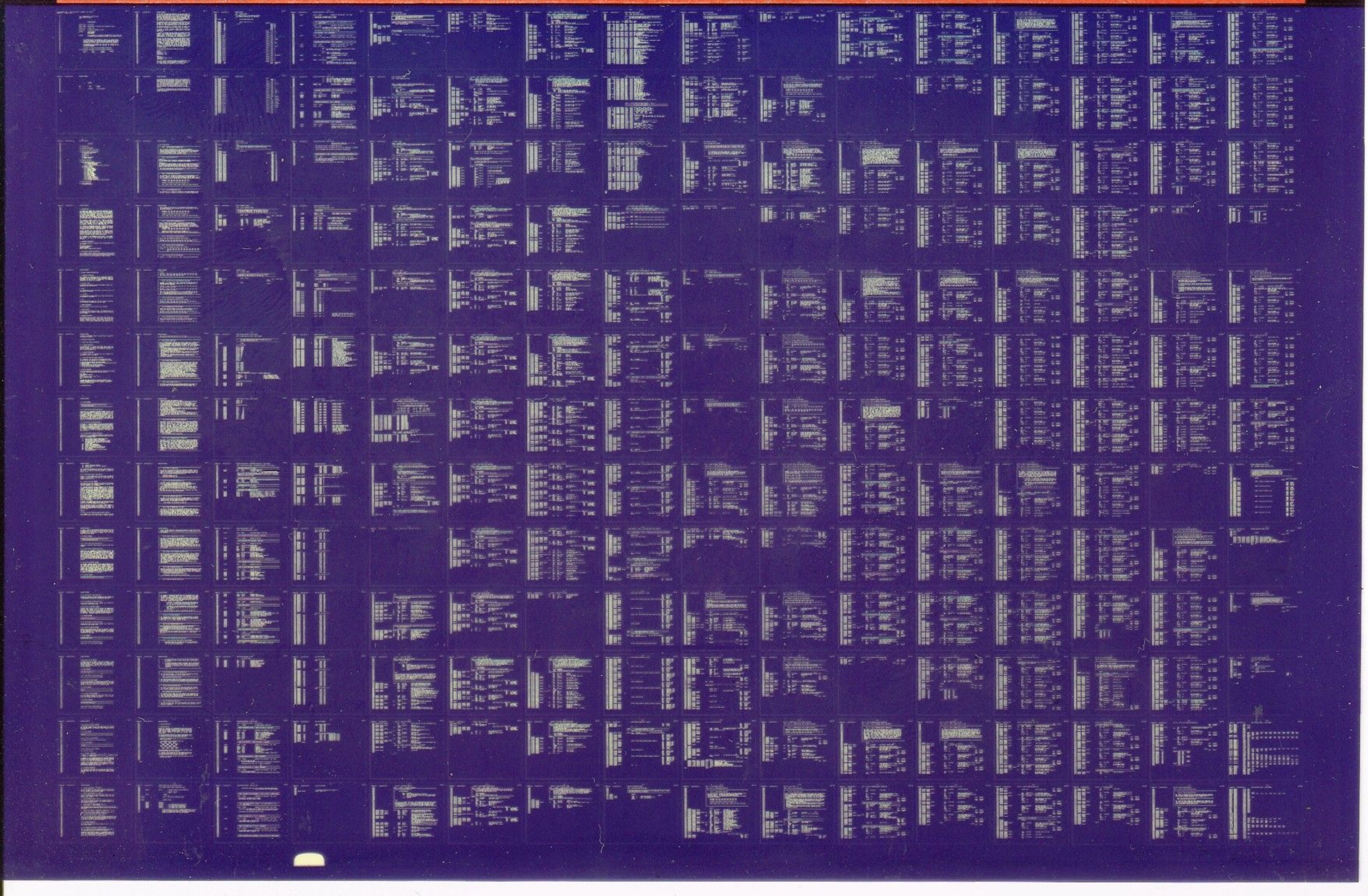
DMV11 DMV11 LINE UNIT DIAG 1 AH-F268A MC
M8053 M8064 CVDMCA0 FICHE 1 OF 2



