MASK FOR PCR
1805  002601      200 TDSRNRW: .BYTE  200             ;TDSR NON-R/W BITS
```

```
1806
1807
1808  002602
1809  002602  377          .SBTTL  DATA TEST PATTERNS
1810  002603  377          ;***** DATA PATTERN E *****
1811  002604  377          PATE:
1812  002605  377                  .BYTE   377
1813  002606  377                  .BYTE   377
1814  002607  377                  .BYTE   377
1815  002610  377                  .BYTE   377
1816  002611  366                  .BYTE   377
1817                               .BYTE   377
1818                               .BYTE   377
1819  002612                       .BYTE   366
1820  002612  000
1821  002613  000          ;***** DATA PATTERN F *****
1822  002614  000          PATF:
1823  002615  000                  .BYTE   000
1824  002616  000                  .BYTE   000
1825  002617  000                  .BYTE   000
1826  002620  000                  .BYTE   000
1827  002621  110                  .BYTE   000
1828                               .BYTE   000
1829                               .BYTE   000
1830  002622                       .BYTE   110
1831  002622  000
1832  002623  001          ;***** DATA PATTERN G *****
1833  002624  003          PATG:
1834  002625  004                  .BYTE   000
1835  002626  005                  .BYTE   001
1836  002627  007                  .BYTE   003
1837  002630  100                  .BYTE   004
1838  002631  101                  .BYTE   005
1839  002632  103                  .BYTE   007
1840  002633  104                  .BYTE   100
1841  002634  105                  .BYTE   101
1842  002635  107                  .BYTE   103
1843  002636  000                  .BYTE   104
1844  002637  017                  .BYTE   105
1845  002640  027                  .BYTE   107
1846  002641  041                  .BYTE   000
1847  002642  200                  .BYTE   017
1848  002643  277                  .BYTE   027
1849  002644  103                  .BYTE   041
1850  002645  144                  .BYTE   200
1851  002646  115                  .BYTE   277
1852  002647  157                  .BYTE   103
1853  002650  000                  .BYTE   144
1854                               .BYTE   115
1855                               .BYTE   157
1856  002651                       .BYTE   000
1857  002651  125
1858  002652  252          ;***** DATA PATTERN X1 *****
1859  002653  000          PATX1:
1860  002654  377                  .BYTE   125
1861  002655  001                  .BYTE   252
                                   .BYTE   000
                                   .BYTE   377
                                   .BYTE   001
```

```
1862  002656      002                   .BYTE   002
1863  002657      004                   .BYTE   004
1864  002660      010                   .BYTE   010
1865  002661      020                   .BYTE   020
1866  002662      040                   .BYTE   040
1867  002663      100                   .BYTE   100
1868  002664      200                   .BYTE   200
1869  002665      376                   .BYTE   376
1870  002666      375                   .BYTE   375
1871  002667      373                   .BYTE   373
1872  002670      367                   .BYTE   367
1873  002671      357                   .BYTE   357
1874  002672      337                   .BYTE   337
1875  002673      277                   .BYTE   277
1876  002674      177                   .BYTE   177
1877  002675      176                   .BYTE   176
1878
1879
1880  002676                            ;***** DATA PATTERN I *****
1881  002676      000                   PATI:
1882  002677      041                   .BYTE   000
1883  002700      102                   .BYTE   041
1884  002701      143                   .BYTE   102
1885  002702      204                   .BYTE   143
1886  002703      245                   .BYTE   204
1887  002704      306                   .BYTE   245
1888  002705      347                   .BYTE   306
1889  002706      000                   .BYTE   347
1890  002707      001                   .BYTE   000
1891  002710      002                   .BYTE   001
1892  002711      004                   .BYTE   002
1893  002712      040                   .BYTE   004
1894  002713      100                   .BYTE   040
1895  002714      200                   .BYTE   100
1896  002715      000                   .BYTE   200
1897  002716      346                   .BYTE   000
1898  002717      345                   .BYTE   346
1899  002720      343                   .BYTE   345
1900  002721      307                   .BYTE   343
1901  002722      247                   .BYTE   307
1902  002723      147                   .BYTE   247
1903  002724      347                   .BYTE   147
1904  002725      242                   .BYTE   347
1905  002726      105                   .BYTE   242
1906  002727      347                   .BYTE   105
1907  002730      010                   .BYTE   347
1908  002731      020                   .BYTE   010
1909  002732      367                   .BYTE   020
1910  002733      357                   .BYTE   367
1911  002734      030                   .BYTE   357
1912  002735      027                   .BYTE   030
1913  002736      377                   .BYTE   027
1914                                    .BYTE   377
1915
1916  002737                            ;***** DATA PATTERN J *****
1917  002737      000                   PATJ:
                                        .BYTE   000
```

```
1918  002740  000                .BYTE   000
1919  002741  001                .BYTE   001
1920  002742  002                .BYTE   002
1921  002743  004                .BYTE   004
1922  002744  020                .BYTE   020
1923  002745  040                .BYTE   040
1924  002746  010                .BYTE   010
1925
1926                             ;***** DATA PATTERN K *****
1927  002747                     PATK:
1928  002747  000                .BYTE   000
1929  002750  377                .BYTE   377
1930  002751  376                .BYTE   376
1931  002752  375                .BYTE   375
1932  002753  373                .BYTE   373
1933  002754  376                .BYTE   376
1934  002755  177                .BYTE   177
1935  002756  377                .BYTE   377
1936  002757  000                .BYTE   000
1937  002760  001                .BYTE   001
1938  002761  002                .BYTE   002
1939  002762  004                .BYTE   004
1940  002763  010                .BYTE   010
1941  002764  200                .BYTE   200
1942  002765  125                .BYTE   125
1943  002766  252                .BYTE   252
1944  002767  000                .BYTE   000
1945
1946                             ;***** DATA PATTERN L *****
1947  002770                     PATL:
1948  002770  000                .BYTE   000
1949  002771  017                .BYTE   017
1950  002772  016                .BYTE   016
1951  002773  015                .BYTE   015
1952  002774  013                .BYTE   013
1953  002775  016                .BYTE   016
1954  002776  017                .BYTE   017
1955  002777  017                .BYTE   017
1956  003000  000                .BYTE   000
1957  003001  001                .BYTE   001
1958  003002  002                .BYTE   002
1959  003003  004                .BYTE   004
1960  003004  010                .BYTE   010
1961  003005  000                .BYTE   000
1962  003006  005                .BYTE   005
1963  003007  012                .BYTE   012
1964  003010  000                .BYTE   000
```

```

                          ;***** DATA PATTERN Q *****
003011      000           PATQ:   .BYTE   000
003012      003                   .BYTE   003
003013      014                   .BYTE   014
003014      060                   .BYTE   060
003015      001                   .BYTE   001
003016      007                   .BYTE   007
003017      037                   .BYTE   037
003020      177                   .BYTE   177

                          ;***** DATA PATTERN INVERTED Q *****
003021      000           PATQB:  .BYTE   000        ;INVERTED 000 (7 BIT)
003022      140                   .BYTE   140        ;INVERTED 003 (7 BIT)
003023      030                   .BYTE   030        ;INVERTED 014 (7 BIT)
003024      006                   .BYTE   006        ;INVERTED 060 (7 BIT)
003025      100                   .BYTE   100        ;INVERTED 001 (7 BIT)
003026      160                   .BYTE   160        ;INVERTED 007 (7 BIT)
003027      174                   .BYTE   174        ;INVERTED 037 (7 BIT)
003030      177                   .BYTE   177        ;INVERTED 177 (7 BIT)

003031                    ENDPAT:
         003032           .EVEN

                          ;*** RECEIVED DATA BUFFER (64. WORDS) ***
003032   000100           RCVBUF: .BLKW   64.
```

```
                                          .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
                                          ;;*****************************************************************
003232                                            DEVTYP  <M8053 OR M8064>
003232                                                                    L$DVTYP::
003232  034115  032460  020063                                                    .ASCIZ  /M8053 OR M8064/
003240  051117  046440  030070
003246  032066     000
        003252                                                                    .EVEN

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

        000012                            .RADIX  10.
003252                                            DESCRIPT        <DMV-11 LINE UNIT TESTS - PART 2 OF 3>
003252                                                                    L$DESC::
003252  046504  026526  030461                                                    .ASCIZ  /DMV-11 LINE UNI
003260  046040  047111  020105
003266  047125  052111  052040
003274  051505  051524  026440
003302  050040  051101  020124
003310  020062  043117  031440
003316     000
        003320                                                                    .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>
                                        ;
                                        ;--*********************************************************************

003320  112777  000301  177076  MSTCLR: MOVB    #RUN!MCLR!MREQ,@BSEL1   ;INITIATE M-LOOP

003326  010346                          MOV     R3,-(SP)
003330  012703  000030                  MOV     #24.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003334  077301                  1$:     SOB     R3,1$
003336  012603                          MOV     (SP)+,R3

003340  132777  000200  177060          BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
003346  001023                          BNE     5$              ;YES,  GOOD.  RETURN
003350  004737  004134                  JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003354  012737  000301  002330          MOV     #RUN!MCLR!MREQ,GDATA    ;IDENTIFY REQUESTED FUNCTION
003362                                  GTDF    EM3,ERR4        ;'MRDY'' TIMEOUT
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 1
003362  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
003370  012737  000001  002200                                                  MOV     #1,ERRNBR
003376  012737  014115  002202                                                  MOV     #EM3,ERRMSG
003404  012737  021274  002204                                                  MOV     #ERR4,ERRBLK
003412  000261                          SEC                     ;SET CARRY TO INDICATE ERROR
003414  000401                          BR      9$              ;EXIT WITH THE 'ERROR'' FLAG (CARRY BIT) SET
003416  000241                  5$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
003420  000207                  9$:     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>
                                ;
                                ;--*************************************************************************

003422  012577  177004        READ:   MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003426  112777  000001  176772        MOVB    #REDLOC,@BSEL2  ;TELL M-LOOP TO GIVE US THE REQUESTED DATA

003434  010346                        MOV     R3,-(SP)
003436  012703  000050                MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003442  077301                1$:     SOB     R3,1$
003444  012603                        MOV     (SP)+,R3

003446  132777  000200  176752        BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
003454  001023                        BNE     5$              ;YES,  GOOD.  RETURN

003456  004737  004134                JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003462  012737  000001  002330        MOV     #REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
003470                                GTDF    EM4,ERR4        ;'MRDY' TIMEOUT
                                                ;       QUEUE 'DEVICE FATAL' ERROR # 2
003470  012737  000001  002176                                        MOV     #T.EDF,ERRTYP
003476  012737  000002  002200                                        MOV     #2,ERRNBR
003504  012737  014141  002202                                        MOV     #EM4,ERRMSG
003512  012737  021274  002204                                        MOV     #ERR4,ERRBLK
003520  000261                        SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
003522  000401                        BR      6$              ;RETURN WITH THAT INDICATION

003524  000241                5$:     CLC                     ;INDICATE 'NO ERROR'
003526  117735  176704        6$:     MOVB    @BSEL6,@(R5)+   ;PUT DATA WHERE CALLER WANTS IT
003532  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>
                                          ;
                                          ;--*****************************************************************

003534                                    READI:
003534  012577  176672                            MOV     (R5)+,@SEL4        ;SETUP SOURCE POINTER
003540  112777  000001  176660                    MOVB    #REDLOC,@SEL2      ;TELL M-LOOP TO GIVE US THE REQUESTED DATA

003546  010346                                    MOV     R3,-(SP)
003550  012703  000050                            MOV     #40.,R3            ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003554  077301                            1$:     SOB     R3,1$
003556  012603                                    MOV     (SP)+,R3

003560  132777  000200  176640                    BITB    #MRDY,@SEL2        ;DID THE M-LOOP FINISH
003566  001023                                    BNE     5$                 ;YES,  GOOD.  RETURN

003570  004737  004134                            JSR     PC,GETWSR          ;GET BYTE SELECT REGISTERS
003574  012737  000001  002330                    MOV     #REDLOC,GDATA      ;IDENTIFY REQUESTED FUNCTION
003602                                            GTDF    EM4,ERR4           ;'MRDY'' TIMEOUT
                                                                             ;       QUEUE 'DEVICE FATAL'' ERROR # 3
003602  012737  000001  002176                                                                       MOV     #T.EDF,ERRTYP
003610  012737  000003  002200                                                                       MOV     #3,ERRNBR
003616  012737  014141  002202                                                                       MOV     #EM4,ERRMSG
003624  012737  021274  002204                                                                       MOV     #ERR4,ERRBLK
003632  000261                                    SEC                        ;INDICATE AN ERROR HAS BEEN STACKED
003634  000401                                    BR      6$                 ;RETURN WITH THAT INDICATION

003636  000241                            5$:     CLC                        ;INDICATE 'NO ERROR''
003640  017725  176572                    6$:     MOV     @SEL6,(R5)+        ;PUT DATA WHERE CALLER WANTS IT
003644  000205                                    RTS     R5                 ;RETURN
```

```
2170                                    .SBTTL ....M-LOOP -- WRITE
2171                                    ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2172                                    ; WRITE - WRITE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
2173                                    ;
2174                                    ;  CALLING SEQUENCE:
2175                                    ;
2176                                    ;       JSR     R5,WRITE
2177                                    ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
2178                                    ;       .WORD   <ADDRESS OF DATA BYTE>
2179                                    ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2180                                    ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
2181                                    ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
2182                                    ;
2183                                    ; N$:   <RESUMPTION OF NORMAL PROCESSING>
2184                                    ;
2185                                    ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2186
2187  003646  012577  176560    WRITE:  MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
2188  003652  113577  176560            MOVB    @(R5)+,@SEL6    ;MAKE DATA AVAILABLE TO M-LOOP
2189  003656  000404                    BR      MLWRI           ;THE REST OF THIS ROUTINE IS THE SAME AS 'WRITEI'
2190
2191
2192
2193
```

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

003660                          WRITEI:
003660  012577  176546                  MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003664  012577  176546                  MOV     (R5)+,@SEL6     ;MAKE DATA AVAILABLE TO M-LOOP
003670  112777  000002  176530  MLWRI:  MOVB    #WRILOC,@BSEL2  ;TELL M-LOOP TO WRITE THE DATA

003676  010346                  MOV     R3,-(SP)
003700  012703  000050          MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003704  077301          1$:     SOB     R3,1$
003706  012603                  MOV     (SP)+,R3

003710  132777  000200  176510          BITB    #RDY,@BSEL2     ;DID THE M-LOOP FINISH
003716  001023                  BNE     5$              ;YES.  GOOD.  RETURN
003720  004737  004134          JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003724  012737  0C0002  002330  MOV     #WRILOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
003732                          GTDF    EM4,ERR4        ;''RDY'' TIMEOUT
                                ;       QUEUE ''DEVICE FATAL'' ERROR # 4
003732  012737  000001  002176                                  MOV     #T.EDF,ERRTYP
003740  012737  000004  002200                                  MOV     #4,ERRNBR
003746  012737  014141  002202                                  MOV     #EM4,ERRMSG
003754  012737  021274  002204                                  MOV     #ERR4,ERRBLK
003762  000261                  SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
003764  000401                  BR      6$              ;RETURN WITH THAT INDICATION

003766  000241          5$:     CLC                     ;INDICATE ''NO ERROR''
003770  000205          6$:     RTS     R5              ;RETURN
```

```
                                        .SBTTL ....GETBSR -- GET BYTE SELECT REGISTERS
                                        ;+************************************************************************
                                        ;
                                        ;       GET THE CONT.NTS 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       ##  ####  ####  #   #  #   #
                                        ;
                                        ;-************************************************************************

003772  117737  176424  002206  GETBSR: MOVB    @BSEL0,BSR0     ;PUT THE CURRENT CSR VALUES INTO THE PRINT-OUT
004000  117737  176420  002210          MOVB    @BSEL1,BSR1     ;TABLE
004006  117737  176414  002212          MOVB    @BSEL2,BSR2
004014  117737  176410  002214          MOVB    @BSEL3,BSR3
004022  117737  176404  002216          MOVB    @BSEL4,BSR4
004030  117737  176400  002220          MOVB    @BSEL5,BSR5
004036  117737  176374  002222          MOVB    @BSEL6,BSR6
004044  117737  176370  002224          MOVB    @BSEL7,BSR7
004052  117737  176364  002226          MOVB    @BSEL10,BSR10
004060  117737  176360  002230          MOVB    @BSEL11,BSR11
004066  117737  176354  002232          MOVB    @BSEL12,BSR12
004074  117737  176350  002234          MOVB    @BSEL13,BSR13
004102  117737  176344  002236          MOVB    @BSEL14,BSR14
004110  117737  176340  002240          MOVB    @BSEL15,BSR15
004116  117737  176334  002242          MOVB    @BSEL16,BSR16
004124  117737  176330  002244          MOVB    @BSEL17,BSR17
004132  000207                          RTS     PC              ;RETURN TO CALLER

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

004134  017737  176262  002206  GETWSR: MOV     @SEL0,WSR0      ;MOVE THE 4 WORD REGISTERS TO THE OTHERWISE
004142  017737  176260  002210          MOV     @SEL2,WSR2      ;BYTE TABLE
004150  017737  176256  002212          MOV     @SEL4,WSR4
004156  017737  176254  002214          MOV     @SEL6,WSR6
004164  017737  176252  002216          MOV     @SEL10,WSR10
004172  017737  176250  002220          MOV     @SEL12,WSR12
004200  017737  176246  002222          MOV     @SEL14,WSR14
004206  017737  176244  002224          MOV     @SEL16,WSR16
004214  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>
                                        ;
                                        ;--*************************************************************************

004216  010037  004232                  STUREG: MOV     R0,2$                   ;PUT SPECIFIED REGISTER'S ADDRESS IN I/O CALLS
004222  010037  004250                          MOV     R0,4$

004226  004537  003646                          JSR     R5,WRITE                ;WRITE IT
004232  000000                          2$:     .WORD   0                       ;*** MODIFIED FROM ABOVE ***
004234  002326                                  .WORD   TDATA                   ;
004236  103431                                  BCS     10$                     ;ON ERROR, EXIT

004240  005037  002332                          CLR     BDATA                   ;CLEAR BOTH BYTES -- JUST IN CASE....
004244  004537  003422                          JSR     R5,READ                 ;READ IT BACK AGAIN
004250  000000                          4$:     .WORD   0                       ;*** MODIFIED FROM ABOVE ***
004252  002332                                  .WORD   BDATA
004254  103422                                  BCS     10$                     ;ON ERROR, EXIT

004256  123737  002330  002332                  CMPB    GDATA,BDATA             ;DID WE READ WHAT WE WROTE?
004264  000241                                  CLC                             ; (THIS ISN'T NEEDED FOR THE ERROR TEST BUT
                                                                                ; MUST BE CLEARED ON EXIT IF NO ERROR OCCURED)
004266  001415                                  BEQ     10$                     ;YES, EXIT FROM SUBTEST
004270                                          GTDF    EM25,ERR7A              ;REPORT READ/WRITE ERROR
                                                                                ;       QUEUE ''DEVICE FATAL'' ERROR # 5
004270  012737  000001  002176                                                                          MOV     #T.EDF,ERRTYP
004276  012737  000005  002200                                                                          MOV     #5,ERRNBR
004304  012737  014347  002202                                                                          MOV     #EM25,ERRMSG
004312  012737  021420  002204                                                                          MOV     #ERR7A,ERRBLK
004320  000261                                  SEC                             ;INDICATE THAT AN ERROR WAS DETECTED
004322  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
                                        ;--*************************************************************************

004324  000207                          STALL:  RTS     PC
```

2342

```
2343                                    .SBTTL
2344
2345                                    ;*****************************************************************************
2346                                    ;* GETURS - LOAD INTO THE 8 WORD STORAGE AREA (UREGS) THE CONTENTS OF THE
2347                                    ;*      VARIOUS USYRT REGISTERS
2348                                    ;*
2349                                    ;*        CALLING SEQUENCE:
2350                                    ;*
2351                                    ;*****************************************************************************
2352  004326  012737  002246  004370  GETURS: MOV     #UREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
2353  004334  012737  120400  004366          MOV     #USYRT,4$       ;INIT POINTER TO REGISTER ADDRESSES
2354
2355  004342  005037  002264                  CLR     UREGS+14.       ;CLEAR STORAGE WORD
2356  004346  004537  003422                  JSR     R5,READ         ;READ THE USYRT STATUS REGISTER
2357  004352  122000                          .WORD   USTATR          ;STATUS REGISTER'S ADDRESS WITHIN DMV-11
2358  004354  002264                          .WORD   UREGS+14.       ;ADDRESS ALLOCATED TO THAT REG. W/IN 'UREGS''
2359
2360  004356  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
2361  004362  004537  003422                  JSR     R5,READ         ;READ A LINE UNIT REG
2362  004366  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
2363  004370  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
2364
2365  004372  005237  004366          6$:     INC     4$              ;INCREMENT REG NO.
2366  004376  023727  004366  120406          CMP     4$,#USYRT+6     ;THIS IS NOT A VALID REGISTER ADDRESS
2367  004404  001772                          BEQ     6$              ;SO IT MUST BE BYPASSED
2368
2369  004406  062737  000002  004370          ADD     #2,5$           ;ADVANCE ADDRESS OF STORAGE AREA POINTER
2370  004414  023727  004366  120410          CMP     4$,#USYRT+10    ;SEE IF ALL REGS READ YET
2371  004422  001355                          BNE     3$              ;BR IF NOT
2372
2373  004424  000207                          RTS     PC              ;RETURN
2374
2375
2376
2377                                    ;*****************************************************************************
2378                                    ;* GETVRS: - LOAD INTO THE 16 WORD STORAGE AREA (VREGS) THE CONTENTS OF THE
2379                                    ;*      VARIOUS VIA REGISTERS.
2380                                    ;*
2381                                    ;*      CALLING SEQUENCE :
2382                                    ;*****************************************************************************
2383  004426  012737  002266  004454  GETVRS: MOV     #VREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
2384  004434  012737  120000  004452          MOV     #VIA,4$         ;INIT POINTER TO REGISTER ADDRESSES
2385  004442  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
2386  004446  004537  003422                  JSR     R5,READ         ;READ A VIA REG
2387  004452  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
2388  004454  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
2389  004456  005237  004452          6$:     INC     4$              ;INCREMENT REG NO.
2390  004462  062737  000002  004454          ADD     #2,5$           ;INCREMENT STORAGE ADRS
2391  004470  023727  004452  120020          CMP     4$,#VIA+16.     ;SEE IF ALL VIA REGS READ YET
2392  004476  001361                          BNE     3$              ;BR IF NOT
2393  004500  000207                          RTS     PC              ;RETURN
```

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

004502 010146           INITT1: MOV     R1,-(SP)        ;SAVE THE REGISTER WE WILL BE USING
004504 012537 004626            MOV     (R5)+,7$        ;SETUP VALUE TO BE WRITTEN IN LATCH
004510 012537 004654            MOV     (R5)+,10$       ;SETUP VALUE TO BE WRITTEN IN COUNTER
004514 111501                   MOVB    (R5),R1         ;GET & PROCESS BITS FOR ACR 6 & 7
004516 143701 000077            BICB    077,R1
004522 010137 004616            MOV     R1,4$           ;SETUP CALL SET ACR'S BITS 6 & 7
004526 112501                   MOVB    (R5)+,R1        ;NOW, GET THE BIT TO BE USED IN SETTING OR
                                                        ;CLEARING BIT 6 OF ''IER''
004530 106301                   ASLB    R1              ;THE PASSED BIT IS IN THE WRONG POSITION
004532 106301                   ASLB    R1              ;BUT, THE PASSED BIT SHOULD CONTROL THE OPERATION.
                                                        ;WE KNOW WE ARE SETTING OR CLEARING BIT 6 --
                                                        ;THUS, THE PASSED BIT WILL BECOME THE CONTROLLING
                                                        ;BIT 7 AND WE WILL ''OR'' IN THE BIT WE WISH TO
                                                        ;BE CONTROLLED (BIT 6).
004534 143701 000177            BICB    177,R1          ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
004540 153701 000100            BISB    100,R1          ;THEN SET BIT 6
004544 010137 004556            MOV     R1,2$           ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE

004550 004537 003660            JSR     R5,WRITEI       ;WRITE TO
004554 120016                   VIAIER                  ;THE VIA'S IER
004556 000000           2$:     .WORD   0               ;INTERRUPT ENABLE/DISABLE INFORMATION

004560 004537 003534            JSR     R5,READI        ;READ THE CURRENT SETTING OF
004564 120013                   VIAACR                  ;THE VIA'S ACR
004566 000000           3$:     .WORD   0               ;INTO ''3$''

004570 013701 004566            MOV     3$,R1           ;GET THAT VALUE
004574 143701 000300            BICB    300,R1          ;CLEAR THE CURRENT SETTING OF BITS 6 & 7
004600 053701 004616            BIS     4$,R1           ;SET THEM ACCORDING TO THE PASSED VALUES
```

```
2450  004604  010137  004616                MOV     R1,4$           ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
2451
2452  004610  004537  003660                JSR     R5,WRITEI       ;WRITE TO
2453  004614  120013                        VIAACR                  ;THE VIA'S ACR
2454  004616  000000            4$:         .WORD   0               ;THE NEW REGISTER SETTING
2455
2456  004620  004537  003660                JSR     R5,WRITEI       ;WRITE TO
2457  004624  120006                        VIAT1C                  ;LOW ORDER LATCH REGISTER (T1L-L)
2458  004626  000000            7$:         .WORD   0               ;THE VALUE PASSED
2459
2460  004630  113737  004627  004644        MOVB    7$+1,8$         ;SETUP FOR AND
2461  004636  004537  003660                JSR     R5,WRITEI       ;WRITE TO
2462  004642  120007                        VIAT1D                  ;HIGH ORDER LATCH REGESTER (T1L-H)
2463  004644  000000            8$:         .WORD   0               ;THE VALUE PASSED
2464
2465  004646  004537  003660                JSR     R5,WRITEI       ;WRITE TO
2466  004652  120004                        VIAT1A                  ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2467  004654  000000            10$:        .WORD   0               ;THE VALUE PASSED
2468
2469  004656  113737  004655  004672        MOVB    10$+1,11$       ;SETUP FOR AND
2470  004664  004537  003660                JSR     R5,WRITEI       ;WRITE TO
2471  004670  120005                        VIAT1B                  ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2472  004672  000000            11$:        .WORD   0               ;THE VALUE PASSED
2473
2474                                        ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2475
2476  004674  012601                        MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2477  004676  005205                        INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2478                                                                ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2479
2480  004700  000205                        RTS     R5              ;NOW, RETURN
2481
2482
```

```
                        .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''.)
                        ;*
                        ;*********************************************************************

004702  010146          INITT2: MOV     R1,-(SP)        ;SAVE THE REGISTER WE WILL BE USING
004704  012537  005024          MOV     (R5)+,10$       ;SETUP VALUE TO BE WRITTEN IN COUNTER
004710  111501                MOVB    (R5),R1         ;GET & PROCESS BIT FOR ACR 5
004712  143701  000337          BICB    337,R1
004716  010137  005014          MOV     R1,4$           ;SETUP CALL TO SET OR CLEAR ACR'S BIT 5
004722  112501                MOVB    (R5)+,R1        ;NOW, GET THE BIT TO BE USED IN SETTING OR
                                                        ;CLEARING BIT 5 OF ''IER''
004724  106301                ASLB    R1              ;THE PASSED BIT IS IN THE WRONG POSITION
004726  106301                ASLB    R1              ;BUT, THE PASSED BIT SHOULD CONTROL THE
004730  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).
004732  143701  000177          BICB    177,R1          ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
004736  153701  000040          BISB    040,R1          ;THEN SET BIT 5
004742  010137  004754          MOV     R1,2$           ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE

004746  004537  003660          JSR     R5,WRITEI       ;WRITE TO
004752  120016                  VIAIER                  ;THE VIA'S IER
004754  000000          2$:     .WORD   0               ;INTERRUPT ENABLE/DISABLE INFORMATION

004756  004537  003534          JSR     R5,READI        ;READ THE CURRENT SETTING OF
004762  120013                  VIAACR                  ;THE VIA'S ACR
004764  000000          3$:     .WORD   0               ;INTO ''3$''

004766  013701  004764          MOV     3$,R1           ;GET THAT VALUE
004772  143701  000040          BICB    040,R1          ;CLEAR THE CURRENT SETTING OF BIT 5
004776  053701  005014          BIS     4$,R1           ;SET IT ACCORDING TO THE PASSED VALUE
005002  010137  005014          MOV     R1,4$           ;PASS NEW REG. SETTING TO APPROPRIATE CALL
```

```
2539  005006  004537  003660                  JSR     R5,WRITEI       ;WRITE TO
2540  005012  120013                          VIAACR                  ;THE VIA'S ACR
2541  005014  000000                  4$:     .WORD   0               ;THE NEW REGISTER SETTING
2542
2543  005016  004537  003660                  JSR     R5,WRITEI       ;WRITE TO
2544  005022  120010                          VIAT2A                  ;LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
2545  005024  000000                  10$:    .WORD   0               ;THE VALUE PASSED
2546
2547  005026  113737  005025  005042          MOVB    10$+1,11$       ;SETUP FOR AND
2548  005034  004537  003660                  JSR     R5,WRITEI       ;WRITE TO
2549  005040  120011                          VIAT2B                  ;HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
2550  005042  000000                  11$:    .WORD   0               ;THE VALUE PASSED
2551
2552                                  ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2553
2554  005044  012601                          MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2555  005046  005205                          INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2556                                                                  ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2557
2558  005050  000205                          RTS     R5              ;THEN RETURN
2559
```

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

005052                  RSTCHK:
005052  010146                  MOV     R1,-(SP)        ;SAVE R1
005054  010246                  MOV     R2,-(SP)        ;SAVE R2

005056  004537  003660          JSR     R5,WRITEI       ;SET PROGRAM RESET BIT IN VIA ORB REG
005062  120000                  VIAORB
005064  000031                  DTR!RTSND!PRESET
005066  004537  003660          JSR     R5,WRITEI       ;CLEAR PROGRAM RESET BIT IN VIA ORB REG
005072  120000                  VIAORB
005074  000030                  DTR!RTSND

005076  005001                  CLR     R1              ;INIT USYRT REG ADRS PTR
005100  012702  002612          MOV     #PATF,R2        ;INIT DATA PATTERN POINTER
005104  016137  002502  005116  6$:     MOV     USYREG(R1),7$   ;SET USYRT READ ADDRESS
005112  004537  003534          JSR     R5,READI        ;READ A USYRT REG
005116  000000                  7$:     .WORD   0       ;USYRT REG ADRS GOES HERE
005120  000000                  8$:     .WORD   0       ;DATA READ IS RETURNED HERE
005122  123722  005120          CMPB    8$,(R2)+        ;SEE IF REG CONTAINS EXPECTED DATA
005126  001432                  BEQ     9$              ;BR IF MATCH

005130  010137  002342          MOV     R1,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
005134  006237  002342          ASR     REGNUM          ;GET WORD OFFSET
005140  005037  002330          CLR     GDATA           ;GET EXPECTED DATA
005144  116237  177777  002330  MOVB    -1(R2),GDATA
005152  013737  005120  002332  MOV     8$,BDATA        ;GET ACTUAL DATA
                        ;STACK 'USYRT NOT CLEARED BY PROGRAM RESET'' MSG
005160                  GTDF    EM2,ERR10
                                ;       QUEUE 'DEVICE FATAL'' ERROR # 6
005160  012737  000001  002176                  MOV     #T.EDF,ERRTYP
005166  012737  000006  002200                  MOV     #6,ERRNBR
005174  012737  014046  002202                  MOV     #EM2,ERRMSG
005202  012737  021540  002204                  MOV     #ERR10,ERRBLK
005210  000261                  SEC             ;SET C BIT TO FLAG ERROR
005212  000406                  BR      10$     ;TAKE ERROR EXIT

005214  062701  000002  9$:     ADD     #2,R1           ;INCR USYRT REG ADRS PTR
005220  020127  000020          CMP     R1,#16.         ;SEE IF ALL REGS READ YET
005224  002727                  BLT     6$              ;BR IF NOT
005226  000241                  CLC                     ;** CLEAR C BIT FOR NO ERRORS
005230  012602  10$:     MOV     (SP)+,R2        ;RESTORE R2
005232  012601                  MOV     (SP)+,R1        ;RESTORE R1
005234  000205                  RTS     R5              ;** RETURN
```

```
                                ;*****************************************************************************
                                ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
                                ;*****************************************************************************
005236  010146                  WAIT50: MOV     R1,-(SP)        ;SAVE R1
005240  012701  000005                  MOV     #5.,R1          ;INIT COUNTER
005244  077101                  3$:     SOB     R1,3$           ;DELAY HERE FOR 23.8 MICRO-SEC'S
005246  012601                          MOV     (SP)+,R1        ;RESTORE R1
005250  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
                                ;*****************************************************************************
005252                          SETVIA:
005252  004537  003660                  JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
005256  120002                          VIADPB
005260  000377                          377
005262  004537  003660                  JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
005266  120003                          VIADPA                  ; (BIT0 IS ONLY OUTPUT BIT)
005270  000001                          001
005272  004537  003660                  JSR     R5,WRITEI       ;DISABLE USYRT INTERNAL LOOPBACK
005276  120017                          VIAORA
005300  000000                          000
005302  004537  003660                  JSR     R5,WRITEI       ;INIT PORT B
005306  120000                          VIAORB
005310  000030                          DTR!RTSND
005312  004537  003660                  JSR     R5,WRITEI       ;SET ACR FOR :  T1 SQUARE WAVE OUTPUT MODE,
005316  120013                          VIAACR                  ;               T2 ONE-SHOT OUTPUT MODE,
005320  000350                          350                     ;               SR AT SYS CLOCK RATE ON CB1
005322  004537  003660                  JSR     R5,WRITEI       ;SET PCR FOR :  CB1 NEG TRANS INPUT MODE,
005326  120014                          VIAPCR                  ;               CA2 NEG TRANS INPUT MODE,
005330  000022                          022                     ;               CA1 NEG TRANS INPUT MODE
005332  004537  003660                  JSR     R5,WRITEI       ;DISABLE ALL MICRO-INTRPTS
005336  120016                          VIAIER
005340  000177                          177
005342  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL ....INIDMV -- INIT DMV (MCLR, VIA SETUP)
                                ;*************************************************************************
                                ;* INIDMV - THIS SUBROUTINE INITIALIZES THE DMV-11, BY DOING A MASTER CLEAR,
                                ;*      ENTERING THE M-LOOP, AND PROGRAMMING THE VIA REGS FOR DEFAULT
                                ;*      OPERATION.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*         JSR  PC,INIDMV
                                ;*************************************************************************
005344  004737  003320        INIDMV: JSR     PC,MSTCLR       ;MASTER CLR, M-LOOP
005350  004737  005252                JSR     PC,SETVIA       ;PROGRAM VIA
005354  000207                        RTS     PC              ;RETURN



                                .SBTTL ....CKUSTS -- CHECK USYRT STATUS REGISTERS
                                ;*************************************************************************
                                ;* CKUSTS - THIS SUBROUTINE CHECKS THE USYRT STATUS BY READING THE USYRT
                                ;*      STATUS REGISTER AND COMPARING IT TO THE LOW BYTE OF THE WORD FOLLOWING
                                ;*      THE CALL. IF THERE IS A MISMATCH, THE SUBROUTINE STACKS THE ERROR
                                ;*      INFORMATION, AND SETS THE "C" BIT AND RETURNS.
                                ;*************************************************************************
005356                          CKUSTS:
005356  004537  003534                JSR     R5,READI        ;READ USYRT STATUS REGISTER
005362  122000                        USTATR
005364  000000                1$:     .WORD   0
005366  122537  005364                CMPB    (R5)+,1$        ;SEE IF STATUS MATCHES EXPECTED
005372  000241                        CLC                     ;CLEAR C BIT
005374  001430                        BEQ     2$              ;BR IF STATUS OK
005376  012737  000007  002342        MOV     #7,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
005404  016537  177777  002330        MOV     -1(R5),GDATA    ;GET EXPECTED DATA
005412  005037  002332                CLR     BDATA           ;GET ACTUAL DATA
005416  113737  005364  002332        MOVB    1$,BDATA
                                ;STACK "USYRT STATUS INCORRECT" ERROR
005424                                  GTDF    EM68,ERR10
                                ;       QUEUE 'DEVICE FATAL" ERROR # 7
005424  012737  000001  002176                                          MOV     #T.EDF,ERRTYP
005432  012737  000007  002200                                          MOV     #7,ERRNBR
005440  012737  015500  002202                                          MOV     #EM68,ERRMSG
005446  012737  021540  002204                                          MOV     #ERR10,ERRBLK
005454  000261                        SEC                     ;SET C BIT FOR ERROR
005456  005205                2$:     INC     R5              ;INCREMENT R5 PAST ARGUMENT
005460  000205                        RTS     R5              ;RETURN
```

```
                                .SBTTL ....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)
                                ;*********************************************************************
                                ;* CKTACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TXACT IN THE USYRT
                                ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,CKTACT
                                ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TXACT>
                                ;*********************************************************************
005462                          CKTACT:
005462  012737  000007  002342          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
005470  004537  003534          1$:     JSR     R5,READI        ;READ USYRT STATUS
005474  122000                          USTATR
005476  000000                          .WORD   0
005500  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TXACT
005504  001422                          BEQ     2$              ;BR IF EXPECTED TXACT = 0
005506  132737  000004  005476          BITB    #TXACT,1$       ;SEE IF TXACT = 1
005514  001040                          BNE     3$              ;BR IF TXACT = 1
                                ;STACK 'TXACT NOT SET'' MSG
005516                                  GTDF    EM69,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 8
005516  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
005524  012737  000010  002200                                                  MOV     #8,ERRNBR
005532  012737  015527  002202                                                  MOV     #EM69,ERRMSG
005540  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
005546  000261                          SEC                     ;SET C BIT TO FLAG ERROR
005550  000423                          BR      4$              ;TAKE ERROR EXIT
005552  132737  000004  005476  2$:     BITB    #TXACT,1$       ;SEE IF TXACT = 0
005560  001416                          BEQ     3$              ;BR IF TXACT = 0
                                ;STACK 'TXACT NOT CLEARED'' MSG
005562                                  GTDF    EM70,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 9
005562  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
005570  012737  000011  002200                                                  MOV     #9,ERRNBR
005576  012737  015545  002202                                                  MOV     #EM70,ERRMSG
005604  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
005612  000261                          SEC                     ;SET C BIT TO FLAG ERROR
005614  000401                          BR      4$              ;TAKE ERROR EXIT
005616  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
005620  000205                  4$:     RTS     R5              ;RETURN
```

```
                                .SBTTL ....CKRACT -- CHECK RECEIVER ACTIVE (RXACT)
                                ;*****************************************************************************
                                ;* CKRACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RXACT IN THE USYRT
                                ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,CKRACT
                                ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF RXACT>
                                ;*****************************************************************************
005622                          CKRACT:
005622  012737  000007  002342          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
005630  004537  003534                  JSR     R5,READI        ;READ USYRT STATUS
005634  122000                          .STATR
005636  000000                  1$:     .WORD   0
005640  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RXACT
005644  001422                          BEQ     2$              ;BR IF EXPECTED RXACT = 0
005646  132737  000040  005636          BITB    #RXACT,1$       ;SEE IF RXACT = 1
005654  001040                          BNE     3$              ;BR IF RXACT = 1
                                ;STACK 'RXACT NOT SET'' MSG
005656                                  GTDF    EM71,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 10
005656  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
005664  012737  000012  002200                                                  MOV     #10,ERRNBR
005672  012737  015567  002202                                                  MOV     #EM71,ERRMSG
005700  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
005706  000261                          SEC                     ;SET C BIT TO FLAG ERROR
005710  000423                          BR      4$              ;TAKE ERROR EXIT
005712  132737  000040  005636  2$:     BITB    #RXACT,1$       ;SEE IF RXACT = 0
005720  001416                          BEQ     3$              ;BR IF RXACT = 0
                                ;STACK 'RXACT NOT CLEARED'' MSG
005722                                  GTDF    EM72,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 11
005722  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
005730  012737  000013  002200                                                  MOV     #11,ERRNBR
005736  012737  015605  002202                                                  MOV     #EM72,ERRMSG
005744  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
005752  000261                          SEC                     ;SET C BIT TO FLAG ERROR
005754  000401                          BR      4$              ;TAKE ERROR EXIT
005756  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
005760  000205                  4$:     RTS     R5              ;RETURN
```

```
2803                                    .SBTTL ....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY
2804                                    ;*****************************************************************************
2805                                    ;* CKTBMT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TBMT IN THE USYRT
2806                                    ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
2807                                    ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
2808                                    ;*
2809                                    ;*      CALLING SEQUENCE :
2810                                    ;*      JSR     R5,CKTBMT
2811                                    ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TBMT>
2812                                    ;*****************************************************************************
2813  005762                            CKTBMT:
2814  005762  012737  000007  002342            MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
2815  005770  004537  003534                    JSR     R5,READI        ;READ USYRT STATUS
2816  005774  122000                            USTATR
2817  005776  000000                    1$:     .WORD   0
2818  006000  032725  000001                    BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TBMT
2819  006004  001422                            BEQ     2$              ;BR IF EXPECTED TBMT = 0
2820  006006  132737  000100  005776          BITB    #TBMT,1$        ;SEE IF TBMT = 1
2821  006014  001040                            BNE     3$              ;BR IF TBMT = 1
2822                                    ;STACK 'TBMT NOT SET'' MSG
2823  006016                                    GTDF    EM73,ERR12
2824                                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 12
2825  006016  012737  000001  002176                                                    MOV     #T.EDF,ERRTYP
2826  006024  012737  000014  002200                                                    MOV     #12,ERRNBR
2827  006032  012737  015627  002202                                                    MOV     #EM73,ERRMSG
2828  006040  012737  021714  002204                                                    MOV     #ERR12,ERRBLK
2829  006046  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2830  006050  000423                            BR      4$              ;TAKE ERROR EXIT
2831  006052  132737  000100  005776  2$:      BITB    #TBMT,1$        ;SEE IF TBMT = 0
2832  006060  001416                            BEQ     3$              ;BR IF TBMT = 0
2833                                    ;STACK 'TBMT NOT CLEARED'' MSG
2834  006062                                    GTDF    EM74,ERR12
2835                                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 13
2836  006062  012737  000001  002176                                                    MOV     #T.EDF,ERRTYP
2837  006070  012737  000015  002200                                                    MOV     #13,ERRNBR
2838  006076  012737  015644  002202                                                    MOV     #EM74,ERRMSG
2839  006104  012737  021714  002204                                                    MOV     #ERR12,ERRBLK
2840  006112  000261                            SEC                     ;SET C BIT TO FLAG ERROR
2841  006114  000401                            BR      4$              ;TAKE ERROR EXIT
2842  006116  000241                    3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
2843  006120  000205                    4$:     RTS     R5              ;RETURN
2844
2845
2846
2847
```

```
                                .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>
                                ;*****************************************************************************
006122                          CKRDA:
006122  012737  000007  002342          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006130  004537  003534                  JSR     R5,READI        ;READ USYRT STATUS
006134  122000                          USTATR
006136  000000                  1$:     .WORD   0
006140  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RDA
006144  001422                          BEQ     2$              ;BR IF EXPECTED RDA = 0
006146  132737  000200  006136          BITB    #RDA,1$         ;SEE IF RDA = 1
006154  001040                          BNE     3$              ;BR IF RDA = 1
                                ;STACK 'RDA NOT SET'' MSG
006156                                  GTDF    EM75,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 14
006156  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006164  012737  000016  002200                                                  MOV     #14,ERRNBR
006172  012737  015665  002202                                                  MOV     #EM75,ERRMSG
006200  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006206  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006210  000423                          BR      4$              ;TAKE ERROR EXIT
006212  132737  000200  006136  2$:     BITB    #RDA,1$         ;SEE IF RDA = 0
006220  001416                          BEQ     3$              ;BR IF RDA = 0
                                ;STACK 'RDA NOT CLEARED'' MSG
006222                                  GTDF    EM76,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 15
006222  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006230  012737  000017  002200                                                  MOV     #15,ERRNBR
006236  012737  015701  002202                                                  MOV     #EM76,ERRMSG
006244  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006252  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006254  000401                          BR      4$              ;TAKE ERROR EXIT
006256  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006260  000205                  4$:     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>
                                ;*****************************************************************************
006262                          CKRSA:
006262  012737  000007  002342          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006270  004537  003534                  JSR     R5,READI        ;READ USYRT STATUS
006274  122000                          USTATR
006276  000000                  1$:     .WORD   0
006300  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RSA
006304  001422                          BEQ     2$              ;BR IF EXPECTED RSA = 0
006306  132737  000020  006276          BITB    #RSA,1$         ;SEE IF RSA = 1
006314  001040                          BNE     3$              ;BR IF RSA = 1
                                ;STACK 'RSA NOT SET'' MSG
006316                                  GTDF    EM77,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 16
006316  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006324  012737  000020  002200                                                  MOV     #16,ERRNBR
006332  012737  015721  002202                                                  MOV     #EM77,ERRMSG
006340  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006346  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006350  000423                          BR      4$              ;TAKE ERROR EXIT
006352  132737  000020  006276  2$:     BITB    #RSA,1$         ;SEE IF RSA = 0
006360  001416                          BEQ     3$              ;BR IF RSA = 0
                                ;STACK 'RSA NOT CLEARED'' MSG
006362                                  GTDF    EM78,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 17
006362  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006370  012737  000021  002200                                                  MOV     #17,ERRNBR
006376  012737  015735  002202                                                  MOV     #EM78,ERRMSG
006404  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006412  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006414  000401                          BR      4$              ;TAKE ERROR EXIT
006416  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006420  000205                  4$:     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>
                                ;*********************************************************************
006422                          CKROR:
006422  012737  000001  002342          MOV     #1,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006430  004537  003534                  JSR     R5,READI        ;READ RECEIVER STATUS
006434  120401                          RDSRH
006436  000000                  1$:     .WORD   0
006440  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF ROR
006444  001422                          BEQ     2$              ;BR IF EXPECTED ROR = 0
006446  132737  000010  006436          BITB    #ROR,1$         ;SEE IF ROR = 1
006454  001040                          BNE     3$              ;BR IF ROR = 1
                                ;STACK 'RECEIVER OVRN NOT SET'' MSG
006456                                  GTDF    EM90,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 18
006456  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006464  012737  000022  002200                                                  MOV     #18,ERRNBR
006472  012737  016300  002202                                                  MOV     #EM90,ERRMSG
006500  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006506  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006510  000423                          BR      4$              ;TAKE ERROR EXIT
006512  132737  000010  006436  2$:     BITB    #ROR,1$         ;SEE IF ROR = 0
006520  001416                          BEQ     3$              ;BR IF ROR = 0
                                ;STACK 'ROR NOT CLEARED'' MSG
006522                                  GTDF    EM91,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 19
006522  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006530  012737  000023  002200                                                  MOV     #19,ERRNBR
006536  012737  016331  002202                                                  MOV     #EM91,ERRMSG
006544  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006552  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006554  000401                          BR      4$              ;TAKE ERROR EXIT
006556  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006560  000205                  4$:     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>
                                ;*********************************************************************
006562                          CKSEOM:
006562  012737  000007  002342          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006570  004537  003534                  JSR     R5,READI        ;READ USYRT RECEIVER STATUS
006574  120401                          RDSRH
006576  000000                  1$:     .WORD   0
006600  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RSOM
006604  001422                          BEQ     2$              ;BR IF EXPECTED RSOM = 0
006606  132737  000001  006576          BITB    #RSOM,1$        ;SEE IF RSOM = 1
006614  001040                          BNE     3$              ;BR IF RSOM = 1
                                ;STACK 'RSOM NOT SET'' MSG
006616                                  GTDF    EM29,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 20
006616  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006624  012737  000024  002200                                                  MOV     #20,ERRNBR
006632  012737  014501  002202                                                  MOV     #EM29,ERRMSG
006640  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006646  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006650  000473                          BR      6$              ;TAKE ERROR EXIT
006652  132737  000001  006576  2$:     BITB    #RSOM,1$        ;SEE IF RSOM = 0
006660  001416                          BEQ     3$              ;BR IF RSOM = 0
                                ;STACK 'RSOM NOT CLEARED'' MSG
006662                                  GTDF    EM28,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 21
006662  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006670  012737  000025  002200                                                  MOV     #21,ERRNBR
006676  012737  014460  002202                                                  MOV     #EM28,ERRMSG
006704  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006712  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006714  000451                          BR      6$              ;TAKE ERROR EXIT
006716  032765  000002  177776  3$:     BIT     #BIT1,-2(R5)    ;GET EXPECTED STATE OF REOM
006724  001422                          BEQ     4$              ;BR IF EXPECTED REOM = 0
006726  132737  000002  006576          BITB    #REOM,1$        ;SEE IF REOM = 1
006734  001040                          BNE     5$              ;BR IF REOM = 1
                                ;STACK 'REOM NOT SET'' MSG
006736                                  GTDF    EM31,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 22
006736  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
006744  012737  000026  002200                                                  MOV     #22,ERRNBR
006752  012737  014537  002202                                                  MOV     #EM31,ERRMSG
006760  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
006766  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006770  000423                          BR      6$              ;TAKE ERROR EXIT
006772  132737  000002  006576  4$:     BITB    #REOM,1$        ;SEE IF REOM = 0
```

```
3036  007000  001416                        BEQ     5$                ;BR IF REOM = 0
3037                                ;STACK 'REOM NOT CLEARED'' MSG
3038  007002                                GTDF    EM30,ERR12
3039                                                                  ;       QUEUE 'DEVICE FATAL'' ERROR # 23
3040  007002  012737  000001  002176                                                MOV     #T.EDF,ERRTYP
3041  007010  012737  000027  002200                                                MOV     #23,ERRNBR
3042  007016  012737  014516  002202                                                MOV     #EM30,ERRMSG
3043  007024  012737  021714  002204                                                MOV     #ERR12,ERRBLK
3044  007032  000261                        SEC                       ;SET C BIT TO FLAG ERROR
3045  007034  000401                        BR      6$                ;TAKE ERROR EXIT
3046  007036  000241                  5$:   CLC                       ;CLEAR C BIT FOR NO ERRORS
3047  007040  000205                  6$:   RTS     R5                ;RETURN
3048
3049
```

```
                                .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>
                                ;*************************************************************************
007042                          CHKTSO:
007042  012737  000007  002342          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
007050  004537  003534                  JSR     R5,READI        ;READ USYRT STATUS
007054  122000                          USTATR
007056  000000                  1$:     .WORD   0
007060  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TSO
007064  001422                          BEQ     2$              ;BR IF EXPECTED TSO = 0
007066  132737  000010  007056          BITB    #TSO,1$         ;SEE IF TSO = 1
007074  001040                          BNE     3$              ;BR IF TSO = 1
                                        ;*** STACK 'TSO NOT SET' ERROR ***
007076                                  GTDF    EM100,ERR12
                                                                ;       QUEUE 'DEVICE FATAL' ERROR # 24
007076  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
007104  012737  000030  002200                                                  MOV     #24,ERRNBR
007112  012737  016422  002202                                                  MOV     #EM100,ERRMSG
007120  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
007126  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007130  000423                          BR      4$              ;TAKE ERROR EXIT

007132  132737  000010  007056  2$:     BITB    #TSO,1$         ;SEE IF TSO = 0
007140  001416                          BEQ     3$              ;BR IF TSO = 0
                                        ;*** STACK 'TSO NOT CLEARED' ERROR ***
007142                                  GTDF    EM101,ERR12
                                                                ;       QUEUE 'DEVICE FATAL' ERROR # 25
007142  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
007150  012737  000031  002200                                                  MOV     #25,ERRNBR
007156  012737  016442  002202                                                  MOV     #EM101,ERRMSG
007164  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
007172  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007174  000401                          BR      4$              ;TAKE ERROR EXIT
007176  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
007200  000205                  4$:     RTS     R5              ;RETURN
```

```
                                .SBTTL ....SERIAL -- READ/CHECK TX CHARACTER VIA TSO BIT
                                ;*********************************************************************
                                ;* SERIAL - THIS SUBROUTINE SERIALLY READS/CLOCKS/CHECKS A CHARACTER FROM
                                ;*      THE TRANSMIT SERIAL OUT (TSO) BIT OF THE USYRT STATUS REGISTER,
                                ;*      AND STACKS MESSAGE/SETS 'C'' BIT IF AN INCORRECT CHARACTER IS READ.
                                ;*      NOTE: 'EXPECTED VALUE'' ARGUMENT IS ALWAYS READ RIGHT-TO-LEFT.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,SERIAL
                                ;*      .WORD   <# OF BITS TO BE READ>
                                ;*      .WORD   <EXPECTED VALUE OF SERIAL BIT STREAM>
                                ;*********************************************************************
007202                          SERIAL:
007202  010146                          MOV     R1,-(SP)        ;SAVE R1
007204  010246                          MOV     R2,-(SP)        ;SAVE R2 (TICKS)
007206  010346                          MOV     R3,-(SP)        ;SAVE R3 (EXPECTED_WORD)
                                
007210  005001                          CLR     R1              ;CLEAR ASSEMBLED_WORD
007212  012502                          MOV     (R5)+,R2        ;GET # OF TICKS
                                
007214  006301                  1$:     ASL     R1              ;SHIFT ASSEMBLED_WORD
007216  004537  011540                  JSR     R5,STEPLU       ;CLOCK USYRT ONCE
007222  000001                          1
                                
007224  004537  007042                  JSR     R5,CHKTSO       ;CHECK FOR TSO=1
007230  000001                          1
007232  103401                          BCS     2$              ;BR IF TSO=0
007234  005201                          INC     R1              ;TSO=1: SET LSB OF ASSEMBLED_WORD
007236  077212                  2$:     SOB     R2,1$           ;LOOP UNTIL NO MORE TICKS
                                
007240  012503                          MOV     (R5)+,R3        ;GET EXPECTED_WORD
007242  020103                          CMP     R1,R3           ;COMPARE EXPECTED  AND ASSEMBLED_WORD
007244  001422                          BEQ     3$              ;BR IF CORRECT VALUE READ
                                
007246  010337  002330                  MOV     R3,GDATA        ;EXPECTED_WORD => GDATA
007252  010137  002332                  MOV     R1,BDATA        ;ASSEMBLED_WORD => BDATA
                                        ;*** STACK ''TRANSMISSION ERROR'' MSG ***
007256                                  GTDF    EM106,ERR13
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 26
007256  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
007264  012737  000032  002200                                                  MOV     #26,ERRNBR
007272  012737  016735  002202                                                  MOV     #EM106,ERRMSG
007300  012737  022030  002204                                                  MOV     #ERR13,ERRBLK
007306  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007310  000401                          BR      .+4             ;TAKE ERROR EXIT
                                
007312  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
007314  012603                          MOV     (SP)+,R3        ;RESTORE REGISTERS
007316  012602                          MOV     (SP)+,R2
007320  012601                          MOV     (SP)+,R1
007322  000205                  4$:     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)>
                                ;**********************************************************************
3163  007324                    INITRN:
3164  007324  010146                   MOV      R1,-(SP)          ;SAVE R1
3165  007326  004537  003660           JSR      R5,WRITEI         ;RESET THE USYRT
3166  007332  120000                   VIAORB
3167  007334  000031                   RTSND!DTR!PRESET
3168  007336  004537  003660           JSR      R5,WRITEI         ;CLEAR USYRT RESET BIT
3169  007342  120000                   VIAORB
3170  007344  000030                   RTSND!DTR
3171  007346  112537  007360           MOVB     (R5)+,1$          ;GET VALUE TO LOAD INTO USYRT PCSARL
3172  007352  004537  003660           JSR      R5,WRITEI         ;LOAD USYRT PCSARL
3173  007356  120404                   PCSARL
3174  007360  000000           1$:     .WORD    0
3175  007362  112537  007374           MOVB     (R5)+,2$          ;GET VALUE TO LOAD INTO PCSARH
3176  007366  004537  003660           JSR      R5,WRITEI         ;LOAD USYRT PCSARH
3177  007372  120405                   PCSARH
3178  007374  000000           2$:     .WORD    0
3179  007376  112537  007422           MOVB     (R5)+,3$          ;GET VALUE TO LOAD INTO PCR
3180  007402  005037  002406           CLR      SAVLEN
3181  007406  113737  007422  002406   MOVB     3$,SAVLEN         ;SAVE CHAR LENGTH BITS
3182  007414  004537  003660           JSR      R5,WRITEI         ;LOAD USYRT PCR
3183  007420  120407                   PCR
3184  007422  000000           3$:     .WORD    0
3185  007424  004537  003660           JSR      R5,WRITEI         ;SET ACR FOR T1 ONE-SHOT MODE
3186  007430  120013                   VIAACR
3187  007432  000200                   200
3188  007434  004537  003660           JSR      R5,WRITEI         ;LOAD VIA T1L-L
3189  007440  120006                   VIAT1C
3190  007442  000300                   300
3191  007444  004537  003660           JSR      R5,WRITEI         ;LOAD VIA T1L-H
3192  007450  120007                   VIAT1D
3193  007452  000000                   000
3194  007454  004537  005356           JSR      R5,CKUSTS         ;CHK USYRT STATUS FOR INIT'D STATE
3195  007460  000110                   110                        ; TBMT = 1, TSO = 1
3196  007462  103454                   BCS      7$                ;IF ERROR, EXIT SUBROUTINE
3197
3198  007464  013737  007620  007504   MOV      #0$,13$           ;* SET UP DEFAULT VIAORB PARAMETERS
3199  007472  142537  007504           BICB     (R5)+,13$         ;* CLEAR ANY SPECIFIED VIAORB BITS.
3200
3201  007476  004537  003660           JSR      R5,WRITEI         ;SET UP USYRT
```

```
3202  007502  120000                      VIAORB
3203  007504  000142              13$:    TXEN!RXEN!TTLOOP        ;* THIS VALUE MIGHT BE MODIFIED ABOVE
3204
3205  007506  004537  003660              JSR     R5,WRITEI       ;SET TSOM IN USYRT
3206  007512  120403                      TDSRH
3207  007514  000001                      TSOM
3208  007516  004537  003660              JSR     R5,WRITEI       ;LOAD SYNCH CHAR INTO TX BUF
3209  007522  120402                      TDSRL
3210  007524  000226                      SYNCH
3211  007526  004537  005762              JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
3212  007532  000000                      0
3213  007534  103427                      BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3214  007536  005001                      CLR     R1              ;INIT CYCLE COUNTER
3215  007540  004537  011540      4$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3216  007544  000001                      1
3217  007546  004537  003534              JSR     R5,READI        ;READ USYRT STATUS REG
3218  007552  122000                      USTATR
3219  007554  000000              5$:     .WORD   0
3220  007556  132737  000100  007554      BITB    #TBMT,5$        ;SEE IF TBMT IS SET YET
3221  007564  001010                      BNE     6$              ;BR IF YES
3222  007566  005201                      INC     R1              ;INCR CYCLE COUNTER
3223  007570  020127  000003              CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
3224  007574  002761                      BLT     4$              ;BR IF LESS THAN 3 CYCLES
3225  007576  004537  005762              JSR     R5,CKTBMT       ;GO STACK 'TBMT NOT SET' MSG
3226  007602  000001                      1
3227  007604  103403                      BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3228  007606  004537  005462      6$:     JSR     R5,CKTACT       ;CHK FOR TXACT = 1
3229  007612  000001                      1
3230  007614  012601              7$:     MOV     (SP)+,R1        ;RESTORE R1
3231  007616  000205                      RTS     R5              ;RETURN (IF C = 1, WE HAD AN ERROR)
3232
3233  007620  000142              20$:    TXEN!RXEN!TTLOOP        ;DEFAULT VALUE FOR VIAORB: ENABLE
3234                                                              ;TX AND RX ON USYRT, ASSERT RTS, DTR
3235
```

```
                                        .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)>
                                        ;*********************************************************************************
007622                                  TXCHAR:
007622  010146                          MOV     R1,-(SP)        ;SAVE R1
007624  010246                          MOV     R2,-(SP)        ;SAVE R2
007626  012537  007640                  MOV     (R5)+,1$        ;GET DATA FOR TDSRL
007632  004537  003660                  JSR     R5,WRITEI       ;LOAD DATA INTO TDSRL
007636  120402                          TDSRL
007640  000000                  1$:     .WORD   0
007642  005001                          CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
007644  005002                          CLR     R2              ;CLEAR REQ'D CYCLE COUNT
007646  112502                          MOVB    (R5)+,R2        ;GET DESIRED NO. OF CYCLES
007650  001425                          BEQ     6$              ;BR IF NO CLOCKING DONE
007652  004537  005462          3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
007656  000001                          1
007660  103421                          BCS     6$              ;BR TO EXIT IF ERROR
007662  020102                          CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
007664  001414                          BEQ     5$              ;BR IF YES
                                        
007666  131527  000200                  BITB    (R5),#NCTBMT    ;* CHECK FOR 'TBMT=0 CHECK'' DISABLE
007672  001004                          BNE     7$              ;* BR IF MSB IS NOT SET
                                        
007674  004537  005762                  JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
007700  000000                          0
007702  103410                          BCS     6$              ;BR TO EXIT IF ERROR
007704  004537  011540          7$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
007710  000001                          1
007712  005201                          INC     R1              ;INCR CYCLE COUNT
007714  000756                          BR      3$              ;KEEP CLOCKING
007716  004537  005762          5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
007722  000001                          1
007724  012602                  6$:     MOV     (SP)+,R2        ;RESTORE R2
007726  012601                          MOV     (SP)+,R1        ;RESTORE R1
007730  005205                          INC     R5              ;ADJUST R5 FOR SANE RETURN
007732  000205                          RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