DMV11 DMV11 LINE UNIT DIAG 1 AH-F268A-MC
M8053 M8064 CVDMCA0 FICHE 2 OF 2

```
                        .TITLE CVDMCA0 DMV11 LINE UNIT DIAG1
                        .SBTTL PROGRAM DOCUMENT
                        .REM ^
```

IDENTIFICATION

PRODUCT CODE: AC-F267A-MC

PRODUCT NAME: CVDMCA0 DMV-11 LINE UNIT STATIC DIAGNOSTIC PART 1

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)

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 BASIC DIAGNOSTIC TESTING OF THE VIA, FIFO, AND USYRT (BCP/BOP MODES) ON THESE BOARDS. THE FOLLOWING FUNCTIONS WILL BE PERFORMED: USYRT REGISTER ADDRESSING, USYRT REGISTER STATIC BIT INTERACTION AND READ/WRITE TESTING, AND BASIC BOP AND BCP TX TESTING (USING THE TSO STATUS BIT).

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 (CVDMC) SHOULD BE THE THIRD OF THE FIVE DMV-11 STATIC DIAGNOSTICS TO BE RUN (CVDMA/CVDMB SHOULD BE RUN FIRST). ERRORS FOUND IN THIS PROGRAM SHOULD BE CORRECTED BEFORE RUNNING ANY OF THE OTHER LINE UNIT DIAGNOSTICS (CVDMD OR 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 FIVE SECONDS PER PASS FOR EACH UNIT.

4.3 XXDP+

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

4.4 ACT/SLIDE

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

4.5 APT

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

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM.

4.7 ERROR LOGGING

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

5.0 PROGRAM LOAD MEDIA

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

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

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

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

6.1.2 STARTING PROCEDURES

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

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

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

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

6.2 INITIAL DIALOGUE

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

```
DRS LOADED
DIAG. RUN-TIME SERVICES
CVDMC-A-0
DMV-11 LINE UNIT TESTS - PART 1 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

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

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

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

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

6.3.1.5 EFFECT OF START COMMAND

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

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

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

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

EXAMPLE:

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

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

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

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

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

6.3.2.3 EFFECT OF RESTART COMMAND

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

6.3.3 CONTINUE COMMAND

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

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

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

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

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

6.3.3.3 EFFECT OF CONTINUE COMMAND

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

6.3.4 PROCEED COMMAND

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

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

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

6.3.5 ADD COMMAND

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

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

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

6.3.5.2 EFFECT OF ADD COMMAND

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

6.3.6 DROP COMMAND

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

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

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

6.3.6.2 EFFECT OF DROP COMMAND

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

6.3.7 PRINT COMMAND

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

6.3.7.1 EFFECT OF PRINT COMMAND

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

6.3.8 DISPLAY COMMAND

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

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

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

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

1. DEVICE CSR ADDRESS : (0) 160020?

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

2. DEVICE VECTOR ADDRESS : (0) 300 ?

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

3. DEVICE PRIORITY LEVEL : (0) 4 ?

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

4. SWITCH PACK # 1 (BOOT ADDRESS): (0) 0 ?
5. SWITCH PACK # 2 (DDCMP ADDRESS): (0) 0 ?

THESE REPRESENT THE TWO USER MODIFIABLE 8 POSITION DIP-SWITCHES. THE ALLOWABLE RANGES FOR BOTH ARE 000-377 (OCTAL), AND THE DEFAULTS ARE BOTH 0.

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

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

7. BAUD RATE (0=LOW (19.2K), 1=HIGH (56K)):

THIS IS THE SPEED AT WHICH THE DMV TRANSMITS AND RECEIVES DATA. IN THE UNIT IS AN M8064, THE ANSWER IS IGNORED (DEFAULTS TO 56K). IF THE UNIT IS AN M8053, THEN THE ANSWER SHOULD BE BASED ON THE 'SPEED SELECT SWITCH' LOCATED ON THE BOARD.

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).