```
3290                                    .SBTTL ....TXCTRL -- CONTROL MESSAGE TRANSMISSION (TDSRH)
3291                                    ;*****************************************************************************
3292                                    ;* TXCTRL - THIS SUBROUTINE ALLOWS CONTROL OF MESSAGE TRANSMISSION BY LOADING
3293                                    ;*       THE USYRT TDSRH WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
3294                                    ;*       FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
3295                                    ;*       PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
3296                                    ;*       MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
3297                                    ;*       IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
3298                                    ;*       IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
3299                                    ;*       DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
3300                                    ;*
3301                                    ;*       CALLING SEQUENCE :
3302                                    ;*       JSR     R5,TXCTRL
3303                                    ;*       .WORD   <DATA FOR TDSRH IN LO BYTE>
3304                                    ;*       .WORD   <NUMBER OF CYCLES TO CLOCK>
3305                                    ;*****************************************************************************
3306  007734                            TXCTRL:
3307  007734  010146                           MOV     R1,-(SP)        ;SAVE R1
3308  007736  010246                           MOV     R2,-(SP)        ;SAVE R2
3309  007740  012537  007752                   MOV     (R5)+,2$        ;GET DATA FOR TDSRH
3310  007744  004537  003660                   JSR     R5,WRITEI       ;LOAD DATA INTO TDSRH
3311  007750  120403                           TDSRH
3312  007752  000000                    2$:    .WORD   0
3313  007754  005001                           CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
3314  007756  012502                           MOV     (R5)+,R2        ;GET DESIRED NO. OF CYCLES
3315  007760  001422                           BEQ     6$              ;BR IF NO CLOCKING DONE
3316  007762  004537  005462            3$:    JSR     R5,CKTACT       ;CHECK TXACT = 1
3317  007766  000001                           1
3318  007770  103416                           BCS     6$              ;BR TO EXIT IF ERROR
3319  007772  020102                           CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3320  007774  001411                           BEQ     5$              ;BR IF YES
3321  007776  004537  005762                   JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
3322  010002  000000                           0
3323  010004  103410                           BCS     6$              ;BR TO EXIT IF ERROR
3324  010006  004537  011540                   JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3325  010012  000001                           1
3326  010014  005201                           INC     R1              ;INCR CYCLE COUNT
3327  010016  000761                           BR      3$              ;KEEP CLOCKING
3328  010020  004537  005762            5$:    JSR     R5,CKTBMT       ;CHK TBMT = 1
3329  010024  000001                           1
3330  010026  012602                    6$:    MOV     (SP)+,R2        ;RESTORE R2
3331  010030  012601                           MOV     (SP)+,R1        ;RESTORE R1
3332  010032  000205                           RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
3333
3334
```

```
                                .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)>
                                ;*************************************************************************
010034                          RXCHAR:
010034  010146                          MOV     R1,-(SP)        ;SAVE R1
010036  010246                          MOV     R2,-(SP)        ;SAVE R2
010040  004537  003534                  JSR     R5,READI        ;READ RDSRH
010044  120401                          RDSRH
010046  000000                  2$:     .WORD   0
010050  004537  003534                  JSR     R5,READI        ;READ RDSRL
010054  120400                          RDSRL
010056  000000                  1$:     .WORD   0
010060  111501                          MOVB    (R5),R1         ;GET EXPECTED RDSRL
010062  042701  177400                  BIC     #177400,R1      ;MASK OFF UNUSED BITS
010066  023727  002406  000347          CMP     SAVLEN,#TXDL!RXDL ;SEE IF 7-BIT CHARS BEING USED
010074  001005                          BNE     3$              ;BR IF NOT 7-BIT CHARS
010076  142737  000200  010056          BICB    #BIT7,1$        ;CLEAR 8TH BIT FOR COMPARE
010104  142701  000200                  BICB    #BIT7,R1
010110  123701  010056          3$:     CMPB    1$,R1           ;COMPARE RCV'D CHAR TO EXPECTED
010114  001462                          BEQ     6$              ;BR IF MATCH
010116  004537  003534                  JSR     R5,READI        ;READ USYRT STATUS REG
010122  122000                          USTATR
010124  000000                  4$:     .WORD   0
010126  132737  000002  010124          BITB    #TXU,4$         ;SEE IF TX UNDERRUN OCCURRED
010134  001421                          BEQ     5$              ;BR IF NOT
010136  012737  000007  002342          MOV     #7,REGNUM       ;SET USYRT REG NO. FOR STATUS REG
                                ;STACK 'TX UNDERRUN'' ERROR
010144                                  GTDF    EM54,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 27
010144  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
010152  012737  000033  002200                                                  MOV     #27,ERRNBR
010160  012737  015330  002202                                                  MOV     #EM54,ERRMSG
010166  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
010174  000137  011274                  JMP     20$             ;TAKE ERROR EXIT
010200  005037  002342          5$:     CLR     REGNUM          ;SET USYRT REG NO. FOR RDSRL
010204  005037  002330                  CLR     GDATA           ;SET EXPECTED DATA
010210  110137  002330                  MOVB    R1,GDATA
010214  005037  002332                  CLR     BDATA           ;SET ACTUAL DATA
010220  113737  010056  002332          MOVB    1$,BDATA
                                ;STACK 'RCV'D DATA MISCOMPARE'' ERROR
```

```
3391  010226                                        GTDF    EM34,ERR10
3392                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 28
3393  010226  012737  000001  002176                                    MOV     #T.EDF,ERRTYP
3394  010234  012737  000034  002200                                    MOV     #28,ERRNBR
3395  010242  012737  014626  002202                                    MOV     #EM34,ERRMSG
3396  010250  012737  021540  002204                                    MOV     #ERR10,ERRBLK
3397  010256  000137  011274                        JMP     20$             ;TAKE ERROR EXIT
3398  010262  116501  000001              6$:     MOVB    1(R5),R1        ;GET RDSRH
3399  010266  042701  177400                      BIC     #177400,R1      ;MASK OFF UNUSED BITS
3400  010272  123701  010046                      CMPB    2$,R1           ;COMPARE RCV'D STATUS TO EXPECTED
3401  010276  001016                              BNE     7$              ;BR IF MISMATCH
3402  010300  000137  011160                      JMP     17$             ;CONTINUE
3403  010304  012737  000001  002342              MOV     #1,REGNUM       ;SET USYRT REG NO. FOR RDSRH
3404  010312  005037  002330                      CLR     GDATA           ;SET EXPECTED DATA
3405  010316  110137  002330                      MOVB    R1,GDATA
3406  010322  005037  002332                      CLR     BDATA           ;SET ACTUAL DATA
3407  010326  113737  010046  002332              MOVB    2$,BDATA
3408  010334  012737  000001  002342      7$:     MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
3409  010342  032765  000001  000002              BIT     #RERCHK,2(R5)   ;SEE IF RCV ERROR BIT SHOULD BE IGNORED
3410  010350  001447                              BEQ     9$              ;BR IF YES
3411                                      ;CHECK RERR BIT
3412  010352  132701  000200                      BITB    #RERR,R1        ;SEE IF EXPECTED BIT = 1
3413  010356  001022                              BNE     8$              ;BR IF YES
3414  010360  132737  000200  010046              BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 0
3415  010366  001440                              BEQ     9$              ;BR IF YES
3416                                      ;STACK 'RERR NOT CLEARED'' MSG
3417  010370                                        GTDF    EM35,ERR12
3418                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 29
3419  010370  012737  000001  002176                                    MOV     #T.EDF,ERRTYP
3420  010376  012737  000035  002200                                    MOV     #29,ERRNBR
3421  010404  012737  014654  002202                                    MOV     #EM35,ERRMSG
3422  010412  012737  021714  002204                                    MOV     #ERR12,ERRBLK
3423  010420  000137  011274                        JMP     20$             ;TAKE ERROR EXIT
3424  010424  132737  000200  010046      8$:     BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 1
3425  010432  001016                              BNE     9$              ;BR IF YES
3426                                      ;STACK 'RERR NOT SET'' MSG
3427  010434                                        GTDF    EM36,ERR12
3428                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 30
3429  010434  012737  000001  002176                                    MOV     #T.EDF,ERRTYP
3430  010442  012737  000036  002200                                    MOV     #30,ERRNBR
3431  010450  012737  014675  002202                                    MOV     #EM36,ERRMSG
3432  010456  012737  021714  002204                                    MOV     #ERR12,ERRBLK
3433  010464  000137  011274                        JMP     20$             ;TAKE ERROR EXIT
3434                                      ;CHECK ROR BIT
3435  010470  132701  000010              9$:     BITB    #ROR,R1         ;SEE IF EXPECTED BIT = 1
3436  010474  001022                              BNE     10$             ;BR IF YES
3437  010476  132737  000010  010046              BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 0
3438  010504  001440                              BEQ     11$             ;BR IF YES
3439                                      ;STACK 'ROR NOT CLEARED'' MSG
3440  010506                                        GTDF    EM16,ERR12
3441                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 31
3442  010506  012737  000001  002176                                    MOV     #T.EDF,ERRTYP
3443  010514  012737  000037  002200                                    MOV     #31,ERRNBR
3444  010522  012737  014327  002202                                    MOV     #EM16,ERRMSG
3445  010530  012737  021714  002204                                    MOV     #ERR12,ERRBLK
3446  010536  000137  011274                        JMP     20$             ;TAKE ERROR EXIT
```

```
3447  010542  132737  000010  010046  10$:    BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 1
3448  010550  001016                          BNE     11$             ;BR IF YES
3449                                  ;STACK 'ROR NOT SET' MSG
3450  010552                                  GTDF    EM14,ERR12
3451                                                                  ;       QUEUE 'DEVICE FATAL' ERROR # 32
3452  010552  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
3453  010560  012737  000040  002200                                                  MOV     #32,ERRNBR
3454  010566  012737  014265  002202                                                  MOV     #EM14,ERRMSG
3455  010574  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
3456  010602  000137  011274                  JMP     20$             ;TAKE ERROR EXIT
3457                                  ;CHECK RABGA BIT
3458  010606  132701  000004          11$:    BITB    #RABGA,R1       ;SEE IF EXPECTED BIT = 1
3459  010612  001022                          BNE     12$             ;BR IF YES
3460  010614  132737  000004  010046        BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 0
3461  010622  001440                          BEQ     13$             ;BR IF YES
3462                                  ;STACK 'RABGA NOT CLEARED' MSG
3463  010624                                  GTDF    EM39,ERR12
3464                                                                  ;       QUEUE 'DEVICE FATAL' ERROR # 33
3465  010624  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
3466  010632  012737  000041  002200                                                  MOV     #33,ERRNBR
3467  010640  012737  014712  002202                                                  MOV     #EM39,ERRMSG
3468  010646  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
3469  010654  000137  011274                  JMP     20$             ;TAKE EPROR EXIT
3470  010660  132737  000004  0100?  12$:    BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 1
3471  010666  001016                          BNE     13$             ;BR IF YES
3472                                  ;STACK 'RABGA NOT SET' MSG
3473  010670                                  GTDF    EM40,ERR12
3474                                                                  ;       QUEUE 'DEVICE FATAL' ERROR # 34
3475  010670  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
3476  010676  012737  000042  002200                                                  MOV     #34,ERRNBR
3477  010704  012737  014734  002202                                                  MOV     #EM40,ERRMSG
3478  010712  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
3479  010720  000137  011274                  JMP     20$             ;TAKE ERROR EXIT
3480                                  ;CHECK REOM BIT
3481  010724  132701  000002          13$:    BITB    #REOM,R1        ;SEE IF EXPECTED BIT = 1
3482  010730  001022                          BNE     14$             ;BR IF YES
3483  010732  132737  000002  010046        BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 0
3484  010740  001440                          BEQ     15$             ;BR IF YES
3485                                  ;STACK 'REOM NOT CLEARED' MSG
3486  010742                                  GTDF    EM30,ERR12
3487                                                                  ;       QUEUE 'DEVICE FATAL' ERROR # 35
3488  010742  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
3489  010750  012737  000043  002200                                                  MOV     #35,ERRNBR
3490  010756  012737  014516  002202                                                  MOV     #EM30,ERRMSG
3491  010764  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
3492  010772  000137  011274                  JMP     20$             ;TAKE ERROR EXIT
3493  010776  132737  000002  010046  14$:    BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 1
3494  011004  001016                          BNE     15$             ;BR IF YES
3495                                  ;STACK 'REOM NOT SET' MSG
3496  011006                                  GTDF    EM31,ERR12
3497                                                                  ;       QUEUE 'DEVICE FATAL' ERROR # 36
3498  011006  012737  000001  002176                                                  MOV     #T.EDF,ERRTYP
3499  011014  012737  000044  002200                                                  MOV     #36,ERRNBR
3500  011022  012737  014537  002202                                                  MOV     #EM31,ERRMSG
3501  011030  012737  021714  002204                                                  MOV     #ERR12,ERRBLK
3502  011036  000137  011274                  JMP     20$             ;TAKE ERROR EXIT
```

```
3503                                    ;CHECK RSOM BIT
3504  011042  132701  000001            15$:    BITB    #RSOM,R1        ;SEE IF EXPECTED BIT = 1
3505  011046  001022                            BNE     16$             ;BR IF YES
3506  011050  132737  000001  010046            BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 0
3507  011056  001440                            BEQ     17$             ;BR IF YES
3508                                    ;STACK 'RSOM NOT CLEARED'' MSG
3509  011060                                    GTDF    EM28,ERR12
3510                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 37
3511  011060  012737  000001  002176                                                    MOV     #T.EDF,ERRTYP
3512  011066  012737  000045  002200                                                    MOV     #37,ERRNBR
3513  011074  012737  014460  002202                                                    MOV     #EM28,ERRMSG
3514  011102  012737  021714  002204                                                    MOV     #ERR12,ERRBLK
3515  011110  000137  011274                    JMP     20$             ;TAKE ERROR EXIT
3516  011114  132737  000001  010046  16$:     BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 1
3517  011122  001016                            BNE     17$             ;BR IF YES
3518                                    ;STACK 'RSOM NOT SET'' MSG
3519  011124                                    GTDF    EM29,ERR12
3520                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 38
3521  011124  012737  000001  002176                                                    MOV     #T.EDF,ERRTYP
3522  011132  012737  000046  002200                                                    MOV     #38,ERRNBR
3523  011140  012737  014501  002202                                                    MOV     #EM29,ERRMSG
3524  011146  012737  021714  002204                                                    MOV     #ERR12,ERRBLK
3525  011154  000137  011274                    JMP     20$             ;TAKE ERROR EXIT
3526
3527  011160  116502  000004            17$:    MOVB    4(R5),R2        ;GET DESIRED NO. OF CYCLES
3528  011164  005001                            CLR     R1              ;INIT CYCLE COUNT
3529
3530  011166  136527  000005  000040  18$:     BITB    5(R5),#BIT5     ;* IS RXACT CHECK TO BE DISABLED ?
3531  011174  001004                            BNE     31$             ;* BR IF YES
3532  011176  004537  005622                    JSR     R5,CKRACT       ;CHK FOR RACT = 1
3533  011202  000001                            1
3534  011204  103433                            BCS     20$             ;BR TO EXIT IF ERROR
3535
3536  011206  020102            31$:    CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3537  011210  001415                            BEQ     19$             ;BR IF YES
3538
3539  011212  136527  000005  000200          BITB    5(R5),#BIT7     ;* SEE IF INITIAL RDA CHECK DESIRED
3540  011220  001004                            BNE     22$             ;* BR IF NO
3541  011222  004537  006122                    JSR     R5,CKRDA        ;CHK FOR RDA = 0
3542  011226  000000                            0
3543  011230  103421                            BCS     20$             ;BR TO EXIT IF ERROR
3544
3545  011232  004537  011540            22$:    JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3546  011236  000001                            1
3547  011240  005201                            INC     R1              ;INCR CYCLE COUNT
3548  011242  000751                            BR      18$             ;CONTINUE CLOCKING
3549
3550  011244  136527  000005  000100  19$:     BITB    5(R5),#BIT6     ;* IS FINAL RDA CHECK TO BE SKIPPED ?
3551  011252  001004                            BNE     30$             ;* BR IF YES
3552  011254  004537  006122                    JSR     R5,CKRDA        ;CHK RDA = 1
3553  011260  000001                            1
3554  011262  103404                            BCS     20$             ;BR IF ERROR
3555
3556  011264  062705  000006            30$:    ADD     #6,R5           ;FIX UP RETURN ADRS
3557  011270  000241                            CLC                     ;SET C = 0 FOR NO ERROR
3558  011272  000403                            BR      21$             ;TAKE ERROR-FREE EXIT
```

```
3559  011274  062705  000006  20$:    ADD     #6,R5           ;FIX UP RETURN ADDRESS
3560  011300  000261                  SEC                     ;SET C BIT FOR ERROR
3561  011302  012602          21$:    MOV     (SP)+,R2        ;RESTORE R2
3562  011304  012601                  MOV     (SP)+,R1        ;RESTORE R1
3563  011306  000205                  RTS     R5              ;RETURN
3564
```

```
                                .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>
                                ;*****************************************************************************
011310                          RCV1ST:
011310  010146                          MOV     R1,-(SP)        ;SAVE R1
011312  010246                          MOV     R2,-(SP)        ;SAVE R2
011314  005001                          CLR     R1              ;INIT CYCLE COUNT
011316  012502                          MOV     (R5)+,R2                ;GET CYCLE COUNT LIMIT
011320  062702  000003                  ADD     #3,R2
011324  004537  005622                  JSR     R5,CKRACT       ;CHK FOR RXACT = 0
011330  000000                          0
011332  103446                          BCS     6$              ;BR TO EXIT IF ERROR
011334  004537  006122                  JSR     R5,CKRDA        ;CHK FOR RDA = 0
011340  000000                          0
011342  103442                          BCS     6$              ;BR TO EXIT IF ERROR
011344  004537  006562                  JSR     R5,CKSEOM       ;CHK FOR RSOM = 0, REOM = 0
011350  000000                          0
011352  103436                          BCS     6$              ;BR TO EXIT IF ERROR
011354  004537  011540          1$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
011360  000001                          1
011362  005201                          INC     R1              ;INCREMENT CYCLE COUNT
011364  004537  003534                  JSR     R5,READI        ;READ USYRT STATUS REG
011370  122000                          USTATR
011372  000000                  2$:     .WORD   0
011374  132737  000200  011372          BITB    #RDA,2$         ;SEE IF RDA SET YET
011402  001006                          BNE     3$              ;BR IF YES
011404  020102                          CMP     R1,R2           ;SEE IF LIMIT EXCEEDED
011406  002762                          BLT     1$              ;BR IF NOT YET
011410  004537  006122                  JSR     R5,CKRDA        ;GO STACK 'RDA NOT SET'' MSG
011414  000001                          1
011416  103414                          BCS     6$              ;BR TO EXIT IF ERROR
011420  020165  177776          3$:     CMP     R1,-2(R5)       ;SEE IF LESS THAN REQUIRED CYCLES
011424  002004                          BGE     4$              ;BR IF NOT
011426  004537  006122                  JSR     R5,CKRDA        ;GO STACK 'RDA NOT CLEARED'' MSG
011432  000000                          0
011434  103405                          BCS     6$              ;BR TO EXIT IF ERROR
011436  004537  005622          4$:     JSR     R5,CKRACT       ;CHK FOR RXACT = 1
011442  000001                          1
011444  103401                          BCS     6$              ;BR TO EXIT IF ERROR
011446  000241                  5$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
011450  012602                  6$:     MOV     (SP)+,R2        ;RESTORE R2
011452  012601                          MOV     (SP)+,R1        ;RESTORE R1
011454  000205                          RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

CVDMDA.P11 10-DEC-80 09:15 ....RCV1ST -- RECEIVE FIRST CHARACTER OF MESSAGE SEQ 83

3621

```
                                .SBTTL ....ENDTRN -- SHUT DOWN TRANSMITTER/RECEIVER
                                ;*****************************************************************************
                                ;* ENDTRN - 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.
                                ;*        NOTE: THIS ROUTINE ASSUMES THAT TTLOOP SHOULD BE ENABLED.
                                ;*
                                ;*       CALLING SEQUENCE :
                                ;*         JSR   R5,ENDTRN
                                ;*         <NO. OF CYCLES TO CLOCK>
                                ;*****************************************************************************
011456                          ENDTRN:
011456  012537  011516                  MOV     (R5)+,2$        ;GET DESIRED NO. OF CYCLES TO CLOCK
011462  004537  005462                  JSR     R5,CKTACT       ;CHK FOR TXACT = 1
000001                                  1
011470  103422                          BCS     6$              ;BR IF ERROR
011472  004537  005622                  JSR     R5,CKRACT       ;CHK FOR RXACT = 1
000001                                  1
011500  103416                          BCS     6$
011502  004537  003660                  JSR     R5,WRITEI       ;CLEAR TXEN AND RXEN IN USYRT
120000                                  VIAORB                  ; ** BUT LEAVE TTLOOP ENABLED **
000002                                  TTLOOP                  ;
011512  004537  011540                  JSR     R5,STEPLU       ;CLOCK LU FOR DESIRED NO. OF CYCLES
011516  000000                  2$:     .WORD   0
011520  004537  005462                  JSR     R5,CKTACT       ;CHK FOR TXACT = 0
011524  000000                          0
011526  103403                          BCS     6$              ;BR IF ERROR
011530  004537  005622                  JSR     R5,CKRACT       ;CHK FOR RXACT = 0
011534  000000                          0
011536  000205                  6$:     RTS     R5
```

```
CVDMDA.P11    10-DEC-80 09:15          ....STEPLU -- CLOCK THE USYRT N TIMES

  3658                                 .SBTTL ....STEPLU -- CLOCK THE USYRT N TIMES
  3659                                 ;*************************************************************************
  3660                                 ;* STEPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NUMBER OF CYCLES
  3661                                 ;*      PASSED IN THE WORD FOLLOWING THE CALL. THE VIA ACR MUST BE PREVIOUSLY
  3662                                 ;*      SET UP FOR T1 ONE-SHOT MODE, AND THE T1 LATCHES MUST BE PREVIOUSLY SET
  3663                                 ;*      TO CONTROL THE WIDTH OF THE CLOCK PULSE. ALL THAT THIS SUBROUTINE
  3664                                 ;*      DOES IS TO LOAD 000 INTO THE HI BYTE OF THE T1 COUNTER, FOR THE
  3665                                 ;*      DESIRED NUMBER OF TIMES.
  3666                                 ;*
  3667                                 ;*      CALLING SEQUENCE :
  3668                                 ;*      JSR     R5,STEPLU
  3669                                 ;*      .WORD   <NUMBER OF CYCLES TO CLOCK>
  3670                                 ;*************************************************************************
  3671  011540                         STEPLU:
  3672  011540  010146                         MOV     R1,-(SP)        ;SAVE R1
  3673  011542  012501                         MOV     (R5)+,R1        ;INIT CYCLE COUNTER
  3674  011544  004537  003660         1$:     JSR     R5,WRITEI       ;LOAD T1C-H, START COUNTER, CLOCK 1 CYCLE
  3675  011550  120005                         VIAT1B
  3676  011552  000000                         000
  3677  011554  005301                         DEC     R1              ;DECR CYCLE COUNTER
  3678  011556  001372                         BNE     1$              ;BR IF ALL CYCLES NOT DONE YET
  3679  011560  012601                         MOV     (SP)+,R1        ;RESTORE R1
  3680  011562  000205                         RTS     R5              ;RETURN
  3681
  3682
  3683
  3684
  3685
  3686
```

```
CVDMDA.P11    10-DEC-80 09:15          GLOBAL ERROR REPORT SECTION

  3687                                 .SBTTL  GLOBAL ERROR REPORT SECTION
  3688
  3689                                 ;////////////////////////////////////////////////////////////////
  3690                                 ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
  3691                                 ;/      THAT ARE USED IN MORE THAN ONE TEST.
  3692                                 ;////////////////////////////////////////////////////////////////
  3693
  3694                                 .NLIST  BEX
      011564  047045  047045     000   ENDEMB: .ASCIZ  /%N%N/
      011571     045  000116           NEWLIN: .ASCIZ  /%N/            ;USED TO TERMINATE ERROR MESSAGES

      011574  047045  040445  040506  FMT2:   .ASCIZ  /%N%AFAILING REG = %T%ASEL%01/
      011631     045  022516  020101  FMT3:   .ASCIZ  /%N%A  EXPECTED:  %03%A   ACTUAL:  %03%A   XOR:  %03/
      011715     045  022516  052101  FMT4:   .ASCIZ  /%N%ATHE CONTENTS OF ALL%T%N%T/
      011753     045  022516  030523  FMT4A:  .ASCIZ  /%N%S1%03%S5%03%S5%03%S5%03/
      012006  047045  052045     000  FMT4B:  .ASCIZ  /%N%T/
      012013     045  022516  032523  FMT4C:  .ASCIZ  /%N%S5%03%S5%03%S5%03%S5%03/
      012046  047045  040445  020040  FMT5:   .ASCIZ  /%N%A  WHEN %03%A LOADED INTO BSEL1/
      012111     045  022516  020101  FMT5A:  .ASCIZ  /%N%A  ATTEMPTING 'M-LOOP'' FUNCTION CODE %02%A (%T%A)/
      012176  047045  040445  042115  FMT7:   .ASCIZ  /%N%AMDIAG #%03%A FAILED/

      012226  047045  040445  020040  FMT10:  .ASCIZ  /%N%A  EXPECTED:%08%A  ACTUAL:%08%A  XOR:%08/
      012302  040445  020040  051514  FMT10A: .ASCIZ  /%A  LSI ADDR:%08/
      012323     045  022516  034117  FMT11:  .ASCIZ  /%N%08%08%08%08/
      012342  047045  047045  052045  FMT12:  .ASCIZ  /%N%N%T/
      012351     045  022516  022524  FMT13:  .ASCIZ  /%N%T%03%S2%03%S2%03%S2%03%S2%03%S2%03/
      012417     045  031123  047445  FMT14:  .ASCIZ  /%S2%03%S2%03/
      012434  040445  020040  042504  FMT15:  .ASCIZ  /%A  DETECTED IN %T%T%A --/
      012466  040445  020040  042504  FMT15A: .ASCIZ  /%A  DETECTED @ TEST PATTERN ELEMENT # %D2/
      012540  047045  052045  047445  FMT16:  .ASCIZ  /%N%T%03%S4%03%S%03/
      012563     045  022516  022524  FMT16A: .ASCIZ  /%N%T%03%S%03%S%03%S4%03%S%03%S%03/
      012625     045  020101  020040  FMT17:  .ASCIZ  /%A    VALUE SENT TO NPR CONTROL REGISTER:  %03/
      012704  047045  040445  020040  FMT17A: .ASCIZ  /%N%A    VALUE READ FROM CONTROL REGISTER:    %03/
      012765     045  022516  020101  FMT17B: .ASCIZ  /%N%A    LSI-11 MEMORY ADDRESS ACCESSED:%08/
      013040  047045  040445  020040  FMT17C: .ASCIZ  /%N%A  INFORMATION ON THE FIRST OF %D5%A ERRORS:/

      013120  047045  040445  042524  FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
      013151     045  022524  033117  FMT21:  .ASCIZ  /%T%06%N/
      013161     045  022516  043101  FMT22:  .ASCIZ  /%N%AFAILING REG: /
      013203     045  042501  050130  FMT23:  .ASCIZ  /%AEXPECTED: %03%S5%AACTUAL: %03%S5%AXOR: %03%N/
      013262  047045  052045  047045  FMT24:  .ASCIZ  /%N%T%N%T%N/
      013275     045  031517  051445  FMT25:  .ASCIZ  /%03%S5%03%S5%03%S5%03%N/
      013325     045  032123  047445  FMT26:  .ASCIZ  /%S4%03%S5%03%S5%03%S5%03%N/
      013360  052045  052045  047045  FMT27:  .ASCIZ  /%T%T%N/
      013367     045  042501  052130  FMT28:  .ASCIZ  /%AEXTENDED REG AX%01%A-%T%N/
      013423     045  022524  000116  FMT29:  .ASCIZ  /%T%N/
      013430  047045  040445  047506  FMT30:  .ASCIZ  /%N%AFOR BAUD RATE SPECIFIED./
      013465     045  022516  044501  FMT31:  .ASCIZ  /%N%AIMPROPER CONNECTOR TYPE SPECIFIED/
      013533     045  022516  043101  FMT32:  .ASCIZ  /%N%AFOR OPTION SPECIFIED./
      013565     045  022516  052101  FMT39:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
      013616  047045  040445  040506  FMT40:  .ASCIZ  /%N%AFAILING RAM ADRS: %06%A (OCT)%N/
      013662  047045  040445  042522  FMT50:  .ASCIZ  /%N%ARESPONDING ADRS: %03%A (OCT)%N/
      013725     045  022516  042501  FMT51:  .ASCIZ  /%N%AEXPECTED COUNT: %D1%A     ACTUAL COUNT: %D1%N/

      014007     122  043505  047040  EM1:    .ASCIZ  /REG NOT INITIALIZED BY MST CLR/
      014046  051525  051131  020124  EM2:    .ASCIZ  /USYRT NOT INITIALIZED BY PROGRAM RESET/
```

```
014115     115  041511  047522  EM3:    .ASCIZ  /MICRO-DIAG. FAILURE/
014141     115  042122  020131  EM4:    .ASCIZ  /MRDY TIMEOUT/
014156  052516  046114  041440  EM5:    .ASCIZ  /NULL CLK BIT STUCK AT 0/
014206  052516  046114  041440  EM6:    .ASCIZ  /NULL CLK BIT STUCK AT 1/
014236  047522  020122  047516  EM13:   .ASCIZ  /ROR NOT CLEARED BY SOM/
014265     122  051117  047040  EM14:   .ASCIZ  /ROR NOT SET/
014301     122  051117  047040  EM15:   .ASCIZ  /ROR NOT CLEARED BY OC/
014327     122  051117  047040  EM16:   .ASCIZ  /ROR NOT CLEARED/
014347     122  040505  027504  EM25:   .ASCIZ  'READ/WRITE DATA ERROR'
014375     111  041516  051117  EM26:   .ASCIZ  /INCORRECT DATA CHAR RCV'D/
014427     111  041516  051117  EM27:   .ASCIZ  /INCORRECT CRC BYTE RCV'D/
014460  051522  046517  047040  EM28:   .ASCIZ  /RSOM NOT CLEARED/
014501     122  047523  020115  EM29:   .ASCIZ  /RSOM NOT SET/
014516  042522  046517  047040  EM30:   .ASCIZ  /REOM NOT CLEARED/
014537     122  047505  020115  EM31:   .ASCIZ  /REOM NOT SET/
014554  054124  040504  040524  EM32:   .ASCIZ  /TXDATA BIT NOT CLEARED/
014603     124  042130  052101  EM33:   .ASCIZ  /TXDATA BIT NOT SET/
014626  041522  023526  020104  EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
014654  042522  051122  047040  EM35:   .ASCIZ  /RERR NOT CLEARED/
014675     122  051105  020122  EM36:   .ASCIZ  /RERR NOT SET/
014712  040522  043502  020101  EM39:   .ASCIZ  /RABGA NOT CLEARED/
014734  040522  043502  020101  EM40:   .ASCIZ  /RABGA NOT SET/
014752  053117  051122  047040  EM41:   .ASCIZ  /OVRR NOT CLEARED/
014773     117  051126  020122  EM42:   .ASCIZ  /OVRR NOT SET/
015010  053523  050040  041501  EM43:   .ASCIZ  /SW PACK #1 INCORRECT/
015035     123  020127  040520  EM44:   .ASCIZ  /SW PACK #2 INCORRECT/
015062  053523  050040  041501  EM45:   .ASCIZ  /SW PACK #3 INCORRECT/
015107     101  051523  046505  EM47:   .ASCIZ  /ASSEMB BIT COUNT INCORRECT/
015142  042117  020104  051126  EM48:   .ASCIZ  'ODD VRC PARITY BIT NOT SET/
015175     117  042104  053040  EM49:   .ASCIZ  /ODD VRC PARITY BIT NOT CLEARED/
015234  053105  047105  053040  EM50:   .ASCIZ  /EVEN VRC PARITY BIT NOT SET/
015270  053105  047105  053040  EM51:   .ASCIZ  /EVEN VRC PARITY BIT NOT CLEARED/
015330  054124  052440  042116  EM54:   .ASCIZ  /TX UNDERRUN ERROR/
015352  052122  020123  047516  EM60:   .ASCIZ  /RTS NOT SET/
015366  052122  020123  047516  EM65:   .ASCIZ  /RTS NOT CLEARED/
015406  042522  020107  044515  EM66:   .ASCIZ  /REG MISCOMPARE/
015425     122  043505  047040  EM67:   .ASCIZ  /REG NOT INITIALIZED BY UNIBUS RESET (INIT)/
015500  051525  051131  020124  EM68:   .ASCIZ  /USYRT STATUS INCORRECT/
015527     124  040530  052103  EM69:   .ASCIZ  /TXACT NOT SET/
015545     124  040530  052103  EM70:   .ASCIZ  /TXACT NOT CLEARED/
015567     122  040530  052103  EM71:   .ASCIZ  /RXACT NOT SET/
015605     122  040530  052103  EM72:   .ASCIZ  /RXACT NOT CLEARED/
015627     124  046502  020124  EM73:   .ASCIZ  /TBMT NOT SET/
015644  041124  052115  047040  EM74:   .ASCIZ  /TBMT NOT CLEARED/
015665     122  040504  047040  EM75:   .ASCIZ  /RDA NOT SET/
015701     122  040504  047040  EM76:   .ASCIZ  /RDA NOT CLEARED/
015721     122  040523  047040  EM77:   .ASCIZ  /RSA NOT SET/
015735     122  040523  047040  EM78:   .ASCIZ  /RSA NOT CLEARED/
015755     122  046501  042440  EM79:   .ASCIZ  /RAM ERROR LOADING MICROCODE/
016011     103  051101  044522  EM80:   .ASCIZ  /CARRIER NOT SET/
016031     103  051101  044522  EM81:   .ASCIZ  /CARRIER NOT CLEARED/
016055     111  053116  046101  EM82:   .ASCIZ  /INVALID ERROR CODE FROM 6502/
016112  047515  042504  020115  EM83:   .ASCIZ  /MODEM STATUS INCORRECT/
016141     103  051524  047040  EM84:   .ASCIZ  /CTS NOT CLRD/
016156  052103  020123  047516  EM85:   .ASCIZ  /CTS NOT SET/
016172  040503  051122  042511  EM86:   .ASCIZ  /CARRIER NOT CLRD/
```

```
016213  103  051101  044522  EM87:   .ASCIZ  /CARRIER NOT SET/
016233  115  042117  046505  EM88:   .ASCIZ  /MODEM RDY NOT CLRD/
016256  047515  042504  020115  EM89:   .ASCIZ  /MODEM RDY NOT SET/
016300  042522  042503  053111  EM90:   .ASCIZ  /RECEIVER OVERRUN NOT SET/
016331  122  041505  044505  EM91:   .ASCIZ  /RECEIVER OVERRUN NOT CLEARED/
016366  041124  052115  044440  EM92:   .ASCIZ  /TBMT INTERRUPT TEST FAILURE/
016422  051524  020117  044502  EM100:  .ASCIZ  /TSO BIT NOT SET/
016442  051524  020117  044502  EM101:  .ASCIZ  /TSO BIT NOT CLEARED/
016466  051525  051131  020124  EM102:  .ASCIZ  /USYRT RESPONDED TO THE WRONG ADDR/
016530  051525  051131  020124  EM103:  .ASCIZ  /USYRT DIDN'T RESPOND TO SECONDARY STATION ADDR/
016607  125  054523  052122  EM104:  .ASCIZ  /USYRT DIDN'T RESPOND TO ALL-PARTIES-ADDR(377)/
016665  125  054523  052122  EM105:  .ASCIZ  /USYRT ASSEMBLED BIT COUNT WAS INCORRECT/
016735  124  040522  051516  EM106:  .ASCIZ  /TRANSMISSION ERROR (AS READ BY TSO BIT)/

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

017005  102  042523  030114  TXT1:   .ASCIZ  /BSEL0   BSEL1   BSEL2   BSEL3/
017043  040  020040  041040  TXT2:   .ASCIZ  /    BSEL4   BSEL5   BSEL6   BSEL7/
017105  102  042523  030514  TXT2A:  .ASCIZ  /BSEL10  BSEL11  BSEL12  BSEL13/
017144  020040  020040  051502  TXT2B:  .ASCIZ  /    BSEL14  BSEL15  BSEL16  BSEL17/
017207  040  054502  042524  TXT3:   .ASCIZ  / BYTE SELECT REG'S ARE:/
017237  040  020040  042523  TXT4:   .ASCIZ  /   SEL0    SEL2    SEL4    SEL6/
017277  040  020040  042523  TXT4A:  .ASCIZ  /   SEL10   SEL12   SEL14   SEL16/
017340  000102  TXT5:   .ASCIZ  /B/
017342  051440  046105  041505  TXT6:   .ASCIZ  / SELECT REG'S ARE:/
017365  040  042522  044507  TXT7:   .ASCIZ  / REGISTERS     ORB  ORA  DDRB DDRA T1CL T1CH T1LL T1LH /
017455  040  020040  020040  TXT7A:  .ASCIZ  /                T2CL T2CH SR   ACR  PCR  IFR  IER  ORA  /
017545  040  054105  042520  TXT8:   .ASCIZ  / EXPECTED:     /
017565  040  041501  052524  TXT9:   .ASCIZ  / ACTUAL:       /
017605  040  047530  035122  TXT10:  .ASCIZ  / XOR:          /
017625  040  047040  020040  TXT11:  .ASCIZ  /  N  P  R      R  E  G  I  S  T  E  R  S:/
017677  040  020040  020040  TXT11A: .ASCIZ  /              CONTROL  D A T A/
017735  040  020040  020040  TXT11B: .ASCIZ  /                 OUT ADDR.      IN ADDR./
020005  104  053105  041511  TXT12:  .ASCIZ  /DEVICE CSR ADDRESS : /
020033  125  054523  052122  TXT13:  .ASCIZ  /USYRT REGS :/
020050  042122  051123  020114  TXT14:  .ASCIZ  /RDSRL   RDSRH   TDSRL   TDSRH/
020106  020040  020040  041520  TXT15:  .ASCIZ  /    PCSARL  PCSARH  PCR     USTAT/
020150  044526  020101  042522  TXT16:  .ASCIZ  /VIA REGS :/
020163  117  041122  020040  TXT17:  .ASCIZ  /ORB     ORA     DDRB    DDRA/
020220  020040  020040  030524  TXT18:  .ASCIZ  /    T1CL    T1CH    T1LL    T1LH/
020261  124  041462  020114  TXT19:  .ASCIZ  /T2CL    T2CH    SR      ACR/
020315  040  020040  050040  TXT20:  .ASCIZ  /    PCR     IFR     IER     ORA/

020355  021  000  TXTNUL: .BYTE   21,0                    ;CTL-Q -- THIS (WE HOPE) IS HARMLESS

020357  116  050117  000  TXTML0: .ASCIZ  /NOP/
020363  122  040505  020104  TXTML1: .ASCIZ  /READ 1 BYTE/
020377  127  044522  042524  TXTML2: .ASCIZ  /WRITE 1 BYTE/
020414  050116  026522  052517  TXTML3: .ASCIZ  /NPR-OUT 256 BYTES/
020436  050116  026522  047111  TXTML4: .ASCIZ  /NPR-IN 256 BYTES/
020457  123  052105  046440  TXTML5: .ASCIZ  /SET MICROPROCESSOR'S PC/
020507  125  042116  043105  TXTML6: .ASCIZ  /UNDEFINED/
020521  101  046114  053517  TXTML7: .ASCIZ  /ALLOW U-PROCESSOR INTERRUPTS/
```

```
020556  044526  020101  042522  TXTVR:  .ASCIZ  /VIA REGISTER /
020574  051117  000102          TXTVR0: .ASCIZ  /ORB/
020600  051117  000101          TXTVR1: .ASCIZ  /ORA/
020604  042104  041122     000  TXTVR2: .ASCIZ  /DDRB/
020611     104  051104  000101  TXTVR3: .ASCIZ  /DDRA/
020616  030524  046103     000  TXTVR4: .ASCIZ  /T1CL/
020623     124  041461  000110  TXTVR5: .ASCIZ  /T1CH/
020630  030524  046114     000  TXTVR6: .ASCIZ  /T1LL/
020635     124  046061  000110  TXTVR7: .ASCIZ  /T1LH/
020642  031124  046103     000  TXTVR8: .ASCIZ  /T2CL/
020647     124  041462  000110  TXTVR9: .ASCIZ  /T2CH/
020654  051123     000          TXTVRA: .ASCIZ  /SR/
020657     101  051103     000  TXTVRB: .ASCIZ  /ACR/
020663     120  051103     000  TXTVRC: .ASCIZ  /PCR/
020667     111  051106     000  TXTVRD: .ASCIZ  /IFR/
020673     111  051105     000  TXTVRE: .ASCIZ  /IER/
020677     117  040522     000  TXTVRF: .ASCIZ  /ORA/

020703     116  051120  000040  TXTNP:  .ASCIZ  /NPR /
020710  047503  052116  047522  TXTNP0: .ASCIZ  /CONTROL/
020720  040504  040524  044040  TXTNP1: .ASCIZ  /DATA HI/
020730  040504  040524  046040  TXTNP2: .ASCIZ  /DATA LO/
020740  042101  051104  020056  TXTNP3: .ASCIZ  /ADDR. OUT EX/
020755     101  042104  027122  TXTNP4: .ASCIZ  /ADDR. OUT HI/
020772  042101  051104  020056  TXTNP5: .ASCIZ  /ADDR. OUT LO/
021007     101  042104  027122  TXTNP6: .ASCIZ  /ADDR. IN EX/
021023     101  042104  027122  TXTNP7: .ASCIZ  /ADDR. IN HI/
021037     101  042104  027122  TXTNP8: .ASCIZ  /ADDR. IN LO/

021053     125  054523  052122  TXTUR:  .ASCIZ  /USYRT REG /
021066  042122  051123  000114  TXTUR0: .ASCIZ  /RDSRL/
021074  042122  051123  000110  TXTUR1: .ASCIZ  /RDSRH/
021102  042124  051123  000114  TXTUR2: .ASCIZ  /TDSRL/
021110  042124  051123  000110  TXTUR3: .ASCIZ  /TDSRH/
021116  041520  040523  046122  TXTUR4: .ASCIZ  /PCSARL/
021125     120  051503  051101  TXTUR5: .ASCIZ  /PCSARH/
021134  041520  000122          TXTUR6: .ASCIZ  /PCR/
021140  051525  040524  000124  TXTUR7: .ASCIZ  /USTAT/
                                .LIST   BEX
3695                            .EVEN
3696
3697
3698                            .SBTTL ....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- 'TXT__T'
3699                            ;-----------------------------------------------------------------
3700                            ;---------- TEXT ADDRESS TABLES USED BY ERROR HANDLERS ------------
3701                            ;-----------------------------------------------------------------
3702
3703  021146  020357  020363  020377  TXTMLT: .WORD   TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7
3704  021154  020414  020436  020457
3705  021162  020507  020521
3706
3707  021166  020556                          .WORD   TXTVR
3708  021170  020574  020600  020604  TXTVRT: .WORD   TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7
3709  021176  020611  020616  020623
3710  021204  020630  020635
```

```
 3711  021210  020642  020647  020654          .WORD   TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF
 3712  021216  020657  020663  020667
 3713  021224  020673  020677
 3714
 3715  021230  020703                          .WORD   TXTNP
 3716  021232  020710  020720  020730  TXTNPT: .WORD   TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8
 3717  021240  020740  020755  020772
 3718  021246  021007  021023  021037
 3719  021254  021066  021074  021102  TXTURT: .WORD   TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7
 3720  021262  021110  021116  021125
 3721  021270  021134  021140
 3722
 3723
```

```
3724
3725
3726
3727    021274
3728    021274
3729    021274  105037  002331          ;----------------------------------------------------------------------
3730    021300  010146                  ;SBTTL ....ERROR HANDLER -- ERR4 -- M-LOOP TIMEOUT ERROR HANDLING
3731    021302  013701  002330          ;----------------------------------------------------------------------
3732    021306  022701  000017                  BGNMSG  ERR4
3733    021312  002012
3734    021314                                                          ERR4::
3735    021314  010146                  CLRB    GDATA+1         ;MAKE SURE BIT 8 DOESN'T PRINT!
3736    021316  012746  012046          MOV     R1,-(SP)        ;SAVE THE WORKING REGISTER
3737    021322  012746  000002          MOV     GDATA,R1        ;SAVE THIS FOR LATER
3738    021326  010600                  CMP     #17,R1          ;WAS THIS AN M-LOOP REQUEST?
3739    021330  104415                  BGE     5$              ;YES, THEN REPORT THE FUNCTION CODE
3740    021332  062706  000006          PRINTX  #FMT5,R1        ;NO,  THEN IT MUST BE A BSEL1 SETTING
3741    021336  000424                                                  MOV     R1,-(SP)
3742                                                                    MOV     #FMT5,-(SP)
3743    021340  001001                                                  MOV     #2,-(SP)
3744    021342  005001                                                  MOV     SP,R0
3745    021344  022701  000007                                          TRAP    C$PNTX
3746    021350  002002                                                  ADD     #6,SP
3747    021352  012701  000006          BR      20$
3748    021356  006301
3749    021360
3750    021360  016146  021146  5$:     BNE     6$              ;IF IT WAS A 17, THIS IS A 'NOP' AND
3751    021364  013746  002330          CLR     R1              ;  THE TEXT POINTER MUST SO REFLECT.
3752    021370  012746  012111  6$:     CMP     #7,R1           ;IS FUNCTION CODE > 7?
3753    021374  012746  000003          BGE     7$              ;NO,  THEN WE CAN HANDLE IT
3754    021400  010600                  MOV     #6,R1           ;YES, THEN IT'S UNDEFINED -- SAY SO
3755    021402  104415          7$:     ASL     R1              ;CONVERT TO A WORD OFFSET
3756    021404  062706  000010          PRINTX  #FMT5A,GDATA,TXTMLT(R1) ;REPORT THE FAILING FUNCTION
3757                                                                    MOV     TXTMLT(R1),-(SP)
3758    021410  012601  20$:     MOV     (SP)+,R1        ;RESTORE THE WORKING REGISTER
3759    021412  004737  022724          JSR     PC,ERR5$        ;DUMP THE SELECT REGISTERS
3760    021416                          ENDMSG
3761    021416                                                          L10002:
3762    021416  104423                                                  TRAP    C$MSG
3763                                    ;----------------------------------------------------------------------
3764                                    ;SBTTL ....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS
3765                                    ;----------------------------------------------------------------------
3766    021420                                  BGNMSG  ERR7A
3767    021420                                                          ERR7A::
3768    021420  113701  002342                  MOVB    REGNUM,R1
3769    021424  006301                          ASL     R1              ;AS PASSED, THIS WAS A BYTE OFFSET
3770    021426                                  PRINTB  #FMT15,#TXTUR,TXTURT(R1)
3771    021426  016146  021254                                                  MOV     TXTURT(R1),-(SP)
3772    021432  012746  021053                                                  MOV     #TXTUR,-(SP)
3773    021436  012746  012434                                                  MOV     #FMT15,-(SP)
3774    021442  012746  000003                                                  MOV     #3,-(SP)
3775    021446  010600                                                          MOV     SP,R0
3776    021450  104414                                                          TRAP    C$PNTB
3777    021452  062706  000010                                                  ADD     #10,SP
3778    021456  004737  022352                  JSR     PC,XORGB
3779    021462                                  PRINTB  #FMT3,GDATA,BDATA,XDATA
```

```
021462  013746  002334                                                          MOV     XDATA,-(SP)
021466  013746  002332                                                          MOV     BDATA,-(SP)
021472  013746  002330                                                          MOV     GDATA,-(SP)
021476  012746  011631                                                          MOV     #FMT3,-(SP)
021502  012746  000004                                                          MOV     #4,-(SP)
021506  010600                                                                  MOV     SP,R0
021510  104414                                                                  TRAP    C$PNTB
021512  062706  000012                                                          ADD     #12,SP
021516                          PRINTB  #ENDEMB
021516  012746  011564                                                          MOV     #ENDEMB,-(SP)
021522  012746  000001                                                          MOV     #1,-(SP)
021526  010600                                                                  MOV     SP,R0
021530  104414                                                                  TRAP    C$PNTB
021532  062706  000004                                                          ADD     #4,SP
021536                          ENDMSG
021536                                                                  L10003:
021536  104423                                                                  TRAP    C$MSG

                        ;---------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)
                        ;---------------------------------------------------------------------

021540                          BGNMSG  ERR10
021540                                                                  ERR10::
021540                          PRINTB  #FMT21,#TXT12,MPCSR
021540  013746  002422                                                          MOV     MPCSR,-(SP)
021544  012746  020005                                                          MOV     #TXT12,-(SP)
021550  012746  013151                                                          MOV     #FMT21,-(SP)
021554  012746  000003                                                          MOV     #3,-(SP)
021560  010600                                                                  MOV     SP,R0
021562  104414                                                                  TRAP    C$PNTB
021564  062706  000010                                                          ADD     #10,SP
021570                          PRINTB  #FMT22
021570  012746  013161                                                          MOV     #FMT22,-(SP)
021574  012746  000001                                                          MOV     #1,-(SP)
021600  010600                                                                  MOV     SP,R0
021602  104414                                                                  TRAP    C$PNTB
021604  062706  000004                                                          ADD     #4,SP
021610  013701  002342          MOV     REGNUM,R1
021614  006301                  ASL     R1                      ;GET PTR TO USYRT REG ASCII
021616                          PRINTB  #FMT27,#TXTUR,TXTURT(R1)
021616  016146  021254                                                          MOV     TXTURT(R1),-(SP)
021622  012746  021053                                                          MOV     #TXTUR,-(SP)
021626  012746  013360                                                          MOV     #FMT27,-(SP)
021632  012746  000003                                                          MOV     #3,-(SP)
021636  010600                                                                  MOV     SP,R0
021640  104414                                                                  TRAP    C$PNTB
021642  062706  000010                                                          ADD     #10,SP
021646  004737  022352          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
021652                          PRINTB  #FMT23,GDATA,BDATA,XDATA
021652  013746  002334                                                          MOV     XDATA,-(SP)
021656  013746  002332                                                          MOV     BDATA,-(SP)
021662  013746  002330                                                          MOV     GDATA,-(SP)
021666  012746  013203                                                          MOV     #FMT23,-(SP)
021672  012746  000004                                                          MOV     #4,-(SP)
021676  010600                                                                  MOV     SP,R0
021700  104414                                                                  TRAP    C$PNTB
```

```
                        ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)

                                                                                ADD     #12,SP
                        JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
                        ENDMSG
                                                                        L10004:
                                                                                TRAP    C$MSG
                        ;-------------------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)
                        ;-------------------------------------------------------------------------------
                        BGNMSG  ERR12
                                                                        ERR12::
                        PRINTB  #FMT21,#TXT12,MPCSR
                                                                                MOV     MPCSR,-(SP)
                                                                                MOV     #TXT12,-(SP)
                                                                                MOV     #FMT21,-(SP)
                                                                                MOV     #3,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTB
                                                                                ADD     #10,SP
                        PRINTB  #FMT22
                                                                                MOV     #FMT22,-(SP)
                                                                                MOV     #1,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTB
                                                                                ADD     #4,SP
                        MOV     REGNUM,R1
                        ASL     R1                      ;GET PTR TO USYRT REG ASCII
                        PRINTB  #FMT27,#TXTUR,TXTURT(R1)
                                                                                MOV     TXTURT(R1),-(SP)
                                                                                MOV     #TXTUR,-(SP)
                                                                                MOV     #FMT27,-(SP)
                                                                                MOV     #3,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTB
                                                                                ADD     #10,SP
                        JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
                        ENDMSG
                                                                        L10005:
                                                                                TRAP    C$MSG

                        ;-------------------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR13 -- RAM ADDRESS ERRORS
                        ;-------------------------------------------------------------------------------
                        BGNMSG  ERR13
                                                                        ERR13::
                        PRINTB  #FMT21,#TXT12,MPCSR
                                                                                MOV     MPCSR,-(SP)
                                                                                MOV     #TXT12,-(SP)
                                                                                MOV     #FMT21,-(SP)
                                                                                MOV     #3,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTB
                                                                                ADD     #10,SP
                        PRINTB  #FMT40,REGNUM
                                                                                MOV     REGNUM,-(SP)
```