```
739
740
741                     7.0 TEST DESCRIPTIONS
742
743                     ;*****************************************************************************
744                     ;*      TEST 1  <TBMT MICROCODE INTERRUPT TEST>
745                     ;*
746                     ;*  THIS TEST CHECKS THE OPERATION OF THE TBMT (IRQ) INTERRUPT.
747                     ;*  THIS IS DONE BY ISSUEING THE 'SET MAINTENANCE INTERRUPT FLAG AND CLEAR
748                     ;*  INTERRUPT DISABLE IN PROCESSOR STATUS'' COMMAND WHILE IN THE MAINTENANCE
749                     ;*  LOOP AND THEN CHECKING FOR BIT 7 OF BSEL3 TO BE SET (THE BIT IS SET
750                     ;*  BY THE MICROCODE WHEN THE TBMT INTERRUPT OCCURS).
751                     ;*****************************************************************************
752
753
754                     ;*****************************************************************************
755                     ;*      TEST 2  <SWITCH SETTING TEST>
756                     ;*
757                     ;* SUBTEST #1:
758                     ;* THE TWO READABLE SWITCH PACKS WILL BE SAMPLED AND COMPARED AGAINST THE 2
759                     ;* VALUES IN THE P-TABLE. AN ERROR IS REPORTED ON A MISMATCH.
760                     ;*
761                     ;* SUBTEST #2:
762                     ;* THE SPEED SELECT SWITCH (SPDSEL) IS READ VIA THE VIAORA REGISTER (BIT PA4)
763                     ;* AND COMPARED AGAINST THE BAUD RATE VALUE IN THE P-TABLE. IF A MISMATCH
764                     ;* OCCURS IT WILL BE REPORTED. NOTE: THIS SUBROUTINE IS NOT RUN IF AN M8064
765                     ;* BOARD IS BEING TESTED (IT ONLY RUNS 56K... MAKING A SPEED SWITCH USELESS).
766                     ;*
767                     ;*  THIS TEST IS ONLY RUN ON THE FIRST PASS AFTER A 'START'' OR 'RESTART''.
768                     ;*  ALL SUCCESSIVE PASSES WILL SKIP THIS TEST.
769                     ;*****************************************************************************
770
771
772                     ;*****************************************************************************
773                     ;*      TEST 3  <USYRT MASTER CLEAR TEST>
774                     ;*
775                     ;*  ALL  REGISTERS  ARE  LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR
776                     ;*  PATTERN  F  BELOW.  THE USYRT IS THEN CLEARED BY A MASTER CLEAR
777                     ;*  (BIT 6 OF BSEL 1). ALL REGISTERS ARE THEN CHECKED FOR THE PROPER CONTENTS.
778                     ;*  THE INITIALIZED STATE OF THE REGISTERS IS CHECKED AGAINST DATA PATTERN F.
779                     ;*
780                     ;*      PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
781                     ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.
782                     ;*
783                     ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:
784                     ;*
785                     ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
786                     ;*
787                     ;*****************************************************************************
788
789
790                     ;*****************************************************************************
791                     ;*      TEST 4  <USYRT PROGRAM RESET TEST>
792                     ;*
793                     ;*  ALL  REGISTERS  ARE  LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR
794                     ;*  PATTERN  F  BELOW.  THE USYRT IS THEN RESET BY ASSERTING PROGRAM RESET
```