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

3892  022064  012746  013616                                        MOV     #FMT40,-(SP)
3893  022070  012746  000002                                        MOV     #2,-(SP)
3894  022074  010600                                               MOV     SP,R0
3895  022076  104414                                               TRAP    C$PNTB
3896  022100  062706  000006                                       ADD     #6,SP
3897  022104  004737  022352          JSR     PC,XORGB        ;COMPUTE XOR OF GOOD AND BAD DATA
3898  022110                          PRINTB  #FMT23,GDATA,BDATA,XDATA
3899  022110  013746  002334                                        MOV     XDATA,-(SP)
3900  022114  013746  002332                                        MOV     BDATA,-(SP)
3901  022120  013746  002330                                        MOV     GDATA,-(SP)
3902  022124  012746  013203                                        MOV     #FMT23,-(SP)
3903  022130  012746  000004                                        MOV     #4,-(SP)
3904  022134  010600                                               MOV     SP,R0
3905  022136  104414                                               TRAP    C$PNTB
3906  022140  062706  000012                                       ADD     #12,SP
3907  022144                          ENDMSG
3908  022144                                                L10006:
3909  022144  104423                                               TRAP    C$MSG
3910
3911
3912                    ;-------------------------------------------------------------------------
3913                    .SBTTL ....ERROR HANDLER -- ERR20 -- USYRT REG DUMP
3914  022146           ;-------------------------------------------------------------------------
3915  022146                          BGNMSG  ERR20
3916  022146                                                ERR20::
3917  022146  013746  002422          PRINTB  #FMT21,#TXT12,MPCSR
3918  022152  012746  020005                                        MOV     MPCSR,-(SP)
3919  022156  012746  013151                                        MOV     #TXT12,-(SP)
3920  022162  012746  000003                                        MOV     #FMT21,-(SP)
3921  022166  010600                                               MOV     #3,-(SP)
3922  022170  104414                                               MOV     SP,R0
3923  022172  062706  000010                                       TRAP    C$PNTB
3924  022176  004737  023454                                       ADD     #10,SP
3925  022202                          JSR     PC,ERR12$       ;GET & PRINT USYRT REGISTERS
3926  022202                          ENDMSG
3927  022202  104423                                        L10007:
3928                                                                TRAP    C$MSG
3929                   ;-------------------------------------------------------------------------
3930                   .SBTTL ....ERROR HANDLER -- ERR21 -- USYRT 'WRONG ADDR' ERROR
3931                   ;-------------------------------------------------------------------------
3932  022204           BGNMSG  ERR21
3933  022204                                                ERR21::
3934  022204           PRINTB  #FMT21,#TXT12,MPCSR
3935  022204  013746  002422                                        MOV     MPCSR,-(SP)
3936  022210  012746  020005                                        MOV     #TXT12,-(SP)
3937  022214  012746  013151                                        MOV     #FMT21,-(SP)
3938  022220  012746  000003                                        MOV     #3,-(SP)
3939  022224  010600                                               MOV     SP,R0
3940  022226  104414                                               TRAP    C$PNTB
3941  022230  062706  000010                                       ADD     #10,SP
3942  022234           PRINTB  #FMT50,R3               ;GET/PRINT RESPONDING ADDRESS
3943  022234  010346                                               MOV     R3,-(SP)
3944  022236  012746  013662                                        MOV     #FMT50,-(SP)
3945  022242  012746  000002                                        MOV     #2,-(SP)
3946  022246  010600                                               MOV     SP,R0
3947  022250  104414                                               TRAP    C$PNTB
```

```
022252  062706  000006                                                          ADD     #6,SP
022256  004737  023454            JSR     PC,ERR12$              ;GET & PRINT USYRT REGISTERS
022262                            ENDMSG
022262                                                                  L10010:
022262  104423                                                                  TRAP    C$MSG


                          ;-------------------------------------------------------------------
                          .SBTTL ....ERROR HANDLER -- ERR22 -- ASSEMBLED BIT COUNT ERROR
                          ;-------------------------------------------------------------------
022264                            BGNMSG  ERR22
022264                                                                  ERR22::
022264                            PRINTB  #FMT21,#TXT12,MPCSR
022264  013746  002422                                                          MOV     MPCSR,-(SP)
022270  012746  020005                                                          MOV     #TXT12,-(SP)
022274  012746  013151                                                          MOV     #FMT21,-(SP)
022300  012746  000003                                                          MOV     #3,-(SP)
022304  010600                                                                  MOV     SP,R0
022306  104414                                                                  TRAP    C$PNTB
022310  062706  000010                                                          ADD     #10,SP
022314                            PRINTB  #FMT51,GDATA,BDATA     ;GET/PRINT GOOD/BAD BIT COUNTS
022314  013746  002332                                                          MOV     BDATA,-(SP)
022320  013746  002330                                                          MOV     GDATA,-(SP)
022324  012746  013725                                                          MOV     #FMT51,-(SP)
022330  012746  000003                                                          MOV     #3,-(SP)
022334  010600                                                                  MOV     SP,R0
022336  104414                                                                  TRAP    C$PNTB
022340  062706  000010                                                          ADD     #10,SP
022344  004737  023454            JSR     PC,ERR12$              ;GET & PRINT USYRT REGISTERS
022350                            ENDMSG
022350                                                                  L10011:
022350  104423                                                                  TRAP    C$MSG


                          .SBTTL ....ERROR HANDLER SUBROUTINES
                          ;*******************************************************************
                          ;************* SUBROUTINES USED ONLY BY ERROR HANDLERS *************
                          ;*******************************************************************
                          ;-------------------------------------------------------------------
                          .SBTTL ........ERROR HANDLER SUBROUTINE -- XORGB
                          ;-------------------------------------------------------------------
                          ;       PERFORM EXCLUSIVE OR BETWEEN ''GDATA'' & ''BDATA'' PUTTING
                          ;       THE RESULT IN ''XDATA''

022352  010146                    XORGB:  MOV     R1,-(SP)        ;PRESERVE WORKING REGISTER
022354  013701  002330                    MOV     GDATA,R1        ;GET ''GOOD'' DATA
022360  013737  002332  002334            MOV     BDATA,XDATA     ;AND ''BAD'' DATA
022366  074137  002334                    XOR     R1,XDATA        ;PERFORM EXCLUSIVE OR
022372  012601                            MOV     (SP)+,R1        ;RESTORE R1
022374  000207                            RTS     PC              ;RETURN

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

022376                    ERR4$:  PRINTX  #FMT4,#TXT3,#TXT1
```

```
4004  022376  012746  017005                    MOV     #TXT1,-(SP)
4005  022402  012746  017207                    MOV     #TXT3,-(SP)
4006  022406  012746  011715                    MOV     #FMT4,-(SP)
4007  022412  012746  000003                    MOV     #3,-(SP)
4008  022416  010600                            MOV     SP,R0
4009  022420  104415                            TRAP    C$PNTX
4010  022422  062706  000010                    ADD     #10,SP
4011  022426
4012  022426  013746  002214                    MOV     BSR3,-(SP)
4013  022432  013746  002212                    MOV     BSR2,-(SP)
4014  022436  013746  002210                    MOV     BSR1,-(SP)
4015  022442  013746  002206                    MOV     BSR0,-(SP)
4016  022446  012746  011753                    MOV     #FMT4A,-(SP)
4017  022452  012746  000005                    MOV     #5,-(SP)
4018  022456  010600                            MOV     SP,R0
4019  022460  104415                            TRAP    C$PNTX
4020  022462  062706  000014                    ADD     #14,SP
4021  022466
4022  022466  012746  017043                    MOV     #TXT2,-(SP)
4023  022472  012746  012006                    MOV     #FMT4B,-(SP)
4024  022476  012746  000002                    MOV     #2,-(SP)
4025  022502  010600                            MOV     SP,R0
4026  022504  104415                            TRAP    C$PNTX
4027  022506  062706  000006                    ADD     #6,SP
4028  022512
4029  022512  013746  002224                    MOV     BSR7,-(SP)
4030  022516  013746  002222                    MOV     BSR6,-(SP)
4031  022522  013746  002220                    MOV     BSR5,-(SP)
4032  022526  013746  002216                    MOV     BSR4,-(SP)
4033  022532  012746  012013                    MOV     #FMT4C,-(SP)
4034  022536  012746  000005                    MOV     #5,-(SP)
4035  022542  010600                            MOV     SP,R0
4036  022544  104415                            TRAP    C$PNTX
4037  022546  062706  000014                    ADD     #14,SP
4038  022552
4039  022552  012746  017105                    MOV     #TXT2A,-(SP)
4040  022556  012746  012006                    MOV     #FMT4B,-(SP)
4041  022562  012746  000002                    MOV     #2,-(SP)
4042  022566  010600                            MOV     SP,R0
4043  022570  104415                            TRAP    C$PNTX
4044  022572  062706  000006                    ADD     #6,SP
4045  022576
4046  022576  013746  002234                    MOV     BSR13,-(SP)
4047  022602  013746  002232                    MOV     BSR12,-(SP)
4048  022606  013746  002230                    MOV     BSR11,-(SP)
4049  022612  013746  002226                    MOV     BSR10,-(SP)
4050  022616  012746  011753                    MOV     #FMT4A,-(SP)
4051  022622  012746  000005                    MOV     #5,-(SP)
4052  022626  010600                            MOV     SP,R0
4053  022630  104415                            TRAP    C$PNTX
4054  022632  062706  000014                    ADD     #14,SP
4055  022636
4056  022636  012746  017144                    MOV     #TXT2B,-(SP)
4057  022642  012746  012006                    MOV     #FMT4B,-(SP)
4058  022646  012746  000002                    MOV     #2,-(SP)
4059  022652  010600                            MOV     SP,R0
```

```
        PRINTX  #FMT4A,BSR0,BSR1,BSR2,BSR3

        PRINTX  #FMT4B,#TXT2

        PRINTX  #FMT4C,BSR4,BSR5,BSR6,BSR7

        PRINTX  #FMT4B,#TXT2A

        PRINTX  #FMT4A,BSR10,BSR11,BSR12,BSR13

        PRINTX  #FMT4B,#TXT2B
```

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

4060  022654  104415                                                TRAP    C$PNTX
4061  022656  062706  000006                                        ADD     #6,SP
4062  022662                  PRINTX  #FMT4C,BSR14,BSR15,BSR16,BSR17
4063  022662  013746  002244                                        MOV     BSR17,-(SP)
4064  022666  013746  002242                                        MOV     BSR16,-(SP)
4065  022672  013746  002240                                        MOV     BSR15,-(SP)
4066  022676  013746  002236                                        MOV     BSR14,-(SP)
4067  022702  012746  012013                                        MOV     #FMT4C,-(SP)
4068  022706  012746  000005                                        MOV     #5,-(SP)
4069  022712  010600                                                MOV     SP,R0
4070  022714  104415                                                TRAP    C$PNTX
4071  022716  062706  000014                                        ADD     #14,SP
4072  022722  000207                  RTS     PC
4073
4074                    ;-------------------------------------------------------------------
4075                    .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR5$
4076                    ;-------------------------------------------------------------------
4077                    ;       COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS
4078  022724            ERR5$:
4079  022724                  PRINTX  #FMT4,#TXT6,#TXT4
4080  022724  012746  017237                                        MOV     #TXT4,-(SP)
4081  022730  012746  017342                                        MOV     #TXT6,-(SP)
4082  022734  012746  011715                                        MOV     #FMT4,-(SP)
4083  022740  012746  000003                                        MOV     #3,-(SP)
4084  022744  010600                                                MOV     SP,R0
4085  022746  104415                                                TRAP    C$PNTX
4086  022750  062706  000010                                        ADD     #10,SP
4087  022754                  PRINTX  #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS
4088  022754  013746  002214                                        MOV     WSR6,-(SP)
4089  022760  013746  002212                                        MOV     WSR4,-(SP)
4090  022764  013746  002210                                        MOV     WSR2,-(SP)
4091  022770  013746  002206                                        MOV     WSR0,-(SP)
4092  022774  012746  012323                                        MOV     #FMT11,-(SP)
4093  023000  012746  000005                                        MOV     #5,-(SP)
4094  023004  010600                                                MOV     SP,R0
4095  023006  104415                                                TRAP    C$PNTX
4096  023010  062706  000014                                        ADD     #14,SP
4097  023014                  PRINTX  #FMT4B,#TXT4A
4098  023014  012746  017277                                        MOV     #TXT4A,-(SP)
4099  023020  012746  012006                                        MOV     #FMT4B,-(SP)
4100  023024  012746  000002                                        MOV     #2,-(SP)
4101  023030  010600                                                MOV     SP,R0
4102  023032  104415                                                TRAP    C$PNTX
4103  023034  062706  000006                                        ADD     #6,SP
4104  023040                  PRINTX  #FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS
4105  023040  013746  002224                                        MOV     WSR16,-(SP)
4106  023044  013746  002222                                        MOV     WSR14,-(SP)
4107  023050  013746  002220                                        MOV     WSR12,-(SP)
4108  023054  013746  002216                                        MOV     WSR10,-(SP)
4109  023060  012746  012323                                        MOV     #FMT11,-(SP)
4110  023064  012746  000005                                        MOV     #5,-(SP)
4111  023070  010600                                                MOV     SP,R0
4112  023072  104415                                                TRAP    C$PNTX
4113  023074  062706  000014                                        ADD     #14,SP
4114  023100                  PRINTB  #ENDEMB
4115  023100  012746  011564                                        MOV     #ENDEMB,-(SP)
```

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

                                                                        MOV     #1,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #4,SP
                        RTS     PC

                        ;-------------------------------------------------------------------------------
                        .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR11$
                        ;-------------------------------------------------------------------------------
                        ;       COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS

                        ERR11$: JSR     PC,GETVRS               ;GET VIA REGS FOR PRINTOUT
                                PRINTX  #FMT24,#TXT16,#TXT17
                                                                        MOV     #TXT17,-(SP)
                                                                        MOV     #TXT16,-(SP)
                                                                        MOV     #FMT24,-(SP)
                                                                        MOV     #3,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #10,SP
                                PRINTX  #FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6
                                                                        MOV     VREGS+6,-(SP)
                                                                        MOV     VREGS+4,-(SP)
                                                                        MOV     VREGS+2,-(SP)
                                                                        MOV     VREGS+0,-(SP)
                                                                        MOV     #FMT25,-(SP)
                                                                        MOV     #5,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #14 SP
                        PRINTX  #FMT29  #TXT18
                                                                        MOV     #TXT18,-(SP)
                                                                        MOV     #FMT29,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #6,SP
                        PRINTX  #FMT26,VREGS+8.,VREGS+10.,VREGS+12.,VREGS+14.
                                                                        MOV     VREGS+14.,-(SP)
                                                                        MOV     VREGS+12.,-(SP)
                                                                        MOV     VREGS+10.,-(SP)
                                                                        MOV     VREGS+8.,-(SP)
                                                                        MOV     #FMT26,-(SP)
                                                                        MOV     #5,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #14,SP
                        PRINTX  #FMT29,#TXT19
                                                                        MOV     #TXT19,-(SP)
                                                                        MOV     #FMT29,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #6,SP
                        PRINTX  #FMT25,VREGS+16.,VREGS+18.,VREGS+20.,VREGS+22.
                                                                        MOV     VREGS+22.,-(SP)
```

```
023332 013746 002312                                                MOV     VREGS+20.,-(SP)
023336 013746 002310                                                MOV     VREGS+18.,-(SP)
023342 013746 002306                                                MOV     VREGS+16.,-(SP)
023346 012746 013275                                                MOV     #FMT25,-(SP)
023352 012746 000005                                                MOV     #5,-(SP)
023356 010600                                                       MOV     SP,R0
023360 104415                                                       TRAP    C$PNTX
023362 062706 000014                                                ADD     #14,SP
023366                          PRINTX  #FMT29,#TXT20
023366 012746 020315                                                MOV     #TXT20,-(SP)
023372 012746 013423                                                MOV     #FMT29,-(SP)
023376 012746 000002                                                MOV     #2,-(SP)
023402 010600                                                       MOV     SP,R0
023404 104415                                                       TRAP    C$PNTX
023406 062706 000006                                                ADD     #6,SP
023412                          PRINTX  #FMT26,VREGS+24.,VREGS+26.,VREGS+28.,VREGS+30.
023412 013746 002324                                                MOV     VREGS+30.,-(SP)
023416 013746 002322                                                MOV     VREGS+28.,-(SP)
023422 013746 002320                                                MOV     VREGS+26.,-(SP)
023426 013746 002316                                                MOV     VREGS+24.,-(SP)
023432 012746 013325                                                MOV     #FMT26,-(SP)
023436 012746 000005                                                MOV     #5,-(SP)
023442 010600                                                       MOV     SP,R0
023444 104415                                                       TRAP    C$PNTX
023446 062706 000014                                                ADD     #14,SP
023452 000207                   RTS     PC


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

023454 004737 004326  ERR12$: JSR     PC,GETURS             ;GET USYRT REGS FOR PRINTOUT
023460                        PRINTX  #FMT24,#TXT13,#TXT14
023460 012746 020050                                                MOV     #TXT14,-(SP)
023464 012746 020033                                                MOV     #TXT13,-(SP)
023470 012746 013262                                                MOV     #FMT24,-(SP)
023474 012746 000003                                                MOV     #3,-(SP)
023500 010600                                                       MOV     SP,R0
023502 104415                                                       TRAP    C$PNTX
023504 062706 000010                                                ADD     #10,SP
023510                        PRINTX  #FMT25,UREGS+0,UREGS+2,UREGS+4,UREGS+6
023510 013746 002254                                                MOV     UREGS+6,-(SP)
023514 013746 002252                                                MOV     UREGS+4,-(SP)
023520 013746 002250                                                MOV     UREGS+2,-(SP)
023524 013746 002246                                                MOV     UREGS+0,-(SP)
023530 012746 013275                                                MOV     #FMT25,-(SP)
023534 012746 000005                                                MOV     #5,-(SP)
023540 010600                                                       MOV     SP,R0
023542 104415                                                       TRAP    C$PNTX
023544 062706 000014                                                ADD     #14,SP
023550                        PRINTX  #FMT29,#TXT15
023550 012746 020106                                                MOV     #TXT15,-(SP)
023554 012746 013423                                                MOV     #FMT29,-(SP)
023560 012746 000002                                                MOV     #2,-(SP)
023564 010600                                                       MOV     SP,R0
```

```
4228 023566 104415                                                          TRAP    C$PNTX
4229 023570 062706 000006                                                   ADD     #6,SP
4230 023574                       PRINTX  #FMT26,UREGS+10,UREGS+12,UREGS+14,UREGS+16
4231 023574 013746 002264                                                   MOV     UREGS+16,-(SP)
4232 023600 013746 002262                                                   MOV     UREGS+14,-(SP)
4233 023604 013746 002260                                                   MOV     UREGS+12,-(SP)
4234 023610 013746 002256                                                   MOV     UREGS+10,-(SP)
4235 023614 012746 013325                                                   MOV     #FMT26,-(SP)
4236 023620 012746 000005                                                   MOV     #5,-(SP)
4237 023624 010600                                                          MOV     SP,R0
4238 023626 104415                                                          TRAP    C$PNTX
4239 023630 062706 000014                                                   ADD     #14,SP
4240 023634 000207                RTS     PC
4241
4242                     .EVEN
```

```
                                .SBTTL  LOAD DEVICE PROTECTION TABLE

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

023636                                  BGNPROT
023636                                                                  L$PROT::
023636  177777                          .WORD   -1      ;DON'T CHK CSR ADRS
023640  177777                          .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
023642  177777                          .WORD   -1      ;DON'T CHK DRIVE NO.
023644                                  ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

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

023644                                  BGNINIT
023644                                                                          L$INIT::

023644  010637  002344                  MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
023650  005037  002350                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
023654  005037  002404                  CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
023660  005037  002402                  CLR     ERROR1          ;CLEAR ERROR FLAG
023664  005037  002406                  CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
023670  005737  002374                  TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
023674  001007                          BNE     6$              ;BR IF NOT
023676  013737  000004  002376          MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
023704  013737  000006  002400          MOV     @#6,SAVE6
023712  000406                          BR      9$

023714  013737  002376  000004  6$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
023722  013737  002400  000006          MOV     SAVE6,@#6

023730  012737  000001  002374  9$:     MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH

                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
023736                                  READEF  #EF.START
023736  012700  000040                                          MOV     #EF.START,R0
023742  104447                                                  TRAP    C$REFG
023744                                  BCOMPLETE       STARST
023744  103415                                                  BCS     STARST

                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
023746                                  READEF  #EF.RESTART
023746  012700  000037                                          MOV     #EF.RESTART,R0
023752  104447                                                  TRAP    C$REFG
023754                                  BCOMPLETE       STARST
023754  103411                                                  BCS     STARST

                                ;SEE IF THIS IS A NEW PASS, BR IF YES
023756                                  READEF  #EF.NEW
023756  012700  000035                                          MOV     #EF.NEW,R0
023762  104447                                                  TRAP    C$REFG
023764                                  BCOMPLETE       NEWST
023764  103411                                                  BCS     NEWST

                                ;SEE IF PROGRAM WAS JUST CONTINUED
023766                                  READEF  #EF.CONTINUE
023766  012700  000036                                          MOV     #EF.CONTINUE,R0
023772  104447                                                  TRAP    C$REFG
023774                                  BCOMPLETE       ENDIT
023774  103470
023776  000414                          BR      GETPRM

                                STARST:

024000
```

```
4312  024000  005037  002416                  CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
4313
4314                                          ;CLEAR DEVICE MAP
4315  024004  005037  002410                  CLR     DEVMAP
4316  024010                                NEWST:
4317  024010  012737  177777  002340          MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
4318  024016  005237  002416                  INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
4319  024022  012737  000001  002412          MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
4320
4321                                          ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
4322                                          ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
4323  024030                                GETPRM:
4324  024030  005237  002340                  INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
4325  024034                                  GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
4326  024034  013700  002340                                                          MOV     LOGDEV,R0
4327  024040  104442                                                                  TRAP    C$GPHRD
4328  024042  010001                                                                  MOV     R0,R1
4329  024044                                  BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
4330  024044  103403                                                                  BCS     10$
4331  024046  006337  002412                  ASL     DEVPTR          ;SHIFT DEVICE POINTER
4332  024052  000766                          BR      GETPRM          ;SKIP THIS DEVICE
4333  024054  053737  002412  002410  10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
4334  024062  006337  002412                  ASL     DEVPTR          ;SHIFT BIT POINTER
4335
4336  024066  012102                          MOV     (R1)+,R2        ;R2=CSR ADDR VALUE
4337  024070  012703  002422                  MOV     #MPCSR,R3       ;R3=POINTER TO CSR ADDR STORAGE AREA
4338
4339  024074  010223                  11$:    MOV     R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
4340  024076  005202                          INC     R2              ;BUMP BSEL ADDR
4341  024100  022703  002462                  CMP     #BSEL17+2,R3    ;ALL 16 ADDRESSES MOVED ?
4342  024104  001373                          BNE     11$             ;NO: DO ANOTHER ADDRESS
4343                                                                  ;YES: CONTINUE
4344
4345  024106  011137  002462                  MOV     (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
4346  024112  012137  002464                  MOV     (R1)+,MPOVEC
4347  024116  062737  000004  002464          ADD     #4,MPOVEC       ;GET DMV11 OUTPUT INTRPT VECTOR
4348  024124  012137  002466                  MOV     (R1)+,MPRIOR    ;GET DMV11 DEVICE PRIORITY
4349  024130  012137  002470                  MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
4350  024134  012137  002472                  MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
4351  024140  012137  002474                  MOV     (R1)+,BRDTYP    ;GET DMV-11 BOARD TYPE
4352  024144  012137  002476                  MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
4353  024150  011137  002500                  MOV     (R1),BDRATE     ;GET BAUD RATE FOR THIS DEVICE
4354                                        ;ISSUE LSI BUS RESET, TO INIT DMV11
4355  024154                                  BRESET
4356  024154  104433                                                                  TRAP    C$RESET
4357  024156                                ENDIT:
4358  024156                                  ENDINIT
4359  024156                                                                L10013:
4360  024156  104411                                                                  TRAP    C$INIT
```

```
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377  024160                          .SBTTL AUTO DROP UNIT SECTION
4378  024160
4379
4380  024160                          ;////////////////////////////////////////////////////////////////////////////
4381  024160                          ;/ THE AUTO DROP CODING DETERMINES WHETHER OR NOT THE DEVICE WHOSE P-TABLE
4382  024160                          ;/ WAS JUST OBTAINED IS READY FOR TESTING, AND IT IS DROPPED IF NOT READY.
4383  024160                          ;////////////////////////////////////////////////////////////////////////////
4384  024160
4385  024160                          ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
4386  024160                          ;
4387  024160                          ;       THIS ALGORITHM IS THE SAME A CVDMA TEST # 1 EXCEPT THAT TEST
4388  024160                          ;       WILL JUST REPORT THE FAILURE AND GO ON -- THIS ROUTINE WILL CAUSE THE
4389  024160                          ;       DEVICE TO BE DROPPED IF A BUS-TIMEOUT OCCURS WHEN ANY OF THE CSR'S
4390  024160                          ;       ARE ACCESSED WITH EITHER A 'TST'' OR 'TSTB'' INSTRUCTION.
4391  024160                          ;
4392  024160                          ;---------------------------------------------------------------------------
4393  024160                                  BGNAUTO
4394  024160                                                                          L$AUTO::
4395  024160                                  SETVEC  #4,#AD.HIT,#0   ;SETUP INVALID-ADDRESS TRAP VECTOR
4396  024160  012746  000000                                                                MOV     #0,-(SP)
4397  024164  012746  024276                                                                MOV     #AD.HIT,-(SP)
4398  024170  012746  000004                                                                MOV     #4,-(SP)
4399  024174  012746  000003                                                                MOV     #3,-(SP)
4400  024200  104437                                                                        TRAP    C$SVEC
4401  024202  062706  000010                                                                ADD     #10,SP
4402  024206  005037  002552                 CLR     TMP0            ;INITIALIZE TRAP FLAG REGISTER
4403  024212  012702  000001                 MOV     #1,R2           ;FLAG BIT
4404  024216  013703  002422                 MOV     BSEL0,R3        ;INIT ADDRESS POINTER
4405  024222                          1$:     TSTB    (R3)+           ;ACCESS THE CSR'S BY BYTES.
4406  024222  105723
4407  024224  006302                          ASL     R2
4408  024226  103375                          BCC     1$
4409  024230                                  MOV     BSEL0,R3        ;RE-INIT ADDRESS POINTER
4410  024230  013703  002422                 MOV     #1,R2           ;RE-INIT FLAG BIT
4411  024234  012702  000001          2$:     TST     (R3)+           ;ACCESS THE CSR'S BY WORDS.
4412  024240  005723                          ASL     R2
4413  024242  006302                          ASL     R2
4414  024244  006302                          BCC     2$
4415  024246  103374                                  CLRVEC  #4              ;RESTORE THE VECTOR TO DS
4416  024250                                                                          MOV     #4,R0
4417  024250  012700  000004                                                        TRAP    C$CVEC
4418  024254  104436                          TST     TMP0            ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP?
4419  024256  005737  002552                 BEQ     AD.OK           ;NO,  EXIT TEST
4420  024262  001403                          DODU    LOGDEV          ;YES, DROP THIS LOGICAL DEV.
4421  024264                                                                          MOV     LOGDEV,R0
4422  024264  013700  002340                                                        TRAP    C$DODU
4423  024270  104451
4424  024272  000240                  AD.OK:  NOP                     ;(FOR PATCHING IN A HALT IF NECESSARY)
4425  024274                                  ENDAUTO
4426  024274                                                                  L10014:
4427  024274  104461                                                                TRAP    C$AUTO
4428
```

```
CVDMDA.P11    10-DEC-80 09:15          AUTO DROP UNIT SECTION

  4417  024276  050237  002552        AD.HIT: BIS     R2,TMP0         ;FLAG THE HIT IF WE GET IT!
  4418  024302  000002                        RTI                     ;RETURN
  4419
```

```
4420
4421
4422                    ;////////////////////////////////////////////////////////////////////////
4423                    ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4424                    ;/ AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
4425                    ;////////////////////////////////////////////////////////////////////////
4426
4427  024304                    BGNCLN
4428  024304                                                          L$CLEAN::
4429
4430
4431  024304                    ENDCLN
4432  024304                                                          L10015:
4433  024304  104412                                                          TRAP    C$CLEAN
```

```
4434
4435
4436
4437
4438
4439
4440
4441  024306
4442  024306
4443
4444  024306
4445  024306  104433
4446  024310
4447  024310
4448  024310  104453
```