```
795                     ;*  (BIT 0 @ A000) IN THE 6522 VIA. ALL REGISTERS ARE THEN CHECKED FOR
796                     ;*  THE PROPER CONTENTS. THE INITIALIZED STATE OF THE REGISTERS IS CHECKED
797                     ;*  AGAINST DATA PATTERN F.
798                     ;*
799                     ;*      PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
800                     ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.
801                     ;*
802                     ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:
803                     ;*
804                     ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
805                     ;*
806                     ;*************************************************************************
807
808
809                     ;*************************************************************************
810                     ;*      TEST 5  <USYRT REGISTER ADDRESSING TEST>
811                     ;*
812                     ;* FIRST, A MASTER CLEAR IS ISSUED, TO INITIALIZE THE USYRT REGS TO
813                     ;* PATTERN F. THEN, EACH REGISTER IS WRITTEN WITH A BYTE OF PATTERN J,
814                     ;* AND AFTER EACH IS WRITTEN, ALL ARE READ AND COMPARED TO THE CURRENT
815                     ;* EXPECTED VALUES. THIS IS PERFORMED FOR ALL REGISTERS -- INCLUDING THE
816                     ;* READ ONLY REGS -- IN ORDER TO MAKE SURE THAT EACH REGISTER ONLY RESPONDS
817                     ;* TO ITS OWN ADDRESS.
818                     ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110
819                     ;*      PATTERN J: 000, 000, 001, 002, 004, 020, 040, 010
820                     ;*
821                     ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS F & J:
822                     ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
823                     ;*************************************************************************
824
825
826                     ;*************************************************************************
827                     ;*      TEST 6  <R/W BIT TEST OF PCSAR HIGH BYTE>
828                     ;*
829                     ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN G.
830                     ;*
831                     ;*  PATTERN  G:  000,  001, 003, 004, 005, 007, 100, 101, 103, 104, 105,
832                     ;*  107, 000, 017, 027, 041, 200, 277, 103, 144, 115, 157, 000.
833                     ;*************************************************************************
834
835
836                     ;*************************************************************************
837                     ;*      TEST 7  <R/W BIT TEST OF S/AR REGISTER>
838                     ;*
839                     ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
840                     ;*
841                     ;*  PATTERN  H:  125, 252, 000, 377, 000, 001, 002, 004, 010, 020, 040, 100,
842                     ;*               200, 000, 377, 376, 375, 373, 367, 357, 337, 277, 177, 377,
843                     ;*               000
844                     ;*************************************************************************
845
846
847                     ;*************************************************************************
848                     ;*      TEST 8  <R/W BIT TEST OF PCR REGISTER>
849                     ;*
850                     ;*  PATTERN  I  IS  LOADED  INTO  PCR  (HIGH)  AND THE DATA READ BACK AND
```

```
851                     ;*  CHECKED.
852                     ;*
853                     ;*  PATTERN  I:  000,  041, 102, 143, 204, 245, 306, 347, 000, 001, 002,
854                     ;*  004,  040,  100,  200,  000, 346, 345, 343, 307, 247, 147, 347, 242,
855                     ;*  105, 347, 010, 020, 367, 357, 030, 027, 377.
856                     ;*
857                     ;*****************************************************************************
858
859
860                     ;*****************************************************************************
861                     ;*      TEST 9  <R/W BIT TEST OF TDSR REGISTER'S HIGH BYTE>
862                     ;*
863                     ;*  PATTERN  K  IS  LOADED  INTO  TDSR  (HIGH) AND THE DATA READ BACK IS
864                     ;*  COMPARED  AGAINST PATTERN L. (UNPREDICTABLE BITS ARE MASKED OFF TO 0
865                     ;*  WHEN READING FOR COMPARISON.)
866                     ;*
867                     ;*  PATTERN  K:  000,  377, 376, 375, 373, 376, 177, 377, 000, 001, 002,
868                     ;*  004, 010, 200, 125, 252, 000.
869                     ;*
870                     ;*  PATTERN  L:  000,  017, 016, 015, 013, 016, 017, 017, 000, 001, 002,
871                     ;*  004, 010, 000, 005, 012, 000.
872                     ;*
873                     ;*  NOTE  THAT  THE  UNDEFINED BITS (12, 13, & 14) ARE MASKED OFF TO 0'S
874                     ;*  FOR THE COMPARISON. ALSO THAT BIT 15 IS A READ/ONLY BIT AND CAN'T BE
875                     ;*  SET -- THEREFORE SHOULD ALWAYS BE READ AS A 0 BY THIS TEST.
876                     ;*****************************************************************************
877
878
879                     ;*****************************************************************************
880                     ;*      TEST 10 <R/W BIT TEST OF TXDB REGISTER>
881                     ;*
882                     ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
883                     ;*
884                     ;*  PATTERN  H:  000,  001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
885                     ;*  376, 375, 373, 367, 357, 337, 277, 177, 377, 000
886                     ;*****************************************************************************
887
888
889                     ;*****************************************************************************
890                     ;*      TEST 11 <PSEUDO R/W BIT TEST OF RXDB>
891                     ;*
892                     ;*  WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
893                     ;*  PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
894                     ;*
895                     ;*  PATTERN  H:  000,  001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
896                     ;*  376, 375, 373, 367, 357, 337, 277, 177, 377, 000
897                     ;*****************************************************************************
898
899
900                     ;*****************************************************************************
901                     ;*      TEST 12 <PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE>
902                     ;*
903                     ;*  WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
904                     ;*  PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
905                     ;*
906                     ;*  PATTERN  H:  000,  001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
```