```
.SBTTL  DROP UNIT SECTION

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

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

```
                                .SBTTL  ADD UNIT SECTION

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

                                        BGNAU

                                                                                LSAU::

                                        ENDAU

                                                                                L10017:
                                                                                        TRAP    CSAU
```

```
                                    .SBTTL  TEST 1 -- VRC PARITY GENERATION TEST

                                    ;++*********************************************************************
                                    ;*
                                    ;*      TEST 1 -- VRC PARITY GENERATION TEST
                                    ;*
                                    ;* SUBTEST 1 - TEST OF CORRECT ODD VRC PARITY GENERATION :
                                    ;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH ODD VRC. AND 7-BIT CHARS SELECTED.
                                    ;* THE DATA CHARS IN PATTERN Q ARE LOADED/TRANSMITTED/READ. AS THE 8TH BIT
                                    ;* (PARITY BIT) OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TSO FOR THE PROPER
                                    ;* STATE. FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 1, FOR THE
                                    ;* LAST 4 CHARACTERS IT SHOULD = 0.
                                    ;*
                                    ;* SUBTEST 2 - TEST OF CORRECT EVEN VRC PARITY GENERATION :
                                    ;* THE LINE UNIT IS PLACED IN CHAR MODE, WITH EVEN VRC AND 7-BIT CHARS SELECTED.
                                    ;* THE DATA CHARS IN PATTERN Q ARE LOADED/TRANSMITTED/READ. AS THE 8TH BIT
                                    ;* (PARITY BIT) OF EACH DATA CHAR IS SENT THE PROGRAM CHECKS TSO FOR THE PROPER
                                    ;* STATE. FOR THE FIRST 4 CHARS IN PATTERN Q THE PARITY BIT SHOULD = 0, FOR THE
                                    ;* LAST 4 CHARACTERS IT SHOULD = 1.
                                    ;*
                                    ;*    DATA PATTERN Q = 000,003,014,060,001,007,037,177
                                    ;*
                                    ;* NOTE: SINCE THE ROUTINE 'SERIAL'' TREATS THE FIRST BIT RECEIVED FROM THE
                                    ;*       USYRT AS THE MSB, THE 'EXPECTED BIT SEQUENCE'' WILL HAVE A REVERSED
                                    ;*       BIT ORDER.
                                    ;*
                                    ;--*********************************************************************
                                    ;
                                    ;       BGNTST
 024314                                                                     T1::
                                    ;-----------------------------------------------------------------
                                    ; SUBTEST #1: ODD VRC PARITY CHECK
                                    ;-----------------------------------------------------------------
 024314                                     BGNSUB
 024314                                                                     T1.1:
 024314  104402                                                                     TRAP    C$BSUB
 024316  004737  005344                     JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

 024322  004537  007324                     JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
 024326  042226                             DDCMP!OVRC!226          ;SET DDCMP,ODD VRC CHECK,SYNCH=226
 024330  000340                             TXDL                    ;USE 7 BIT TX CHARS
 024332  103003                             BCC     .+8.            ;BR IF NO ERROR
 024334                                     ERROR                   ;REPORT STACKED ERROR
 024334  104460                                                                     TRAP    C$ERROR
 024336                                     ESCAPE  SUB             ;SKIP REMAINDER OF THIS SUBTEST
 024336  104410                                                                     TRAP    C$ESCAPE
 024340  000310                                                                     .WORD   L10021-.

 024342  004537  007734                     JSR     R5,TXCTRL       ;CLEAR TSOM
 024346  000000                             000
 024350  000000                             0

 024352  004537  003660                     JSR     R5,WRITEI       ;LOAD 1ST DATA CHARACTER (000)
 024356  120402                             TDSRL
 024360  000000                             000
 024362  103003                             BCC     .+8.            ;BR IF NO ERROR
```

```
4518  024364
4519  024364  104460                TEST 1 -- VRC PARITY GENERATION TEST
4520  024366
4521  024366  104410                        ERROR                   ;PRINT STACKED ERROR MESSAGE
4522  024370  000260                                                                    TRAP    C$ERROR
4523                                        ESCAPE  SUB             ;AND EXIT SUBTEST
4524  024372  004537  007042                                                          TRAP    C$ESCAPE
4525  024376  000000                                                                  .WORD   L10021-.
4526  024400  103003                ;---------- READ SYNCH CHARACTER ----------
4527  024402                              JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'SYNCH'
4528  024402  104460                      0                       ; CHARACTER (SHOULD BE 0)
4529  024404                              BCC     .+8.            ;BR IF NO ERROR
4530  024404  104410                      ERROR                   ;REPORT STACKED ERROR
4531  024406  000242                                                                  TRAP    C$ERROR
4532                                      ESCAPE  SUB             ;AND EXIT SUBTEST
4533  024410  004537  007202                                                          TRAP    C$ESCAPE
4534  024414  000007                                                                  .WORD   L10021-.
4535  024416  000150
4536  024420  103003                      JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF 'SYNCH' CHARACTER
4537  024422                              7.                      ; (OFF OF TSO BIT)
4538  024422  104460                      150                     ; EXPECTED BIT SEQUENCE (0010110)
4539  024424                              BCC     .+8.            ;BR IF NO ERROR
4540  024424  104410                      ERROR                   ;REPORT STACKED ERROR
4541  024426  000222                                                                  TRAP    C$ERROR
4542                                      ESCAPE  SUB             ;AND EXIT SUBTEST
4543  024430  012703  003012                                                          TRAP    C$ESCAPE
4544  024434  012704  003021                                                          .WORD   L10021-.
4545  024440  112337  024456
4546  024444  112437  024476    ;---------- LOAD/TX/READ PARITY BIT=1 CHARACTERS ----------
4547                              MOV     #PAT0+1,R3      ;SET UP TX CHARACTER POINTER
4548  024450  004537  003660     MOV     #PAT0B,R4       ;SET UP RX CHARACTER POINTER
4549  024454  120402     1$:     MOVB    (R3)+,2$        ;SET UP NEXT TX CHAR
4550  024456  000000             MOVB    (R4)+,3$        ;SET UP NEXT RX CHARACTER
4551  024460  103003
4552  024462                      JSR     R5,WRITEI       ;LOAD NEXT TX CHARACTER
4553  024462  104460             TDSRL
4554  024464                2$:   000                     ;** HOLE FOR TX CHARACTER
4555  024464  104410              BCC     .+8.            ;BR IF NO ERROR
4556  024466  000162              ERROR                   ;PRINT STACKED ERROR MESSAGE
4557                                                                          TRAP    C$ERROR
4558  024470  004537  007202      ESCAPE  SUB             ;AND EXIT SUBTEST
4559  024474  000007                                                        TRAP    C$ESCAPE
4560  024476  000000                                                        .WORD   L10021-.
4561  024500  103003
4562  024502                      JSR     R5,SERIAL       ;CLOCK/CHECK PREVIOUS TX CHAR (1 CHAR BUFFER)
4563  024502  104460             7
4564  024504               3$:    000                     ;** HOLE FOR EXPECTED BIT SEQUENCE
4565  024504  104410              BCC     .+8.            ;BR IF NO ERROR
4566  024506  000142              ERROR                   ;REPORT STACKED ERROR
4567                                                                          TRAP    C$ERROR
4568  024510  004537  011540      ESCAPE  SUB             ;SKIP REMAINDER OF THIS SUBTEST
4569  024514  000001                                                        TRAP    C$ESCAPE
4570                                                                        .WORD   L10021-.
4571  024516  004537  007042
4572  024522  000001              JSR     R5,STEPLU       ;CLOCK PARITY BIT TO TSO
4573  024524  103006              1
                                  JSR     R5,CHKTSO       ;CHECK STATE OF PARITY BIT
                                  1                       ; (SHOULD BE 1)
                                  BCC     4$              ;BR IF NO ERROR
```

```
4574  024526                          GEDF    EM48,ERR12      ;REPORT 'ODD VRC PARITY BIT NOT SET''
4575                                                          ;       'DEVICE FATAL'' ERROR # 39
4576  024526  104455                                                          TRAP    C$ERDF
4577  024530  000047                                                          .WORD   39
4578  024532  015142                                                          .WORD   EM48
4579  024534  021714                                                          .WORD   ERR12
4580  024536                          ESCAPE  SUB             ;SKIP REMAINDER OF THIS SUBTEST
4581  024536  104410                                                          TRAP    C$ESCAPE
4582  024540  000110                                                          .WORD   L10021-.
4583
4584  024542  020327  003016    4$:   CMP     R3,#PATQ+5      ;
4585  024546  001334                  BNE     1$              ;BR IF TSO=1 CHECKS ARE NOT COMPLETE
4586                            ;---------- LOAD/TX/READ PARITY BIT=0 CHARACTERS ----------
4587  024550  112337  024566    11$:  MOVB    (R3)+,12$       ;SET UP NEXT TX CHAR
4588  024554  112437  024576          MOVB    (R4)+,13$       ;SET UP NEXT RX CHARACTER
4589
4590  024560  004537  003660          JSR     R5,#WRITEI      ;LOAD NEXT TX CHARACTER
4591  024564  120402                  TDSRL
4592  024566  000000    12$:  000                     ;** HOLE FOR TX CHARACTER
4593
4594  024570  004537  007202          JSR     R5,SERIAL       ;CLOCK/CHECK PREVIOUS TX CHAR (1 CHAR BUFFER)
4595  024574  000007                  7
4596  024576  000000    13$:  000                     ;** HOLE FOR EXPECTED BIT SEQUENCE
4597  024600  103003                  BCC     .+8.            ;BR IF NO ERROR
4598  024602                          ERROR                   ;REPORT STACKED ERROR
4599  024602  104460                                                          TRAP    C$ERROR
4600  024604                          ESCAPE  SUB             ;SKIP REMAINDER OF THIS SUBTEST
4601  024604  104410                                                          TRAP    C$ESCAPE
4602  024606  000042                                                          .WORD   L10021-.
4603
4604  024610  004537  011540          JSR     R5,STEPLU       ;CLOCK PARITY BIT TO TSO
4605  024614  000001                  1
4606
4607  024616  004537  007042          JSR     R5,CHKTSO       ;CHECK STATE OF PARITY BIT
4608  024622  000000                  0                       ; (SHOULD BE 0)
4609  024624  103006                  BCC     14$             ;BR IF NO ERROR
4610  024626                          GEDF    EM49,ERR12      ;REPORT 'ODD VRC PARITY BIT NOT CLEARED''
4611                                                          ;       'DEVICE FATAL'' ERROR # 40
4612  024626  104455                                                          TRAP    C$ERDF
4613  024630  000050                                                          .WORD   40
4614  024632  015175                                                          .WORD   EM49
4615  024634  021714                                                          .WORD   ERR12
4616  024636                          ESCAPE  SUB             ;SKIP REMAINDER OF SUBTEST
4617  024636  104410                                                          TRAP    C$ESCAPE
4618  024640  000010                                                          .WORD   L10021-.
4619
4620  024642  020327  003022    14$:  CMP     R3,#PATQ+9.     ;
4621  024646  001340                  BNE     11$             ;BR IF TSO=0 CHECKS ARE NOT COMPLETE
4622  024650                          ENDSUB
4623  024650                                                                  L10021:
4624  024650  104403                                                          TRAP    C$ESUB
4625
4626                            ;-------------------------------------------------------------
4627                            ; SUBTEST #2: EVEN VRC PARITY CHECK
4628                            ;-------------------------------------------------------------
4629  024652                          BGNSUB
```

```
4630  024652
4631  024652  104402                                                    T1.2:
4632  024654  004737  005344          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP  TRAP    C$BSUB
4633
4634  024660  004537  007324          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
4635  024664  042626                  DDCMP!EVRC!226          ;SET DDCMP,EVEN VRC CHECK,SYNCH=226
4636  024666  000340                  TXDL                    ;USE 7 BIT TX CHARS
4637  024670  103003                  BCC     .+8.            ;BR IF NO ERROR
4638  024672                          ERROR                   ;REPORT STACKED ERROR
4639  024672  104460                                                      TRAP    C$ERROR
4640  024674                          ESCAPE  SUB             ;SKIP TO END OF SUBTEST
4641  024674  104410                                                      TRAP    C$ESCAPE
4642  024676  000320                                                      .WORD   L10022-.
4643
4644  024700  004537  007734          JSR     R5,TXCTRL       ;CLEAR TSOM
4645  024704  000000                  000
4646  024706  000000                  0
4647  024710  004537  003660          JSR     R5,WRITEI       ;LOAD 1ST DATA CHARACTER (000)
4648  024714  120402                  TDSRL
4649  024716  000000                  000
4650  024720  103003                  BCC     .+8.            ;BR IF NO ERROR
4651  024722                          ERROR                   ;PRINT STACKED ERROR MESSAGE
4652  024722  104460                                                      TRAP    C$ERROR
4653  024724                          ESCAPE  SUB             ;AND EXIT SUBTEST
4654  024724  104410                                                      TRAP    C$ESCAPE
4655  024726  000270                                                      .WORD   L10022-.
4656
4657  024730  004537  007042  ;---------- READ SYNCH CHARACTER ----------
                                JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'SYNCH'
4658  024734  000000                  0                       ; CHARACTER (SHOULD BE 0)
4659  024736  103003                  BCC     .+8.            ;BR IF NO ERROR
4660  024740                          ERROR                   ;REPORT STACKED ERROR
4661  024740  104460                                                      TRAP    C$ERROR
4662  024742                          ESCAPE  SUB             ;AND EXIT SUBTEST
4663  024742  104410                                                      TRAP    C$ESCAPE
4664  024744  000252                                                      .WORD   L10022-.
4665
4666  024746  004537  007202          JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF 'SYNCH' CHARACTER
4667  024752  000007                  7.                      ; (OFF OF TSO BIT)
4668  024754  000151                  151                     ; EXPECTED BIT SEQUENCE (0010110)
4669  024756  103003                  BCC     .+8.            ;BR IF NO ERROR
4670  024760                          ERROR                   ;REPORT STACKED ERROR
4671  024760  104460                                                      TRAP    C$ERROR
4672  024762                          ESCAPE  SUB             ;AND EXIT SUBTEST
4673  024762  104410                                                      TRAP    C$ESCAPE
4674  024764  000232                                                      .WORD   L10022-.
4675
4676  024766  012703  003012  ;---------- LOAD/TX/READ PARITY BIT=0 CHARACTERS ----------
                                MOV     #PATQ+1,R3      ;SET UP TX CHARACTER POINTER
4677  024772  012704  003021          MOV     #PATQB,R4       ;SET UP RX CHARACTER POINTER
4678  024776  112337  025014  1$:     MOVB    (R3)+,2$        ;SET UP NEXT TX CHAR
4679  025002  112437  025034          MOVB    (R4)+,3$        ;SET UP NEXT RX CHARACTER
4680
4681  025006  004537  003660          JSR     R5,WRITEI       ;LOAD NEXT TX CHARACTER
4682  025012  120402                  TDSRL
4683  025014  000000          2$:     000                     ;** HOLE FOR TX CHARACTER
4684  025016  103003                  BCC     .+8.            ;BR IF NO ERROR
4685  025020                          ERROR                   ;PRINT STACKED ERROR MESSAGE
```

```
025020  104460                                                          TRAP    C$ERROR
025022                          ESCAPE  SUB             ;AND EXIT SUBTEST
025022  104410                                                          TRAP    C$ESCAPE
025024  000172                                                          .WORD   L10022-.

025026  004537  007202          JSR     R5,SERIAL       ;CLOCK/CHECK PREVIOUS TX CHAR (1 CHAR BUFFER)
025032  000007                  7
025034  000000          3$:     000                     ;** HOLE FOR EXPECTED BIT SEQUENCE
025036  103003                  BCC     .+8.            ;BR IF NO ERROR
025040                          ERROR                   ;REPORT STACKED ERROR
025040  104460                                                          TRAP    C$ERROR
025042                          ESCAPE  SUB             ;SKIP REMAINDER OF THIS SUBTEST
025042  104410                                                          TRAP    C$ESCAPE
025044  000152                                                          .WORD   L10022-.

025046  004537  011540          JSR     R5,STEPLU       ;CLOCK PARITY BIT TO TSO
025052  000001                  1

025054  004537  007042          JSR     R5,CHKTSO       ;CHECK STATE OF PARITY BIT
025060  000000                  0                       ; (SHOULD BE 0)
025062  103006                  BCC     4$              ;BR IF NO ERROR
025064                          GEDF    EM51,ERR12      ;REPORT ''EVEN VRC PARITY NOT CLEARED''
                                                        ;        ''DEVICE FATAL'' ERROR # 41
025064  104455                                                          TRAP    C$ERDF
025066  000051                                                          .WORD   41
025070  015270                                                          .WORD   EM51
025072  021714                                                          .WORD   ERR12
025074                          ESCAPE  SUB             ;SKIP REMAINDER OF THIS SUBTEST
025074  104410                                                          TRAP    C$ESCAPE
025076  000120                                                          .WORD   L10022-.

025100  020327  003016  4$:     CMP     R3,#PATQ+5      ;
025104  001334                  BNE     1$              ;BR IF TSO=0 CHECKS ARE NOT COMPLETE
                        ;---------- LOAD/TX/READ PARITY BIT=1 CHARACTERS ----------
025106  112337  025124  11$:    MOVB    (R3)+,12$       ;SET UP NEXT TX CHAR
025112  112437  025144          MOVB    (R4)+,13$       ;SET UP NEXT RX CHARACTER

025116  004537  003660          JSR     R5,WRITEI       ;LOAD NEXT TX CHARACTER
025122  120402                  TDSRL
025124  000000          12$:    000                     ;** HOLE FOR TX CHARACTER
025126  103003                  BCC     .+8.            ;BR IF NO ERROR
025130                          ERROR                   ;PRINT STACKED ERROR MESSAGE
025130  104460                                                          TRAP    C$ERROR
025132                          ESCAPE  SUB             ;AND EXIT SUBTEST
025132  104410                                                          TRAP    C$ESCAPE
025134  000062                                                          .WORD   L10022-.

025136  004537  007202          JSR     R5,SERIAL       ;CLOCK/CHECK PREVIOUS TX CHAR (1 CHAR BUFFER)
025142  000007                  7
025144  000000          13$:    000                     ;** HOLE FOR EXPECTED BIT SEQUENCE
025146  103003                  BCC     .+8.            ;BR IF NO ERROR
025150                          ERROR                   ;REPORT STACKED ERROR
025150  104460                                                          TRAP    C$ERROR
025152                          ESCAPE  SUB             ;SKIP REMAINDER OF THIS SUBTEST
025152  104410                                                          TRAP    C$ESCAPE
025154  000042                                                          .WORD   L10022-.
```

```
 4742
 4743  025156  004537  011540          JSR     R5,STEPLU       ;CLOCK PARITY BIT TO TSO
 4744  025162  000001                  1
 4745
 4746  025164  004537  007042          JSR     R5,CHKTSO       ;CHECK STATE OF PARITY BIT
 4747  025170  000001                  1                       ; (SHOULD BE 1)
 4748  025172  103006                  BCC     14$             ;BR IF NO ERROR
 4749  025174                          GEDF    EM50,ERR12      ;REPORT 'EVEN VRC PARITY NOT SET''
 4750                                                          ;       'DEVICE FATAL'' ERROR # 42
 4751  025174  104455                                                          TRAP    C$ERDF
 4752  025176  000052                                                          .WORD   42
 4753  025200  015234                                                          .WORD   EM50
 4754  025202  021714                                                          .WORD   ERR12
 4755  025204                          ESCAPE  SUB             ;SKIP REMAINDER OF SUBTEST
 4756  025204  104410                                                          TRAP    C$ESCAPE
 4757  025206  000010                                                          .WORD   L10022-.
 4758
 4759  025210  020327  003022  14$:    CMP     R3,#PATQ+9.     ;
 4760  025214  001334                  BNE     11$             ;BR IF TSO=1 CHECKS ARE NOT COMPLETE
 4761  025216                          ENDSUB
 4762  025216                                                  L10022:
 4763  025216  104403                                                          TRAP    C$ESUB
 4764  025220                  ENDTST
 4765  025220                                                  L10020:
 4766  025220  104401                                                          TRAP    C$ETST
```

```
                                  .SBTTL  TEST 2 -- VRC ERROR DETECTION TEST

                                  ;++*********************************************************************
                                  ;*
                                  ;*        TEST 2 -- VRC ERROR DETECTION TEST
                                  ;*
                                  ;* SUBTEST 1 - FORCING OF RERR USING ODD VRC
                                  ;* THE USYRT IS PLACED IN CHAR MODE WITH ODD VRC AND BOTH TX AND RX CHAR
                                  ;* LENGTH=7 BITS. THE RECEIVER AND TRANSMITTER ARE THEN SYNC'D. WHEN THE FIRST
                                  ;* DATA CHARACTER IS LOADED INTO TXDB, THE RX CHAR LENGTH IS CHANGED TO 6 BITS.
                                  ;* TWO 7 BIT CHARACTERS (+PARITY) ARE THEN TRANSMITTED, RESULTING IN A 16 BIT
                                  ;* STREAM WHICH THE RECEIVER WILL READ AS TWO 6 BIT CHARS (+PARITY + 2 LEFT).
                                  ;* THE FIRST ''CHARACTER'' READ WILL HAVE THE CORRECT PARITY; THE SECOND WILL
                                  ;* NOT.
                                  ;*
                                  ;* SUBTEST 2 - FORCING OF RERR USING EVEN VRC
                                  ;* THE USYRT IS PLACED IN CHAR MODE WITH EVEN VRC AND BOTH TX AND RX CHAR
                                  ;* LENGTH=7 BITS. THE RECEIVER AND TRANSMITTER ARE THEN SYNC'D. WHEN THE FIRST
                                  ;* DATA CHARACTER IS LOADED INTO TXDB, THE RX CHAR LENGTH IS CHANGED TO 6 BITS.
                                  ;* TWO 7 BIT CHARACTERS (+PARITY) ARE THEN TRANSMITTED, RESULTING IN A 16 BIT
                                  ;* STREAM WHICH THE RECEIVER WILL READ AS TWO 6 BIT CHARS (+PARITY + 2 LEFT).
                                  ;* THE FIRST ''CHARACTER'' READ WILL HAVE THE CORRECT PARITY; THE SECOND WILL
                                  ;* NOT.
                                  ;*
                                  ;*
                                  ;--*********************************************************************
                                  ;
                                  ;       BGNTST
 025222                                                                  T2::
                                  ;-----------------------------------------------------------
                                  ; SUBTEST #1: FORCING ODD VRC ERROR
                                  ;-----------------------------------------------------------
 025222                                 BGNSUB
 025222                                                                  T2.1:
 025222  104402                                                                  TRAP    C$BSUB
 025224  004737  005344                 JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
 025230  004537  007324                 JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
 025234  042226                         DDCMP!OVRC!226          ;SET DDCMP,ODD VRC CHECK,SYNCH=226
 025236  000347                         TXDL!RXDL               ;TX/RX CHAR LENGTH=7 BITS
 025240  103003                         BCC     .+8.            ;BR IF NO ERROR
 025242                                 ERROR                   ;REPORT STACKED ERROR
 025242  104460                                                                  TRAP    C$ERROR
 025244                                 ESCAPE  SUB             ;SKIP TO END OF TEST
 025244  104410                                                                  TRAP    C$ESCAPE
 025246  000256                                                                  .WORD   L10024-.

 025250  004537  007734                 JSR     R5,TXCTRL       ;SET TSOM
 025254  000001                         TSOM
 025256  000007                         7.
 025260  004537  007734                 JSR     R5,TXCTRL       ;SET TSOM AGAIN (KNOCK DOWN TBMT)
 025264  000001                         TSOM
 025266  000010                         8.
 025270  004537  007734                 JSR     R5,TXCTRL       ;CLEAR TSOM
 025274  000000                         000
 025276  000000                         0
 025300  004537  007622                 JSR     R5,TXCHAR       ;LOAD 043, TX 3RD SYNCH
```

```
4823  025304  000043                  043
4824  025306  000010                  8.
4825  025310  103003                  BCC     .+8.            ;BR IF NO ERROR
4826  025312                          ERROR                   ;REPORT STACKED ERROR
4827  025312  104460                                                  TRAP    C$ERROR
4828  025314                          ESCAPE  SUB             ;SKIP TO END OF TEST
4829  025314  104410                                                  TRAP    C$ESCAPE
4830  025316  000206                                                  .WORD   L10024-.
4831
4832  025320  004537  003660          JSR     R5,WRITEI       ;SET RX CHAR LENGTH=6 BITS
4833  025324  120407                  PCR
4834  025326  000346                  TXDL!6                  ;TXCL=7, RXCL=6
4835  025330  103003                  BCC     .+8.            ;BR IF NO ERROR
4836  025332                          ERROR                   ;PRINT STACKED ERROR MESSAGE
4837  025332  104460                                                  TRAP    C$ERROR
4838  025334                          ESCAPE  SUB             ;AND EXIT SUBTEST
4839  025334  104410                                                  TRAP    C$ESCAPE
4840  025336  000166                                                  .WORD   L10024-.
4841
4842  025340  004537  007622          JSR     R5,TXCHAR       ;LOAD 036
4843  025344  000036                  036
4844  025346  000010                  8.
4845  025350  103003                  BCC     .+8.            ;BR IF NO ERROR
4846  025352                          ERROR                   ;REPORT STACKED ERROR
4847  025352  104460                                                  TRAP    C$ERROR
4848  025354                          ESCAPE  SUB             ;SKIP TO END OF TEST
4849  025354  104410                                                  TRAP    C$ESCAPE
4850  025356  000146                                                  .WORD   L10024-.
4851
4852  025360  004537  007622          JSR     R5,TXCHAR       ;LOAD FILLER (000)
4853  025364  000000                  000
4854  025366  000010                  8.
4855  025370  103003                  BCC     .+8.            ;BR IF NO ERROR
4856  025372                          ERROR                   ;REPORT STACKED ERROR
4857  025372  104460                                                  TRAP    C$ERROR
4858  025374                          ESCAPE  SUB             ;SKIP TO END OF TEST
4859  025374  104410                                                  TRAP    C$ESCAPE
4860  025376  000126                                                  .WORD   L10024-.
4861
4862  025400  004537  007622          JSR     R5,TXCHAR       ;LOAD FILLER (000)
4863  025404  000000                  000
4864  025406  000010                  8.
4865  025410  103003                  BCC     .+8.            ;BR IF NO ERROR
4866  025412                          ERROR                   ;REPORT STACKED ERROR
4867  025412  104460                                                  TRAP    C$ERROR
4868  025414                          ESCAPE  SUB             ;SKIP TO END OF TEST
4869  025414  104410                                                  TRAP    C$ESCAPE
4870  025416  000106                                                  .WORD   L10024-.
4871
4872  025420  004537  010034          JSR     R5,RXCHAR       ;READ/CHK SYNCH CHARACTER
4873  025424  000026                  026                     ;
4874  025426  000001                  RERCHK                  ;CHECK RERR (NO VRC ERROR EXPECTED)
4875  025430  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
4876  025432  103003                  BCC     .+8.            ;BR IF NO ERROR
4877  025434                          ERROR                   ;REPORT STACKED ERROR
4878  025434  104460                                                  TRAP    C$ERROR
```

```
4879  025436
4880  025436  104410                                                    TRAP    C$ESCAPE
4881  025440  000064                                                    .WORD   L10024-.
4882
4883  025442  004537  010034          JSR     R5,RXCHAR       ;READ/CHK 6 BIT CHARACTER
4884  025446  000043                  043                     ;EXPECTED 1ST 'CHARACTER' (043)
4885  025450  000001                  RERCHK                  ;CHECK RERR (NO VRC ERROR EXPECTED)
4886  025452  100000                  NOCRDA                  ;DON'T CHECK INITIAL RDA=0
4887  025454  103003                  BCC     .+8.            ;BR IF NO ERROR
4888  025456                          ERROR                   ;REPORT STACKED ERROR
4889  025456  104460                                                    TRAP    C$ERROR
4890  025460                          ESCAPE  SUB             ;SKIP TO END OF TEST
4891  025460  104410                                                    TRAP    C$ESCAPE
4892  025462  000042                                                    .WORD   L10024-.
4893
4894  025464  004537  010034          JSR     R5,RXCHAR       ;READ/CHK 6 BIT CHARACTER
4895  025470  100074                  RXERR!074               ;EXPECTED 2ND 'CHARACTER' (074)
4896  025472  000001                  RERCHK                  ;CHECK RERR (VRC ERROR IS EXPECTED)
4897  025474  100000                  NOCRDA          ;DON'T CHECK INITIAL RDA=0
4898  025476  103003                  BCC     .+8.            ;BR IF NO ERROR
4899  025500                          ERROR                   ;REPORT STACKED ERROR
4900  025500  104460                                                    TRAP    C$ERROR
4901  025502                          ESCAPE  SUB             ;SKIP TO END OF TEST
4902  025502  104410                                                    TRAP    C$ESCAPE
4903  025504  000020                                                    .WORD   L10024-.
4904
4905  025506  004537  011456          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
4906  025512  000011                  9.
4907  025514  103003                  BCC     .+8.            ;BR IF NO ERROR
4908  025516                          ERROR                   ;REPORT STACKED ERROR
4909  025516  104460                                                    TRAP    C$ERROR
4910  025520                          ESCAPE  SUB             ;SKIP TO NEXT SUBTEST
4911  025520  104410                                                    TRAP    C$ESCAPE
4912  025522  000002                                                    .WORD   L10024-.
4913  025524                          ENDSUB
4914  025524                                                  L10024:
4915  025524  104403                                                    TRAP    C$ESUB
4916                        ;-------------------------------------------------------------
4917                        ; SUBTEST #2: FORCING EVEN VRC ERROR
4918                        ;-------------------------------------------------------------
4919  025526                          BGNSUB
4920  025526                                                  T2.2:
4921  025526  104402                                                    TRAP    C$BSUB
4922  025530  004737  005344          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
4923  025534  004537  007324          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
4924  025540  042626                  DDCMP!EVRC!226          ;SET DDCMP,EVEN VRC CHECK,SYNCH=226
4925  025542  000347                  TXDL!RXDL               ;TX/RX CHAR LENGTH=7 BITS
4926  025544  103003                  BCC     .+8.            ;BR IF NO ERROR
4927  025546                          ERROR                   ;REPORT STACKED ERROR
4928  025546  104460                                                    TRAP    C$ERROR
4929  025550                          ESCAPE  SUB             ;SKIP TO END OF TEST
4930  025550  104410                                                    TRAP    C$ESCAPE
4931  025552  000256                                                    .WORD   L10025-.
4932
4933  025554  004537  007734          JSR     R5,TXCTRL       ;SET TSOM
4934  025560  000001                  TSOM
```

```
4935  025562  000007                  7.
4936  025564  004537  007734          JSR     R5,TXCTRL       ;SET TSOM AGAIN (KNOCK DOWN TBMT)
4937  025570  000001                  TSOM
4938  025572  000010                  8.
4939  025574  004537  007734          JSR     R5,TXCTRL       ;CLEAR TSOM
4940  025600  000000                  000
4941  025602  000000                  0
4942  025604  004537  007622          JSR     R5,TXCHAR       ;LOAD 143, TX 3RD SYNCH
4943  025610  000143                  143
4944  025612  000010                  8.
4945  025614  103003                  BCC     .+8.            ;BR IF NO ERROR
4946  025616                          ERROR                   ;REPORT STACKED ERROR
4947  025616  104460                                                  TRAP    C$ERROR
4948  025620                          ESCAPE  SUB             ;SKIP TO END OF TEST
4949  025620  104410                                                  TRAP    C$ESCAPE
4950  025622  000206                                                  .WORD   L10025-.
4951
4952  025624  004537  003660          JSR     R5,WRITEI       ;SET RX CHAR LENGTH=6 BITS
4953  025630  120407                  PCR
4954  025632  000346                  TXDL!6                  ;TXCL=7, RXCL=6
4955  025634  103003                  BCC     .+8.            ;BR IF NO ERROR
4956  025636                          ERROR                   ;PRINT STACKED ERROR MESSAGE
4957  025636  104460                                                  TRAP    C$ERROR
4958  025640                          ESCAPE  SUB             ;AND EXIT SUBTEST
4959  025640  104410                                                  TRAP    C$ESCAPE
4960  025642  000166                                                  .WORD   L10025-.
4961
4962  025644  004537  007622          JSR     R5,TXCHAR       ;LOAD 026
4963  025650  000026                  026
4964  025652  000010                  8.
4965  025654  103003                  BCC     .+8.            ;BR IF NO ERROR
4966  025656                          ERROR                   ;REPORT STACKED ERROR
4967  025656  104460                                                  TRAP    C$ERROR
4968  025660                          ESCAPE  SUB             ;SKIP TO END OF TEST
4969  025660  104410                                                  TRAP    C$ESCAPE
4970  025662  000146                                                  .WORD   L10025-.
4971
4972  025664  004537  007622          JSR     R5,TXCHAR       ;LOAD FILLER (000)
4973  025670  000000                  000
4974  025672  000010                  8.
4975  025674  103003                  BCC     .+8.            ;BR IF NO ERROR
4976  025676                          ERROR                   ;REPORT STACKED ERROR
4977  025676  104460                                                  TRAP    C$ERROR
4978  025700                          ESCAPE  SUB             ;SKIP TO END OF TEST
4979  025700  104410                                                  TRAP    C$ESCAPE
4980  025702  000126                                                  .WORD   L10025-.
4981
4982  025704  004537  007622          JSR     R5,TXCHAR       ;LOAD FILLER (000)
4983  025710  000000                  000
4984  025712  000010                  8.
4985  025714  103003                  BCC     .+8.            ;BR IF NO ERROR
4986  025716                          ERROR                   ;REPORT STACKED ERROR
4987  025716  104460                                                  TRAP    C$ERROR
4988  025720                          ESCAPE  SUB             ;SKIP TO END OF TEST
4989  025720  104410                                                  TRAP    C$ESCAPE
4990  025722  000106                                                  .WORD   L10025-.
```

```
4991
4992  025724  004537  010034          JSR     R5,RXCHAR       ;READ/CHK SYNCH CHARACTER
4993  025730  000026                  026                     ;
4994  025732  000001                  RERCHK                  ;CHECK RERR (NO VRC ERROR EXPECTED)
4995  025734  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
4996  025736  103003                  BCC     .+8.            ;BR IF NO ERROR
4997  025740                          ERROR                   ;REPORT STACKED ERROR
4998  025740  104460                                                  TRAP    C$ERROR
4999  025742                          ESCAPE  SUB             ;SKIP TO END OF TEST
5000  025742  104410                                                  TRAP    C$ESCAPE
5001  025744  000064                                                  .WORD   L10025-.
5002
5003  025746  004537  010034          JSR     R5.RXCHAR       ;READ/CHK 6 BIT CHARACTER
5004  025752  000043                  043                     ;EXPECTED 1ST ''CHARACTER'' (043)
5005  025754  000001                  RERCHK                  ;CHECK RERR (NO VRC ERROR EXPECTED)
5006  025756  100000                  NOCRDA                  ;DON'T CHECK INITIAL RDA=0
5007  025760  103003                  BCC     .+8.            ;BR IF NO ERROR
5008  025762                          ERROR                   ;REPORT STACKED ERROR
5009  025762  104460                                                  TRAP    C$ERROR
5010  025764                          ESCAPE  SUB             ;SKIP TO END OF TEST
5011  025764  104410                                                  TRAP    C$ESCAPE
5012  025766  000042                                                  .WORD   L10025-.
5013
5014  025770  004537  010034          JSR     R5,RXCHAR       ;READ/CHK 6 BIT CHARACTER
5015  025774  100054                  RXERR!054               ;EXPECTED 2ND ''CHARACTER'' (054)
5016  025776  000001                  RERCHK                  ;CHECK RERR (VRC ERROR IS EXPECTED)
5017  026000  100000                  NOCRDA          ;DON'T CHECK INITIAL RDA=0
5018  026002  103003                  BCC     .+8.            ;BR IF NO ERROR
5019  026004                          ERROR                   ;REPORT STACKED ERROR
5020  026004  104460                                                  TRAP    C$ERROR
5021  026006                          ESCAPE  SUB             ;SKIP TO END OF TEST
5022  026006  104410                                                  TRAP    C$ESCAPE
5023  026010  000020                                                  .WORD   L10025-.
5024
5025  026012  004537  011456          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
5026  026016  000011                  9.
5027  026020  103003                  BCC     .+8.            ;BR IF NO ERROR
5028  026022                          ERROR                   ;REPORT STACKED ERROR
5029  026022  104460                                                  TRAP    C$ERROR
5030  026024                          ESCAPE  SUB             ;SKIP TO NEXT SUBTEST
5031  026024  104410                                                  TRAP    C$ESCAPE
5032  026026  000002                                                  .WORD   L10025-.
5033  026030                          ENDSUB
5034  026030                                                  L10025:
5035  026030  104403                                                  TRAP    C$ESUB
5036  026032                  ENDTST
5037  026032                                                  L10023:
5038  026032  104401                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 3 -- BCP CRC GENERATION/DETECTION TEST

                                ;+**********************************************************************
                                ;*
                                ;*      TEST 3 -- BCP CRC GENERATION/DETECTION TEST
                                ;*
                                ;*  THIS  TEST  IS COMPOSED OF 2 SUBTESTS -- #1 EXPECTS GOOD CRC
                                ;*  GENERATION  AND  REPORT  ERRORS  -- #2 FORCES AN ERROR AND ONLY
                                ;*  REPORT  WHEN THE CRC IS ACCEPTED AS GOOD. EACH IS
                                ;*  RUN  AT  THE CHARACTER LENGTHS OF 8 BITS FOR THE ENTIRITY
                                ;*  OF  EACH  MESSAGE.  BOTH THE TRANSMITTER AND RECEIVER WILL BE SET TO
                                ;*  THE  SAME  CHARACTER  LENGTH.  ERROR  LOOPING WILL BE ON THE FAILING
                                ;*  SUBTEST. TEXT STRINGS WILL BE LIMITED TO 5 CHARACTERS.
                                ;*
                                ;--**********************************************************************
                                ;
                                ;       BGNTST
026034                                                                          T3::
                                ;-----------------------------------------------------------------
                                ; SUBTEST #1 : GOOD CRC-16 GENERATION
                                ;-----------------------------------------------------------------
026034                                  BGNSUB
026034                                                                          T3.1:
026034  104402                                                                          TRAP    C$BSUB
026036  004737  005344                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
026042  004537  007324                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026046  065626                          DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
026050  000000                          0                       ;USE 8 BIT CHARS
026052  103003                          BCC     .+8.            ;BR IF NO ERROR
026054                                  ERROR                   ;REPORT STACKED ERROR
026054  104460                                                                          TRAP    C$ERROR
026056                                  ESCAPE  TST             ;SKIP TO END OF TEST
026056  104410                                                                          TRAP    C$ESCAPE
026060  000544                                                                          .WORD   L10026-.

026062  004537  007734                  JSR     R5,TXCTRL       ;SET TSOM, TX 1ST SYNCH
026066  000001                          TSOM
026070  000007                          7.
026072  004537  007734                  JSR     R5,TXCTRL       ;CLEAR TSOM
026076  000000                          000
026100  000000                          0
                                ;=================================================================
                                ; NOW TRANSMIT THE FIVE 8-BIT DATA CHARACTERS TO THE RECEIVER/FIFO
                                ;=================================================================
026102  012703  026626                  MOV     #T01TBL,R3      ;SET UP DATA TABLE POINTER
026106  112337  026116          10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER

026112  004537  007622                  JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
026116  000000                  1$:     000                     ;** HOLE FOR NEXT CHARACTER **
026120  000010                          8.
026122  103003                          BCC     .+8.            ;BR IF NO ERROR
026124                                  ERROR                   ;REPORT STACKED ERROR
026124  104460                                                                          TRAP    C$ERROR
026126                                  ESCAPE  TST             ;SKIP TO END OF TEST
026126  104410                                                                          TRAP    C$ESCAPE
026130  000474                                                                          .WORD   L10026-.
```

```
5095
5096  026132  022703  026633          CMP     #T01TBL+5,R3    ;ALL CHARACTERS TRANSMITTED ?
5097  026136  001363                  BNE     10$             ; IF NOT, TX ANOTHER ONE
5098                          ;=================================================================
5099  026140  004537  007734          JSR     R5,TXCTRL       ;LOAD 1ST TEOM
5100  026144  000002                  TEOM
5101  026146  000010                  8.
5102  026150  004537  007734          JSR     R5,TXCTRL       ;LOAD 2ND TEOM
5103  026154  000002                  TEOM
5104  026156  000010                  8.
5105  026160  004537  011540          JSR     R5,STEPLU
5106  026164  000016                  14.
5107
5108  026166  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 000, RCV 125
5109  026172  000000                  000
5110  026174  000000                  0
5111  026176  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
5112  026200  103003                  BCC     .+8.            ;BR IF NO ERROR
5113  026202                          ERROR                   ;REPORT STACKED ERROR
5114  026202  104460                                                          TRAP    C$ERROR
5115  026204                          ESCAPE  TST             ;SKIP TO END OF TEST
5116  026204  104410                                                          TRAP    C$ESCAPE
5117  026206  000416                                                          .WORD   L10026-.
5118
5119  026210  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 125, RCV 252
5120  026214  000125                  125
5121  026216  000000                  0
5122  026220  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
5123  026222  103003                  BCC     .+8.            ;BR IF NO ERROR
5124  026224                          ERROR                   ;REPORT STACKED ERROR
5125  026224  104460                                                          TRAP    C$ERROR
5126  026226                          ESCAPE  TST             ;SKIP TO END OF TEST
5127  026226  104410                                                          TRAP    C$ESCAPE
5128  026230  000374                                                          .WORD   L10026-.
5129
5130  026232  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 252, RCV 377
5131  026236  000252                  252
5132  026240  000000                  0
5133  026242  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
5134  026244  103003                  BCC     .+8.            ;BR IF NO ERROR
5135  026246                          ERROR                   ;REPORT STACKED ERROR
5136  026246  104460                                                          TRAP    C$ERROR
5137  026250                          ESCAPE  TST             ;SKIP TO END OF TEST
5138  026250  104410                                                          TRAP    C$ESCAPE
5139  026252  000352                                                          .WORD   L10026-.
5140
5141  026254  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 377, RCV 000
5142  026260  000377                  377
5143  026262  000000                  0
5144  026264  100010                  NOCRDA!8.               ;NO INITIAL CHECK OF RDA=0
5145  026266  103003                  BCC     .+8.            ;BR IF NO ERROR
5146  026270                          ERROR                   ;REPORT STACKED ERROR
5147  026270  104460                                                          TRAP    C$ERROR
5148  026272                          ESCAPE  TST             ;SKIP TO END OF TEST
5149  026272  104410                                                          TRAP    C$ESCAPE
5150  026274  000330                                                          .WORD   L10026-.
```

```
5151
5152  026276  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 000, CHECK CRC :
5153  026302  100000                  RXERR!000               ; RERR=1  (IF CRC-16 WAS OK).
5154  026304  000001                  RERCHK
5155  026306  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
5156  026310  103003                  BCC     .+8.            ;BR IF NO ERROR
5157  026312                          ERROR                   ;REPORT STACKED ERROR
5158  026312  104460                                                          TRAP    C$ERROR
5159  026314                          ESCAPE  TST             ;SKIP TO END OF TEST
5160  026314  104410                                                          TRAP    C$ESCAPE
5161  026316  000306                                                          .WORD   L10026-.
5162
5163  026320  004537  011456          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
5164  026324  000011                  9.
5165  026326  103003                  BCC     .+8.            ;BR IF NO ERROR
5166  026330                          ERROR                   ;REPORT STACKED ERROR
5167  026330  104460                                                          TRAP    C$ERROR
5168  026332                          ESCAPE  TST             ;SKIP TO END OF TEST
5169  026332  104410                                                          TRAP    C$ESCAPE
5170  026334  000270                                                          .WORD   L10026-.
5171  026336                  ENDSUB
5172  026336                                                          L10027:
5173  026336  104403                                                          TRAP    C$ESUB
5174                          ;-------------------------------------------------------------
5175                          ; SUBTEST #2 : BAD CRC-16 GENERATION
5176                          ;-------------------------------------------------------------
5177  026340                          BGNSUB
5178  026340                                                          T3.2:
5179  026340  104402                                                          TRAP    C$BSUB
5180  026342  004737  005344          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
5181  026346  004537  007324          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
5182  026352  065626                  DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
5183  026354  000000                  0                       ;USE 8 BIT CHARS
5184  026356  103003                  BCC     .+8.            ;BR IF NO ERROR
5185  026360                          ERROR                   ;REPORT STACKED ERROR
5186  026360  104460                                                          TRAP    C$ERROR
5187  026362                          ESCAPE  TST             ;SKIP TO END OF TEST
5188  026362  104410                                                          TRAP    C$ESCAPE
5189  026364  000240                                                          .WORD   L10026-.
5190
5191  026366  004537  007734          JSR     R5,TXCTRL       ;SET TSOM, TX 1ST SYNCH
5192  026372  000001                  TSOM
5193  026374  000007                  7.
5194  026376  004537  007734          JSR     R5,TXCTRL       ;CLEAR TSOM
5195  026402  000000                  000
5196  026404  000000                  0
5197                          ;=============================================================
5198                          ; NOW TRANSMIT THE FIVE 8-BIT DATA CHARACTERS PLUS THE ADDITIONAL
5199                          ; TWO BAD CRC (ALL 1'S) CHARACTERS TO THE RECEIVER/FIFO
5200                          ;=============================================================
5201  026406  012703  026626          MOV     #T01TBL,R3      ;SET UP DATA TABLE POINTER
5202  026412  112337  026422  10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
5203
5204  026416  004537  007622          JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
5205  026422  000000          1$:     000                     ;** HOLE FOR NEXT CHARACTER **
5206  026424  000010                  8.
```

```
026426 103003                 BCC     .+8.            ;BR IF NO ERROR
026430                        ERROR                   ;REPORT STACKED ERROR
026430 104460                                                           TRAP    C$ERROR
026432                        ESCAPE  TST             ;SKIP TO END OF TEST
026432 104410                                                           TRAP    C$ESCAPE
026434 000170                                                           .WORD   L10026-.

026436 022703 026635          CMP     #T01TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
026442 001363                 BNE     10$             ; IF NOT, TX ANOTHER ONE
                       ;==================================================================
026444 004537 011540          JSR     R5,STEPLU
026450 000010                 10

026452 004537 010034          JSR     R5,RXCHAR       ;READ & CHK 000, RCV 125
026456 000000                 000
026460 000000                 0
026462 100000                 NOCRDA                  ;NO INITIAL CHECK OF RDA=0
026464 103003                 BCC     .+8.            ;BR IF NO ERROR
026466                        ERROR                   ;REPORT STACKED ERROR
026466 104460                                                           TRAP    C$ERROR
026470                        ESCAPE  TST             ;SKIP TO END OF TEST
026470 104410                                                           TRAP    C$ESCAPE
026472 000132                                                           .WORD   L10026-.

026474 004537 010034          JSR     R5,RXCHAR       ;READ & CHK 125, RCV 252
026500 000125                 125
026502 000000                 0
026504 100000                 NOCRDA                  ;NO INITIAL CHECK OF RDA=0
026506 103003                 BCC     .+8.            ;BR IF NO ERROR
026510                        ERROR                   ;REPORT STACKED ERROR
026510 104460                                                           TRAP    C$ERROR
026512                        ESCAPE  TST             ;SKIP TO END OF TEST
026512 104410                                                           TRAP    C$ESCAPE
026514 000110                                                           .WORD   L10026-.

026516 004537 010034          JSR     R5,RXCHAR       ;READ & CHK 252, RCV 377
026522 000252                 252
026524 000000                 0
026526 100010                 NOCRDA!8.               ;NO INITIAL CHECK OF RDA=0
026530 103003                 BCC     .+8.            ;BR IF NO ERROR
026532                        ERROR                   ;REPORT STACKED ERROR
026532 104460                                                           TRAP    C$ERROR
026534                        ESCAPE  TST             ;SKIP TO END OF TEST
026534 104410                                                           TRAP    C$ESCAPE
026536 000066                                                           .WORD   L10026-.

026540 004537 010034          JSR     R5,RXCHAR       ;READ & CHK 377, RCV 000
026544 000377                 377
026546 000000                 0
026550 100010                 NOCRDA!8.               ;NO INITIAL CHECK OF RDA=0
026552 103003                 BCC     .+8.            ;BR IF NO ERROR
026554                        ERROR                   ;REPORT STACKED ERROR
026554 104460                                                           TRAP    C$ERROR
026556                        ESCAPE  TST             ;SKIP TO END OF TEST
026556 104410                                                           TRAP    C$ESCAPE
026560 000044                                                           .WORD   L10026-.
```

```
5263
5264  026562  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 000, CHECK CRC :
5265  026566  000000                  000                     ; RERR=0 IF BAD CRC-'S (EXPECTED).
5266  026570  000001                  RERCHK
5267  026572  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
5268  026574  103003                  BCC     .+8.            ;BR IF NO ERROR
5269  026576                          ERROR                   ;REPORT STACKED ERROR
5270  026576  104460                                                          TRAP    C$ERROR
5271  026600                          ESCAPE  TST             ;SKIP TO END OF TEST
5272  026600  104410                                                          TRAP    C$ESCAPE
5273  026602  000022                                                          .WORD   L10026-.
5274
5275  026604  004537  011456          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
5276  026610  000011                  9.
5277  026612  103003                  BCC     .+8.            ;BR IF NO ERROR
5278  026614                          ERROR                   ;REPORT STACKED ERROR
5279  026614  104460                                                          TRAP    C$ERROR
5280  026616                          ESCAPE  TST             ;SKIP TO END OF TEST
5281  026616  104410                                                          TRAP    C$ESCAPE
5282  026620  000004                                                          .WORD   L10026-.
5283  026622
5284  026622                        ENDSUB
5285  026622  104403                                                  L10030:
                                                                              TRAP    C$ESUB
5286  026624                      ENDTST
5287  026624
5288  026624  104401                                                  L10026:
                                                                              TRAP    C$ETST
5289
5290  026626      000     ;-------------------------------
5291  026627      125     T01TBL: .BYTE   000     ;D1
5292  026630      252             .BYTE   125     ;D2
5293  026631      377             .BYTE   252     ;D3
5294  026632      000             .BYTE   377     ;D4
5295  026633      377             .BYTE   000     ;D5
5296  026634      377             .BYTE   377     ;BAD CRC1
5297          026636              .BYTE   377     ;BAD CRC2
5298                              .EVEN
                                ;-------------------------------
```

```
CVDMDA.P11    10-DEC-80 09:15          TEST 4 -- BOP RX BASIC RECEIVE/FLAG RECOGNITION TEST

 5299                                  .SBTTL  TEST 4 -- BOP RX BASIC RECEIVE/FLAG RECOGNITION TEST
 5300
 5301                                  ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
 5302                                  ;*
 5303                                  ;*      TEST 4 -- BOP RX BASIC RECEIVE/FLAG RECOGNITION TEST
 5304                                  ;*
 5305                                  ;*  THE  USYRT IS INITIALIZED FOR BOP MODE WITH TTL LOOPBACK SELECTED.
 5306                                  ;*  'SECONDARY STATION ADDRESS'' IS NOT USED AND NO CRC/VRC IS CALCULATED.
 5307                                  ;*  A PATTERN  IS TRANSMITTED AND TERMINATED FOLLOWED BY A SECOND MESSAGE.
 5308                                  ;*  TERMINATION  OF  THE  FIRST  MESSAGE  IS  ACCOMPLISHED  WITH  A FLAG
 5309                                  ;*  CHARACTER  BUT  RXE IS NOT DROPPED SO THAT THE SECOND MESSAGE CAN BE
 5310                                  ;*  SENT WITHOUT RE-SYNCRONIZITION. SEVERAL FLAG'S ARE IDLED BETWEEN THE
 5311                                  ;*  TWO MESSAGES. DURING THE SECOND MESSAGE A RECEIVER OVERRUN CONDITION
 5312                                  ;*  IS FORCED. THROUGHOUT THIS TEST, BASIC RECEIVER OPERATION AND TIMING
 5313                                  ;*  IS CHECKED. TRANSMITTED INFORMATION IS VERIFIED BY CHECKING THE DATA
 5314                                  ;*  MADE AVAILABLE AT RXDB.
 5315                                  ;*
 5316                                  ;*     TRANSMITTED  PATTERN:  FLAG FLAG 123 321 000 377 101 FLAG... FLAG
 5317                                  ;*                                 321 123 377 000 276.
 5318                                  ;*
 5319                                  ;*     RECEIVED PATTERN: 123 321 000 377 101 ..... 321 123.
 5320                                  ;*
 5321                                  ;----------------------------------------------------------------------------------------------
 5322                                  ;
 5323                                          BGNTST
 5324  026636                                                                                  T4::
 5325  026636  004737  005344                 JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
 5326
 5327  026642  004537  007324                 JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
 5328  026646  003626                         NOCHK!SYNCH             ;SET BOP MODE,SYNCH REG=226
 5329  026650  000000                         0                       ;USE 8 BIT CHARS
 5330  026652  103003                         BCC     .+8.            ;BR IF NO ERROR
 5331  026654                                 ERROR                   ;REPORT STACKED ERROR
 5332  026654  104460                                                                 TRAP    C$ERROR
 5333  026656                                 ESCAPE  TST             ;SKIP TO END OF TEST
 5334  026656  104410                                                                 TRAP    C$ESCAPE
 5335  026660  000602                                                                 .WORD   L10031-.
 5336
 5337  026662  004537  007734                 JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
 5338  026666  000001                         TSOM
 5339  026670  000007                         7.
 5340  026672  004537  007734                 JSR     R5,TXCTRL       ;CLEAR TSOM
 5341  026676  000000                         000
 5342  026700  000000                         0
 5343
 5344  026702  004537  007622                 JSR     R5,TXCHAR       ;LOAD 123(DATA1), TX 2ND FLAG
 5345  026706  000123                         123
 5346  026710  000010                         8.
 5347  026712  103003                         BCC     .+8.            ;BR IF NO ERROR
 5348  026714                                 ERROR                   ;REPORT STACKED ERROR
 5349  026714  104460                                                                 TRAP    C$ERROR
 5350  026716                                 ESCAPE  TST             ;SKIP TO END OF TEST
 5351  026716  104410                                                                 TRAP    C$ESCAPE
 5352  026720  000542                                                                 .WORD   L10031-.
 5353
 5354  026722  004537  007622                 JSR     R5,TXCHAR       ;LOAD 321(DATA2), TX 123(DATA1)
```

```
5355  026726  000321                  321
5356  026730  000010                  8.
5357  026732  103003                  BCC     .+8.            ;BR IF NO ERROR
5358  026734                          ERROR                   ;REPORT STACKED ERROR
5359  026734  104460                                                          TRAP    C$ERROR
5360  026736                          ESCAPE  TST             ;SKIP TO END OF TEST
5361  026736  104410                                                          TRAP    C$ESCAPE
5362  026740  000522                                                          .WORD   L10031-.
5363
5364  026742  004537  007622          JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 321(DATA2)
5365  026746  000000                  000
5366  026750  000010                  8.
5367  026752  103003                  BCC     .+8.            ;BR IF NO ERROR
5368  026754                          ERROR                   ;REPORT STACKED ERROR
5369  026754  104460                                                          TRAP    C$ERROR
5370  026756                          ESCAPE  TST             ;SKIP TO END OF TEST
5371  026756  104410                                                          TRAP    C$ESCAPE
5372  026760  000502                                                          .WORD   L10031-.
5373
5374  026762  004537  007622          JSR     R5,TXCHAR       ;LOAD 377(DATA4)
5375  026766  000377                  377
5376  026770  000000                  0
5377  026772  103003                  BCC     .+8.            ;BR IF NO ERROR
5378  026774                          ERROR                   ;REPORT STACKED ERROR
5379  026774  104460                                                          TRAP    C$ERROR
5380  026776                          ESCAPE  TST             ;SKIP TO END OF TEST
5381  026776  104410                                                          TRAP    C$ESCAPE
5382  027000  000462                                                          .WORD   L10031-.
5383
5384  027002  004537  011310          JSR     R5,RCV1ST       ;CLOCK AND RCV 123(DATA1)
5385  027006  000000                  0
5386  027010  103003                  BCC     .+8.            ;BR IF NO ERROR
5387  027012                          ERROR                   ;REPORT STACKED ERROR
5388  027012  104460                                                          TRAP    C$ERROR
5389  027014                          ESCAPE  TST             ;SKIP TO END OF TEST
5390  027014  104410                                                          TRAP    C$ESCAPE
5391  027016  000444                                                          .WORD   L10031-.
5392
5393  027020  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 123(DATA1), RCV 321(DATA2)
5394  027024  000523                  RXSOM!123               ; & CHECK RSOM=1
5395  027026  000000                  0
5396  027030  000010                  8.
5397  027032  103003                  BCC     .+8.            ;BR IF NO ERROR
5398  027034                          ERROR                   ;REPORT STACKED ERROR
5399  027034  104460                                                          TRAP    C$ERROR
5400  027036                          ESCAPE  TST             ;SKIP TO END OF TEST
5401  027036  104410                                                          TRAP    C$ESCAPE
5402  027040  000422                                                          .WORD   L10031-.
5403
5404  027042  004537  007622          JSR     R5,TXCHAR       ;LOAD 101(DATA5)
5405  027046  000101                  101
5406  027050  000000                  0
5407  027052  103003                  BCC     .+8.            ;BR IF NO ERROR
5408  027054                          ERROR                   ;REPORT STACKED ERROR
5409  027054  104460                                                          TRAP    C$ERROR
5410  027056                          ESCAPE  TST             ;SKIP TO END OF TEST
```

```
5411  027056  104410                                                          TRAP    C$ESCAPE
5412  027060  000402                                                          .WORD   L10031-.
5413
5414  027062  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 321(DATA2),RCV 000(DATA3)
5415  027066  000321                321
5416  027070  000000                0
5417  027072  000010                8.
5418  027074  103003                BCC     .+8.            ;BR IF NO ERROR
5419  027076                        ERROR                   ;REPORT STACKED ERROR
5420  027076  104460                                                          TRAP    C$ERROR
5421  027100                        ESCAPE  TST             ;SKIP TO END OF TEST
5422  027100  104410                                                          TRAP    C$ESCAPE
5423  027102  000360                                                          .WORD   L10031-.
5424
5425  027104  004537  007734        JSR     R5,TXCTRL       ;LOAD TEOM
5426  027110  000002                TEOM
5427  027112  000000                0
5428
5429  027114  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 000(DATA3),RCV 377(DATA4)
5430  027120  000000                000
5431  027122  000000                0
5432  027124  000010                8.
5433  027126  103003                BCC     .+8.            ;BR IF NO ERROR
5434  027130                        ERROR                   ;REPORT STACKED ERROR
5435  027130  104460                                                          TRAP    C$ERROR
5436  027132                        ESCAPE  TST             ;SKIP TO END OF TEST
5437  027132  104410                                                          TRAP    C$ESCAPE
5438  027134  000326                                                          .WORD   L10031-.
5439
5440  027136  004537  007734        JSR     R5,TXCTRL       ;LOAD TEOM
5441  027142  000002                TEOM
5442  027144  000000                0
5443
5444  027146  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 377(DATA4),RCV 101(DATA5)
5445  027152  000377                377                     ; AND TX (FLAG1)
5446  027154  000000                0
5447  027156  020010                NCRACT!8.               ;DON'T CHECK RECEIVER ACTIVE
5448  027160  103003                BCC     .+8.            ;BR IF NO ERROR
5449  027162                        ERROR                   ;REPORT STACKED ERROR
5450  027162  104460                                                          TRAP    C$ERROR
5451  027164                        ESCAPE  TST             ;SKIP TO END OF TEST
5452  027164  104410                                                          TRAP    C$ESCAPE
5453  027166  000274                                                          .WORD   L10031-.
5454
5455  027170  004537  007734        JSR     R5,TXCTRL       ;LOAD TEOM
5456  027174  000002                TEOM
5457  027176  000000                0
5458
5459  027200  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 101(DATA5),RCV (FLAG1)
5460  027204  001101                RXEOM!101               ; TX (FLAG2) & CHECK REOM
5461  027206  000000                0
5462  027210  060010                NFCRDA!NCRACT!8.        ;DON'T CHECK FOR FINAL RDA=RXACT=1
5463  027212  103003                BCC     .+8.            ;BR IF NO ERROR
5464  027214                        ERROR                   ;REPORT STACKED ERROR
5465  027214  104460                                                          TRAP    C$ERROR
5466  027216                        ESCAPE  TST             ;SKIP TO END OF TEST
```

```
5467  027216  104410                                                          TRAP    C$ESCAPE
5468  027220  000242                                                          .WORD   L10031-.
5469
5470  027222  004537  007734        JSR     R5,TXCTRL       ;CLEAR TEOM
5471  027226  000000                000
5472  027230  000000                0
5473
5474  027232  004537  007622        JSR     R5,TXCHAR       ;LOAD 321(DATA6),TX (FLAG3)
5475  027236  000321                321
5476  027240  100010                NCTBMT*256.!8.          ;DON'T CHECK TBMT
5477  027242  103003                BCC     .+8.            ;BR IF NO ERROR
5478  027244                        ERROR                   ;REPORT STACKED ERROR
5479  027244  104460                                                          TRAP    C$ERROR
5480  027246                        ESCAPE  TST             ;SKIP TO END OF TEST
5481  027246  104410                                                          TRAP    C$ESCAPE
5482  027250  000212                                                          .WORD   L10031-.
5483
5484  027252  004537  007622        JSR     R5,TXCHAR       ;LOAD 123(DATA7),TX(DATA6)
5485  027256  000123                123
5486  027260  100010                NCTBMT*256.!8.          ;DON'T CHECK TBMT
5487  027262  103003                BCC     .+8.            ;BR IF NO ERROR
5488  027264                        ERROR                   ;REPORT STACKED ERROR
5489  027264  104460                                                          TRAP    C$ERROR
5490  027266                        ESCAPE  TST             ;SKIP TO END OF TEST
5491  027266  104410                                                          TRAP    C$ESCAPE
5492  027270  000172                                                          .WORD   L10031-.
5493
5494  027272  004537  007622        JSR     R5,TXCHAR       ;LOAD 377(DATA8),TX(DATA7)
5495  027276  000377                377
5496  027300  100010                NCTBMT*256.!8.          ;DON'T CHECK FINAL TBMT
5497  027302  103003                BCC     .+8.            ;BR IF NO ERROR
5498  027304                        ERROR                   ;REPORT STACKED ERROR
5499  027304  104460                                                          TRAP    C$ERROR
5500  027306                        ESCAPE  TST             ;SKIP TO END OF TEST
5501  027306  104410                                                          TRAP    C$ESCAPE
5502  027310  000152                                                          .WORD   L10031-.
5503
5504  027312  004537  007622        JSR     R5,TXCHAR       ;LOAD 000(DATA9)
5505  027316  000000                000
5506  027320  000000                0
5507  027322  103003                BCC     .+8.            ;BR IF NO ERROR
5508  027324                        ERROR                   ;REPORT STACKED ERROR
5509  027324  104460                                                          TRAP    C$ERROR
5510  027326                        ESCAPE  TST             ;SKIP TO END OF TEST
5511  027326  104410                                                          TRAP    C$ESCAPE
5512  027330  000132                                                          .WORD   L10031-.
5513
5514  027332  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 321(DATA6),RCV 123(DATA7)
5515  027336  000721                RXSOM!321
5516  027340  000000                0
5517  027342  000010                8.
5518  027344  103003                BCC     .+8.            ;BR IF NO ERROR
5519  027346                        ERROR                   ;REPORT STACKED ERROR
5520  027346  104460                                                          TRAP    C$ERROR
5521  027350                        ESCAPE  TST             ;SKIP TO END OF TEST
5522  027350  104410                                                          TRAP    C$ESCAPE
```

```
5523  027352  000110                                                          .WORD   L10031-.
5524
5525  027354  004537  007622          JSR     R5,TXCHAR       ;LOAD 276(DATA10)
5526  027360  000276                  276
5527  027362  000000                  0
5528  027364  103003                  BCC     .+8.            ;BR IF NO ERROR
5529  027366                          ERROR                   ;REPORT STACKED ERROR
5530  027366  104460                                                          TRAP    C$ERROR
5531  027370                          ESCAPE  TST             ;SKIP TO END OF TEST
5532  027370  104410                                                          TRAP    C$ESCAPE
5533  027372  000070                                                          .WORD   L10031-.
5534
5535  027374  004537  010034          JSR     R5.RXCHAR       ;READ/CHECK 123(DATA7),RCV 377(DATA8)
5536  027400  000123                  123
5537  027402  000000                  0
5538  027404  100014                  NOCRDA!12.              ;NO CHECK OF INITIAL RDA=0
5539  027406  103003                  BCC     .+8.            ;BR IF NO ERROR
5540  027410                          ERROR                   ;REPORT STACKED ERROR
5541  027410  104460                                                          TRAP    C$ERROR
5542  027412                          ESCAPE  TST             ;SKIP TO END OF TEST
5543  027412  104410                                                          TRAP    C$ESCAPE
5544  027414  000046                                                          .WORD   L10031-.
5545                          ;-----------------------------------------------------
5546  027416  012704  000010          MOV     #8.,R4          ;INIT CHARACTER COUNT
5547  027422  004537  007622  5$:     JSR     R5,TXCHAR       ;LOAD/TX FILLER (OVERFLOW FIFO)
5548  027426  000000                  000
5549  027430  100010                  NCTBMT*256.!8.          ;DON'T CHECK FINAL TBMT
5550  027432  103003                  BCC     .+8.            ;BR IF NO ERROR
5551  027434                          ERROR                   ;REPORT STACKED ERROR
5552  027434  104460                                                          TRAP    C$ERROR
5553  027436                          ESCAPE  TST             ;SKIP TO END OF TEST
5554  027436  104410                                                          TRAP    C$ESCAPE
5555  027440  000022                                                          .WORD   L10031-.
5556
5557  027442  077411                  SOB     R4,5$           ;FILL TO OVERFLOW
5558                          ;-----------------------------------------------------
5559
5560  027444  004537  006422          JSR     R5,CKROR        ;CHECK RECEIVER OVERRUN BIT
5561  027450  000001                  1                       ;(IT SHOULD BE SET)
5562  027452  103003                  BCC     .+8.            ;BR IF NO ERROR
5563  027454                          ERROR                   ;REPORT STACKED ERROR
5564  027454  104460                                                          TRAP    C$ERROR
5565  027456                          ESCAPE  TST             ;SKIP TO END OF TEST
5566  027456  104410                                                          TRAP    C$ESCAPE
5567  027460  000002                                                          .WORD   L10031-.
5568  027462                  ENDTST
5569  027462                                                          L10031:
5570  027462  104401                                                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST 5 -- BOP RX SECONDARY STATION ADDRESSING

                                        ;+*********************************************************************
                                        ;*
                                        ;*      TEST 5 -- BOP RX SECONDARY STATION ADDRESSING
                                        ;*
                                        ;*  THE  USYRT IS INITIALIZED FOR BOP MODE WITH TTL LEVEL LOOPBACK,
                                        ;*  SAM = 1, APA=0, AND ECM = 7. USING SHORT MESSAGES, THE ADDRESSES
                                        ;*  000, 125, 252, 176, AND 177 ARE CHECKED TO SEE THAT THE RECEIVER
                                        ;*  RECOGNIZES THEM CORRECTLY. IN EACH CASE (AT EACH ADDRESS), A SERIES OF
                                        ;*  20  DIFFERENT  MESSAGES ARE SENT TO VERIFY THAT THE USYRT WILL ONLY
                                        ;*  RESPOND TO THE SPECIFIED VALUE.
                                        ;*
                                        ;*  TEST PATTERN: ADR 000 OCR ADR
                                        ;*  WHERE ADR IS THE ADDRESS BEING TESTED AND OCA IS THE ONE'S
                                        ;*  COMPLEMENT OF THAT ADDRESS.
                                        ;*
                                        ;*
                                        ;-*********************************************************************
                                        ;
                                        ;       BGNTST
 027464                                                                                 T5::
 027464  004767  005344                         JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
 027470  005004                                 CLR     R4              ;CLEAR TEST ADDR INDEX (0,125,252,176,177)

 027472                                 OLOOP:  BGNSUB
 027472                                                                                 T5.1:
 027472  104402                                                                                 TRAP    C$BSUB
 027474  005002                                 CLR     R2              ;CLEAR TX ADDRESS INDEX (0 => 20.)
 027476  116437  030170  027512                 MOVB    ADPAT(R4),NWSAR ;INSTALL NEW S/AR VALUE IN "INNER LOOP"
                                        ;**********************************************************************
                                        ; INNER LOOP: TEST ONE 'TEST ADDRESS" (0,125,252,176, OR 177)
                                        ;**********************************************************************
 027504                                         BGNSEG
 027504  104404                                                                                 TRAP    C$BSEG
 027506  004537  007324                 ILOOP:  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
 027512  013400                         NWSAR:  SECADR!NOCHK!000        ;SET BOP MODE,SAM=1,###S/AR IS VARIABLE###
 027514  000000                                 0                       ;USE 8 BIT CHARS
 027516  103003                                 BCC     .+8.            ;BR IF NO ERROR
 027520                                         ERROR                   ;REPORT STACKED ERROR
 027520  104460                                                                                 TRAP    C$ERROR
 027522                                         ESCAPE  SEG             ;SKIP TO END OF TEST
 027522  104410                                                                                 TRAP    C$ESCAPE
 027524  000422                                                                                 .WORD   10000$-.
                                        ;---------------------------------------------------------------------
                                        ;  SETUP TX/RX STRINGS (ADR 000 OCA ADR) -- ADR TAKEN FROM PATX1
                                        ;---------------------------------------------------------------------
 027526  116203  002651                         MOVB    PATX1(R2),R3    ;ADR => R3
 027532  110337  027614                         MOVB    R3,1$           ;ADDRESS(ADR) => 1$,3$,4$,6$
 027536  110337  027674                         MOVB    R3,3$
 027542  110337  027754                         MOVB    R3,4$
 027546  110337  030072                         MOVB    R3,6$
 027552  110337  027654                         MOVB    R3,2$           ;ADDRESS_NOT(OCA) => 2$,5$
 027556  105137  027654                         COMB    2$
 027562  113737  027654  030040                 MOVB    2$,5$
```

```
5627
5628                            ;-----------------------------------------------------------------
5629                            ;  NOW TRANSMIT TEST PATTERN
5630                            ;-----------------------------------------------------------------
5630  027570  004537  007734          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
5631  027574  000001                  TSOM
5632  027576  000007                  7.
5633  027600  004537  007734          JSR     R5,TXCTRL       ;CLEAR TSOM
5634  027604  000000                  000
5635  027606  000000                  0
5636  027610  004537  007622          JSR     R5,TXCHAR       ;LOAD ADDRESS, TX 2ND FLAG
5637  027614  000000          1$:     000                     ;** HOLE FOR SECONDARY STATION ADDRESS
5638  027616  000010                  8.
5639  027620  103003                  BCC     .+8.            ;BR IF NO ERROR
5640  027622                          ERROR                   ;REPORT STACKED ERROR
5641  027622  104460                                                          TRAP    C$ERROR
5642  027624                          ESCAPE  SEG             ;SKIP TO END OF TEST
5643  027624  104410                                                          TRAP    C$ESCAPE
5644  027626  000320                                                          .WORD   10000$-.
5645  027630  004537  007622          JSR     R5,TXCHAR       ;LOAD 000, TX ADDRESS
5646  027634  000000                  000
5647  027636  100011                  NCTBMT*256.!9.          ;DON'T CHECK TBMT
5648  027640  103003                  BCC     .+8.            ;BR IF NO ERROR
5649  027642                          ERROR                   ;REPORT STACKED ERROR
5650  027642  104460                                                          TRAP    C$ERROR
5651  027644                          ESCAPE  SEG             ;SKIP TO END OF TEST
5652  027644  104410                                                          TRAP    C$ESCAPE
5653  027646  000300                                                          .WORD   10000$-.
5654  027650  004537  007622          JSR     R5,TXCHAR       ;LOAD ADDR_NOT, TX 000
5655  027654  000000          2$:     000                     ;** HOLE FOR COMPLEMENTED ADDRESS
5656  027656  100010                  NCTBMT*256.!8.          ;DON'T CHECK TBMT
5657  027660  103003                  BCC     .+8.            ;BR IF NO ERROR
5658  027662                          ERROR                   ;REPORT STACKED ERROR
5659  027662  104460                                                          TRAP    C$ERROR
5660  027664                          ESCAPE  SEG             ;SKIP TO END OF TEST
5661  027664  104410                                                          TRAP    C$ESCAPE
5662  027666  000260                                                          .WORD   10000$-.
5663  027670  004537  007622          JSR     R5,TXCHAR       ;LOAD ADDRESS AGAIN
5664  027674  000000          3$:     000                     ;** HOLE FOR ADDRESS (AGAIN)
5665  027676  000000                  0
5666  027700  103003                  BCC     .+8.            ;BR IF NO ERROR
5667  027702                          ERROR                   ;REPORT STACKED ERROR
5668  027702  104460                                                          TRAP    C$ERROR
5669  027704                          ESCAPE  SEG             ;SKIP TO END OF TEST
5670  027704  104410                                                          TRAP    C$ESCAPE
5671  027706  000240                                                          .WORD   10000$-.
5672  027710  004537  011540          JSR     R5,STEPLU       ;CLOCK/RCV ADDRESS FIELD
5673  027714  000003                  3
5674
5675  027716  004537  006122          JSR     R5,CKRDA        ;DID USYRT RESPOND TO ADDRESS ??
5676  027722  000001                  1
5677  027724  103471                  BCS     10$             ;BR IF RDA=0
5678                            ;-----------------------------------------------------------------
5679                            ;  USYRT RESPONDED TO MESSAGE (RDA=1): SHOULD IT HAVE?
5680                            ;-----------------------------------------------------------------
5681  027726  123703  027512          CMPB    NWSAR,R3        ;IS ADDRESS = S/AR ?
5682  027732  001406                  BEQ     40$             ;BR IF YES
```

```
                                  ;------------------------------------------------------------
                                  ; ...NO, REPORT ERROR : 'USYRT RESPONDED TO WRONG ADDRESS''
                                  ;------------------------------------------------------------
027734                                    GEDF    EM102,ERR21
                                                                  ;       'DEVICE FATAL'' ERROR # 43
027734  104455                                                                                   TRAP    C$ERDF
027736  000053                                                                                   .WORD   43
027740  016466                                                                                   .WORD   EM102
027742  022204                                                                                   .WORD   ERR21
027744                                    ESCAPE  SEG
027744  104410                                                                                   TRAP    C$ESCAPE
027746  000200                                                                                   .WORD   10000$-.
                                  ;------------------------------------------------------------
                                  ; ...YES, READ AND VERIFY RECEIVED MESSAGE
                                  ;------------------------------------------------------------
027750  004537  010034            40$:    JSR     R5,RXCHAR       ;READ & CHK ADDRESS, RCV 000
027754  000400                    4$:     RXSOM!000               ; & CHECK RSOM=1
027756  000000                            0
027760  100010                            NOCRDA!8.               ;NO INITIAL CHECK OF RDA=0
027762  103003                            BCC     .+8.            ;BR IF NO ERROR
027764                                    ERROR                   ;REPORT STACKED ERROR
027764  104460                                                                                   TRAP    C$ERROR
027766                                    ESCAPE  SEG             ;SKIP TO END OF TEST
027766  104410                                                                                   TRAP    C$ESCAPE
027770  000156                                                                                   .WORD   10000$-.
027772  004537  007734                    JSR     R5,TXCTRL       ;SET TEOM
027776  000002                            TEOM
030000  000000                            0
030002  004537  010034                    JSR     R5,RXCHAR       ;READ/CHECK 000
030006  000000                            000
030010  000000                            0
030012  100010                            NOCRDA!8.               ;NO INITIAL CHECK OF RDA=0
030014  103003                            BCC     .+8.            ;BR IF NO ERROR
030016                                    ERROR                   ;REPORT STACKED ERROR
030016  104460                                                                                   TRAP    C$ERROR
030020                                    ESCAPE  SEG             ;SKIP TO END OF TEST
030020  104410                                                                                   TRAP    C$ESCAPE
030022  000124                                                                                   .WORD   10000$-.
030024  004537  007734                    JSR     R5,TXCTRL       ;SET TEOM
030030  000002                            TEOM
030032  000000                            0
030034  004537  010034                    JSR     R5,RXCHAR       ;READ/CHECK COMPLEMENTED ADDRESS
030040  000000                    5$:     000                     ;** HOLE FOR ADDRESS_NOT
030042  000000                            0                       ;NO INITIAL CHECK OF RDA=0
030044  120010                            NOCRDA!NCRACT!8.        ;DON'T CHECK FINAL RXACT=1
030046  103003                            BCC     .+8.            ;BR IF NO ERROR
030050                                    ERROR                   ;REPORT STACKED ERROR
030050  104460                                                                                   TRAP    C$ERROR
030052                                    ESCAPE  SEG             ;SKIP TO END OF TEST
030052  104410                                                                                   TRAP    C$ESCAPE
030054  000072                                                                                   .WORD   10000$-.
030056  004537  007734                    JSR     R5,TXCTRL       ;SET TSOM
030062  000001                            TSOM
030064  000000                            0
030066  004537  010034                    JSR     R5,RXCHAR       ;READ/CHECK ADDRESS (AGAIN)
030072  001000                    6$:     RXEOM!000               ;** HOLE FOR FINAL ADDRESS
```

```
5739  030074  000000                    0
5740  030076  060000                    NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
5741  030100  103014                    BCC     50$             ;BR IF NO ERROR (TO CONTINUE TEST)
5742  030102                            ERROR                   ;REPORT STACKED ERROR
5743  030102  104460                                                            TRAP    C$ERROR
5744  030104                            ESCAPE  SEG             ;SKIP TO END OF TEST
5745  030104  104410                                                            TRAP    C$ESCAPE
5746  030106  000040                                                            .WORD   10000$-.
5747                                    ;-------------------------------------------------------
5748                                    ; USYRT DIDN'T RESPOND TO MESSAGE (RDA=0): SHOULD IT HAVE ?
5749                                    ;-------------------------------------------------------
5750  030110  123703  027512    10$:    CMPB    NWSAR,R3        ;WAS NON-RESPONDING ADDR=S/AR ?
5751  030114  001006                    BNE     50$             ;BR IF NO
5752                                    ;-------------------------------------------------------
5753                                    ; ...NO, REPORT ERROR : 'USYRT DIDN'T RESPOND TO SECONDARY STATION ADDR''
5754                                    ;-------------------------------------------------------
5755  030116                            GEDF    EM103,ERR20
5756                                                    ;       'DEVICE FATAL'' ERROR # 44
5757  030116  104455                                                            TRAP    C$ERDF
5758  030120  000054                                                            .WORD   44
5759  030122  016530                                                            .WORD   EM103
5760  030124  022146                                                            .WORD   ERR20
5761  030126                            ESCAPE  SEG
5762  030126  104410                                                            TRAP    C$ESCAPE
5763  030130  000016                                                            .WORD   10000$-.
5764                                    ;-------------------------------------------------------
5765                                    ;...YES, UPDATE ADDRESS AND CONTINUE TESTING
5766                                    ;-------------------------------------------------------
5767  030132  005202              50$:    INC     R2              ;INCREMENT TESTING ADDRESS INDEX
5768  030134  022702  000025            CMP     #21.,R2
5769  030140  001402                    BEQ     .+6
5770  030142  000137  027506            JMP     ILOOP           ;IF INDEX .LE. 20 THEN CHECK IT
5771  030146                            ENDSEG                  ;OTHERWISE END INNER LOOP...
5772  030146                                                                  10000$:
5773  030146  104405                                                            TRAP    C$ESEG
5774                                  ;*************************************************************************
5775  030150  005204                    INC     R4              ;INCREMENT ACTUAL TEST ADDRESS INDEX
5776  030152  020427  000005            CMP     R4,#5           ;ALL 5 TEST ADDRESSES CHECKED?
5777  030156  001402                    BEQ     .+6             ; BR IF DONE
5778  030160  000137  027472            JMP     OLOOP           ; NOT DONE: DO NEXT ADDRESS
5779  030164                          ENDSUB
5780  030164                                                                  L10033:
5781  030164  104403                                                            TRAP    C$ESUB
5782  030166                          ENDTST
5783  030166                                                                  L10032:
5784  030166  104401                                                            TRAP    C$ETST
5785                                  ;-------------------------------------------------------
5786  030170     000          ADPAT:  .BYTE   000
5787  030171     125                  .BYTE   125
5788  030172     252                  .BYTE   252
5789  030173     176                  .BYTE   176
5790  030174     177                  .BYTE   177
5791        030176                    .EVEN
5792                                  ;-------------------------------------------------------
```

```
                                        .SBTTL  TEST 6 -- BOP RX ALL PARTIES ADDRESS TEST

                                        ;*****************************************************************
                                        ;*
                                        ;*      TEST 6 -- BOP RX ALL PARTIES ADDRESS TEST
                                        ;*
                                        ;*   INITIALIZE  THE  USYRT  FOR BOP MODE WITH TTL LEVEL LOOPBACK
                                        ;*   SAM = 1, S/AR = 123(OCT.), APA = 1, AND ECM = 7.
                                        ;*   A SERIES OF 256 DIFFERENT SHORT MESSAGES ARE SENT TO VERIFY THAT
                                        ;*   THE USYRT WILL ONLY RESPOND TO THE SPECIFIED VALUE AND ALSO 377 (FF
                                        ;*   HEX.).
                                        ;*
                                        ;*   TEST PATTERN: ADR 000 OCA ADR
                                        ;*   WHERE  ADR  IS  THE  ADDRESS  BEING  TESTED  AND  OCA  IS THE ONE'S
                                        ;*   COMPLEMENT OF THAT ADDRESS.
                                        ;*
                                        ;-****************************************************************
                                        ;
                                        ;       BGNTST
030176                                                                                  T6::
030176  004737  005344                          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
030202  005003                                  CLR     R3              ;CLEAR ADDRESS

030204                                  LOOP:   BGNSUB
030204                                                                                  T6.1:
030204  104402                                                                            TRAP    C$BSUB
030206  004537  007324                          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030212  113523                                  APAD!SECADR!NOCHK!123   ;SET BOP MODE,APA=1,SAM=1,ECM=7,S/AR=123
030214  000000                                  0                       ;USE 8 BIT CHARS
030216  103003                                  BCC     .+8.            ;BR IF NO ERROR
030220                                          ERROR                   ;REPORT STACKED ERROR
030220  104460                                                                            TRAP    C$ERROR
030222                                          ESCAPE  SUB             ;SKIP TO END OF TEST
030222  104410                                                                            TRAP    C$ESCAPE
030224  000444                                                                            .WORD   L10035-.
                                        ;---------------------------------------------------------
                                        ;  SETUP TX/RX STRINGS (ADR 000 OCA ADR)
                                        ;---------------------------------------------------------
030226  110337  030310                          MOVB    R3,1$           ;ADDRESS(ADR) => 1$,3$,4$,6$
030232  110337  030370                          MOVB    R3,3$
030236  110337  030456                          MOVB    R3,4$
030242  110337  030574                          MOVB    R3,6$
030246  110337  030350                          MOVB    R3,2$           ;ADDRESS_NOT(OCA) => 2$,5$
030252  105137  030350                          COMB    2$
030256  113737  030350  030542                  MOVB    2$,5$
                                        ;---------------------------------------------------------
                                        ;  NOW TRANSMIT TEST PATTERN
                                        ;---------------------------------------------------------
030264  004537  007734                          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
030270  000001                                  TSOM
030272  000007                                  7.
030274  004537  007734                          JSR     R5,TXCTRL       ;CLEAR TSOM
030300  000000                                  000
030302  000000                                  0
030304  004537  007622                          JSR     R5,TXCHAR       ;LOAD ADDRESS, TX 2ND FLAG
030310  000000                          1$:     000                     ;** HOLE FOR SECONDARY STATION ADDRESS
```

```
5849  030312  000010                    8.
5850  030314  103003                    BCC     .+8.            ;BR IF NO ERROR
5851  030316                            ERROR                   ;REPORT STACKED ERROR
5852  030316  104460                                                    TRAP    C$ERROR
5853  030320                            ESCAPE  SUB             ;SKIP TO END OF TEST
5854  030320  104410                                                    TRAP    C$ESCAPE
5855  030322  000346                                                    .WORD   L10035-.
5856  030324  004537  007622            JSR     R5,TXCHAR       ;LOAD 000, TX ADDRESS
5857  030330  000000                    000
5858  030332  100011                    NCTBMT*256.!9.
5859  030334  103003                    BCC     .+8.            ;BR IF NO ERROR
5860  030336                            ERROR                   ;REPORT STACKED ERROR
5861  030336  104460                                                    TRAP    C$ERROR
5862  030340                            ESCAPE  SUB             ;SKIP TO END OF TEST
5863  030340  104410                                                    TRAP    C$ESCAPE
5864  030342  000326                                                    .WORD   L10035-.
5865  030344  004537  007622            JSR     R5,TXCHAR       ;LOAD ADDR NOT, TX 000
5866  030350  000000            2$.     000                     ;** HOLE FOR COMPLEMENTED ADDRESS
5867  030352  100010                    NCTBMT*256.!8.
5868  030354  103003                    BCC     .+8.            ;BR IF NO ERROR
5869  030356                            ERROR                   ;REPORT STACKED ERROR
5870  030356  104460                                                    TRAP    C$ERROR
5871  030360                            ESCAPE  SUB             ;SKIP TO END OF TEST
5872  030360  104410                                                    TRAP    C$ESCAPE
5873  030362  000306                                                    .WORD   L10035-.
5874  030364  004537  007622            JSR     R5,TXCHAR       ;LOAD ADDRESS AGAIN
5875  030370  000000            3$:     000                     ;** HOLE FOR ADDRESS (AGAIN)
5876  030372  000000                    0
5877  030374  103003                    BCC     .+8.            ;BR IF NO ERROR
5878  030376                            ERROR                   ;REPORT STACKED ERROR
5879  030376  104460                                                    TRAP    C$ERROR
5880  030400                            ESCAPE  SUB             ;SKIP TO END OF TEST
5881  030400  104410                                                    TRAP    C$ESCAPE
5882  030402  000266                                                    .WORD   L10035-.
5883  030404  004537  011540            JSR     R5,STEPLU       ;CLOCK/RCV ADDRESS FIELD
5884  030410  000002                    2
5885
5886  030412  004537  006122            JSR     R5,CKRDA        ;DID USYRT RESPOND TO ADDRESS ??
5887  030416  000001                    1
5888  030420  103475                    BCS     10$             ;BR IF RDA=0
5889                            ;-----------------------------------------------------------
5890                            ;  USYRT RESPONDED TO MESSAGE (RDA=1): SHOULD IT HAVE?
5891                            ;-----------------------------------------------------------
5892  030422  022703  000123            CMP     #123,R3         ;ADDRESS = 123 ?
5893  030426  001411                    BEQ     40$             ;BR IF YES
5894  030430  022703  000377            CMP     #377,R3         ;ADDRESS = 377 ?
5895  030434  001406                    BEQ     40$
5896                            ;-----------------------------------------------------------
5897                            ; ...NO, REPORT ERROR : 'USYRT RESPONDED TO WRONG ADDRESS''
5898                            ;-----------------------------------------------------------
5899  030436                            GEDF    EM102,ERR21
5900                                                    ;       'DEVICE FATAL'' ERROR # 45
5901  030436  104455                                                    TRAP    C$ERDF
5902  030440  000055                                                    .WORD   45
5903  030442  016466                                                    .WORD   EM102
5904  030444  022204                                                    .WORD   ERR21
```

```
5905  030446
5906  030446  104410                                                    TRAP    C$ESCAPE
5907  030450  000220                                                    .WORD   L10035-.
5908
5909
5910
5911  030452  004537  010034    40$:    JSR     R5,RXCHAR       ;READ & CHK ADDRESS, RCV 000
5912  030456  000400    4$:     RXSOM!000                       ; & CHECK RSOM=1
5913  030460  000000            0
5914  030462  100010            NOCRDA!8.                       ;NO INITIAL CHECK OF RDA=0
5915  030464  103003            BCC     .+8.                    ;BR IF NO ERROR
5916  030466                    ERROR                           ;REPORT STACKED ERROR
5917  030466  104460                                                    TRAP    C$ERROR
5918  030470                    ESCAPE  SUB                     ;SKIP TO END OF TEST
5919  030470  104410                                                    TRAP    C$ESCAPE
5920  030472  000176                                                    .WORD   L10035-.
5921  030474  004537  007734            JSR     R5,TXCTRL       ;SET TEOM
5922  030500  000002            TEOM
5923  030502  000000            0
5924  030504  004537  010034            JSR     R5,RXCHAR       ;READ/CHECK 000
5925  030510  000000            000
5926  030512  000000            0
5927  030514  100010            NOCRDA!8.                       ;NO INITIAL CHECK OF RDA=0
5928  030516  103003            BCC     .+8.                    ;BR IF NO ERROR
5929  030520                    ERROR                           ;REPORT STACKED ERROR
5930  030520  104460                                                    TRAP    C$ERROR
5931  030522                    ESCAPE  SUB                     ;SKIP TO END OF TEST
5932  030522  104410                                                    TRAP    C$ESCAPE
5933  030524  000144                                                    .WORD   L10035-.
5934  030526  004537  007734            JSR     R5,TXCTRL       ;SET TEOM
5935  030532  000002            TEOM
5936  030534  000000            0
5937  030536  004537  010034            JSR     R5,RXCHAR       ;READ/CHECK COMPLEMENTED ADDRESS
5938  030542  000000    5$:     000                             ;** HOLE FOR ADDRESS_NOT
5939  030544  000000            0                               ;NO INITIAL CHECK OF RDA=0
5940  030546  120010            NOCRDA!NCRACT!8.                ;DON'T CHECK FINAL RXACT=1
5941  030550  103003            BCC     .+8.                    ;BR IF NO ERROR
5942  030552                    ERROR                           ;REPORT STACKED ERROR
5943  030552  104460                                                    TRAP    C$ERROR
5944  030554                    ESCAPE  SUB                     ;SKIP TO END OF TEST
5945  030554  104410                                                    TRAP    C$ESCAPE
5946  030556  000112                                                    .WORD   L10035-.
5947  030560  004537  007734            JSR     R5,TXCTRL       ;SET TSOM
5948  030564  000001            TSOM
5949  030566  000000            0
5950  030570  004537  010034            JSR     R5,RXCHAR       ;READ/CHECK ADDRESS (AGAIN)
5951  030574  001000    6$:     RXEOM!000                       ;** HOLE FOR FINAL ADDRESS
5952  030576  000000            0
5953  030600  060000            NFCRDA!NCRACT                   ;DON'T CHECK FOR FINAL RDA=RXACT=1
5954  030602  103003            BCC     .+8.                    ;BR IF NO ERROR
5955  030604                    ERROR                           ;REPORT STACKED ERROR
5956  030604  104460                                                    TRAP    C$ERROR
5957  030606                    ESCAPE  SUB                     ;SKIP TO END OF TEST
5958  030606  104410                                                    TRAP    C$ESCAPE
5959  030610  000060                                                    .WORD   L10035-.
5960  030612  000422            BR      20$                     ;BR TO CONTINUE TEST
```

```
5961
5962                                    ;-------------------------------------------------------------
5963                                    ; USYRT DIDN'T RESPOND TO MESSAGE (RDA=0): SHOULD IT HAVE ?
5964 030614 022703 000123               ;-------------------------------------------------------------
5965 030620 001006              10$:    CMP     #123,R3         ;WAS NON-RESPONDING ADDR=S/AR ?
                                        BNE     50$             ;BR IF NO
5966                                    ;-------------------------------------------------------------
5967                                    ; ...NO, REPORT ERROR : 'USYRT DIDN'T RESPOND TO SECONDARY STATION ADDR''
5968                                    ;-------------------------------------------------------------
5969 030622                             GEDF    EM103,ERR20
5970                                                    ;       'DEVICE FATAL'' ERROR # 46
5971 030622 104455                                                              TRAP    C$ERDF
5972 030624 000056                                                              .WORD   46
5973 030626 016530                                                              .WORD   EM103
5974 030630 022146                                                              .WORD   ERR20
5975 030632                             ESCAPE  SUB
5976 030632 104410                                                              TRAP    C$ESCAPE
5977 030634 000034                                                              .WORD   L10035-.
5978
5979 030636 022703 000377       50$:    CMP     #377,R3         ;WAS NON-RESPONDING ADDR=APA(377) ?
5980 030642 001006                      BNE     20$             ;BR IF NO
5981                                    ;-------------------------------------------------------------
5982                                    ; ...NO, REPORT ERROR : 'USYRT DIDN'T RESPOND TO ALL PARTIES ADDRESS(377)''
5983                                    ;-------------------------------------------------------------
5984 030644                             GEDF    EM104,ERR20
5985                                                    ;       'DEVICE FATAL'' ERROR # 47
5986 030644 104455                                                              TRAP    C$ERDF
5987 030646 000057                                                              .WORD   47
5988 030650 016607                                                              .WORD   EM104
5989 030652 022146                                                              .WORD   ERR20
5990 030654                             ESCAPE  SUB
5991 030654 104410                                                              TRAP    C$ESCAPE
5992 030656 000012                                                              .WORD   L10035-.
5993                                    ;-------------------------------------------------------------
5994                                    ;...YES, UPDATE ADDRESS AND CONTINUE TESTING
5995                                    ;-------------------------------------------------------------
5996 030660 105203              20$:    INCB    R3              ;INCREMENT TESTING ADDRESS
5997 030662 001402                      BEQ     NOLP            ;IF ADDRESS .LE. 377 THEN CHECK IT
5998 030664 000137 030204               JMP     LOOP
5999 030670                     NOLP:   ENDSUB                  ;OTHERWISE END TEST....
6000 030670                                                             L10035:
6001 030670 104403                                                              TRAP    C$ESUB
6002 030672                     ENDTST
6003 030672                                                             L10034:
6004 030672 104401                                                              TRAP    C$ETST
```

```
                                        .SBTTL  TEST 7 -- BOP RX BIT STUFFING TEST

                                        ;*++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                        ;*
                                        ;*      TEST 7 -- BOP RX BIT STUFFING TEST
                                        ;*
                                        ;*    THE  USYRT  IS  INITIALIZED  AND  THE FOLLOWING TEXT IS TRANSMITTED
                                        ;*    (DELIMITED BY THE APPROPRIATE CONTROL CHARACTERS -- OF COURSE):
                                        ;*
                                        ;*      000, 017, 036, 074, 170, 360, 037, 076, 174, 370, 077, 176, 374,
                                        ;*      177, 376, 377.
                                        ;*
                                        ;*   NOTE  THAT  THIS  PATTERN  CONSISTS  OF CHARACTERS WHICH REQUIRE BIT
                                        ;*   STUFFING   BOTH   INDIVIDUALLY  AND  IN  COMBINATION  WITH  ADJACENT
                                        ;*   CHARACTERS.  THERE ARE ALSO CHARACTERS WHICH REQUIRE NO BIT STUFFING
                                        ;*   AT  ALL.  ALL 16 CHARACTERS ARE READ BY THE RECEIVER AND COMPARED AS
                                        ;*   THEY ARE MADE AVAILABLE AT RXDB.
                                        ;*
                                        ;--***************************************************************************
                                        ;
                                        ;       BGNTST
030674                                                                                  T7::
030674  004737  005344                          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

030700  004537  007324                          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030704  003626                                  NOCHK!SYNCH             ;SET BOP MODE,SYNCH REG=226
030706  000000                                  0                       ;USE 8 BIT CHARS
030710  103003                                  BCC     .+8.            ;BR IF NO ERROR
030712                                          ERROR                   ;REPORT STACKED ERROR
030712  104460                                                                          TRAP    C$ERROR
030714                                          ESCAPE  TST             ;SKIP TO END OF TEST
030714  104410                                                                          TRAP    C$ESCAPE
030716  001120                                                                          .WORD   L10036-.

030720  004537  007734                          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
030724  000001                                  TSOM
030726  000007                                  7.
030730  004537  007734                          JSR     R5,TXCTRL       ;CLEAR TSOM
030734  000000                                  000
030736  000000                                  0

030740  004537  007622                          JSR     R5,TXCHAR       ;LOAD 000(DATA1), TX 2ND FLAG
030744  000000                                  000
030746  000010                                  8.
030750  103003                                  BCC     .+8.            ;BR IF NO ERROR
030752                                          ERROR                   ;REPORT STACKED ERROR
030752  104460                                                                          TRAP    C$ERROR
030754                                          ESCAPE  TST             ;SKIP TO END OF TEST
030754  104410                                                                          TRAP    C$ESCAPE
030756  001060                                                                          .WORD   L10036-.

030760  004537  007622                          JSR     R5,TXCHAR       ;LOAD 017(DATA2), TX 000(DATA1)
030764  000017                                  017
030766  000010                                  8.
030770  103003                                  BCC     .+8.            ;BR IF NO ERROR
030772                                          ERROR                   ;REPORT STACKED ERROR
```

```
030772  104460                                                                  TRAP    C$ERROR
030774                   ESCAPE  TST             ;SKIP TO END OF TEST
030774  104410                                                                  TRAP    C$ESCAPE
030776  001040                                                                  .WORD   L10036-.

031000  004537  007622   JSR     R5,TXCHAR       ;LOAD 036(DATA3), TX 017(DATA2)
031004  000036           036
031006  000010           8.
031010  103003           BCC     .+8.            ;BR IF NO ERROR
031012                   ERROR                   ;REPORT STACKED ERROR
031012  104460                                                                  TRAP    C$ERROR
031014                   ESCAPE  TST             ;SKIP TO END OF TEST
031014  104410                                                                  TRAP    C$ESCAPE
031016  001020                                                                  .WORD   L10036-.

031020  004537  007622   JSR     R5,TXCHAR       ;LOAD 074(DATA4)
031024  000074           074
031026  000000           0
031030  103003           BCC     .+8.            ;BR IF NO ERROR
031032                   ERROR                   ;REPORT STACKED ERROR
031032  104460                                                                  TRAP    C$ERROR
031034                   ESCAPE  TST             ;SKIP TO END OF TEST
031034  104410                                                                  TRAP    C$ESCAPE
031036  001000                                                                  .WORD   L10036-.

031040  004537  011310   JSR     R5,RCV1ST       ;CLOCK AND RCV 000(DATA1)
031044  000000           0
031046  103003           BCC     .+8.            ;BR IF NO ERROR
031050                   ERROR                   ;REPORT STACKED ERROR
031050  104460                                                                  TRAP    C$ERROR
031052                   ESCAPE  TST             ;SKIP TO END OF TEST
031052  104410                                                                  TRAP    C$ESCAPE
031054  000762                                                                  .WORD   L10036-.

031056  004537  010034   JSR     R5,RXCHAR       ;READ & CHK 000(DATA1), RCV 017(DATA2)
031062  000400           RXSOM!000               ; & CHECK RSOM=1
031064  000000           0
031066  000010           8.
031070  103003           BCC     .+8.            ;BR IF NO ERROR
031072                   ERROR                   ;REPORT STACKED ERROR
031072  104460                                                                  TRAP    C$ERROR
031074                   ESCAPE  TST             ;SKIP TO END OF TEST
031074  104410                                                                  TRAP    C$ESCAPE
031076  000740                                                                  .WORD   L10036-.

031100  004537  007622   JSR     R5,TXCHAR       ;LOAD 170(DATA5)
031104  000170           170
031106  000000           0
031110  103003           BCC     .+8.            ;BR IF NO ERROR
031112                   ERROR                   ;REPORT STACKED ERROR
031112  104460                                                                  TRAP    C$ERROR
031114                   ESCAPE  TST             ;SKIP TO END OF TEST
031114  104410                                                                  TRAP    C$ESCAPE
031116  000720                                                                  .WORD   L10036-.

031120  004537  010034   JSR     R5,RXCHAR       ;READ/CHECK 017(DATA2),RCV 036(DATA3)
```

```
6117  031124  000017                  017
6118  031126  000000                  0
6119  031130  000010                  8.
6120  031132  103003                  BCC     .+8.            ;BR IF NO ERROR
6121  031134                          ERROR                   ;REPORT STACKED ERROR
6122  031134  104460                                                  TRAP    C$ERROR
6123  031136                          ESCAPE  TST             ;SKIP TO END OF TEST
6124  031136  104410                                                  TRAP    C$ESCAPE
6125  031140  000676                                                  .WORD   L10036-.
6126
6127  031142  004537  007622          JSR     R5,TXCHAR       ;LOAD 360(DATA6)
6128  031146  000360                  360
6129  031150  000000                  0
6130  031152  103003                  BCC     .+8.            ;BR IF NO ERROR
6131  031154                          ERROR                   ;REPORT STACKED ERROR
6132  031154  104460                                                  TRAP    C$ERROR
6133  031156                          ESCAPE  TST             ;SKIP TO END OF TEST
6134  031156  104410                                                  TRAP    C$ESCAPE
6135  031160  000656                                                  .WORD   L10036-.
6136
6137  031162  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 036(DATA3),RCV 074(DATA4)
6138  031166  000036                  036
6139  031170  000000                  0
6140  031172  000010                  8.
6141  031174  103003                  BCC     .+8.            ;BR IF NO ERROR
6142  031176                          ERROR                   ;REPORT STACKED ERROR
6143  031176  104460                                                  TRAP    C$ERROR
6144  031200                          ESCAPE  TST             ;SKIP TO END OF TEST
6145  031200  104410                                                  TRAP    C$ESCAPE
6146  031202  000634                                                  .WORD   L10036-.
6147
6148  031204  004537  007622          JSR     R5,TXCHAR       ;LOAD 037(DATA7)
6149  031210  000037                  037
6150  031212  000000                  0
6151  031214  103003                  BCC     .+8.            ;BR IF NO ERROR
6152  031216                          ERROR                   ;REPORT STACKED ERROR
6153  031216  104460                                                  TRAP    C$ERROR
6154  031220                          ESCAPE  TST             ;SKIP TO END OF TEST
6155  031220  104410                                                  TRAP    C$ESCAPE
6156  031222  000614                                                  .WORD   L10036-.
6157
6158  031224  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 074(DATA4),RCV 170(DATA5)
6159  031230  000074                  074
6160  031232  000000                  0
6161  031234  000010                  8.
6162  031236  103003                  BCC     .+8.            ;BR IF NO ERROR
6163  031240                          ERROR                   ;REPORT STACKED ERROR
6164  031240  104460                                                  TRAP    C$ERROR
6165  031242                          ESCAPE  TST             ;SKIP TO END OF TEST
6166  031242  104410                                                  TRAP    C$ESCAPE
6167  031244  000572                                                  .WORD   L10036-.
6168
6169  031246  004537  007622          JSR     R5,TXCHAR       ;LOAD 076(DATA8)
6170  031252  000076                  076
6171  031254  000000                  0
6172  031256  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
6173  031260
6174  031260  104460                ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
6175  031262
6176  031262  104410                ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
6177  031264  000552                                                    .WORD   L10036-.
6178
6179  031266  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 170(DATA5),RCV 360(DATA6)
6180  031272  000170                170
6181  031274  000000                0
6182  031276  000010                8.
6183  031300  103003                BCC     .+8.            ;BR IF NO ERROR
6184  031302                        ERROR                   ;REPORT STACKED ERROR
6185  031302  104460                                                   TRAP    C$ERROR
6186  031304
6187  031304  104410                ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
6188  031306  000530                                                    .WORD   L10036-.
6189
6190  031310  004537  007622        JSR     R5,TXCHAR       ;LOAD 174(DATA9)
6191  031314  000174                174
6192  031316  000000                0
6193  031320  103003                BCC     .+8.            ;BR IF NO ERROR
6194  031322                        ERROR                   ;REPORT STACKED ERROR
6195  031322  104460                                                   TRAP    C$ERROR
6196  031324
6197  031324  104410                ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
6198  031326  000510                                                    .WORD   L10036-.
6199
6200  031330  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 360(DATA6),RCV 037(DATA7)
6201  031334  000360                360
6202  031336  000000                0
6203  031340  000010                8.
6204  031342  103003                BCC     .+8.            ;BR IF NO ERROR
6205  031344                        ERROR                   ;REPORT STACKED ERROR
6206  031344  104460                                                   TRAP    C$ERROR
6207  031346
6208  031346  104410                ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
6209  031350  000466                                                    .WORD   L10036-.
6210
6211  031352  004537  007622        JSR     R5,TXCHAR       ;LOAD 370(DATA10)
6212  031356  000370                370
6213  031360  000000                0
6214  031362  103003                BCC     .+8.            ;BR IF NO ERROR
6215  031364                        ERROR                   ;REPORT STACKED ERROR
6216  031364  104460                                                   TRAP    C$ERROR
6217  031366
6218  031366  104410                ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
6219  031370  000446                                                    .WORD   L10036-.
6220
6221  031372  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 037(DATA7),RCV 076(DATA8)
6222  031376  000037                037
6223  031400  000000                0
6224  031402  000014                12.                     ;(EXTRA 4 TICKS FOR BIT-STUFF & FIFO)
6225  031404  103003                BCC     .+8.            ;BR IF NO ERROR
6226  031406                        ERROR                   ;REPORT STACKED ERROR
6227  031406  104460                                                   TRAP    C$ERROR
6228  031410                        ESCAPE  TST             ;SKIP TO END OF TEST
```

```
6229  031410  104410                                                          TRAP    C$ESCAPE
6230  031412  000424                                                          .WORD   L10036-.
6231
6232  031414  004537  007622        JSR     R5,TXCHAR       ;LOAD 077(DATA11)
6233  031420  000077                077
6234  031422  000000                0
6235  031424  103003                BCC     .+8.            ;BR IF NO ERROR
6236  031426                        ERROR                   ;REPORT STACKED ERROR
6237  031426  104460                                                          TRAP    C$ERROR
6238  031430                        ESCAPE  TST             ;SKIP TO END OF TEST
6239  031430  104410                                                          TRAP    C$ESCAPE
6240  031432  000404                                                          .WORD   L10036-.
6241
6242  031434  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 076(DATA8),RCV 174(DATA9)
6243  031440  000076                076
6244  031442  000000                0
6245  031444  000010                8.
6246  031446  103003                BCC     .+8.            ;BR IF NO ERROR
6247  031450                        ERROR                   ;REPORT STACKED ERROR
6248  031450  104460                                                          TRAP    C$ERROR
6249  031452                        ESCAPE  TST             ;SKIP TO END OF TEST
6250  031452  104410                                                          TRAP    C$ESCAPE
6251  031454  000362                                                          .WORD   L10036-.
6252
6253  031456  004537  007622        JSR     R5,TXCHAR       ;LOAD 176(DATA12)
6254  031462  000176                176
6255  031464  000000                0
6256  031466  103003                BCC     .+8.            ;BR IF NO ERROR
6257  031470                        ERROR                   ;REPORT STACKED ERROR
6258  031470  104460                                                          TRAP    C$ERROR
6259  031472                        ESCAPE  TST             ;SKIP TO END OF TEST
6260  031472  104410                                                          TRAP    C$ESCAPE
6261  031474  000342                                                          .WORD   L10036-.
6262
6263  031476  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 174(DATA9),RCV 370(DATA10)
6264  031502  000174                174
6265  031504  000000                0
6266  031506  000010                8.
6267  031510  103003                BCC     .+8.            ;BR IF NO ERROR
6268  031512                        ERROR                   ;REPORT STACKED ERROR
6269  031512  104460                                                          TRAP    C$ERROR
6270  031514                        ESCAPE  TST             ;SKIP TO END OF TEST
6271  031514  104410                                                          TRAP    C$ESCAPE
6272  031516  000320                                                          .WORD   L10036-.
6273
6274  031520  004537  007622        JSR     R5,TXCHAR       ;LOAD 374(DATA13)
6275  031524  000374                374
6276  031526  000000                0
6277  031530  103003                BCC     .+8.            ;BR IF NO ERROR
6278  031532                        ERROR                   ;REPORT STACKED ERROR
6279  031532  104460                                                          TRAP    C$ERROR
6280  031534                        ESCAPE  TST             ;SKIP TO END OF TEST
6281  031534  104410                                                          TRAP    C$ESCAPE
6282  031536  000300                                                          .WORD   L10036-.
6283
6284  031540  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 370(DATA10),RCV 077(DATA11)
```

```
6285  031544  000370                  370
6286  031546  000000                  0
6287  031550  000010                  8.
6288  031552  103003                  BCC     .+8.            ;BR IF NO ERROR
6289  031554                          ERROR                   ;REPORT STACKED ERROR
6290  031554  104460                                                  TRAP    C$ERROR
6291  031556                          ESCAPE  TST             ;SKIP TO END OF TEST
6292  031556  104410                                                  TRAP    C$ESCAPE
6293  031560  000256                                                  .WORD   L10036-.
6294
6295  031562  004537  007622          JSR     R5,TXCHAR       ;LOAD 177(DATA14)
6296  031566  000177                  177
6297  031570  000000                  0
6298  031572  103003                  BCC     .+8.            ;BR IF NO ERROR
6299  031574                          ERROR                   ;REPORT STACKED ERROR
6300  031574  104460                                                  TRAP    C$ERROR
6301  031576                          ESCAPE  TST             ;SKIP TO END OF TEST
6302  031576  104410                                                  TRAP    C$ESCAPE
6303  031600  000236                                                  .WORD   L10036-.
6304
6305  031602  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 077(DATA11),RCV 176(DATA12)
6306  031606  000077                  077
6307  031610  000000                  0
6308  031612  000014                  12.                     ;(EXTRA 4 TICKS FOR BIT-STUFF & FIFO)
6309  031614  103003                  BCC     .+8.            ;BR IF NO ERROR
6310  031616                          ERROR                   ;REPORT STACKED ERROR
6311  031616  104460                                                  TRAP    C$ERROR
6312  031620                          ESCAPE  TST             ;SKIP TO END OF TEST
6313  031620  104410                                                  TRAP    C$ESCAPE
6314  031622  000214                                                  .WORD   L10036-.
6315
6316  031624  004537  007622          JSR     R5,TXCHAR       ;LOAD 376(DATA15)
6317  031630  000376                  376
6318  031632  000000                  0
6319  031634  103003                  BCC     .+8.            ;BR IF NO ERROR
6320  031636                          ERROR                   ;REPORT STACKED ERROR
6321  031636  104460                                                  TRAP    C$ERROR
6322  031640                          ESCAPE  TST             ;SKIP TO END OF TEST
6323  031640  104410                                                  TRAP    C$ESCAPE
6324  031642  000174                                                  .WORD   L10036-.
6325
6326  031644  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 176(DATA12),RCV 374(DATA13)
6327  031650  000176                  176
6328  031652  000000                  0
6329  031654  000010                  8.
6330  031656  103003                  BCC     .+8.            ;BR IF NO ERROR
6331  031660                          ERROR                   ;REPORT STACKED ERROR
6332  031660  104460                                                  TRAP    C$ERROR
6333  031662                          ESCAPE  TST             ;SKIP TO END OF TEST
6334  031662  104410                                                  TRAP    C$ESCAPE
6335  031664  000152                                                  .WORD   L10036-.
6336
6337  031666  004537  007622          JSR     R5,TXCHAR       ;LOAD 377(DATA16)
6338  031672  000377                  377
6339  031674  000000                  0
6340  031676  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
6341  031700                        ERROR                   ;REPORT STACKED ERROR
6342  031700  104460                                                  TRAP    C$ERROR
6343  031702                        ESCAPE  TST             ;SKIP TO END OF TEST
6344  031702  104410                                                  TRAP    C$ESCAPE
6345  031704  000132                                                  .WORD   L10036-.
6346
6347  031706  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 374(DATA13),RCV 177(DATA14)
6348  031712  000374                374
6349  031714  000000                0
6350  031716  000010                8.
6351  031720  103003                BCC     .+8.            ;BR IF NO ERROR
6352  031722                        ERROR                   ;REPORT STACKED ERROR
6353  031722  104460                                                  TRAP    C$ERROR
6354  031724                        ESCAPE  TST             ;SKIP TO END OF TEST
6355  031724  104410                                                  TRAP    C$ESCAPE
6356  031726  000110                                                  .WORD   L10036-.
6357
6358  031730  004537  007734        JSR     R5,TXCTRL       ;LOAD 1ST TEOM
6359  031734  000002                TEOM
6360  031736  000000                0
6361  031740  103003                BCC     .+8.            ;BR IF NO ERROR
6362  031742                        ERROR                   ;RFPORT STACKED ERROR
6363  031742  104460                                                  TRAP    C$ERROR
6364  031744                        ESCAPE  TST             ;SKIP TO END OF TEST
6365  031744  104410                                                  TRAP    C$ESCAPE
6366  031746  000070                                                  .WORD   L10036-.
6367
6368  031750  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 177(DATA14),RCV 376(DATA15)
6369  031754  000177                177
6370  031756  000000                0
6371  031760  000010                8.
6372  031762  103003                BCC     .+8.            ;BR IF NO ERROR
6373  031764                        ERROR                   ;REPORT STACKED ERROR
6374  031764  104460                                                  TRAP    C$ERROR
6375  031766                        ESCAPE  TST             ;SKIP TO END OF TEST
6376  031766  104410                                                  TRAP    C$ESCAPE
6377  031770  000046                                                  .WORD   L10036-.
6378
6379  031772  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 376(DATA15),RCV 377(DATA16)
6380  031776  000376                376
6381  032000  000000                0                       ;DON'T CHECK FOR FINAL RXACT=1
6382  032002  020014                NCRACT!12.              ;(EXTRA 4 TICKS FOR BIT-STUFF/FIFO)
6383  032004  103003                BCC     .+8.            ;BR IF NO ERROR
6384  032006                        ERROR                   ;REPORT STACKED ERROR
6385  032006  104460                                                  TRAP    C$ERROR
6386  032010                        ESCAPE  TST             ;SKIP TO END OF TEST
6387  032010  104410                                                  TRAP    C$ESCAPE
6388  032012  000024                                                  .WORD   L10036-.
6389
6390  032014  004537  010034        JSR     R5,RXCHAR       ;READ/CHECK 377(DATA16)
6391  032020  001377                RXEOM!377               ; & CHECK REOM
6392  032022  000000                0
6393  032024  060000                NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
6394  032026  103003                BCC     .+8.            ;BR IF NO ERROR
6395  032030                        ERROR                   ;REPORT STACKED ERROR
6396  032030  104460                                                  TRAP    C$ERROR
```

```
6397   032032
6398   032032  104410          ESCAPE  TST             ;SKIP TO END OF TEST
6399   032034  000002                                                  TRAP    C$ESCAPE
6400   032036                                                          .WORD   L10036-.
6401   032036          ENDTST                                  L10036:
6402   032036  104401                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 8 -- BOP RX UNDERRUN IDLE ABORTS/FLAGS

                                ;+*********************************************************************
                                ;*
                                ;*      TEST 8 -- BOP RX UNDERRUN IDLE ABORTS/FLAGS
                                ;*
                                ;*  THE  USYRT  IS  INITIALIZED  AND  A  MESSAGE  IS  STARTED.  THEN,  A
                                ;*  TRANSMITTER  UNDERRUN  IS  FORCED  WITH  IDLE  =  0 -- CAUSING ABORT
                                ;*  CHARACTERS  TO  BE  IDLED. THE RECEIVER SHOULD BE RESET BY THE ABORT
                                ;*  CHARACTER(S). VERIFY THAT RAB/GA BIT=1.
                                ;*  REPEAT THE ABOVE WITH IDLE=1.
                                ;*
                                ;--*********************************************************************
                                ;
                                ;       BGNTST
032040                                                                  T8::
                                        ;========== SUBTEST # 1 ==========
032040                                  BGNSUB
032040                                                                  T8.1:
032040  104402                                                                  TRAP    C$BSUB
032042  004737  005344                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

032046  004537  007324                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
032052  003626                          NOCHK!SYNCH             ;SET BOP MODE,SYNCH REG=226
032054  000000                          0                       ;USE 8 BIT CHARS
032056  103003                          BCC     .+8.            ;BR IF NO ERROR
032060                                  ERROR                   ;REPORT STACKED ERROR
032060  104460                                                                  TRAP    C$ERROR
032062                                  ESCAPE  SUB             ;SKIP TO END OF TEST
032062  104410                                                                  TRAP    C$ESCAPE
032064  000300                                                                  .WORD   L10040-.

032066  004537  007734                  JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
032072  000001                          TSOM
032074  000007                          7.
032076  004537  007734                  JSR     R5,TXCTRL       ;CLEAR TSOM
032102  000000                          000
032104  000000                          0

032106  004537  007622                  JSR     R5,TXCHAR       ;LOAD 123(DATA1), TX 2ND FLAG
032112  000123                          123
032114  000010                          8.
032116  103003                          BCC     .+8.            ;BR IF NO ERROR
032120                                  ERROR                   ;REPORT STACKED ERROR
032120  104460                                                                  TRAP    C$ERROR
032122                                  ESCAPE  SUB             ;SKIP TO END OF TEST
032122  104410                                                                  TRAP    C$ESCAPE
032124  000240                                                                  .WORD   L10040-.

032126  004537  007622                  JSR     R5,TXCHAR       ;LOAD 321(DATA2), TX 123(DATA1)
032132  000321                          321
032134  000010                          8.
032136  103003                          BCC     .+8.            ;BR IF NO ERROR
032140                                  ERROR                   ;REPORT STACKED ERROR
032140  104460                                                                  TRAP    C$ERROR
032142                                  ESCAPE  SUB             ;SKIP TO END OF TEST
```

```
6459  032142  104410                                                          TRAP    C$ESCAPE
6460  032144  000220                                                          .WORD   L10040-.
6461
6462  032146  004537  007622          JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 321(DATA2)
6463  032152  000000                  000
6464  032154  000010                  8.
6465  032156  103003                  BCC     .+8.            ;BR IF NO ERROR
6466  032160                          ERROR                   ;REPORT STACKED ERROR
6467  032160  104460                                                          TRAP    C$ERROR
6468  032162                          ESCAPE  SUB             ;SKIP TO END OF TEST
6469  032162  104410                                                          TRAP    C$ESCAPE
6470  032164  000200                                                          .WORD   L10040-.
6471
6472  032166  004537  011310          JSR     R5,RCV1ST       ;CLOCK AND RCV 123(DATA1)
6473  032172  000000                  0
6474  032174  103003                  BCC     .+8.            ;BR IF NO ERROR
6475  032176                          ERROR                   ;REPORT STACKED ERROR
6476  032176  104460                                                          TRAP    C$ERROR
6477  032200                          ESCAPE  SUB             ;SKIP TO END OF TEST
6478  032200  104410                                                          TRAP    C$ESCAPE
6479  032202  000162                                                          .WORD   L10040-.
6480
6481  032204  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 123(DATA1), RCV 321(DATA2)
6482  032210  000523                  RXSOM!123               ; & CHECK RSOM=1
6483  032212  000000                  0
6484  032214  000010                  8.                      ; 8 TICKS OF THE CLOCK
6485  032216  103003                  BCC     .+8.            ;BR IF NO ERROR
6486  032220                          ERROR                   ;REPORT STACKED ERROR
6487  032220  104460                                                          TRAP    C$ERROR
6488  032222                          ESCAPE  SUB             ;SKIP TO END OF TEST
6489  032222  104410                                                          TRAP    C$ESCAPE
6490  032224  000140                                                          .WORD   L10040-.
6491                          ;-------------------------------------------------------------------
6492  032226  004537  005356          JSR     R5,CKUSTS       ;+++ CHECK FOR TXU=1 (& S/F=0) +++
6493  032232  000356                  RDA!TBMT!RXACT!TXU!TSO!TXACT
6494  032234  103003                  BCC     .+8.            ;BR IF NO ERROR
6495  032236                          ERROR                   ;REPORT STACKED ERROR
6496  032236  104460                                                          TRAP    C$ERROR
6497  032240                          ESCAPE  SUB             ;SKIP TO END OF TEST
6498  032240  104410                                                          TRAP    C$ESCAPE
6499  032242  000122                                                          .WORD   L10040-.
6500
6501  032244  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 321(DATA2),DATA3 LOST....
6502  032250  000321                  321
6503  032252  000000                  0
6504  032254  060010                  NFCRDA!NCRACT!8.        ;NO CHECKING OF RDA
6505  032256  103003                  BCC     .+8.            ;BR IF NO ERROR
6506  032260                          ERROR                   ;REPORT STACKED ERROR
6507  032260  104460                                                          TRAP    C$ERROR
6508  032262                          ESCAPE  SUB             ;SKIP TO END OF TEST
6509  032262  104410                                                          TRAP    C$ESCAPE
6510  032264  000100                                                          .WORD   L10040-.
6511
6512  032266  004537  003534          JSR     R5,READI        ;READ RECEIVER STATUS REGISTER
6513  032272  120401                  RDSRH
6514  032274  000000          1$:     000                     ;* RESULTS GO HERE
```

```
6515  032276  132737  000004  032274          BITB    #RABGA,1$       ;+++ CHECK IF RAB/GA BIT = 1 +++
6516  032304  001006                          BNE     10$             ;BR IF BIT SET (IE: IF OK)
6517  032306                                  GEDF    EM40,ERR12      ;** REPORT RAB/GA BIT NOT SET!!!
6518                                                                  ;       'DEVICE FATAL'' ERROR # 48
6519  032306  104455                                                          TRAP    C$ERDF
6520  032310  000060                                                          .WORD   48
6521  032312  014734                                                          .WORD   EM40
6522  032314  021714                                                          .WORD   ERR12
6523  032316                                  ESCAPE  SUB             ;** AND EXIT TEST
6524  032316  104410                                                          TRAP    C$ESCAPE
6525  032320  000044                                                          .WORD   L10040-.
6526  032322  132737  000002  032274  10$:   BITB    #REOM,1$        ;+++ CHECK FOR RXEOM BIT = 1 +++
6527  032330  001006                          BNE     15$             ;BR IF BIT SET (IE: IF OK)
6528  032332                                  GEDF    EM31,ERR12      ;** REPORT REOM BIT NOT SET!!!
6529                                                                  ;       'DEVICE FATAL'' ERROR # 49
6530  032332  104455                                                          TRAP    C$ERDF
6531  032334  000061                                                          .WORD   49
6532  032336  014537                                                          .WORD   EM31
6533  032340  021714                                                          .WORD   ERR12
6534  032342                                  ESCAPE  SUB             ;** AND EXIT TEST
6535  032342  104410                                                          TRAP    C$ESCAPE
6536  032344  000020                                                          .WORD   L10040-.
6537
6538  032346  004537  005356  15$:   JSR     R5,CKUSTS       ;++ CHECK USYRT STATUS ++
6539  032352  000116                          TBMT!TSO!TXACT!TXU
6540  032354  103003                          BCC     .+8.            ;BR IF NO ERROR
6541  032356                                  ERROR                   ;REPORT STACKED ERROR
6542  032356  104460                                                          TRAP    C$ERROR
6543  032360                                  ESCAPE  SUB             ;SKIP TO END OF TEST
6544  032360  104410                                                          TRAP    C$ESCAPE
6545  032362  000002                                                          .WORD   L10040-.
6546  032364                          ENDSUB
6547  032364                                                                  L10040:
6548  032364  104403                                                          TRAP    C$ESUB
6549
6550  032366                          ;========== SUBTEST # 2 ==========
6551  032366                          BGNSUB
6552  032366  104402                                                          T8.2:
6553  032370  004737  005344                                                  TRAP    C$BSUB
                        JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
6554
6555  032374  004537  007324                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
6556  032400  007626                          IDLES!NOCHK!SYNCH       ;SET BOP MODE,IDLE=1,SYNCH REG=226
6557  032402  000000                          0                       ;USE 8 BIT CHARS
6558  032404  103003                          BCC     .+8.            ;BR IF NO ERROR
6559  032406                                  ERROR                   ;REPORT STACKED ERROR
6560  032406  104460                                                          TRAP    C$ERROR
6561  032410                                  ESCAPE  SUB             ;SKIP TO END OF TEST
6562  032410  104410                                                          TRAP    C$ESCAPE
6563  032412  000242                                                          .WORD   L10041-.
6564
6565  032414  004537  007734                  JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
6566  032420  000001                          TSOM
6567  032422  000007                          7.
6568  032424  004537  007734                  JSR     R5,TXCTRL       ;CLEAR TSOM
6569  032430  000000                          000
6570  032432  000000                          0
```

```
6571
6572  032434  004537  007622          JSR     R5,TXCHAR       ;LOAD 123(DATA1), TX 2ND FLAG
6573  032440  000123                  123
6574  032442  000010                  8.
6575  032444  103003                  BCC     .+8.            ;BR IF NO ERROR
6576  032446                          ERROR                   ;REPORT STACKED ERROR
6577  032446  104460                                                  TRAP    C$ERROR
6578  032450                          ESCAPE  SUB             ;SKIP TO END OF TEST
6579  032450  104410                                                  TRAP    C$ESCAPE
6580  032452  000202                                                  .WORD   L10041-.
6581
6582  032454  004537  007622          JSR     R5,TXCHAR       ;LOAD 321(DATA2), TX 123(DATA1)
6583  032460  000321                  321
6584  032462  000010                  8.
6585  032464  103003                  BCC     .+8.            ;BR IF NO ERROR
6586  032466                          ERROR                   ;REPORT STACKED ERROR
6587  032466  104460                                                  TRAP    C$ERROR
6588  032470                          ESCAPE  SUB             ;SKIP TO END OF TEST
6589  032470  104410                                                  TRAP    C$ESCAPE
6590  032472  000162                                                  .WORD   L10041-.
6591
6592  032474  004537  007622          JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 321(DATA2)
6593  032500  000000                  000
6594  032502  000010                  8.
6595  032504  103003                  BCC     .+8.            ;BR IF NO ERROR
6596  032506                          ERROR                   ;REPORT STACKED ERROR
6597  032506  104460                                                  TRAP    C$ERROR
6598  032510                          ESCAPE  SUB             ;SKIP TO END OF TEST
6599  032510  104410                                                  TRAP    C$ESCAPE
6600  032512  000142                                                  .WORD   L10041-.
6601
6602  032514  004537  011310          JSR     R5,RCV1ST       ;CLOCK AND RCV 123(DATA1)
6603  032520  000000                  0
6604  032522  103003                  BCC     .+8.            ;BR IF NO ERROR
6605  032524                          ERROR                   ;REPORT STACKED ERROR
6606  032524  104460                                                  TRAP    C$ERROR
6607  032526                          ESCAPE  SUB             ;SKIP TO END OF TEST
6608  032526  104410                                                  TRAP    C$ESCAPE
6609  032530  000124                                                  .WORD   L10041-.
6610
6611  032532  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 123(DATA1), RCV 321(DATA2)
6612  032536  000523                  RXSOM!123               ; & CHECK RSOM=1
6613  032540  000000                  0
6614  032542  000010                  8.                      ; 8 TICKS OF THE CLOCK
6615  032544  103003                  BCC     .+8.            ;BR IF NO ERROR
6616  032546                          ERROR                   ;REPORT STACKED ERROR
6617  032546  104460                                                  TRAP    C$ERROR
6618  032550                          ESCAPE  SUB             ;SKIP TO END OF TEST
6619  032550  104410                                                  TRAP    C$ESCAPE
6620  032552  000102                                                  .WORD   L10041-.
6621                          ;-------------------------------------------------------------
6622  032554  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 321(DATA2),RCV 000(DATA3)
6623  032560  000321                  321
6624  032562  000000                  0
6625  032564  020010                  NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
6626  032566  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
6627  032570
6628  032570  104460                    ERROR                   ;REPORT STACKED ERROR
                                                                                TRAP    C$ERROR
6629  032572
6630  032572  104410                    ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
6631  032574  000060                                                            .WORD   L10041-.
6632
6633  032576  004537  005356            JSR     R5,CKUSTS       ;+++ CHECK FOR TXU=1 +++
6634  032602  000336                    RDA!TBMT!RSA!TSO!TXACT!TXU
6635  032604  103003                    BCC     .+8.            ;BR IF NO ERROR
6636  032606                            ERROR                   ;REPORT STACKED ERROR
6637  032606  104460                                                            TRAP    C$ERROR
6638  032610                            ESCAPE  SUB             ;SKIP TO END OF TEST
6639  032610  104410                                                            TRAP    C$ESCAPE
6640  032612  000042                                                            .WORD   L10041-.
6641
6642  032614  004537  010034            JSR     R5,RXCHAR       ;READ/CHECK 000(DATA3)
6643  032620  001000                    RXEOM!000               ; & CHECK REOM
6644  032622  000000                    0
6645  032624  060010                    NFCRDA!NCRACT!8.        ;DON'T CHECK FOR FINAL RDA=RXACT=1
6646  032626  103003                    BCC     .+8.            ;BR IF NO ERROR
6647  032630                            ERROR                   ;REPORT STACKED ERROR
6648  032630  104460                                                            TRAP    C$ERROR
6649  032632                            ESCAPE  SUB             ;SKIP TO END OF TEST
6650  032632  104410                                                            TRAP    C$ESCAPE
6651  032634  000020                                                            .WORD   L10041-.
6652
6653  032636  004537  005356            JSR     R5,CKUSTS       ;++ CHECK USYRT STATUS ++
6654  032642  000116                    TBMT!TSO!TXACT!TXU      ;
6655  032644  103003                    BCC     .+8.            ;BR IF NO ERROR
6656  032646                            ERROR                   ;REPORT STACKED ERROR
6657  032646  104460                                                            TRAP    C$ERROR
6658  032650                            ESCAPE  SUB             ;SKIP TO END OF TEST
6659  032650  104410                                                            TRAP    C$ESCAPE
6660  032652  000002                                                            .WORD   L10041-.
6661  032654                    ENDSUB
6662  032654                                                            L10041:
6663  032654  104403                                                            TRAP    C$ESUB
6664  032656                    ENDTST
6665  032656                                                            L10037:
6666  032656  104401                                                            TRAP    C$ETST
```

```
                                 .SBTTL  TEST 9 -- BOP RX LOST RXE TEST

                                 ;+*********************************************************************
                                 ;*
                                 ;*      TEST 9 -- BOP RX LOST RXE TEST
                                 ;*
                                 ;*  THE  USYRT  IS  INITIALIZED  AND  A MESSAGE IS STARTED. WHILE IN THE
                                 ;*  MIDDLE  OF  TEXT, RXE IS DROPPED AND THE REACTION OF THE RECEIVER IS
                                 ;*  MONITORED.
                                 ;*
                                 ;--*********************************************************************
                                 ;
                                 ;       BGNTST
032660                                                                     T9::

032660  004737  005344                   JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

032664  004537  007324                   JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
032670  007626                           IDLES!NOCHK!SYNCH       ;SET BOP MODE,IDLE=1,SYNCH REG=226
032672  000000                           0                       ;USE 8 BIT CHARS
032674  103003                           BCC     .+8.            ;BR IF NO ERROR
032676                                   ERROR                   ;REPORT STACKED ERROR
032676  104460                                                                   TRAP    C$ERROR
032700                                   ESCAPE  TST             ;SKIP TO END OF TEST
032700  104410                                                                   TRAP    C$ESCAPE
032702  000216                                                                   .WORD   L10042-.

032704  004537  007734                   JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
032710  000001                           TSOM
032712  000007                           7.
032714  004537  007734                   JSR     R5,TXCTRL       ;CLEAR TSOM
032720  000000                           000
032722  000000                           0

032724  004537  007622                   JSR     R5,TXCHAR       ;LOAD 123(DATA1), TX 2ND FLAG
032730  000123                           123
032732  000010                           8.
032734  103003                           BCC     .+8.            ;BR IF NO ERROR
032736                                   ERROR                   ;REPORT STACKED ERROR
032736  104460                                                                   TRAP    C$ERROR
032740                                   ESCAPE  TST             ;SKIP TO END OF TEST
032740  104410                                                                   TRAP    C$ESCAPE
032742  000156                                                                   .WORD   L10042-.

032744  004537  007622                   JSR     R5,TXCHAR       ;LOAD 321(DATA2), TX 123(DATA1)
032750  000321                           321
032752  000010                           8.
032754  103003                           BCC     .+8.            ;BR IF NO ERROR
032756                                   ERROR                   ;REPORT STACKED ERROR
032756  104460                                                                   TRAP    C$ERROR
032760                                   ESCAPE  TST             ;SKIP TO END OF TEST
032760  104410                                                                   TRAP    C$ESCAPE
032762  000136                                                                   .WORD   L10042-.

032764  004537  007622                   JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 321(DATA2)
032770  000000                           000
```

```
6723  032772  000010                  8.
6724  032774  103003                  BCC     .+8.            ;BR IF NO ERROR
6725  032776                          ERROR                   ;REPORT STACKED ERROR
6726  032776  104460                                                  TRAP    C$ERROR
6727  033000                          ESCAPE  TST             ;SKIP TO END OF TEST
6728  033000  104410                                                  TRAP    C$ESCAPE
6729  033002  000116                                                  .WORD   L10042-.
6730
6731  033004  004537  011310          JSR     R5,RCV1ST       ;CLOCK AND RCV 123(DATA1)
6732  033010  000000                  0
6733  033012  103003                  BCC     .+8.            ;BR IF NO ERROR
6734  033014                          ERROR                   ;REPORT STACKED ERROR
6735  033014  104460                                                  TRAP    C$ERROR
6736  033016                          ESCAPE  TST             ;SKIP TO END OF TEST
6737  033016  104410                                                  TRAP    C$ESCAPE
6738  033020  000100                                                  .WORD   L10042-.
6739
6740  033022  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 123(DATA1), RCV 321(DATA2)
6741  033026  000523                  RXSOM!123               ; & CHECK RSOM=1
6742  033030  000000                  0
6743  033032  000010                  8.                      ; 8 TICKS OF THE CLOCK
6744  033034  103003                  BCC     .+8.            ;BR IF NO ERROR
6745  033036                          ERROR                   ;REPORT STACKED ERROR
6746  033036  104460                                                  TRAP    C$ERROR
6747  033040                          ESCAPE  TST             ;SKIP TO END OF TEST
6748  033040  104410                                                  TRAP    C$ESCAPE
6749  033042  000056                                                  .WORD   L10042-.
6750                          ;-------------------------------------------------------------
6751  033044  004537  003660          JSR     R5,WRITEI       ;DROP RECEIVER ENABLE (RXEN)
6752  033050  120000                  VIAORB
6753  033052  000072                  TXEN!DTR!RTSND!TTLOOP
6754
6755  033054  004537  005356          JSR     R5,CKUSTS       ;+++ CHECK USYRT STATUS REGISTER +++
6756  033060  000116                  TBMT!TSO!TXACT!TXU
6757  033062  103003                  BCC     .+8.            ;BR IF NO ERROR
6758  033064                          ERROR                   ;REPORT STACKED ERROR
6759  033064  104460                                                  TRAP    C$ERROR
6760  033066                          ESCAPE  TST             ;SKIP TO END OF TEST
6761  033066  104410                                                  TRAP    C$ESCAPE
6762  033070  000030                                                  .WORD   L10042-.
6763
6764  033072  004537  007734          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
6765  033076  000001                  TSOM
6766  033100  000010                  8.
6767
6768  033102  004537  005356          JSR     R5,CKUSTS       ;+++ CHECK USYRT STATUS REGISTER +++
6769  033106  000104                  TBMT!TXACT
6770  033110  103003                  BCC     .+8.            ;BR IF NO ERROR
6771  033112                          ERROR                   ;REPORT STACKED ERROR
6772  033112  104460                                                  TRAP    C$ERROR
6773  033114                          ESCAPE  TST             ;SKIP TO END OF TEST
6774  033114  104410                                                  TRAP    C$ESCAPE
6775  033116  000002                                                  .WORD   L10042-.
6776  033120                  ENDTST
6777  033120                                                  L10042:
6778  033120  104401                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 10 -- BOP RX GA (GO-AHEAD) RECOGNITION

                        ;+*************************************************************************
                        ;*
                        ;*      TEST 10 -- BOP RX GA (GO-AHEAD) RECOGNITION
                        ;*
                        ;*  A  SHORT  MESSAGE IS TRANSMITTED FOLLOWED BY A GA CHARACTER (INSTEAD
                        ;*  OF  A  FLAG CHARACTER). THE RECEIVER IS OBSERVED FOR PROPER HANDLING
                        ;*  OF  BOTH  THE  MESSAGE  AND  THE GA CHARACTER. THE RAB/GA STATUS BIT
                        ;*  SHOULD BE SET BY THE RECEIVER UPON RECOGNITION OF THE GA CHARACTER.
                        ;*
                        ;-*************************************************************************
                        ;
                        ;       BGNTST
033122                                                                  T10::
033122  004737  005344          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

033126  004537  007324          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
033132  023400                  STRIPS!NOCHK            ;SET BOP MODE,NO ERROR CHECKING,SS/GA=1
033134  000000                  0                       ;USE 8 BIT CHARS
033136  103003                  BCC     .+8.            ;BR IF NO ERROR
033140                          ERROR                   ;REPORT STACKED ERROR
033140  104460                                                          TRAP    C$ERROR
033142                          ESCAPE  TST             ;SKIP TO END OF TEST
033142  104410                                                          TRAP    C$ESCAPE
033144  000216                                                          .WORD   L10043-.

033146  004537  007734          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
033152  000001                  TSOM
033154  000007                  7.
033156  004537  007734          JSR     R5,TXCTRL       ;CLEAR TSOM
033162  000000                  000
033164  000000                  0

033166  004537  007622          JSR     R5,TXCHAR       ;LOAD 123(DATA1), TX 2ND FLAG
033172  000123                  123
033174  000010                  8.
033176  103003                  BCC     .+8.            ;BR IF NO ERROR
033200                          ERROR                   ;REPORT STACKED ERROR
033200  104460                                                          TRAP    C$ERROR
033202                          ESCAPE  TST             ;SKIP TO END OF TEST
033202  104410                                                          TRAP    C$ESCAPE
033204  000156                                                          .WORD   L10043-.

033206  004537  007622          JSR     R5,TXCHAR       ;LOAD 321(DATA2), TX 123(DATA1)
033212  000321                  321
033214  000010                  8.
033216  103003                  BCC     .+8.            ;BR IF NO ERROR
033220                          ERROR                   ;REPORT STACKED ERROR
033220  104460                                                          TRAP    C$ERROR
033222                          ESCAPE  TST             ;SKIP TO END OF TEST
033222  104410                                                          TRAP    C$ESCAPE
033224  000136                                                          .WORD   L10043-.

033226  004537  007622          JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 321(DATA2)
033232  000000                  000
```

```
6835  033234  000010                  8.
6836  033236  103003                  BCC     .+8.            ;BR IF NO ERROR
6837  033240                          ERROR                   ;REPORT STACKED ERROR
6838  033240  104460                                                          TRAP    C$ERROR
6839  033242                          ESCAPE  TST             ;SKIP TO END OF TEST
6840  033242  104410                                                          TRAP    C$ESCAPE
6841  033244  000116                                                          .WORD   L10043-.
6842
6843  033246  004537  011310          JSR     R5,RCV1ST       ;CLOCK AND RCV 123(DATA1)
6844  033252  000000                  0
6845  033254  103003                  BCC     .+8.            ;BR IF NO ERROR
6846  033256                          ERROR                   ;REPORT STACKED ERROR
6847  033256  104460                                                          TRAP    C$ERROR
6848  033260                          ESCAPE  TST             ;SKIP TO END OF TEST
6849  033260  104410                                                          TRAP    C$ESCAPE
6850  033262  000100                                                          .WORD   L10043-.
6851
6852  033264  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 123(DATA1), RCV 321(DATA2)
6853  033270  000523                  RXSOM!123               ; & CHECK RSOM=1
6854  033272  000000                  0
6855  033274  000010                  8.                      ; 8 TICKS OF THE CLOCK
6856  033276  103003                  BCC     .+8.            ;BR IF NO ERROR
6857  033300                          ERROR                   ;REPORT STACKED ERROR
6858  033300  104460                                                          TRAP    C$ERROR
6859  033302                          ESCAPE  TST             ;SKIP TO END OF TEST
6860  033302  104410                                                          TRAP    C$ESCAPE
6861  033304  000056                                                          .WORD   L10043-.
6862
6863  033306  004537  007734          JSR     R5,TXCTRL       ;SET TEOM AND TGA
6864  033312  000012                  TEOM!TGA
6865  033314  000000                  0
6866
6867  033316  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 321(DATA2),RCV 000(DATA3)
6868  033322  000321                  321
6869  033324  000000                  0
6870  033326  020010                  NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
6871  033330  103003                  BCC     .+8.            ;BR IF NO ERROR
6872  033332                          ERROR                   ;REPORT STACKED ERROR
6873  033332  104460                                                          TRAP    C$ERROR
6874  033334                          ESCAPE  TST             ;SKIP TO END OF TEST
6875  033334  104410                                                          TRAP    C$ESCAPE
6876  033336  000024                                                          .WORD   L10043-.
6877
6878  033340  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 000(DATA3)
6879  033344  003000                  RXABGA!RXEOM!000        ;VERIFY REOM & RABGA = 1
6880  033346  000000                  0
6881  033350  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
6882  033352  103003                  BCC     .+8.            ;BR IF NO ERROR
6883  033354                          ERROR                   ;REPORT STACKED ERROR
6884  033354  104460                                                          TRAP    C$ERROR
6885  033356                          ESCAPE  TST             ;SKIP TO END OF TEST
6886  033356  104410                                                          TRAP    C$ESCAPE
6887  033360  000002                                                          .WORD   L10043-.
6888  033362                  ENDTST
6889  033362                                                          L10043:
6890  033362  104401                                                          TRAP    C$ETST
```

```
                                        .SBTTL  TEST 11 -- BOP RX "ABC" TEST

                                        ;+*******************************************************************
                                        ;*
                                        ;*      TEST 11 -- BOP RX "ABC" TEST
                                        ;*
                                        ;*  THIS TEST IS COMPOSED OF 7 SUBTESTS -- EACH ONE CHECKING A DIFFERENT
                                        ;*  EXPECTED VALUE IN ABC (THE 3 BIT "ASSEMBLED BIT COUNT" FIELD WITHIN
                                        ;*  RDSR). IN EACH SUBTEST THE USYRT IS INITIALIZED AND A SMALL MESSAGE
                                        ;*  IS STARTED. THE LAST CHARACTER IS SENT WITH ITS LENGTH BEING
                                        ;*  SPECIFIED FIRST AS 1 BIT, THEN AS 2 BITS, THEN AS 3 BITS, ETC. IN THE
                                        ;*  TRANSMITTER SIDE OF THE USYRT. IN ALL CASES THE RECEIVER IS LEFT SET
                                        ;*  TO 8 BITS IN LENGTH AND WHEN THE FLAG CHARACTER IS DETECTED, ABC IS
                                        ;*  CHECKED AND SHOULD MATCH TXCL. ERROR LOOPING WILL BE ON THE FAILING
                                        ;*  SUBTEST.
                                        ;*
                                        ;*
                                        ;--*******************************************************************
                                        ;
                                        ;       BGNTST
033364                                                                                  T11::
033364  004737  005344                          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
033370  012704  000001                          MOV     #1,R4           ;INIT GENERAL PURPOSE INDEX
                                        ;-----------------------------------------------------------
                                        ; MAIN PROGRAM LOOP
                                        ;-----------------------------------------------------------
033374                                  T9.LP:  BGNSUB
033374                                                                                  T11.1:
033374  104402                                                                                  TRAP    C$BSUB
033376  116437  034022  033764                  MOVB    TABLR(R4),30$   ;SET UP EXPECTED FINAL VALUE
033404  116437  034032  033540                  MOVB    LNTBL(R4),5$    ;SET UP FINAL TX CHAR LENGTH (1 => 8 BITS)

033412  004537  007324                          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
033416  003626                                  NOCHK!SYNCH             ;SET BOP MODE,SYNCH REG=226
033420  000000                                  0                       ;USE 8 BIT CHARS
033422  103003                                  BCC     .+8.            ;BR IF NO ERROR
033424                                          ERROR                   ;REPORT STACKED ERROR
033424  104460                                                                                  TRAP    C$ERROR
033426                                          ESCAPE  SUB             ;SKIP TO END OF TEST
033426  104410                                                                                  TRAP    C$ESCAPE
033430  000352                                                                                  .WORD   L10045-.

033432  004537  007734                          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
033436  000001                                  TSOM
033440  000007                                  7.
033442  004537  007734                          JSR     R5,TXCTRL       ;CLEAR TSOM
033446  000000                                  000
033450  000000                                  0

033452  004537  007622                          JSR     R5,TXCHAR       ;LOAD 123(DATA1), TX 2ND FLAG
033456  000123                                  123
033460  000010                                  8.
033462  103003                                  BCC     .+8.            ;BR IF NO ERROR
033464                                          ERROR                   ;REPORT STACKED ERROR
033464  104460                                                                                  TRAP    C$ERROR
033466                                          ESCAPE  SUB             ;SKIP TO END OF TEST
```

```
6947  033466  104410                                                          TRAP    C$ESCAPE
6948  033470  000312                                                          .WORD   L10045-.
6949
6950  033472  004537  007622          JSR     R5,TXCHAR       ;LOAD 321(DATA2), TX 123(DATA1)
6951  033476  000321                  321
6952  033500  000010                  8.
6953  033502  103003                  BCC     .+8.            ;BR IF NO ERROR
6954  033504                          ERROR                   ;REPORT STACKED ERROR
6955  033504  104460                                                          TRAP    C$ERROR
6956  033506                          ESCAPE  SUB             ;SKIP TO END OF TEST
6957  033506  104410                                                          TRAP    C$ESCAPE
6958  033510  000272                                                          .WORD   L10045-.
6959
6960  033512  004537  007622          JSR     R5,TXCHAR       ;LOAD 000(DATA3), TX 321(DATA2)
6961  033516  000000                  000
6962  033520  000010                  8.
6963  033522  103003                  BCC     .+8.            ;BR IF NO ERROR
6964  033524                          ERROR                   ;REPORT STACKED ERROR
6965  033524  104460                                                          TRAP    C$ERROR
6966  033526                          ESCAPE  SUB             ;SKIP TO END OF TEST
6967  033526  104410                                                          TRAP    C$ESCAPE
6968  033530  000252                                                          .WORD   L10045-.
6969
6970  033532  004537  003660          JSR     R5,WRITEI       ;CHANGE BIT LENGTH OF FINAL CHAR
6971  033536  120407                  PCR
6972  033540  000000          5$.     000                     ;** HOLE FOR RX/TX CHAR LENGTH **
6973
6974  033542  004537  007622          JSR     R5,TXCHAR       ;LOAD 377(DATA4); TX 000(DATA3)
6975  033546  000377                  377                     ;
6976  033550  000010                  8.                      ;
6977  033552  103003                  BCC     .+8.            ;BR IF NO ERROR
6978  033554                          ERROR                   ;REPORT STACKED ERROR
6979  033554  104460                                                          TRAP    C$ERROR
6980  033556                          ESCAPE  SUB             ;SKIP TO END OF TEST
6981  033556  104410                                                          TRAP    C$ESCAPE
6982  033560  000222                                                          .WORD   L10045-.
6983
6984  033562  004537  007734          JSR     R5,TXCTRL       ;TX DATA4 (ONLY # OF BITS SPECIFIED IN
6985  033566  000002                  TEOM                    ; R4 WILL GET TRANSMITTED) + SOME OF THE
6986  033570  000020                  16.                     ; CRC CHARACTER
6987  033572  004537  011540          JSR     R5,STEPLU       ;TX REMAINING CRC CHAR + PUT SOME EXTRA BITS
6988  033576  000040                  32.                     ;ON THE FIFO
6989
6990  033600  004537  010034          JSR     R5,RXCHAR       ;READ & CHK 123(DATA1), RCV 321(DATA2)
6991  033604  000523                  RXSOM!123               ; & CHECK RSOM=1
6992  033606  000000                  0
6993  033610  100000                  NOCRDA
6994  033612  103003                  BCC     .+8.            ;BR IF NO ERROR
6995  033614                          ERROR                   ;REPORT STACKED ERROR
6996  033614  104460                                                          TRAP    C$ERROR
6997  033616                          ESCAPE  SUB             ;SKIP TO END OF TEST
6998  033616  104410                                                          TRAP    C$ESCAPE
6999  033620  000162                                                          .WORD   L10045-.
7000
7001  033622  004537  010034          JSR     R5,RXCHAR       ;READ/CHECK 321(DATA2),RCV 000(DATA3)
7002  033626  000321                  321
```

```
7003  033630  000000                    0
7004  033632  120000                    NOCRDA!NCRACT
7005  033634  103003                    BCC     .+8.            ;BR IF NO ERROR
7006  033636                            ERROR                   ;REPORT STACKED ERROR
7007  033636  104460                                                    TRAP    C$ERROR
7008  033640                            ESCAPE  SUB             ;SKIP TO END OF TEST
7009  033640  104410                                                    TRAP    C$ESCAPE
7010  033642  000140                                                    .WORD   L10045-.
7011
7012  033644  004537  010034            JSR     R5,RXCHAR       ;READ/CHECK 000(DATA3),RCV DATA4
7013  033650  000000                    000
7014  033652  000000                    0
7015  033654  120000                    NOCRDA!NCRACT
7016  033656  103003                    BCC     .+8.            ;BR IF NO ERROR
7017  033660                            ERROR                   ;REPORT STACKED ERROR
7018  033660  104460                                                    TRAP    C$ERROR
7019  033662                            ESCAPE  SUB             ;SKIP TO END OF TEST
7020  033662  104410                                                    TRAP    C$ESCAPE
7021  033664  000116                                                    .WORD   L10045-.
7022
7023  033666  004537  003534            JSR     R5,READI        ;GET READ STATUS REGISTER
7024  033672  120401                    RDSRH
7025  033674  000000            20$:    000                     ;** HOLE FOR RDSRH VALUE **
7026  033676  042737  177617  033674    BIC     #177617,20$     ;MASK ''ABC'' VALUE
7027  033704  006237  033674            ASR     20$             ; AND RIGHT JUSTIFY IT
7028  033710  006237  033674            ASR     20$
7029  033714  006237  033674            ASR     20$
7030  033720  006237  033674            ASR     20$
7031  033724  020437  033674            CMP     R4,20$          ;IS ASSEMBLED BIT COUNT CORRECT ?
7032  033730  001413                    BEQ     31$
7033                                    ;---------------------------------------------------------------
7034                                    ;       ERROR REPORTING GOES HERE
7035                                    ;---------------------------------------------------------------
7036  033732  010437  002330            MOV     R4,GDATA        ;EXPECTED BIT COUNT
7037  033736  013737  033674  002332    MOV     20$,BDATA       ;ACTUAL (ERRONEOUS) BIT COUNT
7038  033744                            GEDF    EM105,ERR22
7039                                    ;       ''DEVICE FATAL'' ERROR # 50
7040  033744  104455                                                    TRAP    C$ERDF
7041  033746  000062                                                    .WORD   50
7042  033750  016665                                                    .WORD   EM105
7043  033752  022264                                                    .WORD   ERR22
7044  033754                            ESCAPE  SUB
7045  033754  104410                                                    TRAP    C$ESCAPE
7046  033756  000024                                                    .WORD   L10045-.
7047                                    ;---------------------------------------------------------------
7048  033760  004537  010034    31$:    JSR     R5,RXCHAR       ;READ/CHECK DATA4 (SHORT CHARACTER)
7049  033764  000001            30$:    001                     ;** HOLE FOR DATA4 VALUE **
7050  033766  000000                    0
7051  033770  060000                    NCRACT!NFCRDA           ;DON'T CHECK RECEIVER ACTIVE/FINAL RDA.
7052  033772  103003                    BCC     .+8.            ;BR IF NO ERROR
7053  033774                            ERROR                   ;REPORT STACKED ERROR
7054  033774  104460                                                    TRAP    C$ERROR
7055  033776                            ESCAPE  SUB             ;SKIP TO END OF TEST
7056  033776  104410                                                    TRAP    C$ESCAPE
7057  034000  000002                                                    .WORD   L10045-.
7058  034002                    ENDSUB
```

```
7059  034002
7060  034002  104403
7061  034004  005204                    INC     R4              ;BUMP GENERAL PURPOSE INDEX
7062  034006  020427  000010            CMP     R4,#8.          ;ARE WE DONE WITH THIS TEST ?
7063  034012  001402                    BEQ     .+6             ;EXIT IF YES
7064  034014  000137  033374            JMP     T9.LP           ;OTHERWISE DO THE NEXT COUNT
7065  034020                    ENDTST
7066  034020                                                                    L10045:
7067  034020  104401                                                            TRAP    C$ESUB
7068
7069  034022     377            ;-----------------------------
7070  034023     001            TABLR:  .BYTE   377
7071  034024     003                    .BYTE   001
7072  034025     007                    .BYTE   003
7073  034026     017                    .BYTE   007
7074  034027     037                    .BYTE   017
7075  034030     077                    .BYTE   037
7076  034031     177                    .BYTE   077
7077                                    .BYTE   177
7078  034032     000            ;-----------------------------
7079  034033     040            LNTBL:  .BYTE   000
7080  034034     100                    .BYTE   040
7081  034035     140                    .BYTE   100
7082  034036     200                    .BYTE   140
7083  034037     240                    .BYTE   200
7084  034040     300                    .BYTE   240
7085  034041     340                    .BYTE   300
7086                                    .BYTE   340
                                ;-----------------------------
```

```
                                                                          L10044:
                                                                          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.
                                ;////////////////////////////////////////////////////////////////////

034042                                  BGNHRD
034042  000015                                                          .WORD L10046-L$HARD/2
034044                                                          L$HARD::
034044                                  GPRMA   ADDRES,0,0,160020,177776,YES
034044  000031                                                          .WORD   T$CODE
034046  034076                                                          .WORD   ADDRES
034050  160020                                                          .WORD   T$LOLIM
034052  177776                                                          .WORD   T$HILIM
034054                                  GPRMA   VECTOR,2,0,0,674,YES
034054  001031                                                          .WORD   T$CODE
034056  034124                                                          .WORD   VECTOR
034060  000000                                                          .WORD   T$LOLIM
034062  000674                                                          .WORD   T$HILIM
034064                                  GPRMD   PRIRTY,4,0,7000,4,7,YES
034064  002032                                                          .WORD   T$CODE
034066  034155                                                          .WORD   PRIRTY
034070  007000                                                          .WORD   7000
034072  000004                                                          .WORD   T$LOLIM
034074  000007                                                          .WORD   T$HILIM

034076                                  ENDHRD
                                                                .EVEN
034076                                                          L10046:

                                .NLIST  BEX
034076  042504  044520  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
034124  042504  044526  042503  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
034155     104  053105  041511  PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
                                .LIST   BEX
                                .EVEN
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION


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

034206                                  BGNSFT
034206  000000                                                          .WORD L10047-L$SOFT/2
034210                                                          L$SOFT::
                                
034210                                  ENDSFT
                                
                                                                        .EVEN
034210                                                          L10047:
```