```
907                     ;*  376, 375, 373, 367, 357, 337, 277, 177, 377, 000
908                     ;*****************************************************************************
909
910
911                     ;*****************************************************************************
912                     ;*      TEST 13 <NULL CLOCK TEST>
913                     ;*
914                     ;* FIRST, A MASTER CLEAR IS DONE TO INIT THE DMV. THEN, THE T1 TIMER ON THE
915                     ;* VIA CHIP IS PROGRAMMED FOR SQUARE WAVE CLOCK GENERATION ON PB7 (BIT 7
916                     ;* OF VIA OUTPUT REG B), WITH A BAUD RATE = 56 KBAUD. THIS IS THE MODE OF
917                     ;* VIA OPERATION WHICH IS USED TO GENERATE THE NULL CLOCK. THEN, THE PROGRAM
918                     ;* SCANS ORB REPEATEDLY TO MONITOR THE NULL CLOCK BIT, IN THE FOLLOWING
919                     ;* SEQUENCE :
920                     ;* - THE PROGRAM REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE, AND
921                     ;*   IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC (A GROSS TIMEOUT
922                     ;*   INTERVAL), AN ERROR IS REPORTED. (AT 56 KBAUD, THE CLOCK SHOULD
923                     ;*   HAVE A PERIOD OF ABOUT 18 MICRO-SEC.)
924                     ;* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 0 STATE,
925                     ;*   AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC, AN ERROR IS
926                     ;*   REPORTED.
927                     ;* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE
928                     ;*   AGAIN, AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC,
929                     ;*   AN ERROR IS REPORTED.
930                     ;*****************************************************************************
931
932
933                     ;*****************************************************************************
934                     ;*      TEST 14 <BCP TX RESET W/IDLE = 0>
935                     ;*
936                     ;*  THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL'' (BCP) WITH IDLE
937                     ;*  SET TO ZERO AND A 125 SYNC CHARACTER IS LOADED INTO S/AR. A 226 SYNC
938                     ;*  CHARACTER IS LOADED INTO TXDB SO THAT THE SOURCE OF SYNC CHARACTERS
939                     ;*  CAN BE LATER DETERMINED. THE VALID STATE OF THE USYRT REGISTERS IS
940                     ;*  READ AND CHECKED. TXE IS ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
941                     ;*   THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
942                     ;*  TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
943                     ;*   (TXBE SHOULD GO HIGH; AT THIS TIME THE S/AR'S SYNC CHARACTER SHOULD
944                     ;*  BE LOADED INTO TXSO AND TSOM IS AGAIN SET -- DRIVING TXBE LOW.)
945                     ;*   THREE SYNC CHARACTERS ARE SENT/RECEIVED: THE FIRST TWO SYNCHRONIZE
946                     ;*  THE RECEIVER, THE THIRD IS DIRECTLY READ (STRIP SYNC IS OFF) AND
947                     ;*  COMPARED AGAINST 125 (THE S/AR SYNC CHARACTER).
948                     ;*   IF VALUE READ IS 226, THEN TXDB PROVIDED THE SYNC (IE: ERROR).
949                     ;*   THE USYRT IS THEN RESET AND REGISTERS ARE AGAIN READ AND CHECKED.
950                     ;*   THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
951                     ;*  ONE MARK AND THREE SYNC CHARACTERS (FROM THE S/AR) IS TRANSMITTED.
952                     ;*   ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
953                     ;*  SEQUENCE.
954                     ;*****************************************************************************
955
956
957                     ;*****************************************************************************
958                     ;*      TEST 15 <BCP TX RESET W/IDLE = 1>
959                     ;*
960                     ;*  THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL'' (BCP) WITH IDLE
961                     ;*  SET TO ONE AND A 226 SYNC CHARACTERS LOADED INTO S/AR AND TXDB.
962                     ;*  THE VALID STATE OF THE USYRT REGISTERS IS READ AND CHECKED. TXE IS
```