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

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

034210                          PATCH:
                        034310          .=.+100
034310  000240                  NOP
034312  000240                  NOP
034314  000240                  NOP
                                ;*********************************

                                .SBTTL  'ENDMOD'' STATEMENT

034316                          ENDMOD

                                .SBTTL  'LASTAD'' STATEMENT & END OF PROGRAM
034316                          LASTAD
                                                                        .EVEN
034316  000000                                                          .WORD   0
034320  000000                                                          .WORD   0
034322                          L$LAST::

                        000001  .END
```

```
ABC    = 000160           1261#
ADDRES  034076            7106    7125#
ADPAT   030170            5600    5786#
ADR    = 000020 G         1193#
AD.HIT  024276            4382    4417#
AD.OK   024272            4406    4411#
APA    = 000200           1317#
APAD   = 100000           1326#   5820
ASSEMB= 000010             949
BADDAT  002372            1712#
BDATA   002332            1695#   2314*   2317    2320    2594*   2697*   2698*   3128*   3388*   3389*   3406*   3407*   3781
                          3830    3900    3968    3993    7037*
BDRATE  002500            1759#   4353*
BIT0   = 000001 G         1166#   1211    1223    1265    1274    1293    1322    1348    1364    1382    1398    1540    1554
                          1576    1629    2728    2773    2818    2863    2908    2951    2998    3065    4319
BIT00  = 000001 G         1155#   1166
BIT01  = 000002 G         1154#   1165
BIT02  = 000004 G         1153#   1164
BIT03  = 000010 G         1152#   1163
BIT04  = 000020 G         1151#   1162
BIT05  = 000040 G         1150#   1161
BIT06  = 000100 G         1149#   1160
BIT07  = 000200 G         1148#   1159
BIT08  = 000400 G         1147#   1158
BIT09  = 001000 G         1146#   1157
BIT1   = 000002 G         1165#   1264    1292    1322    1348    1363    1381    1397    1539    1553    1575    1628    3022
BIT10  = 002000 G         1145#   1270    1298    1333    1334    1335
BIT11  = 004000 G         1144#   1269    1297    1330
BIT12  = 010000 G         1143#   1329
BIT13  = 020000 G         1142#   1328    1638
BIT14  = 040000 G         1141#   1327    1631    1637
BIT15  = 100000 G         1140#   1268    1296    1326    1625    1636
BIT2   = 000004 G         1164#   1263    1291    1322    1348    1362    1380    1396    1528    1553    1574
BIT3   = 000010 G         1163#   1262    1290    1321    1347    1361    1379    1395    1528    1553    1573
BIT4   = 000020 G         1162#   1210    1261    1320    1346    1360    1378    1394    1528    1552    1572
BIT5   = 000040 G         1161#   1261    1319    1345    1359    1377    1393    1523    1551    1571    3530
BIT6   = 000100 G         1160#   1222    1261    1318    1345    1358    1376    1392    1513    1551    1570    3550
BIT7   = 000200 G         1159#   1221    1229    1260    1289    1317    1345    1357    1375    1391    1513    1551    1564
                          1586    1634    3367    3368    3539
BIT8   = 000400 G         1158#   1272    1300    1331    1332    1333    1334
BIT9   = 001000 G         1157#   1271    1299    1332    1333    1632
BOE    = 000400 G         1197#
BRDTYP  002474            1757#   4351*
BSEL0   002422            1726#   2256    4389    4395
BSEL1   002424            1729#   2052*   2257
BSEL10  002442            1739#   2264
BSEL11  002444            1741#   2265
BSEL12  002446            1742#   2266
BSEL13  002450            1744#   2267
BSEL14  002452            1745#   2268
BSEL15  002454            1747#   2269
BSEL16  002456            1748#   2270
BSEL17  002460            1750#   2271    4341
BSEL2   002426            1730#   2059    2095*   2102    2142*   2149    2214*   2221    2258
BSEL3   002430            1732#   2259
BSEL4   002432            1733#   2260
```

```
BSEL5   002434          1735#   2261
BSEL6   002436          1736#   2117    2262
BSEL7   002440          1738#   2263
BSR0    002206          1662#   2256*   4015
BSR1    002210          1664#   2257*   4014
BSR10   002226          1677#   2264*   4049
BSR11   002230          1678#   2265*   4048
BSR12   002232          1679#   2266*   4047
BSR13   002234          1680#   2267*   4046
BSR14   002236          1681#   2268*   4066
BSR15   002240          1682#   2269*   4065
BSR16   002242          1683#   2270*   4064
BSR17   002244          1684#   2271*   4063
BSR2    002212          1666#   2258*   4013
BSR3    002214          1668#   2259*   4012
BSR4    002216          1670#   2260*   4032
BSR5    002220          1672#   2261*   4031
BSR6    002222          1674#   2262*   4030
BSR7    002224          1676#   2263*   4029
CARIER= 000100          1392#
CA1CTL= 000001          1554#
CA2CTL= 000016          1553#
CB1CTL= 000020          1552#
CB2CTL= 000340          1551#
CHKTSO  007042          3060#   3117    4524    4571    4607    4657    4704    4746
CHPTYP  002404          1717#   4268*
CKRACT  005622          2768#   3532    3586    3613    3642    3653
CKRDA   006122          2858#   3541    3552    3589    3605    3610    5675    5886
CKROR   006422          2946#   5560
CKRSA   006262          2903#
CKSEOM  006562          2993#   3592
CKTACT  005462          2723#   3228    3264    3316    3639    3650
CKTBMT  005762          2813#   3211    3225    3273    3280    3321    3328
CKUSTS  005356          2683#   3194    6492    6538    6633    6653    6755    6768
CRC0S = 000400          1331#
CRC16 = 001400          1332#   5065    5182
CTS   = 000010          1395#
C$AU  = 000052          949#    4461
C$AUTO= 000061          949#    4415
C$BRK = 000022          949#
C$BSEG= 000004          949#    5605
C$BSUB= 000002          949#    4497    4631    4801    4921    5062    5179    5598    5818    6422    6552    6919
C$CEFG= 000045          949#
C$CLCK= 000062          949#
C$CLEA= 000012          949#    4433
C$CLOS= 000035          949#
C$CLP1= 000006          949#
C$CVEC= 000036          949#    4404
C$DCLN= 000044          949#
C$DODU= 000051          949#    4409
C$DRPT= 000024          949#
C$DU  = 000053          949#    4448
C$EDIT= 000003          949#    1021
C$ERDF= 000055          949#    4576    4612    4709    4751    5688    5757    5901    5971    5986    6519    6530    7040
C$ERHR= 000056          949#
C$ERRO= 000060          949#    4505    4519    4528    4538    4553    4563    4599    4639    4652    4661    4671    4686
```

```
                        4696  4728  4738  4808  4827  4837  4847  4857  4867  4878  4889  4900  4909
                        4928  4947  4957  4967  4977  4987  4998  5009  5020  5029  5069  5091  5114
                        5125  5136  5147  5158  5167  5186  5209  5226  5237  5248  5259  5270  5279
                        5332  5349  5359  5369  5379  5388  5399  5409  5420  5435  5450  5465  5479
                        5489  5499  5509  5520  5530  5541  5552  5564  5611  5641  5650  5659  5668
                        5704  5717  5730  5743  5824  5852  5861  5870  5879  5917  5930  5943  5956
                        6034  6051  6061  6071  6081  6090  6101  6111  6122  6132  6143  6153  6164
                        6174  6185  6195  6206  6216  6227  6237  6248  6258  6269  6279  6290  6300
                        6311  6321  6332  6342  6353  6363  6374  6385  6396  6430  6447  6457  6467
                        6476  6487  6496  6507  6542  6560  6577  6587  6597  6606  6617  6628  6637
                        6648  6657  6689  6706  6716  6726  6735  6746  6759  6772  6801  6818  6828
                        6838  6847  6858  6873  6884  6928  6945  6955  6965  6979  6996  7007  7018
                        7054
C$ERSF= 000054           949#
C$ERSO= 000057           949#
C$ESCA= 000010           949#  4507  4521  4530  4540  4555  4565  4581  4601  4617  4641  4654  4663
                        4673  4688  4698  4714  4730  4740  4756  4810  4829  4839  4849  4859  4869
                        4880  4891  4902  4911  4930  4949  4959  4969  4979  4989  5000  5011  5022
                        5031  5071  5093  5116  5127  5138  5149  5160  5169  5188  5211  5228  5239
                        5250  5261  5272  5281  5334  5351  5361  5371  5381  5390  5401  5411  5422
                        5437  5452  5467  5481  5491  5501  5511  5522  5532  5543  5554  5566  5613
                        5643  5652  5661  5670  5693  5706  5719  5732  5745  5762  5826  5854  5863
                        5872  5881  5906  5919  5932  5945  5958  5976  5991  6036  6053  6063  6073
                        6083  6092  6103  6113  6124  6134  6145  6155  6166  6176  6187  6197  6208
                        6218  6229  6239  6250  6260  6271  6281  6292  6302  6313  6323  6334  6344
                        6355  6365  6376  6387  6398  6432  6449  6459  6469  6478  6489  6498  6509
                        6524  6535  6544  6562  6579  6589  6599  6608  6619  6630  6639  6650  6659
                        6691  6708  6718  6728  6737  6748  6761  6774  6803  6820  6830  6840  6849
                        6860  6875  6886  6930  6947  6957  6967  6981  6998  7009  7020  7045  7056
C$ESEG= 000005           949#  5773
C$ESUB= 000003           949#  4624  4763  4915  5035  5173  5285  5781  6001  6548  6663  7060
C$ETST= 000001           949#  4766  5038  5288  5570  5784  6004  6402  6666  6778  6890  7067
C$EXIT= 000032           949#
C$GETB= 000026           949#
C$GETW= 000027           949#
C$GMAN= 000043           949#
C$GPHR= 000042           949#  4327
C$GPLO= 000030           949#
C$GPRI= 000040           949#
C$INIT= 000011           949#  4360
C$INLP= 000020           949#
C$MANI= 000050           949#
C$MEM = 000031           949#
C$MSG = 000023           949#  3762  3796  3840  3874  3909  3927  3952  3978
C$OPEN= 000034           949#
C$PNTB= 000014           949#  3776  3786  3792  3809  3815  3825  3835  3853  3859  3869  3888  3895
                        3905  3922  3940  3947  3965  3973  4118
C$PNTF= 000017           949#
C$PNTS= 000016           949#
C$PNTX= 000015           949#  3739  3755  4009  4019  4026  4036  4043  4053  4060  4070  4085  4095
                        4102  4112  4134  4144  4151  4161  4168  4178  4185  4195  4211  4221  4228
                        4238
C$QIO = 000377           949#
C$RDBU= 000007           949#
C$REFG= 000047           949#  4285  4292  4299  4306
C$RESE= 000033           949#  4356  4445
```

```
C$REVI= 000003            949#  1020
C$RFLA= 000021            949#
C$RPT = 000025            949#
C$SEFG= 000046            949#
C$SPRI= 000041            949#
C$SVEC= 000037            949#  4385
C$TPRI= 000013            949#
DDCMP = 040000           1327#  4501    4635    4804    4924    5065    5182
DEVMAP  002410           1719#  4315*   4333*
DEVPTR  002412           1720#  4319*   4331*   4333    4334*
DFPTBL  002154 G         1099#
DIAGMC= 000000            949
DOTBMT= 000007           1239#
DTR   = 000020           1378#  2576    2579    2652    3167    3170    6753
DTRL  = 000000           1383#
D.BUG = 000000           1640#
EF.COM= 000036 G         1173#  4305
EF.NEW= 000035 G         1174#  4298
EF.PUR= 000034 G         1175#
EF.RES= 000037 G         1172#  4291
EF.STA= 000040 G         1171#  4284
EIAV35= 000002           1628#
EM1     014007           3694#
EM100   016422           3074   3694#
EM101   016442           3086   3694#
EM102   016466           3694#  5690    5903
EM103   016530           3694#  5759    5973
EM104   016607           3694#  5988
EM105   016665           3694#  7042
EM106   016735           3134   3694#
EM13    014236           3694#
EM14    014265           3454   3694#
EM15    014301           3694#
EM16    014327           3444   3694#
EM2     014046           2600   3694#
EM25    014347           2328   3694#
EM26    014375           3694#
EM27    014427           3694#
EM28    014460           3018   3513    3694#
EM29    014501           3007   3523    3694#
EM3     014115           2067   3694#
EM30    014516           3042   3490    3694#
EM31    014537           3031   3500    3694#   6532
EM32    014554           3694#
EM33    014603           3694#
EM34    014626           3395   3694#
EM35    014654           3421   3694#
EM36    014675           3431   3694#
EM39    014712           3467   3694#
EM4     014141           2111   2158    2229    3694#
EM40    014734           3477   3694#   6521
EM41    014752           3694#
EM42    014773           3694#
EM43    015010           3694#
EM44    015035           3694#
EM45    015062           3694#
```

```
EM47    015107          3694#
EM48    015142          3694#   4578
EM49    015175          3694#   4614
EM5     014156          3694#
EM50    015234          3694#   4753
EM51    015270          3694#   4711
EM54    015330          3382    3694#
EM6     014206          3694#
EM60    015352          3694#
EM65    015366          3694#
EM66    015406          3694#
EM67    015425          3694#
EM68    015500          2704    3694#
EM69    015527          2737    3694#
EM70    015545          2748    3694#
EM71    015567          2782    3694#
EM72    015605          2793    3694#
EM73    015627          2827    3694#
EM74    015644          2838    3694#
EM75    015665          2872    3694#
EM76    015701          2883    3694#
EM77    015721          2917    3694#
EM78    015735          2928    3694#
EM79    015755          3694#
EM80    016011          3694#
EM81    016031          3694#
EM82    016055          3694#
EM83    016112          3694#
EM84    016141          3694#
EM85    016156          3694#
EM86    016172          3694#
EM87    016213          3694#
EM88    016233          3694#
EM89    016256          3694#
EM90    016300          2960    3694#
EM91    016331          2971    3694#
EM92    016366          3694#
ENDEMB  011564          3694#   3789    4115
ENDIT   024156          4308    4357#
ENDPAT  003031          1986#
ENDTRN  011456          3637#   4905    5025    5163    5275
ERRBLK  002204  G       1656#   2068*   2112*   2159*   2230*   2329*   2601*   2705*   2738*   2749*   2783*   2794*   2828*
                        2839*   2873*   2884*   2918*   2929*   2961*   2972*   3008*   3019*   3032*   3043*   3075*   3087*
                        3135*   3383*   3396*   3422*   3432*   3445*   3455*   3468*   3478*   3491*   3501*   3514*   3524*
ERRFLG  002354          1705#
ERRMSG  002202  G       1655#   2067*   2111*   2158*   2229*   2328*   2600*   2704*   2737*   2748*   2782*   2793*   2827*
                        2838*   2872*   2883*   2917*   2928*   2960*   2971*   3007*   3018*   3031*   3042*   3074*   3086*
                        3134*   3382*   3395*   3421*   3431*   3444*   3454*   3467*   3477*   3490*   3500*   3513*   3523*
ERRNBR  002200  G       1654#   2066*   2110*   2157*   2228*   2327*   2599*   2703*   2736*   2747*   2781*   2792*   2826*
                        2837*   2871*   2882*   2916*   2927*   2959*   2970*   3006*   3017*   3030*   3041*   3073*   3085*
                        3133*   3381*   3394*   3420*   3430*   3443*   3453*   3466*   3476*   3489*   3499*   3512*   3522*
ERROR1  002402          1716#   4269*
ERRTYP  002176  G       1653#   2065*   2109*   2156*   2227*   2326*   2598*   2702*   2735*   2746*   2780*   2791*   2825*
                        2836*   2870*   2881*   2915*   2926*   2958*   2969*   3005*   3016*   3029*   3040*   3072*   3084*
                        3132*   3380*   3393*   3419*   3429*   3442*   3452*   3465*   3475*   3488*   3498*   3511*   3521*
ERR10   021540  G       2601    2705    3396    3802#
```

```
ERR11$  023122           4127#
ERR12   021714 G         2738    2749    2783    2794    2828    2839    2873    2884    2918    2929    2961    2972    3008
                         3019    3032    3043    3075    3087    3383    3422    3432    3445    3455    3468    3478    3491
                         3501    3514    3524    3846#   4579    4615    4712    4754    6522    6533
ERR12$  023454           3837    3871    3924    3949    3975    4204#
ERR13   022030 G         3135    3881#
ERR20   022146 G         3915#   5760    5974    5989
ERR21   022204 G         3933#   5691    5904
ERR22   022264 G         3958#   7043
ERR4    021274 G         2068    2112    2159    2230    3728#
ERR4$   022376           4003#
ERR5$   022724           3759    4078#
ERR7A   021420 G         2329    3767#
EVL     = 000004 G       1191#
EVRC    = 002400         1334#   4635    4924
EXADD   = 000020         1346#
EXCON   = 000010         1347#
EXECUT  = 000005         1238#
ESEND   = 002100          949#
ESLOAD  = 000035          949#   1044
FLGCA1  = 000002         1575#
FLGCA2  = 000001         1576#
FLGCB1  = 000020         1572#
FLGCB2  = 000010         1573#
FLGIRQ  = 000200         1564#
FLGSR   = 000004         1574#
FLGT1   = 000100         1570#
FLGT2   = 000040         1571#
FMT10   012226           3694#
FMT10A  012302           3694#
FMT11   012323           3694#   4092    4109
FMT12   012342           3694#
FMT13   012351           3694#
FMT14   012417           3694#
FMT15   012434           3694#   3773
FMT15A  012466           3694#
FMT16   012540           3694#
FMT16A  012563           3694#
FMT17   012625           3694#
FMT17A  012704           3694#
FMT17B  012765           3694#
FMT17C  013040           3694#
FMT19   013120           3694#
FMT2    011574           3694#
FMT21   013151           3694#   3806    3850    3885    3919    3937    3962
FMT22   013161           3694#   3812    3856
FMT23   013203           3694#   3832    3902
FMT24   013262           3694#   4131    4208
FMT25   013275           3694#   4141    4175    4218
FMT26   013325           3694#   4158    4192    4235
FMT27   013360           3694#   3822    3866
FMT28   013367           3694#
FMT29   013423           3694#   4148    4165    4182    4225
FMT3    011631           3694#   3783
FMT30   013430           3694#
FMT31   013465           3694#
```

CVDMDA.P11 10-DEC-80 09:15 CROSS REFERENCE TABLE -- USER SYMBOLS

```
FMT32   013533          3694#
FMT39   013565          3694#
FMT4    011715          3694#   4006    4082
FMT4A   011753          3694#   4016    4050
FMT4B   012006          3694#   4023    4040    4057    4099
FMT4C   012013          3694#   4033    4067
FMT40   013616          3694#   3892
FMT5    012046          3694#   3736
FMT5A   012111          3694#   3752
FMT50   013662          3694#   3944
FMT51   013725          3694#   3970
FMT7    012176          3694#
FRSTIM  002374          1713#   4271    4280*
F$AU  = 000015           949#   4458    4460
F$AUTO= 000020           949#   4378    4414
F$BGN = 000040           949#    952    3728    3767    3802    3846    3881    3915    3933    3958    4251    4264    4378
                        4428    4442    4458    4491    4496    4507    4521    4530    4540    4555    4565    4581    4601
                        4617    4623    4630    4641    4654    4663    4673    4688    4698    4714    4730    4740    4756
                        4762    4765    4795    4800    4810    4829    4839    4849    4859    4869    4880    4891    4902
                        4911    4914    4920    4930    4949    4959    4969    4979    4989    5000    5011    5022    5031
                        5034    5037    5056    5061    5071    5093    5116    5127    5138    5149    5160    5169    5172
                        5178    5188    5211    5228    5239    5250    5261    5272    5281    5284    5287    5324    5334
                        5351    5361    5371    5381    5390    5401    5411    5422    5437    5452    5467    5481    5491
                        5501    5511    5522    5532    5543    5554    5566    5569    5592    5597    5605    5613    5643
                        5652    5661    5670    5693    5706    5719    5732    5745    5762    5780    5783    5812    5817
                        5826    5854    5863    5872    5881    5906    5919    5932    5945    5958    5976    5991    6000
                        6003    6026    6036    6053    6063    6073    6083    6092    6103    6113    6124    6134    6145
                        6155    6166    6176    6187    6197    6208    6218    6229    6239    6250    6260    6271    6281
                        6292    6302    6313    6323    6334    6344    6355    6365    6376    6387    6398    6401    6418
                        6421    6432    6449    6459    6469    6478    6489    6498    6509    6524    6535    6544    6547
                        6551    6562    6579    6589    6599    6608    6619    6630    6639    6650    6659    6662    6665
                        6680    6691    6708    6718    6728    6737    6748    6761    6774    6777    6793    6803    6820
                        6830    6840    6849    6860    6875    6886    6889    6911    6918    6930    6947    6957    6967
                        6981    6998    7009    7020    7045    7056    7059    7066    7101    7140    7158
F$CLEA= 000007           949#   4428    4432
F$DU  = 000016           949#   4442    4447
F$END = 000041           949#    952    3763    3797    3841    3875    3910    3928    3953    3979    4361    4416    4434
                        4449    4462    4491    4496    4507    4521    4530    4540    4555    4565    4581    4601    4617
                        4623    4625    4630    4641    4654    4663    4673    4688    4698    4714    4730    4740    4756
                        4762    4764    4765    4767    4795    4800    4810    4829    4839    4849    4859    4869    4880
                        4891    4902    4911    4914    4916    4920    4930    4949    4959    4969    4979    4989    5000
                        5011    5022    5031    5034    5036    5037    5039    5056    5061    5071    5093    5116    5127
                        5138    5149    5160    5169    5172    5174    5178    5188    5211    5228    5239    5250    5261
                        5272    5281    5284    5286    5287    5289    5324    5334    5351    5361    5371    5381    5390
                        5401    5411    5422    5437    5452    5467    5481    5491    5501    5511    5522    5532    5543
                        5554    5566    5569    5571    5592    5597    5613    5643    5652    5661    5670    5693    5706
                        5719    5732    5745    5762    5774    5780    5782    5783    5785    5812    5817    5826    5854
                        5863    5872    5881    5906    5919    5932    5945    5958    5976    5991    6000    6002    6003
                        6005    6026    6036    6053    6063    6073    6083    6092    6103    6113    6124    6134    6145
                        6155    6166    6176    6187    6197    6208    6218    6229    6239    6250    6260    6271    6281
                        6292    6302    6313    6323    6334    6344    6355    6365    6376    6387    6398    6401    6403
                        6418    6421    6432    6449    6459    6469    6478    6489    6498    6509    6524    6535    6544
                        6547    6549    6551    6562    6579    6589    6599    6608    6619    6630    6639    6650    6659
                        6662    6664    6665    6667    6680    6691    6708    6718    6728    6737    6748    6761    6774
                        6777    6779    6793    6803    6820    6830    6840    6849    6860    6875    6886    6889    6891
                        6911    6918    6930    6947    6957    6967    6981    6998    7009    7020    7045    7056    7059
```

CVDMDA.P11 10-DEC-80 09:15 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                        7061    7066    7068    7124    7146    7158
F$HARD= 000004           949#    7101    7122
F$HW  = 000013           949#    1097    1113
F$INIT= 000006           949#    4264    4359
F$JMP = 000050           949#
F$MOD = 000000           949#     952    7158
F$MSG = 000011           949#    3728    3761    3767    3795    3802    3839    3846    3873    3881    3908    3915    3926
                        3933    3951    3958    3977
F$PROT= 000021           949#    4251    4256
F$PWR = 000017           949#
F$RPT = 000012           949#
F$SEG = 000003           949#    5605    5772
F$SOFT= 000005           949#    7140    7144
F$SRV = 000010           949#
F$SUB = 000002           949#    4497    4623    4631    4762    4801    4914    4921    5034    5062    5172    5179    5284
                        5598    5780    5818    6000    6422    6547    6552    6662    6919    7059
F$SW  = 000014           949#    1122    1127
F$TEST= 000001           949#    4492    4765    4796    5037    5057    5287    5325    5569    5593    5783    5813    6003
                        6027    6401    6419    6665    6681    6777    6794    6889    6912    7066
GDATA   002330          1694#   2062*   2106*   2153*   2224*   2320    2592*   2593*   2696*   3127*   3386*   3387*   3404*
                        3405*   3729*   3731    3751    3782    3831    3901    3969    3992    7036*
GETBSR  003772          2256#
GETPRM  024030          4309    4323#   4332
GETURS  004326          2352#   4204
GETVRS  004426          2383#   4127
GETWSR  004134          2061    2105    2152    2223    2277#
GOODAT  002370          1711#
G$CNTO= 000200           949#
G$DELM= 000372           949#
G$DISP= 000003           949#
G$EXCP= 000400           949#
G$HILI= 000002           949#
G$LOLI= 000001           949#
G$NO  = 000000           949#
G$OFFS= 000400           949#    7105    7110    7115
G$OFSI= 000376           949#    7105    7110    7115
G$PRMA= 000001           949#    7105    7110
G$PRMD= 000002           949#    7115
G$PRML= 000000           949#
G$RADA= 000140           949#
G$RADB= 000000           949#
G$RADD= 000040           949#
G$RADL= 000120           949#
G$RADO= 000020           949#    7105    7110    7115
G$XFER= 000004           949#
G$YES = 000010           949#    7105    7110    7115
HDX   = 000004          1380#
HELP  = 000000           940#     974    1062    1088    2031
HOE   = 100000 G        1204#
IBE   = 010000 G        1201#
IDLE  = 000010          1321#
IDLES = 004000          1330#    5065    5182    6556    6685
IDU   = 000040 G        1194#
IEI   = 000001          1211#
IEO   = 000020          1210#
IER   = 020000 G        1202#
```

```
ILOOP   027506          5606#  5770
INIDMV  005344          2674#  4498   4632   4802   4922   5063   5180   5325   5593   5813   6027   6423   6553
                        6682   6794   6912
INITRN  007324          3163#  4500   4634   4803   4923   5064   5181   5327   5606   5819   6029   6425   6555
                        6684   6796   6923
INITT1  004502          2421#
INITT2  004702          2508#
INTFLG  002352          1703#
INTGRL= 000001          1629#
INTSC = 000200          1586#
ISR   = 000100 G        1195#
IXE   = 004000 G        1200#
I$AU  = 000041           949#  4458#  4462#
I$AUTO= 000041           949#  4378#  4416#
I$CLN = 000041           949#  4428#  4434#
I$DU  = 000041           949#  4442#  4449#
I$HRD = 000041          7101#  7124#
I$INIT= 000041           949#  4264#  4361#
I$MOD = 000041           949#   952#  7158#
I$MSG = 000041           949#  3728#  3763#  3767#  3797#  3802#  3841#  3846#  3875#  3881#  3910#  3915#  3928#
                        3933#  3953#  3958#  3979#
I$PROT= 000040           949#  4251#
I$PTAB= 000041           949#
I$PWR = 000041           949#
I$RPT = 000041           949#
I$SEG = 000041           949#  4491   4496   4630   4795   4800   4920   5056   5061   5178   5324   5592   5597
                        5605#  5613   5643   5652   5661   5670   5693   5706   5719   5732   5745   5762   5774#
                        5812   5817   6026   6418   6421   6551   6680   6793   6911   6918
I$SETU= 000041           949#
I$SFT = 000041          7140#  7146#
I$SRV = 000041           949#
I$SUB = 000041           949#  4491   4496#  4507   4521   4530   4540   4555   4565   4581   4601   4617   4623#
                        4625#  4630#  4641   4654   4663   4673   4688   4698   4714   4730   4740   4756   4762#
                        4764#  4795   4800#  4810   4829   4839   4849   4859   4869   4880   4891   4902   4911
                        4914#  4916#  4920#  4930   4949   4959   4969   4979   4989   5000   5011   5022   5031
                        5034#  5036#  5056   5061#  5172#  5174#  5178#  5284#  5286#  5324   5592   5597#  5780#
                        5782#  5812   5817#  5826   5854   5863   5872   5881   5906   5919   5932   5945   5958
                        5976   5991   6000#  6002#  6026   6418   6421#  6432   6449   6459   6469   6478   6489
                        6498   6509   6524   6535   6544   6547#  6549#  6551#  6562   6579   6589   6599   6608
                        6619   6630   6639   6650   6659   6662#  6664#  6680   6793   6911   6918#  6930   6947
                        6957   6967   6981   6998   7009   7020   7045   7056   7059#  7061#
I$TST = 000041           949#  4491#  4496   4630   4765#  4767#  4795#  4800   4920   5037#  5039#  5056#  5061
                        5071   5093   5116   5127   5138   5149   5160   5169   5178   5188   5211   5228   5239
                        5250   5261   5272   5281   5287#  5289#  5324#  5334   5351   5361   5371   5381   5390
                        5401   5411   5422   5437   5452   5467   5481   5491   5501   5511   5522   5532   5543
                        5554   5566   5569#  5571#  5592#  5597   5783#  5785#  5812#  5817   6003#  6005#  6026#
                        6036   6053   6063   6073   6083   6092   6103   6113   6124   6134   6145   6155   6166
                        6176   6187   6197   6208   6218   6229   6239   6250   6260   6271   6281   6292   6302
                        6313   6323   6334   6344   6355   6365   6376   6387   6398   6401#  6403#  6418#  6421
                        6551   6665#  6667#  6680#  6691   6708   6718   6728   6737   6748   6761   6774   6777#
                        6779#  6793#  6803   6820   6830   6840   6849   6860   6875   6886   6889#  6891#  6911#
                        6918   7066#  7068#
J$JMP = 000167           949#
LNTBL   034032          6921   7078#
LOADAT  002366          1710#
LOE   = 040000 G        1203#
```

```
LOGDEV  002340          1698#   4317*  4324*  4326   4408
LOOP    030204          5816#   5998
LOT   = 000010 G        1192#
LUSWI1  002470          1755#   4349*
LUSWI2  002472          1756#   4350*
LU1MOD  002000 G         952#
LSACP   002110 G        1051#
LSAPT   002036 G        1009#
LSAU    024312 G        1036    4458#
LSAUT   002070 G        1035#
LSAUTO  024160 G        1052    4378#
LSCCP   002106 G        1049#
LSCLEA  024304 G        1050    4428#
LSCO    002032 G        1005#
LSDEPO  002011 G         987#
LSDESC  003252 G        1042    2019#
LSDESP  002076 G        1041#
LSDEVP  002060 G        1027#
LSDISP  002124 G        1012    1075#
LSDLY   002116 G        1057#
LSDTP   002040 G        1011#
LSDTYP  002034 G        1007#
LSDU    024306 G        1038    4442#
LSDUT   002072 G        1037#
LSDVTY  003232 G        1028    2007#
LSEF    002052 G        1022#
LSENVI  002044 G        1015#
LSERRT  002176 G        1046    1652#
LSETP   002102 G        1045#
LSEXP1  002046 G        1017#
LSEXP4  002064 G        1031#
LSEXP5  002066 G        1033#
LSHARD  034044 G         994    7101   7102#
LSHIME  002120 G        1059#
LSHPCP  002016 G         993#
LSHPTP  002022 G         997#
LSHW    002154 G         998    1097   1098#
LSICP   002104 G        1047#
LSINIT  023644 G        1048    4264#
LSLADP  002026 G        1001#
LSLAST  034322 G        1002    7164#
LSLOAD  002100 G        1043#
LSLUN   002074 G        1039#
LSMREV  002050 G        1019#
LSNAME  002000 G         976#
LSPRIO  002042 G        1013#
LSPROT  023636 G        1054    4251#
LSPRT   002112 G        1053#
LSREPP  002062 G        1029#
LSREV   002010 G         985#
LSSOFT  034210 G        7140    7141#
LSSPC   002056 G        1[illegible]#
LSSPCP  002020 G         995#
LSSPTP  002024 G         999#
LSSTA   002030 G        1003#
LSSW    002176 G        1122    1123#
```

```
LSTEST  002114 G        1055#
LSTIML  002014 G         991#
LSUNIT  002012 G         989#
L10000  002174           1097    1113#
L10001  002176           1122    1127#
L10002  021416           3761#
L10003  021536           3795#
L10004  021712           3839#
L10005  022026           3873#
L10006  022144           3908#
L10007  022202           3926#
L10010  022262           3951#
L10011  022350           3977#
L10013  024156           4359#
L10014  024274           4414#
L10015  024304           4432#
L10016  024310           4447#
L10017  024312           4460#
L10020  025220           4765#
L10021  024650           4508    4522    4531    4541    4556    4566    4582    4602    4618    4623#
L10022  025216           4642    4655    4664    4674    4689    4699    4715    4731    4741    4757    4762#
L10023  026032           5037#
L10024  025524           4811    4830    4840    4850    4860    4870    4881    4892    4903    4912    4914#
L10025  026030           4931    4950    4960    4970    4980    4990    5001    5012    5023    5032    5034#
L10026  026624           5072    5094    5117    5128    5139    5150    5161    5170    5189    5212    5229    5240    5251
                         5262    5273    5282    5287#
L10027  026336           5172#
L10030  026622           5284#
L10031  027462           5335    5352    5362    5372    5382    5391    5402    5412    5423    5438    5453    5468    5482
                         5492    5502    5512    5523    5533    5544    5555    5567    5569#
L10032  030166           5783#
L10033  030164           5780#
L10034  030672           6003#
L10035  030670           5827    5855    5864    5873    5882    5907    5920    5933    5946    5959    5977    5992    6000#
L10036  032036           6037    6054    6064    6074    6084    6093    6104    6114    6125    6135    6146    6156    6167
                         6177    6188    6198    6209    6219    6230    6240    6251    6261    6272    6282    6293    6303
                         6314    6324    6335    6345    6356    6366    6377    6388    6399    6401#
L10037  032656           6665#
L10040  032364           6433    6450    6460    6470    6479    6490    6499    6510    6525    6536    6545    6547#
L10041  032654           6563    6580    6590    6600    6609    6620    6631    6640    6651    6660    6662#
L10042  033120           6692    6709    6719    6729    6738    6749    6762    6775    6777#
L10043  033362           6804    6821    6831    6841    6850    6861    6876    6887    6889#
L10044  034020           7066#
L10045  034002           6931    6948    6958    6968    6982    6999    7010    7021    7046    7057    7059#
L10046  034076           7101    7123#
L10047  034210           7140    7145#
MCLR    = 000100         1222#   1224    2052    2062
MDMRDY= 000040           1393#
MLWRI   003670           2189    2214#
MPCSR   002422           1728#   3804    3848    3883    3917    3935    3960    4337
MPIVEC  002462           1752#   4345*
MPOVEC  002464           1753#   4346*   4347*
MPRIOR  002466           1754#   4348*
MRDY    = 000200         1229#   2059    2102    2149    2221
MREQ    = 000001         1223#   1224    2052    2062
MSTCLR  003320           2052#   2674
```

CVDMDA.P11 10-DEC-80 09:15 CROSS REFERENCE TABLE -- USER SYMBOLS

```
NCRACT= 020000          1638#  5447   5462   5727   5740   5940   5953   6382   6393   6504   6625   6645   6870
                        6881   7004   7015   7051
NCTBMT= 000200          1634#  3270   5476   5486   5496   5549   5647   5656   5858   5867
NEWLIN  011571          3694#
NEWST   024010          4301   4316#
NFCRDA= 040000          1637#  5462   5740   5953   6393   6504   6645   6881   7051
NOCHK = 003400          1333#  5328   5607   5820   6030   6426   6556   6685   6797   6924
NOCRDA= 100000          1636#  4875   4886   4897   4995   5006   5017   5111   5122   5133   5144   5155   5223
                        5234   5245   5256   5267   5538   5701   5714   5727   5914   5927   5940   6993   7004
                        7015
NOLOOP= 001000          1632#
NOLP    030670          5997   5999#
NORXEN= 040000          1631#
NULCLK= 000200          1375#
NWSAR   027512          5600*  5607#  5681   5750
OLOOP   027472          5596#  5778
OVRC  = 002000          1335#  4501   4804
O$APTS= 000000           949#  1003
O$AU  = 000001           949#   973#  1035
O$BGNR= 000000           949#  1029
O$BGNS= 000000           949#   995
O$DU  = 000001           949#   973#  1037
O$ERRT= 000001           949#   973#  1045
O$GNSW= 000000           949#   999
O$POIN= 000001           949#   973#  1061
O$SETU= 000000           949#   989   7162
PALENB= 000001          1540#
PATCH   034210          7148#
PATE    002602          1808#
PATF    002612          1819#  2582
PATG    002622          1830#
PATI    002676          1880#
PATJ    002737          1916#
PATK    002747          1927#
PATL    002770          1947#
PATQ    003011          1967#  4543   4584   4620   4676   4717   4759
PATQB   003021          1977#  4544   4677
PATX1   002651          1856#  5618
PBLENB= 000002          1539#
PCR   = 120407          1341#  3183   4833   4953   6971
PCSARH= 120405          1313#  3177
PCSARL= 120404          1306#  3173
PNT   = 001000 G        1198#
PRESET= 000001          1382#  2576   3167
PRI   = 002000 G        1199#
PRIOR   002346          1701#
PRIRTY  034155          7116   7125#
PRI00 = 000000 G        1187#
PRI01 = 000040 G        1186#
PRI02 = 000100 G        1185#
PRI03 = 000140 G        1184#
PRI04 = 000200 G        1183#
PRI05 = 000240 G        1182#
PRI06 = 000300 G        1181#
PRI07 = 000340 G        1180#
PROTO = 000100          1318#
```

```
PSTACK  002344          1700#  4266*
RABGA = 000004          1263#  3458   3460   3470   6515
RAMADR= 001000          1626#
RCVBUF  003032          1990#
RCVDAT= 000002          1397#
RCV1ST  011310          3580#  5384   6086   6472   6602   6731   6843
RDA   = 000200          1357#  2865   2876   3601   6493   6634
RDSRH = 120401          1257#  2949   2996   3358   6513   7024
RDSRL = 120400          1251#  3361
RF D    003422          2094#  2315   2356   2361   2386
READI   003534          2140#  2443   2530   2584   2689   2725   2770   2815   2860   2905   2948   2995   3062
                        3217   3357   3360   3371   3598   3512   7023
REDBYT  002362          1708#
REDDAT  002522          1773#
REDLOC= 000001          1234#  2095   2106   2142   2153
REDPAG= 000003          1236#
REGNUM  002342          1699#  2590*  2591*  2695*  2724*  2769*  2814*  2859*  2904*  2947*  2994*  3061*  3376*
                        3385*  3403*  3408*  3768   3817   3861   3891
REG0    002532          1776#
REG1    002534          1777#
REG2    002536          1778#
REG3    002540          1779#
REG4    002542          1780#
REG5    002544          1781#
REG6    002546          1782#
REG7    002550          1783#
REOM  = 000002          1264#  3024   3035   3481   3483   3493   6526
RERCHK= 000001          1274#  3409   4874   4885   4896   4994   5005   5016   5154   5266
RERR  = 000200          1260#  3412   3414   3424
RETADR  002360          1707#
RING  = 000200          1391#
ROR   = 000010          1262#  2953   2964   3435   3437   3447
RSA   = 000020          1360#  2910   2921   6634
RSOM  = 000001          1265#  3000   3011   3504   3506   3516
RSTCHK  005052          2570#
RTSND = 000010          1379#  2576   2579   2652   3167   3170   6753
RUN   = 000200          1221#  1224   2052   2062
RXABGA= 002000          1270#  6879
RXACT = 000040          1359#  2775   2786   6493
RXCHAR  010034          3354#  4872   4883   4894   4992   5003   5014   5108   5119   5130   5141   5152   5220
                        5231   5242   5253   5264   5393   5414   5429   5444   5459   5514   5535   5698   5711
                        5724   5737   5911   5924   5937   5950   6095   6116   6137   6158   6179   6200   6221
                        6242   6263   6284   6305   6326   6347   6368   6379   6390   6481   6501   6611   6622
                        6642   6740   6852   6867   6878   6990   7001   7012   7048
RXDL  = 000007          1348#  3365   4805   4925
RXEN  = 000100          1376#  3203   3233
RXEOM = 001000          1271#  5460   5738   5951   6391   6643   6879
RXERR = 100000          1268#  4895   5015   5153
RXOR  = 004000          1269#
RXSOM = 000400          1272#  5394   5515   5699   5912   6096   6482   6612   6741   6853   6991
SAVE4   002376          1714#  4273*  4277
SAVE6   002400          1715#  4274*  4278
SAVLEN  002406          1718#  3180*  3181*  3365   4270*
SCRACH  002336          1697#
SECAD = 000020          1320#
SECADR= 010000          1329#  5607   5820
```

CVDMDA.P11 10-DEC-80 09:15 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SEL0     002422          1727#  2277
SEL10    002442          1740#  2281
SEL12    002446          1743#  2282
SEL14    002452          1746#  2283
SEL16    002456          1749#  2284
SEL2     002426          1731#  2278
SEL4     002432          1734#  2094*  2141*  2187*  2212*  2279
SEL6     002436          1737#  2164   2188*  2213*  2280
SERIAL   007202          3105#  4533   4558   4594   4666   4691   4733
SETVIA   005252          2640#  2675
SFPTBL   002176 G        1124#
SFR    = 000001          1364#
SPEED  = 000020          1394#
SRMODE= 000034           1528#
STALL    004324          2339#
STARES   002416          1722#  4312*  4318*
STARST   024000          4287   4294   4311#
STEPLU   011540          3114   3215   3276   3324   3545   3595   3648   3671#  4568   4604   4701   4743   5105
                         5217   5672   5883   6987
STRIP  = 000040          1319#
STRIPS= 020000           1328#  5065   5182   6797
STRTML= 000301           1224#
STUREG   004216          2306#
SUBRPC   002350          1702#  4267*
SVCGBL= 000000            949#   952    959#   976    985    987    989    991    993    995    997    999   1001
                         1003   1005   1007   1009   1011   1013   1015   1017   1019   1022   1025   1027   1029
                         1031   1033   1035   1037   1039   1041   1043   1045   1047   1049   1051   1053   1055
                         1057   1059   1075   1098   1099   1123   1124   1652   2007   2019   3728   3767   3802
                         3846   3881   3915   3933   3958   4251   4264   4378   4428   4442   4458   7102   7141
                         7164#  7165
SVCINS- 000001            949#   956#   977    978    979    980    981    982    983    984    986    988    990
                          992    994    996    998   1000   1002   1004   1006   1008   1010   1012   1014   1016
                         1018   1020   1021   1023   1024   1026   1028   1030   1032   1034   1036   1038   1040
                         1042   1044   1046   1048   1050   1052   1054   1056   1058   1060   1074   1076   1077
                         1078   1079   1080   1081   1082   1083   1084   1085   1086   1097   1122   2008   2011
                         2020   2027   2065   2066   2067   2068   2109   2110   2111   2112   2156   2157   2158
                         2159   2227   2228   2229   2230   2326   2327   2328   2329   2598   2599   2600   2601
                         2702   2703   2704   2705   2735   2736   2737   2738   2746   2747   2748   2749   2780
                         2781   2782   2783   2791   2792   2793   2794   2825   2826   2827   2828   2836   2837
                         2838   2839   2870   2871   2872   2873   2881   2882   2883   2884   2915   2916   2917
                         2918   2926   2927   2928   2929   2958   2959   2960   2961   2969   2970   2971   2972
                         3005   3006   3007   3008   3016   3017   3018   3019   3029   3030   3031   3032   3040
                         3041   3042   3043   3072   3073   3074   3075   3084   3085   3086   3087   3132   3133
                         3134   3135   3380   3381   3382   3383   3393   3394   3395   3396   3419   3420   3421
                         3422   3429   3430   3431   3432   3442   3443   3444   3445   3452   3453   3454   3455
                         3465   3466   3467   3468   3475   3476   3477   3478   3488   3489   3490   3491   3498
                         3499   3500   3501   3511   3512   3513   3514   3521   3522   3523   3524   3735   3736
                         3737   3738   3739   3740   3750   3751   3752   3753   3754   3755   3756   3762   3771
                         3772   3773   3774   3775   3776   3777   3780   3781   3782   3783   3784   3785   3786
                         3787   3789   3790   3791   3792   3793   3796   3804   3805   3806   3807   3808   3809
                         3810   3812   3813   3814   3815   3816   3820   3821   3822   3823   3824   3825   3826
                         3829   3830   3831   3832   3833   3834   3835   3836   3840   3848   3849   3850   3851
                         3852   3853   3854   3856   3857   3858   3859   3860   3864   3865   3866   3867   3868
                         3869   3870   3874   3883   3884   3885   3886   3887   3888   3889   3891   3892   3893
                         3894   3895   3896   3899   3900   3901   3902   3903   3904   3905   3906   3909   3917
                         3918   3919   3920   3921   3922   3923   3927   3935   3936   3937   3938   3939   3940
```

3941	3943	3944	3945	3946	3947	3948	3952	3960	3961	3962	3963	3964
3965	3966	3968	3969	3970	3971	3972	3973	3974	3978	4004	4005	4006
4007	4008	4009	4010	4012	4013	4014	4015	4016	4017	4018	4019	4020
4022	4023	4024	4025	4026	4027	4029	4030	4031	4032	4033	4034	4035
4036	4037	4039	4040	4041	4042	4043	4044	4046	4047	4048	4049	4050
4051	4052	4053	4054	4056	4057	4058	4059	4060	4061	4063	4064	4065
4066	4067	4068	4069	4070	4071	4080	4081	4082	4083	4084	4085	4086
4088	4089	4090	4091	4092	4093	4094	4095	4096	4098	4099	4100	4101
4102	4103	4105	4106	4107	4108	4109	4110	4111	4112	4113	4115	4116
4117	4118	4119	4129	4130	4131	4132	4133	4134	4135	4137	4138	4139
4140	4141	4142	4143	4144	4145	4147	4148	4149	4150	4151	4152	4154
4155	4156	4157	4158	4159	4160	4161	4162	4164	4165	4166	4167	4168
4169	4171	4172	4173	4174	4175	4176	4177	4178	4179	4181	4182	4183
4184	4185	4186	4188	4189	4190	4191	4192	4193	4194	4195	4196	4206
4207	4208	4209	4210	4211	4212	4214	4215	4216	4217	4218	4219	4220
4221	4222	4224	4225	4226	4227	4228	4229	4231	4232	4233	4234	4235
4236	4237	4238	4239	4284	4285	4287	4291	4292	4294	4298	4299	4301
4305	4306	4308	4326	4327	4328	4330	4356	4360	4381	4382	4383	4384
4385	4386	4403	4404	4408	4409	4415	4433	4445	4448	4461	4497	4505
4507	4508	4519	4521	4522	4528	4530	4531	4538	4540	4541	4553	4555
4556	4563	4565	4566	4576	4577	4578	4579	4581	4582	4599	4601	4602
4612	4613	4614	4615	4617	4618	4624	4631	4639	4641	4642	4652	4654
4655	4661	4663	4664	4671	4673	4674	4686	4688	4689	4696	4698	4699
4709	4710	4711	4712	4714	4715	4728	4730	4731	4738	4740	4741	4751
4752	4753	4754	4756	4757	4763	4766	4801	4808	4810	4811	4827	4829
4830	4837	4839	4840	4847	4849	4850	4857	4859	4860	4867	4869	4870
4878	4880	4881	4889	4891	4892	4900	4902	4903	4909	4911	4912	4915
4921	4928	4930	4931	4947	4949	4950	4957	4959	4960	4967	4969	4970
4977	4979	4980	4987	4989	4990	4998	5000	5001	5009	5011	5012	5020
5022	5023	5029	5031	5032	5035	5038	5062	5069	5071	5072	5091	5093
5094	5114	5116	5117	5125	5127	5128	5136	5138	5139	5147	5149	5150
5158	5160	5161	5167	5169	5170	5173	5179	5186	5188	5189	5209	5211
5212	5226	5228	5229	5237	5239	5240	5248	5250	5251	5259	5261	5262
5270	5272	5273	5279	5281	5282	5285	5288	5332	5334	5335	5349	5351
5352	5359	5361	5362	5369	5371	5372	5379	5381	5382	5388	5390	5391
5399	5401	5402	5409	5411	5412	5420	5422	5423	5435	5437	5438	5450
5452	5453	5465	5467	5468	5479	5481	5482	5489	5491	5492	5499	5501
5502	5509	5511	5512	5520	5522	5523	5530	5532	5533	5541	5543	5544
5552	5554	5555	5564	5566	5567	5570	5598	5605	5611	5613	5614	5641
5643	5644	5650	5652	5653	5659	5661	5662	5668	5670	5671	5688	5689
5690	5691	5693	5694	5704	5706	5707	5717	5719	5720	5730	5732	5733
5743	5745	5746	5757	5758	5759	5760	5762	5763	5773	5781	5784	5818
5824	5826	5827	5852	5854	5855	5861	5863	5864	5870	5872	5873	5879
5881	5882	5901	5902	5903	5904	5906	5907	5917	5919	5920	5930	5932
5933	5943	5945	5946	5956	5958	5959	5971	5972	5973	5974	5976	5977
5986	5987	5988	5989	5991	5992	6001	6004	6034	6036	6037	6051	6053
6054	6061	6063	6064	6071	6073	6074	6081	6083	6084	6090	6092	6093
6101	6103	6104	6111	6113	6114	6122	6124	6125	6132	6134	6135	6143
6145	6146	6153	6155	6156	6164	6166	6167	6174	6176	6177	6185	6187
6188	6195	6197	6198	6206	6208	6209	6216	6218	6219	6227	6229	6230
6237	6239	6240	6248	6250	6251	6258	6260	6261	6269	6271	6272	6279
6281	6282	6290	6292	6293	6300	6302	6303	6311	6313	6314	6321	6323
6324	6332	6334	6335	6342	6344	6345	6353	6355	6356	6363	6365	6366
6374	6376	6377	6385	6387	6388	6396	6398	6399	6402	6422	6430	6432
6433	6447	6449	6450	6457	6459	6460	6467	6469	6470	6476	6478	6479
6487	6489	6490	6496	6498	6499	6507	6509	6510	6519	6520	6521	6522

```
                        6524   6525   6530   6531   6532   6533   6535   6536   6542   6544   6545   6548   6552
                        6560   6562   6563   6577   6579   6580   6587   6589   6590   6597   6599   6600   6606
                        6608   6609   6617   6619   6620   6628   6630   6631   6637   6639   6640   6648   6650
                        6651   6657   6659   6660   6663   6666   6689   6691   6692   6706   6708   6709   6716
                        6718   6719   6726   6728   6729   6735   6737   6738   6746   6748   6749   6759   6761
                        6762   6772   6774   6775   6778   6801   6803   6804   6818   6820   6821   6828   6830
                        6831   6838   6840   6841   6847   6849   6850   6858   686v   6861   6873   6875   6876
                        6884   6886   6887   6890   6919   6928   6930   6931   6945   6947   6948   6955   6957
                        6958   6965   6967   6968   6979   6981   6982   6996   6998   6999   7007   7009   7010
                        7018   7020   7021   7040   7041   7042   7043   7045   7046   7054   7056   7057   7060
                        7067   7101   7105   7106   7107   7108   7110   7111   7112   7113   7115   7116   7117
                        7118   7119   7122   7140   7144   7161   7162   7163
SVCSUB= 000001            949#   958#  4496   4630   4800   4920   5061   5178   5597   5817   6421   6551   6918
SVCTAG= 000001            949#   960#  1067   1071   1113   1127   3761   3795   3839   3873   3908   3926   3951
                         3977   4359   4414   4432   4447   4460   4623   4762   4765   4914   5034   5037   5172
                         5284   5287   5569   5772   5780   5783   6000   6003   6401   6547   6662   6665   6777
                         6889   7059   7066   7123   7145
SVCTST= 000001            949#   957#  4491   4795   5056   5324   5592   5812   6026   6418   6680   6793   6911
SWPBOT= 121000           1620#
SWPDDC= 121400           1621#
SYNCH = 000226           1307#  3210   5065   5182   5328   6030   6426   6556   6685   6924
SYSLSYM= 010000           949#  1114#  1128#  3762#  3796#  3840#  3874#  3909#  3927#  3952#  3978#  4360#  4415#
                         4433#  4448#  4461#  4624#  4763#  4766#  4915#  5035#  5038#  5173#  5285#  5288#  5570#
                         5605#  5781#  5784#  6001#  6004#  6402#  6548#  6663#  6666#  6778#  6890#  7060#  7067#
                         7124#  7146#
TAB   = 000004           1291#
TABLR   034022           6920   7069#
TBMT  = 000100           1358#  2820   2831   3220   6493   6539   6634   6654   6756   6769
TCCHEK= 100000           1625#
TDATA   002326           1693#  2311
TDSRH = 120403           1286#  3206   3311
TDSRL = 120402           1280#  3209   3258   4515   4549   4591   4648   4682   4724
TDSRNR  002601           1805#
TEOM  = 000002           1292#  5100   5103   5426   5441   5456   5709   5722   5922   5935   6359   6864   6985
TERR  = 000200           1289#
TGA   = 000010           1290#  6864
TIMFLG  002356           1706#
TM    = 000004           1396#
TMP0    002552           1786#  4387*  4405   4417*
TMP1    002554           1787#
TMP2    002556           1788#
TMP3    002560           1789#
TMP4    002562           1790#
TMP5    002564           1791#
TMP6    002566           1792#
TMP7    002570           1793#
TSO   = 000010           1361#  3067   3079   6493   6539   6634   6654   6756
TSOM  = 000001           1293#  3207   4814   4817   4934   4937   5075   5192   5338   5631   5735   5842   5948
                         6040   6436   6566   6695   6765   6807   6934
TSTCON  002476           1758#  4352*
TSTNUM  002420           1723#
TTLOOP= 000002           1381#  3203   3233   3647   6753
TXAB  = 002000           1298#
TXACT = 000004           1362#  2730   2741   6493   6539   6634   6654   6756   6769
TXCHAR  007622           3253#  4822   4842   4852   4862   4942   4962   4972   4982   5086   5204   5344   5354
                         5364   5374   5404   5474   5484   5494   5504   5525   5547   5636   5645   5654   5663
```

CVDMDA.P11 10-DEC-80 09:15 CROSS REFERENCE TABLE -- USER SYMBOLS

```
TXTVR4  020616          3694#  3708
TXTVR5  020623          3694#  3708
TXTVR6  020630          3694#  3708
TXTVR7  020635          3694#  3708
TXTVR8  020642          3694#  3711
TXTVR9  020647          3694#  3711
TXT1    017005          3694#  4004
TXT10   017605          3694#
TXT11   01 625          3694#
TXT11A  017677          3694#
TXT11B  017735          3694#
TXT12   020005          3694#  3805   3849   3884   3918   3936   3961
TXT13   020033          3694#  4207
TXT14   020050          3694#  4206
TXT15   020106          3694#  4224
TXT16   020150          3694#  4130
TXT17   020163          3694#  4129
TXT18   020220          3694#  4147
TXT19   020261          3694#  4164
TXT2    017043          3694#  4022
TXT2A   017105          3694#  4039
TXT2B   017144          3694#  4056
TXT20   020315          3694#  4181
TXT3    017207          3694#  4005
TXT4    017237          3694#  4080
TXT4A   017277          3694#  4098
TXT5    017340          3694#
TXT6    017342          3694#  4081
TXT7    017365          3694#
TXT7A   017455          3694#
TXT8    017545          3694#
TXT9    017565          3694#
TXU     = 000002        1363#  3374   6493   6539   6634   6654   6756
TSARGC= 000005           977#   978#   979#   980#   981#   982#  3735#  3740  3750#  3756  3771#  3777  3780#
                        3787   3789#  3793   3804#  3810   3812#  3816   3820#  3826   3829#  3836   3848#  3854
                        3856#  3860   3864#  3870   3883#  3889   3891#  3896   3899#  3906   3917#  3923   3935#
                        3941   3943#  3948   3960#  3966   3968#  3974   4004#  4010   4012#  4020   4022#  4027
                        4029#  4037   4039#  4044   4046#  4054   4056#  4061   4063#  4071   4080#  4086   4088#
                        4096   4098#  4103   4105#  4113   4115#  4119   4129#  4135   4137#  4145   4147#  4152
                        4154#  4162   4164#  4169   4171#  4179   4181#  4186   4188#  4196   4206#  4212   4214#
                        4222   4224#  4229   4231#  4239
TSCODE= 002032          7105#  7110#  7115#
TSERRN= 000062           949#  4577#  4613#  4710#  4752#  5689#  5758#  5902#  5972#  5987#  6520#  6531#  7041#
TSEXCP= 000000          7105#  7109   7110#  7114   7115#  7120
TSFLAG= 000040          4507#  4521#  4530#  4540#  4555#  4565#  4581#  4601#  4617#  4641#  4654#  4663#  4673#
                        4688#  4698#  4714#  4730#  4740#  4756#  4810#  4829#  4839#  4849#  4859#  4869#  4880#
                        4891#  4902#  4911#  4930#  4949#  4959#  4969#  4979#  4989#  5000#  5011#  5022#  5031#
                        5071#  5093#  5116#  5127#  5138#  5149#  5160#  5169#  5188#  5211#  5228#  5239#  5250#
                        5261#  5272#  5281#  5334#  5351#  5361#  5371#  5381#  5390#  5401#  5411#  5422#  5437#
                        5452#  5467#  5481#  5491#  5501#  5511#  5522#  5532#  5543#  5554#  5566#  5613#  5643#
                        5652#  5661#  5670#  5693#  5706#  5719#  5732#  5745#  5762#  5826#  5854#  5863#  5872#
                        5881#  5906#  5919#  5932#  5945#  5958#  5976#  5991#  6036#  6053#  6063#  6073#  6083#
                        6092#  6103#  6113#  6124#  6134#  6145#  6155#  6166#  6176#  6187#  6197#  6208#  6218#
                        6229#  6239#  6250#  6260#  6271#  6281#  6292#  6302#  6313#  6323#  6334#  6344#  6355#
                        6365#  6376#  6387#  6398#  6432#  6449#  6459#  6469#  6478#  6489#  6498#  6509#  6524#
                        6535#  6544#  6562#  6579#  6589#  6599#  6608#  6619#  6630#  6639#  6650#  6659#  6691#
```

```
                    6708#  6718#  6728#  6737#  6748#  6761#  6774#  6803#  6820#  6830#  6840#  6849#  6860#
                    6875#  6886#  6930#  6947#  6957#  6967#  6981#  6998#  7009#  7020#  7045#  7056#
T$GMAN= 000000       949#
T$HILI= 000007      7105#  7108   7110#  7113   7115#  7119
T$LAST= 000001       949#  7162#
T$LOLI= 000004      7105#  7107   7110#  7112   7115#  7118
T$LSYM= 010000       949#  1114   1128   3762   3796   3840   3874   3909   3927   3952   3978   4360   4415
                    4433   4448   4461   4624   4763   4766   4915   5035   5038   5173   5285   5288   5570
                    5781   5784   6001   6004   6402   6548   6663   6666   6778   6890   7060   7067   7124
                    7146
T$LTNO= 000013      7165#
T$NEST= 177777       949#   952#  1097#  1113#  1122#  1127#  3728#  3761#  3767#  3795#  3802#  3839#  3846#
                    3873#  3881#  3908#  3915#  3926#  3933#  3951#  3958#  3977#  4251#  4256#  4264#  4359#
                    4378#  4414#  4428#  4432#  4442#  4447#  4458#  4460#  4492#  4497#  4623#  4631#  4762#
                    4765#  4796#  4801#  4914#  4921#  5034#  5037#  5057#  5062#  5172#  5179#  5284#  5287#
                    5325#  5569#  5593#  5598#  5605#  5772#  5780#  5783#  5813#  5818#  6000#  6003#  6027#
                    6401#  6419#  6422#  6547#  6552#  6662#  6665#  6681#  6777#  6794#  6889#  6912#  6919#
                    7059#  7066#  7101#  7122#  7140#  7144#  7158#
T$NS0 = 000000       952#  7158
T$NS1 = 000005      1097#  1113   1122#  1127   3728#  3761   3767#  3795   3802#  3839   3846#  3873   3881#
                    3908   3915#  3926   3933#  3951   3958#  3977   4251#  4256   4264#  4359   4378#  4414
                    4428#  4432   4442#  4447   4458#  4460   4492#  4765   4796#  5037   5057#  5287   5325#
                    5569   5593#  5783   5813#  6003   6027#  6401   6419#  6665   6681#  6777   6794#  6889
                    6912#  7066   7101#  7122   7140#  7144
T$NS2 = 000002      4497#  4623   4631#  4762   4801#  4914   4921#  5034   5062#  5172   5179#  5284   5598#
                    5780   5818#  6000   6422#  6547   6552#  6662   6919#  7059
T$NS3 = 000003      5605#  5772
T$PTNU= 000000       949#
T$SAVL= 177777       949#
T$SEGL= 177777       949#  5605#  5614   5644   5653   5662   5671   5694   5707   5720   5733   5746   5763
                    5772#  5774
T$SEKO= 010000      5605#  5614   5644   5653   5662   5671   5694   5707   5720   5733   5746   5763   5772
T$SUBN= 000001       949#  4491#  4496#  4630#  4795#  4800#  4920#  5056#  5061#  5178#  5324#  5592#  5597#
                    5812#  5817#  6026#  6418#  6421#  6551#  6680#  6793#  6911#  6918#
T$TAGL= 177777       949#
T$TAGN= 010050       949#  1097#  1122#  3728#  3767#  3802#  3846#  3881#  3915#  3933#  3958#  4251#  4264#
                    4378#  4428#  4442#  4458#  4492#  4497#  4631#  4796#  4801#  4921#  5057#  5062#  5179#
                    5325#  5593#  5598#  5813#  5818#  6027#  6419#  6422#  6552#  6681#  6794#  6912#  6919#
                    7101#  7140#
T$TEMP= 000000      1067#  1071#  1076#  1077#  1078#  1079#  1080#  1081#  1082#  1083#  1084#  1085#  1086#
                    1087#  1113#  1127#  3761#  3795#  3839#  3873#  3908#  3926#  3951#  3977#  4256#  4359#
                    4414#  4432#  4447#  4460#  4507#  4508   4521#  4522   4530#  4531   4540#  4541   4555#
                    4556   4565#  4566   4581#  4582   4601#  4602   4617#  4618   4623#  4641#  4642   4654#
                    4655   4663#  4664   4673#  4674   4688#  4689   4698#  4699   4714#  4715   4730#  4731
                    4740#  4741   4756#  4757   4762#  4765#  4810#  4811   4829#  4830   4839#  4840   4849#
                    4850   4859#  4860   4869#  4870   4880#  4881   4891#  4892   4902#  4903   4911#  4912
                    4914#  4930#  4931   4949#  4950   4959#  4960   4969#  4970   4979#  4980   4989#  4990
                    5000#  5001   5011#  5012   5022#  5023   5031#  5032   5034#  5037#  5071#  5072   5093#
                    5094   5116#  5117   5127#  5128   5138#  5139   5149#  5150   5160#  5161   5169#  5170
                    5172#  5188#  5189   5211#  5212   5228#  5229   5239#  5240   5250#  5251   5261#  5262
                    5272#  5273   5281#  5282   5284#  5287#  5334#  5335   5351#  5352   5361#  5362   5371#
                    5372   5381#  5382   5390#  5391   5401#  5402   5411#  5412   5422#  5423   5437#  5438
                    5452#  5453   5467#  5468   5481#  5482   5491#  5492   5501#  5502   5511#  5512   5522#
                    5523   5532#  5533   5543#  5544   5554#  5555   5566#  5567   5569#  5613#  5614#  5643#
                    5644#  5652#  5653#  5661#  5662#  5670#  5671#  5693#  5694#  5706#  5707#  5719#  5720#
                    5732#  5733#  5745#  5746#  5762#  5763#  5772#  5780#  5783#  5826#  5827   5854#  5855
```

```
                    5863#  5864   5872#  5873   5881#  5882   5906#  5907   5919#  5920   5932#  5933   5945#
                    5946   5958#  5959   5976#  5977   5991#  5992   6000#  6003#  6036#  6037   6053#  6054
                    6063#  6064   6073#  6074   6083#  6084   6092#  6093   6103#  6104   6113#  6114   6124#
                    6125   6134#  6135   6145#  6146   6155#  6156   6166#  6167   6176#  6177   6187#  6188
                    6197#  6198   6208#  6209   6218#  6219   6229#  6230   6239#  6240   6250#  6251   6260#
                    6261   6271#  6272   6281#  6282   6292#  6293   6302#  6303   6313#  6314   6323#  6324
                    6334#  6335   6344#  6345   6355#  6356   6365#  6366   6376#  6377   6387#  6388   6398#
                    6399   6401#  6432#  6433   6449#  6450   6459#  6460   6469#  6470   6478#  6479   6489#
                    6490   6498#  6499   6509#  6510   6524#  6525   6535#  6536   6544#  6545   6547#  6562#
                    6563   6579#  6580   6589#  6590   6599#  6600   6608#  6609   6619#  6620   6630#  6631
                    6639#  6640   6650#  6651   6659#  6660   6662#  6665#  6691#  6692   6708#  6709   6718#
                    6719   6728#  6729   6737#  6738   5748#  6749   6761#  6762   6774#  6775   6777#  6803#
                    6804   6820#  6821   6830#  6931   6840#  6841   6849#  6850   6860#  6861   6875#  6876
                    6886#  6887   6889#  6930#  6931   6947#  6948   6957#  6958   6967#  6968   6981#  6982
                    6998#  6999   7009#  7010   7020#  7021   7045#  7046   7056#  7057   7059#  7066#  7105#
                    7110#  7115#  7122#  7144#  7158#
TSTEST= 000013       949#  4491#  4496   4630   4795#  4800   4920   5056#  5061   5178   5324#  5592#  5597
                    5812#  >817   6026#  6418#  6421   6551   6680#  6793#  6911#  6918   7165
TSTSTM= 177777       949#  3739   3755   3762   3776   3786   3792   3796   3809   3815   3825   3835   3840
                    3853   3859   3869   3874   3888   3895   3905   3909   3922   3927   3940   3947   3952
                    3965   3973   3978   4009   4019   4026   4036   4043   4053   4060   4070   4085   4095
                    4102   4112   4118   4134   4144   4151   4161   4168   4178   4185   4195   4211   4221
                    4228   4238   4285   4292   4299   4306   4327   4356   4360   4385   4404   4409   4415
                    4433   4445   4448   4461   4497   4505   4507   4519   4521   4528   4530   4538   4540
                    4553   4555   4563   4565   45     4581   4599   4601   4612   4617   4624   4631   4639
                    4641   4652   4654   4661   4663   4671   4673   4686   4688   4696   4698   4709   4714
                    4728   4730   4738   4740   4751   4756   4763   4766   4801   4808   4810   4827   4829
                    4837   4839   4847   4849   4857   4859   4867   4869   4878   4880   4889   4891   4900
                    4902   4909   4911   4915   4921   4928   4930   4947   4949   4957   4959   4967   4969
                    4977   4979   4987   4989   4998   5000   5009   5011   5020   5022   5029   5031   5035
                    5038   5062   5069   5071   5091   5093   5114   5116   5125   5127   5136   5138   5147
                    5149   5158   5160   5167   5169   5173   5179   5186   5188   5209   5211   5226   5228
                    5237   5239   5248   5250   5259   5261   5270   5272   5279   5281   5285   5288   5332
                    5334   5349   5351   5359   5361   5369   5371   5379   5381   5388   5390   5399   5401
                    5409   5411   5420   5422   5435   5437   5450   5452   5465   5467   5479   5481   5489
                    5491   5499   5501   5509   5511   5520   5522   5530   5532   5541   5543   5552   5554
                    5564   5566   5570   5598   5605   5611   5613   5641   5643   5650   5652   5659   5661
                    5668   5670   5688   5693   5704   5706   5717   5719   5730   5732   5743   5745   5757
                    5762   5773   5781   5784   5818   5824   5826   5852   5854   5861   5863   5870   5872
                    5879   5881   5901   5906   5917   5919   5930   5932   5943   5945   5956   5958   5971
                    5976   5986   5991   6001   6004   6034   6036   6051   6053   6061   6063   6071   6073
                    6081   6083   6090   6092   6101   6103   6111   6113   6122   6124   6132   6134   6143
                    6145   6153   6155   6164   6166   6174   6176   6185   6187   6195   6197   6206   6208
                    6216   6218   6227   6229   6237   6239   6248   6250   6258   6260   6269   6271   6279
                    6281   6290   6292   6300   6302   6311   6313   6321   6323   6332   6334   6342   6344
                    6353   6355   6363   6365   6374   6376   6385   6387   6396   6398   6402   6422   6430
                    6432   6447   6449   6457   6459   6467   6469   6476   6478   6487   6489   6496   6498
                    6507   6509   6519   6524   6530   6535   6542   6544   6548   6552   6560   6562   6577
                    6579   6587   6589   6597   6599   6606   6608   6617   6619   6628   6630   6637   6639
                    6648   6650   6657   6659   6663   6666   6689   6691   6706   6708   6716   6718   6726
                    6728   6735   6737   6746   6748   6759   6761   6772   6774   6778   6801   6803   6818
                    6820   6828   6830   6838   6840   6847   6849   6858   6860   6873   6875   6884   6886
                    6890   6919   6928   6930   6945   6947   6955   6957   6965   6967   6979   6981   6996
                    6998   7007   7009   7018   7020   7040   7045   7054   7056   7060   7067
TSTSTS= 000001        949#  4492#  4796#  5057#  5325#  5593#  5813#  6027#  6419#  6681#  6794#  6912#
TSSAU = 010017       4458#  4460
```

CVDMDA.P11 10-DEC-80 09:15 CROSS REFERENCE TABLE -- USER SYMBOLS

```
T$$AUT= 010014   4378#  4414
T$$CLE= 010015   4428#  4432
T$$DU = 010016   4442#  4447
T$$HAR= 010046   7101#  7123
T$$HW = 010000   1097#  1113
T$$INI= 010013   4264#  4359
T$$MSG= 010011   3728#  3761  3767#  3795  3802#  3839  3846#  3873  3881#  3908  3915#  3926  3933#
                 3951   3958#  3977
T$$PRO= 010012   4251#
T$$SEG= 010000   5605#  5613  5643  5652  5661  5670  5693  5706  5719  5732  5745  5762  5772#
T$$SOF= 010047   7140#  7145
T$$SUB= 010045   4497#  4507  4521  4530  4540  4555  4565  4581  4601  4617  4623  4631#  4641
                 4654   4663  4673  4688  4698  4714  4730  4740  4756  4762  4801#  4810   4829
                 4839   4849  4859  4869  4880  4891  4902  4911  4914  4921#  4930   4949   4959
                 4969   4979  4989  5000  5011  5022  5031  5034  5062#  5172  5179#  5284   5598#
                 5780   5818#  5826  5854  5863  5872  5881  5906  5919  5932  5945  5958   5976
                 5991   6000  6422#  6432  6449  6459  6469  6478  6489  6498  6509  6524   6535
                 6544   6547  6552#  6562  6579  6589  6599  6608  6619  6630  6639  6650   6659
                 6662   6919#  6930  6947  6957  6967  6981  6998  7009  7020  7045  7056   7059
T$$SW = 010001   1122#  1127
T$$TES= 010044   4492#  4765  4796#  5037  5057#  5071  5093  5116  5127  5138  5149  5160   5169
                 5188   5211  5228  5239  5250  5261  5272  5281  5287  5325#  5334  5351   5361
                 5371   5381  5390  5401  5411  5422  5437  5452  5467  5481  5491  5501   5511
                 5522   5532  5543  5554  5566  5569  5593#  5783  5813#  6003  6027#  6036   6053
                 6063   6073  6083  6092  6103  6113  6124  6134  6145  6155  6166  6176   6187
                 6197   6208  6218  6229  6239  6250  6260  6271  6281  6292  6302  6313   6323
                 6334   6344  6355  6365  6376  6387  6398  6401  6419#  6665  6681#  6691   6708
                 6718   6728  6737  6748  6761  6774  6777  6794#  6803  6820  6830  6840   6849
                 6860   6875  6886  6889  6912#  7066

T.EDF = 000001   1640#  2065  2109  2156  2227  2326  2598  2702  2735  2746  2780  2791   2825
                 2836   2870  2881  2915  2926  2958  2969  3005  3016  3029  3040  3072   3084
                 3132   3380  3393  3419  3429  3442  3452  3465  3475  3488  3498  3511   3521
T.EHRD= 000002   1640#
T.ESF = 000000   1640#
T.ESFT= 000003   1640#
T01TBL  026626   5083   5096  5201  5214  5290#
T1      024314 G 1076   4491#
T1MODE= 000300   1513#
T1.1    024314   4495#
T1.2    024652   4630#
T10     033122 G 1085   6793#
T11     033364 G 1086   6911#
T11.1   033374   6918#
T2      025222 G 1077   4795#
T2MODE= 000040   1523#
T2.1    025222   4800#
T2.2    025526   4920#
T3      026034 G 1078   5056#
T3.1    026034   5061#
T3.2    026340   5178#
T4      026636 G 1079   5324#
T5      027464 G 1080   5592#
T5.1    027472   5597#
T6      030176 G 1081   5812#
T6.1    030204   5817#
T7      030674 G 1082   6026#
```

```
T8       032040 G        1083    6418#
T8.1     032040          6421#
T8.2     032366          6551#
T9       032660 G        1084    6680#
T9.LP    033374          6917#   7064
UAM    = 000200 G        1196#
UMAINT= 000001           1398#
UNIT     002414          1721#
UPBITS   002572          1796#
UREGS    002246          1686#   2352    2355*   2358    4214    4215    4216    4217    4231    4232    4233    4234
USTATR= 122000           1353#   2357    2690    2726    2771    2816    2861    2906    3063    3218    3372    3599
USYREG   002502          1763#   2583
USYRT  = 120400          1245#   2353    2366    2370
VECTOR   034124          7111    7125#
VIA    = 120000          1367#   2384    2391
VIAACR= 120013           1511#   2444    2453    2531    2540    2654    3186
VIADPA= 120003           1417#   2645
VIADPB= 120002           1405#   2642
VIAIER= 120016           1584#   2440    2527    2660
VIAIFR= 120015           1562#
VIAMS  = 120001          1389#
VIAORA= 120017           1604#   2648
VIAORB= 120000           1373#   2575    2578    2651    3166    3169    3202    3646    6752
VIAPCR= 120014           1549#   2657
VIASR  = 120012          1501#
VIAT1A= 120004           1431#   2466
VIAT1B= 120005           1443#   2471    3675
VIAT1C= 120006           1456#   2457    3189
VIAT1D= 120007           1467#   2462    3192
VIAT2A= 120010           1478#   2544
VIAT2B= 120011           1490#   2549
VREGS    002266          1689#   2383    4137    4138    4139    4140    4154    4155    4156    4157    4171    4172    4173
                         4174    4188    4189    4190    4191
WAIT50   005236          2617#
WRIBYT   002364          1709#
WRILOC= 000002           1235#   2214    2224
WRIPAG= 000004           1237#
WRITE    003646          2187#   2309
WRITE1   003660          2211#   2439    2452    2456    2461    2465    2470    2526    2539    2543    2548    2574    2577
                         2641    2644    2647    2650    2653    2656    2659    3165    3168    3172    3176    3182    3185
                         3188    3191    3201    3205    3208    3257    3310    3645    3674    4514    4548    4590    4647
                         4681    4723    4832    4952    6751    6970
WSR0     002206          1661#   2277*   4091
WSR10    002216          1669#   2281*   4108
WSR12    002220          1671#   2282*   4107
WSR14    002222          1673#   2283*   4106
WSR16    002224          1675#   2284*   4105
WSR2     002210          1663#   2278*   4090
WSR4     002212          1665#   2279*   4089
WSR6     002214          1667#   2280*   4088
XDATA    002334          1696#   3780    3829    3899    3993*   3994*
XORGB    022352          3778    3827    3897    3991#
XYZ    = 000007          1322#
X$ALWA= 000000           949#
X$FALS= 000040           949#
X$OFFS= 000400           949#
```