```
963                     ;*  ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
964                     ;*   THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
965                     ;*  TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
966                     ;*   (TXBE SHOULD GO HIGH; AT THIS TIME THE TXDB SYNC CHARACTER SHOULD
967                     ;*  BE LOADED INTO TXSO AND TSOM IS AGAIN SET -- DRIVING TXBE LOW.)
968                     ;*   AFTER THE RECEIVER IS SYNCHRONIZED (TWO SYNC CHARACTERS), TXDB IS
969                     ;*  LOADED WITH A 125 AND, WITH TSOM STILL = 1 (SYNC SOURCE = TXDB), THE
970                     ;*  USYRT IS AGAIN CLOCKED.
971                     ;*   AT THIS POINT, IF THE IDLE BIT WORKED, THE VALUE 125 WILL BE READ
972                     ;*  BY THE RECEIVER. OTHERWISE A 226 WILL BE READ, INDICATING TXDB WASN'T
973                     ;*  PROVIDING THE SYNC CHARACTERS.
974                     ;*   WHEN TXBE GOES HIGH AGAIN, THE USYRT IS RESET.
975                     ;*   ALL REGISTERS ARE AGAIN READ AND CHECKED.
976                     ;*   THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
977                     ;*  ONE MARK AND THREE SYNCS (226,226,125 FROM TXDB) ARE TRANSMITTED.
978                     ;*   ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
979                     ;*  SEQUENCE.
980                     ;*****************************************************************************
981
982
983                     ;*****************************************************************************
984                     ;*       TEST 16 <BCP TX UNDERRUN W/TSOM TERMINATION>
985                     ;*
986                     ;*  THE  USYRT  IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY
987                     ;*  CONTROLLED  UNTIL  TWO  SYNC CHARACTERS AND ONE DATA CHARACTER (000)
988                     ;*  HAVE  BEEN  TRANSMITTED.  AT  THIS TIME WHEN TXBE IS ASSERTED BY THE
989                     ;*  USYRT,  NO  DATA  IS  LOADED  INTO  TXDB  --  FORCING  AN  UNDER RUN
990                     ;*  CONDITION.  TXU  AND  TERR  ARE  CHECKED BOTH BEFORE AND AFTER THEIR
991                     ;*  EXPECTED  ASSERTIONS.  AFTER  THE  FIRST  NON-DATA  CHARACTER (WHICH
992                     ;*  SHOULD  BE  THE  MARK CHARACTER) HAS BEEN STARTED, IDLE IS SET TO 0.
993                     ;*  THIS SHOULD FORCE THE NEXT NON-DATA CHARACTER TO BE A SYNC CHARACTER
994                     ;*  FROM  S/AR.  WHILE THIS SYNC CHARACTER IS BEING TRANSMITTED, TSOM IS
995                     ;*  ASSERTED  (CLEARING TXU AND TERR) -- IDLE IS LEFT AT 0. TXBE IS THEN
996                     ;*  CYCLED  THROUGH AT LEAST ONE MORE SYNC CHARACTER AND THE TEST IS
997                     ;*  ABORTED. ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS
998                     ;*  WITHIN THE SEQUENCE.
999                     ;*      NOTE: BITS SHIFT OUT OF TX LSB FIRST.
1000                    ;*****************************************************************************
1001
1002
1003                    ;*****************************************************************************
1004                    ;*       TEST 17 <BCP TX UNDERRUN W/RESET TERMINATION>
1005                    ;*
1006                    ;*  THE  USYRT  IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY
1007                    ;*  CONTROLLED  UNTIL  TWO  SYNC  CHARACTERS AND ONE DATA CHARACTER HAVE
1008                    ;*  BEEN  TRANSMITTED.  AT THIS TIME WHEN TXBE IS ASSERTED BY THE USYRT,
1009                    ;*  NO  DATA  IS LOADED INTO TXDB -- FORCING AN UNDER RUN CONDITION. TXU
1010                    ;*  AND  TERR  ARE  CHECKED  BOTH  BEFORE  AND  AFTER  THEIR  EXPECTED
1011                    ;*  ASSERTIONS.  AFTER THE FIRST NON-DATA CHARACTER (WHICH SHOULD BE THE
1012                    ;*  MARK  CHARACTER)  HAS  BEEN  STARTED,  IDLE IS SET TO 0. THIS SHOULD
1013                    ;*  FORCE  THE  NEXT  NON-DATA  CHARACTER  TO  BE  A  SYNC  CHARACTER.
1014                    ;*  IMMEDIATELY  AFTER  THIS  SYNC  CHARACTER  HAS  BEING TRANSMITTED, A
1015                    ;*  PROGRAM RESET IS ISSUED AND ALL REGISTERS ARE CHECKED. ERROR
1016                    ;*  LOOPING  WILL  DEPEND  ON  WHERE  THE FIRST ERROR OCCURS WITHIN THE
1017                    ;*  SEQUENCE.
1018                    ;*****************************************************************************
```