```
XSTRJE= 000020   949#
$E    = 000062   1640#  2064#  2108#  2155#  2226#  2325#  2597#  2701#  2734#  2745#  2779#  2790#  2824#
                 2835#  2869#  2880#  2914#  2925#  2957#  2968#  3004#  3015#  3028#  3039#  3071#  3083#
                 3131#  3379#  3392#  3418#  3428#  3441#  3451#  3464#  3474#  3487#  3497#  3510#  3520#
                 4575#  4611#  4708#  4750#  5687#  5756#  5900#  5970#  5985#  6518#  6529#  7039#

$LSTIN= 000001   954#
$LSTTA= 000001   955#
$T    = 000013   1640#  4462#  4767#  5039#  5299#  5571#  5793#  6005#  6403#  6667#  6779#  6891#
.     = 034322    945#  1686#  1689#  1773#  1987#  1990#  2011#  2027#  3137   4503   4508   4517   4522
                 4526   4531   4536   4541   4551   4556   4561   4566   4582   4597   4602   4618   4637
                 4642   4650   4655   4659   4664   4669   4674   4684   4689   4694   4699   4715   4726
                 4731   4736   4741   4757   4806   4811   4825   4830   4835   4840   4845   4850   4855
                 4860   4865   4870   4876   4881   4887   4892   4898   4903   4907   4912   4926   4931
                 4945   4950   4955   4960   4965   4970   4975   4980   4985   4990   4996   5001   5007
                 5012   5018   5023   5027   5032   5067   5072   5089   5094   5112   5117   5123   5128
                 5134   5139   5145   5150   5156   5161   5165   5170   5184   5189   5207   5212   5224
                 5229   5235   5240   5246   5251   5257   5262   5268   5273   5277   5282   5297#  5330
                 5335   5347   5352   5357   5362   5367   5372   5377   5382   5386   5391   5397   5402
                 5407   5412   5418   5423   5433   5438   5448   5453   5463   5468   5477   5482   5487
                 5492   5497   5502   5507   5512   5518   5523   5528   5533   5539   5544   5550   5555
                 5562   5567   5609   5614   5639   5644   5648   5653   5657   5662   5666   5671   5694
                 5702   5707   5715   5720   5728   5733   5746   5763   5769   5777   5791#  5822   5827
                 5850   5855   5859   5864   5868   5873   5877   5882   5907   5915   5920   5928   5933
                 5941   5946   5954   5959   5977   5992   6032   6037   6049   6054   6059   6064   6069
                 6074   6079   6084   6088   6093   6099   6104   6109   6114   6120   6125   6130   6135
                 6141   6146   6151   6156   6162   6167   6172   6177   6183   6188   6193   6198   6204
                 6209   6214   6219   6225   6230   6235   6240   6246   6251   6256   6261   6267   6272
                 6277   6282   6288   6293   6298   6303   6309   6314   6319   6324   6330   6335   6340
                 6345   6351   6356   6361   6366   6372   6377   6383   6388   6394   6399   6428   6433
                 6445   6450   6455   6460   6465   6470   6474   6479   6485   6490   6494   6499   6505
                 6510   6525   6536   6540   6545   6558   6563   6575   6580   6585   6590   6595   6600
                 6604   6609   6615   6620   6626   6631   6635   6640   6646   6651   6655   6660   6687
                 6692   6704   6709   6714   6719   6724   6729   6733   6738   6744   6749   6757   6762
                 6770   6775   6799   6804   6816   6821   6826   6831   6836   6841   6845   6850   6856
                 6861   6871   6876   6882   6887   6926   6931   6943   6948   6953   6958   6963   6968
                 6977   6982   6994   6999   7005   7010   7016   7021   7046   7052   7057   7063   7149#
```

BCOMPL	1#	949#	4286	4293	4300	4307	4329					
BERROR	1#	949#										
BGNAU	1#	949#	4457									
BGNAUT	1#	949#	4377									
BGNCLN	1#	949#	4427									
BGNDU	1#	949#	4441									
BGNHRD	1#	949#	7100									
BGNHW	1#	949#	1096									
BGNINI	1#	949#	4263									
BGNMOD	1#	949#	951									
BGNMSG	1#	949#	3727	3766	3801	3845	3880	3914	3932	3957		
BGNPRO	1#	949#	4250									
BGNPTA	1#	949#										
BGNRPT	1#	949#										
BGNSEG	1#	949#	5604									
BGNSET	1#	949#										
BGNSFT	1#	949#	7139									
BGNSRV	1#	949#										
BGNSUB	1#	949#	4495	4629	4799	4919	5060	5177	5596	5816	6420	6550	6917
BGNSW	1#	949#	1121									
BGNTST	1#	949#	4491	4795	5056	5324	5592	5812	6026	6418	6680	6793	6911
BNCOMP	1#	949#										
BNERRO	1#	949#										
BREAK	1#	949#										
BRESET	1#	949#	4355	4444								
CKLOOP	1#	949#										
CLOCK	1#	949#										
CLOSE	1#	949#										
CLRVEC	1#	949#	4402									
COMMEN	1#	949#										
DELAY	1#	949#										
DESCRI	1#	949#	2018									
DEVTYP	1#	949#	2006									
DISPAT	1#	949#	1073									
DISPLA	1#	949#										
DOCLN	1#	949#										
DODU	1#	949#	4407									
DORPT	1#	949#										
ENDAU	1#	949#	4459									
ENDAUT	1#	949#	4413									
ENDCLN	1#	949#	4431									
ENDCOM	1#	949#										
ENDDU	1#	949#	4446									
ENDHRD	1#	949#	7121									
ENDHW	1#	949#	1112									
ENDINI	1#	949#	4358									
ENDMOD	1#	949#	7157									
ENDMSG	1#	949#	3760	3794	3838	3872	3907	3925	3950	3976		
ENDPRO	1#	949#	4255									
ENDPTA	1#	949#										
ENDRPT	1#	949#										
ENDSEG	1#	949#	5771									
ENDSET	1#	949#										
ENDSFT	1#	949#	7143									
ENDSRV	1#	949#										
ENDSUB	1#	949#	4622	4761	4913	5033	5171	5283	5779	5999	6546	6661	7058

```
ENDSW      1#  949#  1126
ENDTST     1#  949#  4764  5036  5286  5568  5782  6002  6400  6664  6776  6888  7065
EQUALS     1#  949#  1136
ERRDF      1#  949#  4576  4612  4709  4751  5688  5757  5901  5971  5986  6519  6530  7040
ERRHRD     1#  949#
ERROR      1#  949#  4504  4518  4527  4537  4552  4562  4598  4638  4651  4660  4670  4685  4695
           4727  4737  4807  4826  4836  4846  4856  4866  4877  4888  4899  4908  4927  4946  4956
           4966  4976  4986  4997  5008  5019  5028  5068  5090  5113  5124  5135  5146  5157  5166
           5185  5208  5225  5236  5247  5258  5269  5278  5331  5348  5358  5368  5378  5387  5398
           5408  5419  5434  5449  5464  5478  5488  5498  5508  5519  5529  5540  5551  5563  5610
           5640  5649  5658  5667  5703  5716  5729  5742  5823  5851  5860  5869  5878  5916  5929
           5942  5955  6033  6050  6060  6070  6080  6089  6100  6110  6121  6131  6142  6152  6163
           6173  6184  6194  6205  6215  6226  6236  6247  6257  6268  6278  6289  6299  6310  6320
           6331  6341  6352  6362  6373  6384  6395  6429  6446  6456  6466  6475  6486  6495  6506
           6541  6559  6576  6586  6596  6605  6616  6627  6636  6647  6656  6688  6705  6715  6725
           6734  6745  6758  6771  6800  6817  6827  6837  6846  6857  6872  6883  6927  6944  6954
           6964  6978  6995  7006  7017  7053
ERRSF      1#  949#
ERRSOF     1#  949#
ERRTBL     1#  949#  1651
ESCAPE     1#  949#  4506  4520  4529  4539  4554  4564  4580  4600  4616  4640  4653  4662  4672
           4687  4697  4713  4729  4739  4755  4809  4828  4838  4848  4858  4868  4879  4890  4901
           4910  4929  4948  4958  4968  4978  4988  4999  5010  5021  5030  5070  5092  5115  5126
           5137  5148  5159  5168  5187  5210  5227  5238  5249  5260  5271  5280  5333  5350  5360
           5370  5380  5389  5400  5410  5421  5436  5451  5466  5480  5490  5500  5510  5521  5531
           5542  5553  5565  5612  5642  5651  5660  5669  5692  5705  5718  5731  5744  5761  5825
           5853  5862  5871  5880  5905  5918  5931  5944  5957  5975  5990  6035  6052  6062  6072
           6032  6091  6102  6112  6123  6133  6144  6154  6165  6175  6186  6196  6207  6217  6228
           6238  6249  6259  6270  6280  6291  6301  6312  6322  6333  6343  6354  6364  6375  6386
           6397  6431  6448  6458  6468  6477  6488  6497  6508  6523  6534  6543  6561  6578  6588
           6598  6607  6618  6629  6638  6649  6658  6690  6707  6717  6727  6736  6747  6760  6773
           6802  6819  6829  6839  6848  6859  6874  6885  6929  6946  6956  6966  6980  6997  7008
           7019  7044  7055
EXIT       1#  949#
FEQUAL     1#  949#
GEDF       1640#  4574  4610  4707  4749  5686  5755  5899  5969  5984  6517  6528  7038
GEHRD      1640#
GESF       1640#
GESFT      1640#
GETBYT     1#  949#
GETPRI     1#  949#
GETWOR     1#  949#
GMANIA     1#  949#
GMANID     1#  949#
GMANIL     1#  949#
GPHARD     1#  949#  4325
GPRMA      1#  949#  7104  7109
GPRMD      1#  949#  7114
GPRML      1#  949#
GTDF       1640#  2063  2107  2154  2225  2324  2596  2700  2733  2744  2778  2789  2823  2834  2868
           2879  2913  2924  2956  2967  3003  3014  3027  3038  3070  3082  3130  3378  3391  3417
           3427  3440  3450  3463  3473  3486  3496  3509  3519
GTHRD      1640#
GTSF       1640#
GTSFT      1640#
HEADER     1#  949#  975
```

CVDMDA.P11 10-DEC-80 09:15 CROSS REFERENCE TABLE -- MACRO NAMES

```
INLOOP      1#   949#
IOSETU      1#   949#
IOSTAR      1#   949#
KT11        1#   949#
LASTAD      1#   949#  7160
MANUAL      1#   949#
MEMORY      1#   949#
MSG      4462#  4468   4767#  477?  5039#  5045   5299#  5305   5571#  5577   5793#  5799   6005#  6011   6403#
         6409   6667#  6673   6779#  6785   6891#  6897
MSBYTE      1#   949#   976#   982    983    984
MSCHEC      1#   949#
MSCNTO      1#   949#  7105#  7110#  7115#
MSCOUN      1#   949#  3735#  3750#  3771#  3780#  3789#  3804#  3812#  3820#  3829#  3848#  3856#  3864#  3883#
         3891#  3899#  3917#  3935#  3943#  3960#  3968#  4004#  4012#  4022#  4029#  4039#  4046#  4056#  4063#
         4080#  4088#  4098#  4105#  4115#  4129#  4137#  4147#  4154#  4164#  4171#  4181#  4188#  4206#  4214#
         4224#  4231#
MSDATA      1#   949#   976#   985    987    989    991    993    995    997    999    1001   1003   1005   1007
         1009   1011   1013   1015#  1017   1019   1022   1025   1027   1029   1031   1033   1035   1037   1039
         1041   1043   1045   1047   1049   1051   1053   1055   1057   1059   2007#  2019#
MSDECR      1#   949#  1113#  1127#  3761#  3795#  3839#  3873#  3908#  3926#  3951#  3977#  4256#  4359#  4414#
         4432#  4447#  4460#  4623#  4762#  4765#  4914#  5034#  5037#  5172#  5284#  5287#  5569#  5772#  5780#
         5783#  6000#  6003#  6401#  6547#  6662#  6665#  6777#  6889#  7059#  7066#  7122#  7144#  7158#
MSDEFA      1#   949#  7105#  7110#  7115#
MSENDE      1#   949#  1113#  1127#  3761#  3795#  3839#  3873#  3908#  3926#  3951#  3977#  4359#  4414#  4432#
         4447#  4460#  4623#  4762#  4765#  4914#  5034#  5037#  5172#  5284#  5287#  5569#  5772#  5780#  5783#
         6000#  6003#  6401#  6547#  6662#  6665#  6777#  6889#  7059#  7066#  7122#  7144#  7158#
MSERRI      1#   949#  4576#  4612#  4709#  4751#  5688#  5757#  5901#  5971#  5986#  6519#  6530#  7040#
MSESCA      1#   949#  4507#  4508   4521#  4522   4530#  4531   4540#  4541   4555#  4556   4565#  4566   4581#
         4582   4601#  4602   4617#  4618   4641#  4642   4654#  4655   4663#  4664   4673#  4674   4688#  4689
         4698#  4699   4714#  4715   4730#  4731   4740#  4741   4756#  4757   4810#  4811   4829#  4830   4839#
         4840   4849#  4850   4859#  4860   4869#  4870   4880#  4881   4891#  4892   4902#  4903   4911#  4912
         4930#  4931   4949#  4950   4959#  4960   4969#  4970   4979#  4980   4989#  4990   5000#  5001   5011#
         5012   5022#  5023   5031#  5032   5071#  5072   5093#  5094   5116#  5117   5127#  5128   5138#  5139
         5149#  5150   5160#  5161   5169#  5170   5188#  5189   5211#  5212   5228#  5229   5239#  5240   5250#
         5251   5261#  5262   5272#  5273   5281#  5282   5334#  5335   5351#  5352   5361#  5362   5371#  5372
         5381#  5382   5390#  5391   5401#  5402   5411#  5412   5422#  5423   5437#  5438   5452#  5453   5467#
         5468   5481#  5482   5491#  5492   5501#  5502   5511#  5512   5522#  5523   5532#  5533   5543#  5544
         5554#  5555   5566#  5567   5613#  5643#  5652#  5661#  5670#  5693#  5706#  5719#  5732#  5745#  5762#
         5826#  5827   5854#  5855   5863#  5864   5872#  5873   5881#  5882   5906#  5907   5919#  5920   5932#
         5933   5945#  5946   5958#  5959   5976#  5977   5991#  5992   6036#  6037   6053#  6054   6063#  6064
         6073#  6074   6083#  6084   6092#  6093   6103#  6104   6113#  6114   6124#  6125   6134#  6135   6145#
         6146   6155#  6156   6166#  6167   6176#  6177   6187#  6188   6197#  6198   6208#  6209   6218#  6219
         6229#  6230   6239#  6240   6250#  6251   6260#  6261   6271#  6272   6281#  6282   6292#  6293   6302#
         6303   6313#  6314   6323#  6324   6334#  6335   6344#  6345   6355#  6356   6365#  6366   6376#  6377
         6387#  6388   6398#  6399   6432#  6433   6449#  6450   6459#  6460   6469#  6470   6478#  6479   6489#
         6490   6498#  6499   6509#  6510   6524#  6525   6535#  6536   6544#  6545   6562#  6563   6579#  6580
         6589#  6590   6599#  6600   6608#  6609   6619#  6620   6630#  6631   6639#  6640   6650#  6651   6659#
         6660   6691#  6692   6708#  6709   6718#  6719   6728#  6729   6737#  6738   6748#  6749   6761#  6762
         6774#  6775   6803#  6804   6820#  6821   6830#  6831   6840#  6841   6849#  6850   6860#  6861   6875#
         6876   6886#  6887   6930#  6931   6947#  6948   6957#  6958   6967#  6968   6981#  6982   6998#  6999
         7009#  7010   7020#  7021   7045#  7046   7056#  7057
MSESCS      1#   949#  4507#  4521#  4530#  4540#  4555#  4565#  4581#  4601#  4617#  4641#  4654#  4663#  4673#
         4688#  4698#  4714#  4730#  4740#  4756#  4810#  4829#  4839#  4849#  4859#  4869#  4880#  4891#  4902#
         4911#  4930#  4949#  4959#  4969#  4979#  4989#  5000#  5011#  5022#  5031#  5071#  5093#  5116#  5127#
         5138#  5149#  5160#  5169#  5188#  5211#  5228#  5239#  5250#  5261#  5272#  5281#  5334#  5351#  5361#
         5371#  5381#  5390#  5401#  5411#  5422#  5437#  5452#  5467#  5481#  5491#  5501#  5511#  5522#  5532#
```

```
        5543#  5554#  5566#  5613#  5614   5643#  5644   5652#  5653   5661#  5662   5670#  5671   5693#  5694
        5706#  5707   5719#  5720   5732#  5733   5745#  5746   5762#  5763   5826#  5854#  5863#  5872#  5881#
        5906#  5919#  5932#  5945#  5958#  5976#  5991#  6036#  6053#  6063#  6073#  6083#  6092#  6103#  6113#
        6124#  6134#  6145#  6155#  6166#  6176#  6187#  6197#  6208#  6218#  6229#  6239#  6250#  6260#  6271#
        6281#  6292#  6302#  6313#  6323#  6334#  6344#  6355#  6365#  6376#  6387#  6398#  6432#  6449#  6459#
        6469#  6478#  6489#  6498#  6509#  6524#  6535#  6544#  6562#  6579#  6589#  6599#  6608#  6619#  6630#
        6639#  6650#  6659#  6691#  6708#  6718#  6728#  6737#  6748#  6761#  6774#  6803#  6820#  6830#  6840#
        6849#  6860#  6875#  6886#  6930#  6947#  6957#  6967#  6981#  6998#  7009#  7020#  7045#  7056#
MSEXCP     1#   949#  7105#  7110#  7115#
MSEXIT     1#   949#
MS XSE     1#   949#
MSEXTJ     1#   949#
MSGEN      1#   949#   952#   976#   985#   987#   989#   991#   993#   995#   997#   999#  1001#  1003#  1005#
        1007#  1009#  1011#  1013#  1015#  1017#  1019#  1022#  1025#  1027#  1029#  1031#  1033#  1035#  1037#
        1039#  1041#  1043#  1045#  1047#  1049#  1051#  1053#  1055#  1057#  1059#  1075#  1098#  1099#  1113#
        1123#  1124#  1127#  1652#  2007#  2019#  3728#  3761#  3767#  3795#  3802#  3839#  3846#  3873#  3881#
        3908#  3915#  3926#  3933#  3951#  3958#  3977#  4251#  4264#  4359#  4378#  4414#  4428#  4432#  4442#
        4447#  4458#  4460#  4491#  4496#  4623#  4630#  4762#  4765#  4795#  4800#  4914#  4920#  5034#  5037#
        5056#  5061#  5172#  5178#  5284#  5287#  5324#  5569#  5592#  5597#  5772#  5780#  5783#  5812#  5817#
        6000#  6003#  6026#  6401#  6418#  6421#  6547#  6551#  6662#  6665#  6680#  6777#  6793#  6889#  6911#
        6918#  7059#  7066#  7102#  7123#  7141#  7145#  7164#
MSGENB     1#   949#
MSGETS     1#   949#  1113#  1127#  3761#  3795#  3839#  3873#  3908#  3926#  3951#  3977#  4256#  4359#  4414#
        4432#  4447#  4460#  4623#  4762#  4765#  4914#  5034#  5037#  5172#  5284#  5287#  5569#  5614#  5644#
        5653#  5662#  5671#  5694#  5707#  5720#  5733#  5746#  5763#  5772#  5780#  5783#  6000#  6003#  6401#
        6547#  6662#  6665#  6777#  6889#  7059#  7066#  7122#  7144#  7158#
MSGETT     1#   949#  4507#  4521#  4530#  4540#  4555#  4565#  4581#  4601#  4617#  4641#  4654#  4663#  4673#
        4688#  4698#  4714#  4730#  4740#  4756#  4810#  4829#  4839#  4849#  4859#  4869#  4880#  4891#  4902#
        4911#  4930#  4949#  4959#  4969#  4979#  4989#  5000#  5011#  5022#  5031#  5071#  5093#  5116#  5127#
        5138#  5149#  5160#  5169#  5188#  5211#  5228#  5239#  5250#  5261#  5272#  5281#  5334#  5351#  5361#
        5371#  5381#  5390#  5401#  5411#  5422#  5437#  5452#  5467#  5481#  5491#  5501#  5511#  5522#  5532#
        5543#  5554#  5566#  5613#  5614   5643#  5644   5652#  5653   5661#  5662   5670#  5671   5693#  5694
        5706#  5707   5719#  5720   5732#  5733   5745#  5746   5762#  5763   5826#  5854#  5863#  5872#  5881#
        5906#  5919#  5932#  5945#  5958#  5976#  5991#  6036#  6053#  6063#  6073#  6083#  6092#  6103#  6113#
        6124#  6134#  6145#  6155#  6166#  6176#  6187#  6197#  6208#  6218#  6229#  6239#  6250#  6260#  6271#
        6281#  6292#  6302#  6313#  6323#  6334#  6344#  6355#  6365#  6376#  6387#  6398#  6432#  6449#  6459#
        6469#  6478#  6489#  6498#  6509#  6524#  6535#  6544#  6562#  6579#  6589#  6599#  6608#  6619#  6630#
        6639#  6650#  6659#  6691#  6708#  6718#  6728#  6737#  6748#  6761#  6774#  6803#  6820#  6830#  6840#
        6849#  6860#  6875#  6886#  6930#  6947#  6957#  6967#  6981#  6998#  7009#  7020#  7045#  7056#
MSGNGB     1#   949#   952#   976#   985#   987#   989#   991#   993#   995#   997#   999#  1001#  1003#  1005#
        1007#  1009#  1011#  1013#  1015#  1017#  1019#  1022#  1025#  1027#  1029#  1031#  1033#  1035#  1037#
        1039#  1041#  1043#  1045#  1047#  1049#  1051#  1053#  1055#  1057#  1059#  1074#  1075   1097#  1098
        1099   1122#  1123   1124   1652#  2007#  2019#  3728#  3767#  3802#  3846#  3881#  3915#  3933#  3958#
        4251#  4264#  4378#  4428#  4442#  4458#  7101#  7102   7140#  7141   7161#  7164
MSGNIN     1#   949#   976#   977    978    979    980    981    982#  983#   984#   985#   986    987#   988
        989#   990    991#   992    993#   994    995#   996    997#  998    999#   1000   1001#  1002   1003#
        1004   1005#  1006   1007#  1008   1009#  1010   1011#  1012   1013#  1014   1015#  1016   1017#  1018
        1019#  1020   1021   1022#  1023   1024#  1025#  1026   1027#  1028   1029#  1030   1031#  1032   1033#
        1034   1035#  1036   1037#  1038   1039#  1040   1041#  1042   1043#  1044   1045#  1046   1047#  1048
        1049#  1050   1051#  1052   1053#  1054   1055#  1056   1057#  1058   1059#  1060   1074#  1076#  1077#
        1078#  1079#  1080#  1081#  1082#  1083#  1084#  1085#  1086#  1097#  1122#  2007#  2008   2011   2019#
        2020   2027   2065   2066   2067   2068   2109   2110   2111   2112   2156   2157   2158   2159   2227
        2228   2229   2230   2326   2327   2328   2329   2598   2599   2600   2601   2702   2703   2704   2705
        2735   2736   2737   2738   2746   2747   2748   2749   2780   2781   2782   2783   2791   2792   2793
        2794   2825   2826   2827   2828   2836   2837   2838   2839   2870   2871   2872   2873   2881   2882
        2883   2884   2915   2916   2917   2918   2926   2927   2928   2929   2958   2959   2960   2961   2969
```

2970	2971	2972	3005	3006	3007	3008	3016	3017	3018	3019	3029	3030	3031	3032
3040	3041	3042	3043	3072	3073	3074	3075	3084	3085	3086	3087	3132	3133	3134
3135	3380	3381	3382	3383	3393	3394	3395	3396	3419	3420	3421	3422	3429	3430
3431	3432	3442	3443	3444	3445	3452	3453	3454	3455	3465	3466	3467	3468	3475
3476	3477	3478	3488	3489	3490	3491	3498	3499	3500	3501	3511	3512	3513	3514
3521	3522	3523	3524	3735#	3736#	3737#	3738	3739#	3740	3750#	3751#	3752#	3753#	3754
3755#	3756	3762#	3771#	3772#	3773#	3774#	3775	3776#	3777	3780#	3781#	3782#	3783#	3784#
3785	3786#	3787	3789#	3790#	3791	3792#	3793	3796#	3804#	3805#	3806#	3807#	3808	3809#
3810	3812#	3813#	3814	3815#	3816	3820#	3821#	3822#	3823#	3824	3825#	3826	3829#	3830#
3831#	3832#	3833#	3834	3835#	3836	3840#	3848#	3849#	3850#	3851#	3852	3853#	3854	3856#
3857#	3858	3859#	3860	3864#	3865#	3866#	3867#	3868	3869#	3870	3874#	3883#	3884#	3885#
3886#	3887	3888#	3889	3891#	3892#	3893#	3894	3895#	3896	3899#	3900#	3901#	3902#	3903#
3904	3905#	3906	3909#	3917#	3918#	3919#	3920#	3921	3922#	3923	3927#	3935#	3936#	3937#
3938#	3939	3940#	3941	3943#	3944#	3945#	3946	3947#	3948	3952#	3960#	3961#	3962#	3963#
3964	3965#	3966	3968#	3969#	3970#	3971#	3972	3973#	3974	3978#	4004#	4005#	4006#	4007#
4008	4009#	4010	4012#	4013#	4014#	4015#	4016#	4017#	4018	4019#	4020	4022#	4023#	4024#
4025	4026#	4027	4029#	4030#	4031#	4032#	4033#	4034#	4035	4036#	4037	4039#	4040#	4041#
4042	4043#	4044	4046#	4047#	4048#	4049#	4050#	4051#	4052	4053#	4054	4056#	4057#	4058#
4059	4060#	4061	4063#	4064#	4065#	4066#	4067#	4068#	4069	4070#	4071	4080#	4081#	4082#
4083#	4084	4085#	4086	4088#	4089#	4090#	4091#	4092#	4093#	4094	4095#	4096	4098#	4099#
4100#	4101	4102#	4103	4105#	4106#	4107#	4108#	4109#	4110#	4111	4112#	4113	4115#	4116#
4117	4118#	4119	4129#	4130#	4131#	4132#	4133	4134#	4135	4137#	4138#	4139#	4140#	4141#
4142#	4143	4144#	4145	4147#	4148#	4149#	4150	4151#	4152	4154#	4155#	4156#	4157#	4158#
4159#	4160	4161#	4162	4164#	4165#	4166#	4167	4168#	4169	4171#	4172#	4173#	4174#	4175#
4176#	4177	4178#	4179	4181#	4182#	4183#	4184	4185#	4186	4188#	4189#	4190#	4191#	4192#
4193#	4194	4195#	4196	4206#	4207#	4208#	4209#	4210	4211#	4212	4214#	4215#	4216#	4217#
4218#	4219#	4220	4221#	4222	4224#	4225#	4226#	4227	4228#	4229	4231#	4232#	4233#	4234#
4235#	4236#	4237	4238#	4239	4284#	4285#	4287#	4291#	4292#	4294#	4298#	4299#	4301#	4305#
4306#	4308#	4326#	4327#	4328#	4330#	4356#	4360#	4381#	4382#	4383#	4384#	4385#	4386	4403#
4404#	4408#	4409#	4415#	4433#	4445#	4448#	4461#	4497#	4505#	4507#	4508#	4519#	4521#	4522#
4528#	4530#	4531#	4538#	4540#	4541#	4553#	4555#	4556#	4563#	4565#	4566#	4576#	4577#	4578#
4579#	4581#	4582#	4599#	4601#	4602#	4612#	4613#	4614#	4615#	4617#	4618#	4624#	4631#	4639#
4641#	4642#	4652#	4654#	4655#	4661#	4663#	4664#	4671#	4673#	4674#	4686#	4688#	4689#	4696#
4698#	4699#	4709#	4710#	4711#	4712#	4714#	4715#	4728#	4730#	4731#	4738#	4740#	4741#	4751#
4752#	4753#	4754#	4756#	4757#	4763#	4766#	4801#	4808#	4810#	4811#	4827#	4829#	4830#	4837#
4839#	4840#	4847#	4849#	4850#	4857#	4859#	4860#	4867#	4869#	4870#	4878#	4880#	4881#	4889#
4891#	4892#	4900#	4902#	4903#	4909#	4911#	4912#	4915#	4921#	4928#	4930#	4931#	4947#	4949#
4950#	4957#	4959#	4960#	4967#	4969#	4970#	4977#	4979#	4980#	4987#	4989#	4990#	4998#	5000#
5001#	5009#	5011#	5012#	5020#	5022#	5023#	5029#	5031#	5032#	5035#	5038#	5062#	5069#	5071#
5072#	5091#	5093#	5094#	5114#	5116#	5117#	5125#	5127#	5128#	5136#	5138#	5139#	5147#	5149#
5150#	5158#	5160#	5161#	5167#	5169#	5170#	5173#	5179#	5186#	5188#	5189#	5209#	5211#	5212#
5226#	5228#	5229#	5237#	5239#	5240#	5248#	5250#	5251#	5259#	5261#	5262#	5270#	5272#	5273#
5279#	5281#	5282#	5285#	5288#	5332#	5334#	5335#	5349#	5351#	5352#	5359#	5361#	5362#	5369#
5371#	5372#	5379#	5381#	5382#	5388#	5390#	5391#	5399#	5401#	5402#	5409#	5411#	5412#	5420#
5422#	5423#	5435#	5437#	5438#	5450#	5452#	5453#	5465#	5467#	5468#	5479#	5481#	5482#	5489#
5491#	5492#	5499#	5501#	5502#	5509#	5511#	5512#	5520#	5522#	5523#	5530#	5532#	5533#	5541#
5543#	5544#	5552#	5554#	5555#	5564#	5566#	5567#	5570#	5598#	5605#	5611#	5613#	5614#	5641#
5643#	5644#	5650#	5652#	5653#	5659#	5661#	5662#	5668#	5670#	5671#	5688#	5689#	5690#	5691#
5693#	5694#	5704#	5706#	5707#	5717#	5719#	5720#	5730#	5732#	5733#	5743#	5745#	5746#	5757#
5758#	5759#	5760#	5762#	5763#	5773#	5781#	5784#	5818#	5824#	5826#	5827#	5852#	5854#	5855#
5861#	5863#	5864#	5870#	5872#	5873#	5879#	5881#	5882#	5901#	5902#	5903#	5904#	5906#	5907#
5917#	5919#	5920#	5930#	5932#	5933#	5943#	5945#	5946#	5956#	5958#	5959#	5971#	5972#	5973#
5974#	5976#	5977#	5986#	5987#	5988#	5989#	5991#	5992#	6001#	6004#	6034#	6036#	6037#	6051#
6053#	6054#	6061#	6063#	6064#	6071#	6073#	6074#	6081#	6083#	6084#	6090#	6092#	6093#	6101#
6103#	6104#	6111#	6113#	6114#	6122#	6124#	6125#	6132#	6134#	6135#	6143#	6145#	6146#	6153#
6155#	6156#	6164#	6166#	6167#	6174#	6176#	6177#	6185#	6187#	6188#	6195#	6197#	6198#	6206#

```
        6208#  6209#  6216#  6218#  6219#  6227#  6229#  6230#  6237#  6239#  6240#  6248#  6250#  6251#  6258#
        6260#  6261#  6269#  6271#  6272#  6279#  6281#  6282#  6290#  6292#  6293#  6300#  6302#  6303#  6311#
        6313#  6314#  6321#  6323#  6324#  6332#  6334#  6335#  6342#  6344#  6345#  6353#  6355#  6356#  6363#
        6365#  6366#  6374#  6376#  6377#  6385#  6387#  6388#  6396#  6398#  6399#  6402#  6422#  6430#  6432#
        6433#  6447#  6449#  6450#  6457#  6459#  6460#  6467#  6469#  6470#  6476#  6478#  6479#  6487#  6489#
        6490#  6496#  6498#  6499#  6507#  6509#  6510#  6519#  6520#  6521#  6522#  6524#  6525#  6530#  6531#
        6532#  6533#  6535#  6536#  6542#  6544#  6545#  6548#  6552#  6560#  6562#  6563#  6577#  6579#  6580#
        6587#  6589#  6590#  6597#  6599#  6600#  6606#  6608#  6609#  6617#  6619#  6620#  6628#  6630#  6631#
        6637#  6639#  6640#  6648#  6650#  6651#  6657#  6659#  6660#  6663#  6666#  6689#  6691#  6692#  6706#
        6708#  6709#  6716#  6718#  6719#  6726#  6728#  6729#  6735#  6737#  6738#  6746#  6748#  6749#  6759#
        6761#  6762#  6772#  6774#  6775#  6778#  6801#  6803#  6804#  6818#  6820#  6821#  6828#  6830#  6831#
        6838#  6840#  6841#  6847#  6849#  6850#  6858#  5860#  6861#  6873#  6875#  6876#  6884#  6886#  6887#
        6890#  6919#  6928#  6930#  6931#  6945#  6947#  6948#  6955#  6957#  6958#  6965#  6967#  6968#  6979#
        6981#  6982#  6996#  6998#  6999#  7007#  7009#  7010#  7018#  7020#  7021#  7040#  7041#  7042#  7043#
        7045#  7046#  7054#  7056#  7057#  7060#  7067#  7101#  7105#  7106   7107   7108   7110#  7111   7112
        7113   7115#  7116   7117   7118   7119   7122#  7140#  7144#  7161#  7162#  7163#
MSGNLS     1#   949#  5772#
MSGNSU     1#   949#  4496#  4630#  4800#  4920#  5061#  5178#  5597#  5817#  6421#  6551#  6918#
MSGNTA     1#   949#  1113#  1127#  3761#  3795#  3839#  3873#  3908#  3926#  3951#  3977#  4359#  4414#  4432#
        4447#  4460#  4623#  4762#  4765#  4914#  5034#  5037#  5172#  5284#  5287#  5569#  5780#  5783#  6000#
        6003#  6401#  6547#  6662#  6665#  6777#  6889#  7059#  7066#  7122#  7123   7144#  7145
MSGNTE     1#   949#  4491#  4795#  5056#  5324#  5592#  5812#  6026#  6418#  6680#  6793#  6911#
MSHAPT     1#   949#   976#
MSHNAP     1#   949#   976#  1015
MSINCR     1#   949#   952#  1097#  1122#  3728#  3739#  3755#  3762#  3767#  3776#  3786#  3792#  3796#  3802#
        3809#  3815#  3825#  3835#  3840#  3846#  3853#  3859#  3869#  3874#  3881#  3888#  3895#  3905#  3909#
        3915#  3922#  3927#  3933#  3940#  3947#  3952#  3958#  3965#  3973#  3978#  4009#  4019#  4026#  4036#
        4043#  4053#  4060#  4070#  4085#  4095#  4102#  4112#  4118#  4134#  4144#  4151#  4161#  4168#  4178#
        4185#  4195#  4211#  4221#  4228#  4238#  4251#  4264#  4285#  4292#  4299#  4306#  4327#  4356#  4360#
        4378#  4385#  4404#  4409#  4415#  4428#  4433#  4442#  4445#  4448#  4458#  4461#  4491#  4492#  4496#
        4497#  4505#  4507#  4519#  4521#  4528#  4530#  4538#  4540#  4553#  4555#  4563#  4565#  4576#  4581#
        4599#  4601#  4612#  4617#  4624#  4630#  4631#  4639#  4641#  4652#  4654#  4661#  4663#  4671#  4673#
        4686#  4688#  4696#  4698#  4709#  4714#  4728#  4730#  4738#  4740#  4751#  4756#  4763#  4766#  4795#
        4796#  4800#  4801#  4808#  4810#  4827#  4829#  4837#  4839#  4847#  4849#  4857#  4859#  4867#  4869#
        4878#  4880#  4889#  4891#  4900#  4902#  4909#  4911#  4915#  4920#  4921#  4928#  4930#  4947#  4949#
        4957#  4959#  4967#  4969#  4977#  4979#  4987#  4989#  4998#  5000#  5009#  5011#  5020#  5022#  5029#
        5031#  5035#  5038#  5056#  5057#  5061#  5062#  5069#  5071#  5091#  5093#  5114#  5116#  5125#  5127#
        5136#  5138#  5147#  5149#  5158#  5160#  5167#  5169#  5173#  5178#  5179#  5186#  5188#  5209#  5211#
        5226#  5228#  5237#  5239#  5248#  5250#  5259#  5261#  5270#  5272#  5279#  5281#  5285#  5288#  5324#
        5325#  5332#  5334#  5349#  5351#  5359#  5361#  5369#  5371#  5379#  5381#  5388#  5390#  5399#  5401#
        5409#  5411#  5420#  5422#  5435#  5437#  5450#  5452#  5465#  5467#  5479#  5481#  5489#  5491#  5499#
        5501#  5509#  5511#  5520#  5522#  5530#  5532#  5541#  5543#  5552#  5554#  5564#  5566#  5570#  5592#
        5593#  5597#  5598#  5605#  5611#  5613#  5641#  5643#  5650#  5652#  5659#  5661#  5668#  5670#  5688#
        5693#  5704#  5706#  5717#  5719#  5730#  5732#  5743#  5745#  5757#  5762#  5773#  5781#  5784#  5812#
        5813#  5817#  5818#  5824#  5826#  5852#  5854#  5861#  5863#  5870#  5872#  5879#  5881#  5901#  5906#
        5917#  5919#  5930#  5932#  5943#  5945#  5956#  5958#  5971#  5976#  5986#  5991#  6001#  6004#  6026#
        6027#  6034#  6036#  6051#  6053#  6061#  6063#  6071#  6073#  6081#  6083#  6090#  6092#  6101#  6103#
        6111#  6113#  6122#  6124#  6132#  6134#  6143#  6145#  6153#  6155#  6164#  6166#  6174#  6176#  6185#
        6187#  6195#  6197#  6206#  6208#  6216#  6218#  6227#  6229#  6237#  6239#  6248#  6250#  6258#  6260#
        6269#  6271#  6279#  6281#  6290#  6292#  6300#  6302#  6311#  6313#  6321#  6323#  6332#  6334#  6342#
        6344#  6353#  6355#  6363#  6365#  6374#  6376#  6385#  6387#  6396#  6398#  6402#  6418#  6419#  6421#
        6422#  6430#  6432#  6447#  6449#  6457#  6459#  6467#  6469#  6476#  6478#  6487#  6489#  6496#  6498#
        6507#  6509#  6519#  6524#  6530#  6535#  6542#  6544#  6548#  6551#  6552#  6560#  6562#  6577#  6579#
        6587#  6589#  6597#  6599#  6606#  6608#  6617#  6619#  6628#  6630#  6637#  6639#  6648#  6650#  6657#
        6659#  6663#  6666#  6680#  6681#  6689#  6691#  6706#  6708#  6716#  6718#  6726#  6728#  6735#  6737#
        6746#  6748#  6759#  6761#  6772#  6774#  6778#  6793#  6794#  6801#  6803#  6818#  6820#  6828#  6830#
```

```
                6838#  6840#  6847#  6849#  6858#  6860#  6873#  6875#  6884#  6886#  6890#  6911#  6912#  6918#  6919#
                6928#  6930#  6945#  6947#  6955#  6957#  6965#  6967#  6979#  6981#  6996#  6998#  7007#  7009#  7018#
                7020#  7040#  7045#  7054#  7056#  7060#  7067#  7101#  7140#
MSIOSE            1#   949#
MSLDRO            1#   949#  4284#  4291#  4298#  4305#  4326#  4403#  4408#
MSMASK            1#   949#
MSMCHI            1#   949#
MSMCLO            1#   949#
MSMSK1            1#   949#
MSPOP             1#   949#  1113#  1127#  3761#  3795#  3839#  3873#  3908#  3926#  3951#  3977#  4256#  4359#  4414#
                4432#  4447#  4460#  4623#  4762#  4765#  4914#  5034#  5037#  5172#  5284#  5287#  5569#  5772#  5780#
                5783#  6000#  6003#  6401#  6547#  6662#  6665#  6777#  6889#  7059#  7066#  7122#  7144#  7158#
MSPRIN            1#   949#  3735#  3750#  3771#  3780#  3789#  3804#  3812#  3820#  3829#  3848#  3856#  3864#  3883#
                3891#  3899#  3917#  3935#  3943#  3960#  3968#  4004#  4012#  4022#  4029#  4039#  4046#  4056#  4063#
                4080#  4088#  4098#  4105#  4115#  4129#  4137#  4147#  4154#  4164#  4171#  4181#  4188#  4206#  4214#
                4224#  4231#
MSPUSH            1#   949#   952#  1097#  1122#  3728#  3767#  3802#  3846#  3881#  3915#  3933#  3958#  4251#  4264#
                4378#  4428#  4442#  4458#  4491#  4492   4496#  4497   4630#  4631   4795#  4796   4800#  4801   4920#
                4921   5056#   057   5061#  5062   5178#  5179   5324#  5325   5592#  5593   5597#  5598   5605#  5812#
                5813   5817#  ,818   6026#  6027   6418#  6419   6421#  6422   6551#  6552   6680#  6681   6793#  6794
                6911#  6912   6918#  6919   7101#  7140#
MSPUT             1#   949#  3735#  3750#  3771#  3780#  3789#  3804#  3812#  3820#  3829#  3848#  3856#  3864#  3883#
                3891#  3899#  3917#  3935#  3943#  3960#  3968#  4004#  4012#  4022#  4029#  4039#  4046#  4056#  4063#
                4080#  4088#  4098#  4105#  4115#  4129#  4137#  4147#  4154#  4164#  4171#  4181#  4188#  4206#  4214#
                4224#  4231#  4381#
MSPUT1            1#   949#  3735#  3736   3737   3750#  3751   3752   3753   3771#  3772   3773   3774   3780#  3781
                3782   3783   3784   3789#  3790   3804#  3805   3806   3807   3812#  3813   3820#  3821   3822   3823
                3829#  3830   3831   3832   3833   3848#  3849   3850   3851   3856#  3857   3864#  3865   3866   3867
                3883#  3884   3885   3886   3891#  3892   3893   3899#  3900   3901   3902   3903   3917#  3918   3919
                3920   3935#  3936   3937   3938   3943#  3944   3945   3960#  3961   3962   3963   3968#  3969   3970
                3971   4004#  4005   4006   4007   4012#  4013   4014   4015   4016   4017   4022#  4023   4024   4029#
                4030   4031   4032   4033   4034   4039#  4040   4041   4046#  4047   4048   4049   4050   4051   4056#
                4057   4058   4063#  4064   4065   4066   4067   4068   4080#  4081   4082   4083   4088#  4089   4090
                4091   4092   4093   4098#  4099   4100   4105#  4106   4107   4108   4109   4110   4115#  4116   4129#
                4130   4131   4132   4137#  4138   4139   4140   4141   4142   4147#  4148   4149   4154#  4155   4156
                4157   4158   4159   4164#  4165   4166   4171#  4172   4173   4174   4175   4176   4181#  4182   4183
                4188#  4189   4190   4191   4192   4193   4206#  4207   4208   4209   4214#  4215   4216   4217   4218
                4219   4224#  4225   4226   4231#  4232   4233   4234   4235   4236   4381#  4382   4383   4384
MSRADI            1#   949#  7105#  7110#  7115#
MSRBRO            1#   949#
MSRNRO            1#   949#  4326#  4328
MSSETS            1#   949#   952#  1097#  1122#  3728#  3767#  3802#  3846#  3881#  3915#  3933#  3958#  4251#  4264#
                4378#  4428#  4442#  4458#  4492#  4497#  4631#  4796#  4801#  4921#  5057#  5062#  5179#  5325#  5593#
                5598#  5605#  5813#  5818#  6027#  6419#  6422#  6552#  6681#  6794#  6912#  6919#  7101#  7140#
MSSTAR            1#   949#
MSSVC             1#   949#  3735#  3739   3750#  3755   3761#  3762   3771#  3776   3780#  3786   3789#  3792   3795#
                3796   3804#  3809   3812#  3815   3820#  3825   3829#  3835   3839#  3840   3848#  3853   3856#  3859
                3864#  3869   3873#  3874   3883#  3888   3891#  3895   3899#  3905   3908#  3909   3917#  3922   3926#
                3927   3935#  3940   3943#  3947   3951#  3952   3960#  3965   3968#  3973   3977#  3978   4004#  4009
                4012#  4019   4022#  4026   4029#  4036   4039#  4043   4046#  4053   4056#  4060   4063#  4070   4080#
                4085   4088#  4095   4098#  4102   4105#  4112   4115#  4118   4129#  4134   4137#  4144   4147#  4151
                4154#  4161   4164#  4168   4171#  4178   4181#  4185   4188#  4195   4206#  4211   4214#  4221   4224#
                4228   4231#  4238   4284#  4285   4291#  4292   4298#  4299   4305#  4306   4326#  4327   4356#  4359#
                4360   4381#  4385   4403#  4404   4408#  4409   4414#  4415   4432#  4433   4445#  4447#  4448   4460#
                4461   4496#  4497   4505#  4507#  4519#  4521#  4528#  4530#  4538#  4540#  4553#  4555#  4563#  4565#
                4576   4581#  4599#  4601#  4612   4617#  4623#  4624   4630#  4631   4639#  4641#  4652#  4654#  4661#
```

```
        6847#  6849#  6858#  6860#  6873#  6875#  6884#  6886#  6890#  6919#  6928#  6930#  6945#  6947#  6955#
        6957#  6965#  6967#  6979#  6981#  6996#  6998#  7007#  7009#  7018#  7020#  7040#  7045#  7054#  7056#
        7060#  7067#
MSTSTL     1#   949#  3739#  3755#  3762#  3776#  3786#  3792#  3796#  3809#  3815#  3825#  3835#  3840#  3853#
        3859#  3869#  3874#  3888#  3895#  3905#  3909#  3922#  3927#  3940#  3947#  3952#  3965#  3973#  3978#
        4009#  4019#  4026#  4036#  4043#  4053#  4060#  4070#  4085#  4095#  4102#  4112#  4118#  4134#  4144#
        4151#  4161#  4168#  4178#  4185#  4195#  4211#  4221#  4228#  4238#  4285#  4292#  4299#  4306#  4327#
        4356#  4360#  4385#  4404#  4409#  4415#  4433#  4445#  4448#  4461#  4497#  4505#  4507#  4519#  4521#
        4528#  4530#  4538#  4540#  4553#  4555#  4563#  4565#  4576#  4581#  4599#  4601#  4612#  4617#  4624#
        4631#  4639#  4641#  4652#  4654#  4661#  4663#  4671#  4673#  4686#  4688#  4696#  4698#  4709#  4714#
        4728#  4730#  4738#  4740#  4751#  4756#  4763#  4766#  4801#  4808#  4810#  4827#  4829#  4837#  4839#
        4847#  4849#  4857#  4859#  4867#  4869#  4878#  4880#  4889#  4891#  4900#  4902#  4909#  4911#  4915#
        4921#  4928#  4930#  4947#  4949#  4957#  4959#  4967#  4969#  4977#  4979#  4987#  4989#  4998#  5000#
        5009#  5011#  5020#  5022#  5029#  5031#  5035#  5038#  5062#  5069#  5071#  5091#  5093#  5114#  5116#
        5125#  5127#  5136#  5138#  5147#  5149#  5158#  5160#  5167#  5169#  5173#  5179#  5186#  5188#  5209#
        5211#  5226#  5228#  5237#  5239#  5248#  5250#  5259#  5261#  5270#  5272#  5279#  5281#  5285#  5288#
        5332#  5334#  5349#  5351#  5359#  5361#  5369#  5371#  5379#  5381#  5388#  5390#  5399#  5401#  5409#
        5411#  5420#  5422#  5435#  5437#  5450#  5452#  5465#  5467#  5479#  5481#  5489#  5491#  5499#  5501#
        5509#  5511#  5520#  5522#  5530#  5532#  5541#  5543#  5552#  5554#  5564#  5566#  5570#  5598#  5605#
        5611#  5613#  5641#  5643#  5650#  5652#  5659#  5661#  5668#  5670#  5688#  5693#  5704#  5706#  5717#
        5719#  5730#  5732#  5743#  5745#  5757#  5762#  5773#  5781#  5784#  5818#  5824#  5826#  5852#  5854#
        5861#  5863#  5870#  5872#  5879#  5881#  5901#  5906#  5917#  5919#  5930#  5932#  5943#  5945#  5956#
        5958#  5971#  5976#  5986#  5991#  6001#  6004#  6034#  6036#  6051#  6053#  6061#  6063#  6071#  6073#
        6081#  6083#  6090#  6092#  6101#  6103#  6111#  6113#  6122#  6124#  6132#  6134#  6143#  6145#  6153#
        6155#  6164#  6166#  6174#  6176#  6185#  6187#  6195#  6197#  6206#  6208#  6216#  6218#  6227#  6229#
        6237#  6239#  6248#  6250#  6258#  6260#  6269#  6271#  6279#  6281#  6290#  6292#  6300#  6302#  6311#
        6313#  6321#  6323#  6332#  6334#  6342#  6344#  6353#  6355#  6363#  6365#  6374#  6376#  6385#  6387#
        6396#  6398#  6402#  6422#  6430#  6432#  6447#  6449#  6457#  6459#  6467#  6469#  6476#  6478#  6487#
        6489#  6496#  6498#  6507#  6509#  6519#  6524#  6530#  6535#  6542#  6544#  6548#  6552#  6560#  6562#
        6577#  6579#  6587#  6589#  6597#  6599#  6606#  6608#  6617#  6619#  6628#  6630#  6637#  6639#  6648#
        6650#  6657#  6659#  6663#  6666#  6689#  6691#  6706#  6708#  6716#  6718#  6726#  6728#  6735#  6737#
        6746#  6748#  6759#  6761#  6772#  6774#  6778#  6801#  6803#  6818#  6820#  6828#  6830#  6838#  6840#
        6847#  6849#  6858#  6860#  6873#  6875#  6884#  6886#  6890#  6919#  6928#  6930#  6945#  6947#  6955#
        6957#  6965#  6967#  6979#  6981#  6996#  6998#  7007#  7009#  7018#  7020#  7040#  7045#  7054#  7056#
        7060#  7067#
MSWORD     1#   949#  1015#  1024   1074#  1076   1077   1078   1079   1080   1081   1082   1083   1084   1085
        1086   4576#  4577   4578   4579   4612#  4613   4614   4615   4709#  4710   4711   4712   4751#  4752
        4753   4754   5688#  5689   5690   5691   5757#  5758   5759   5760   5901#  5902   5903   5904   5971#
        5972   5973   5974   5986#  5987   5988   5989   6519#  6520   6521   6522   6530#  6531   6532   6533
        7040#  7041   7042   7043   7105#  7110#  7115#  7162   7163
MSXFER     1#   949#
NEWTST  1640#  4462   4767   5039   5299   5571   5793   6005   6403   6667   6779   6891
NTST    1640#  4462   4767   5039   5299   5571   5793   6005   6403   6667   6779   6891
OPEN       1#   949#
POINTE     1#   949#   972
PRINTB     1#   949#  3770   3779   3788   3803   3811   3819   3828   3847   3855   3863   3882   3890   3898
        3916   3934   3942   3959   3967   4114
PRINTF     1#   949#
PRINTS     1#   949#
PRINTX     1#   949#  3734   3749   4003   4011   4021   4028   4038   4045   4055   4062   4079   4087   4097
        4104   4128   4136   4146   4153   4163   4170   4180   4187   4205   4213   4223   4230
READBU     1#   949#
READEF     1#   949#  4283   4290   4297   4304
RFLAGS     1#   949#
SETDF   1640#  2065   2109   2156   2227   2326   2598   2702   2735   2746   2780   2791   2825   2836   2870
        2881   2915   2926   2958   2969   3005   3016   3029   3040   3072   3084   3132   3380   3393   3419
```

```
        3429    3442    3452    3465    3475    3488    3498    3511    3521
SETHRD  1640#
SETPRI     1#    949#
SETSF   1640#
SETSFT  1640#
SETVEC     1#    949#    4380
SLASH      1#    949#    1066    1070
STARS      1#    949#
SVC        1#    947#     948
T$GEN   1640#   2065    2109    2156    2227    2326    2598    2702    2735    2746    2780    2791    2825    2836    2870
        2881    2915    2926    2958    2969    3005    3016    3029    3040    3072    3084    3132    3380    3393    3419
        3429    3442    3452    3465    3475    3488    3498    3511    3521
XFER       1#    949#
XFERF      1#    949#
XFERT      1#    949#
$GEDF   1640#   4575    4611    4708    4750    5687    5756    5900    5970    5985    6518    6529    7039
$GEHRD  1640#
$GESF   1640#
$GESFT  1640#
$GTDF   1640#   2064    2108    2155    2226    2325    2597    2701    2734    2745    2779    2790    2824    2835    2869
        2880    2914    2925    2957    2968    3004    3015    3028    3039    3071    3083    3131    3379    3392    3418
        3428    3441    3451    3464    3474    3487    3497    3510    3520
$GTHRD  1640#
$GTSF   1640#
$GTSFT  1640#

. ABS.  034322     000

 ERRORS DETECTED:  0

 CVDMDA.BIN,CVDMDA.SEQ/CRF/SOL=SVC34R.MAC,CVDMDA.P11
 RUN-TIME:  32 41 5 SECONDS
 RUN-TIME RATIO: 130/79=1.6
 CORE USED:  21K  (41 PAGES)
```