```
1019
1020
1021                ;*********************************************************************************
1022                ;*      TEST 18 <BCP TX DISABLE TEST>
1023                ;*
1024                ;*  THE  USYRT IS INITIALIZED FOR BCP AND A MESSAGE IS STARTED. ONCE THE
1025                ;*  SECOND  DATA  CHARACTER  IS LOADED INTO TXDB TXE IS DROPPED. TXSO IS
1026                ;*  WATCHED TO ASSURE THAT THE CHARACTER BEING TRANSMITTED IS COMPLETED.
1027                ;*  WHEN  IT  IS, THE USYRT SHOULD DROP TXA AND STOP TRANSMITTING -- THE
1028                ;*  LAST CHARACTER LOADED INTO TXDB SHOULD BE LOST.
1029                ;*
1030                ;*      CHARACTERS LOADED: 125 252
1031                ;*      CHARACTERS TRANSMITTED: 125
1032                ;*********************************************************************************
1033
1034
1035                ;*********************************************************************************
1036                ;*      TEST 19 <FIFO STACKING CHARACTERS TEST>
1037                ;*
1038                ;*  THE USYRT IS SETUP FOR BCP MODE WITH NO ERROR DETECTION.
1039                ;*  THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS
1040                ;* TO FILL THE 8 CHARACTER RECEIVER FIFO WITH THE CHARACTERS:
1041                ;*      1/2(SYNCH),000,377,125,252,347,030,303,1/2(074).
1042                ;*  THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. NOTE
1043                ;* THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE  CHARACTERS SINCE THEY
1044                ;* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
1045                ;* RDA = 0).
1046                ;*  ALSO NOTE THAT DUE TO FIFO TIMING, TWO 'HALF CHARACTERS' ARE LOADED
1047                ;* INTO THE FIFO (THE 1ST AND LAST CHARACTERS).
1048                ;*
1049                ;*********************************************************************************
1050
1051
1052                ;*********************************************************************************
1053                ;*      TEST 20 <BCP CHARACTER LENGTH TEST>
1054                ;*
1055                ;*  THE USYRT IS INITIALIZED FOR BCP WITH NO ERROR CHECKING. TXC IS MANUALLY
1056                ;*  CONTROLLED UNTIL TWO SYNC CHARACTERS HAVE BEEN TRANSMITTED.  THEN  3
1057                ;*  SUBTESTS FOLLOW, EACH ONE USING A DIFFERENT TRANSMIT CHARACTER LENGTH
1058                ;*  STARTING AT FIVE (5) AND ENDING WITH SEVEN (7).
1059                ;*
1060                ;*       TEST PATTERN: 111 222 333 044 155 266 377
1061                ;*********************************************************************************
1062
1063
1064                ;*********************************************************************************
1065                ;*      TEST 21 <BOP TX TABORT/(IDLE = 0) TEST>
1066                ;*
1067                ;*  THE USYRT IS INITIALIZED FOR "BIT-ORIENTED PROTOCOL" (BOP) WITH IDLE
1068                ;*  SET TO ZERO. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
1069                ;*  OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
1070                ;*  NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
1071                ;*  LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
1072                ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
1073                ;*  ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
1074                ;*  THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
```

```
1075                    ;*  FLAGS, ONE ZERO CHARACTER, AND ONE ABORT CHARACTER IS SENT (INTO THE BIT
1076                    ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
1077                    ;*  THE SEQUENCE.
1078                    ;*****************************************************************************
1079
1080
1081                    ;*****************************************************************************
1082                    ;*      TEST 22 <BOP TX TABORT/(IDLE = 1) TEST>
1083                    ;*
1084                    ;*  THE USYRT IS INITIALIZED FOR "BIT-ORIENTED PROTOCOL" (BOP) WITH IDLE
1085                    ;*  SET TO ONE. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
1086                    ;*  OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
1087                    ;*  NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
1088                    ;*  LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
1089                    ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
1090                    ;*  ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
1091                    ;*  THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
1092                    ;*  FLAGS, ONE ZERO CHARACTER, AND ONE FLAG CHARACTER IS SENT (INTO THE BIT
1093                    ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
1094                    ;*  THE SEQUENCE.
1095                    ;*****************************************************************************
1096
1097
1098                    ;*****************************************************************************
1099                    ;*      TEST 23 <BOP TX TXGA (TRANSMIT GO-AHEAD) TEST>
1100                    ;*
1101                    ;*  THE USYRT IS INITIALIZED FOR BOP AND TXE IS ASSERTED. TSOM IS ASSERTED
1102                    ;*  AND TXA OBSERVED -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
1103                    ;*  NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
1104                    ;*  LOADED  INTO  TXDB DRIVING TXBE LOW. WHEN TXBE GOES HIGH AGAIN, TXGA
1105                    ;*  IS  ASSERTED CAUSING GA TO BE TRANSMITTED. THE SEQUENCE OF EVENTS IS
1106                    ;*  CONTINUALLY  MONITORED  WHILE  TXC  IS  MANUALLY  CONTROLLED AND ALL
1107                    ;*  CHARACTERS  ARE  CHECKED  AT TXSO. THIS TEST WILL GO NO FURTHER INTO
1108                    ;*  THE  TRANSMIT  SEQUENCE  SO  THAT  ONLY TWO FLAGS, ONE ZERO CHARACTER
1109                    ;*  AND  ONE GA CHARACTER IS SENT (INTO THE BIT BUCKET).
1110                    ;*  ERROR  LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
1111                    ;*  THE SEQUENCE.
1112                    ;*****************************************************************************
1113
1114
1115                    ;*****************************************************************************
1116                    ;*      TEST 24 <BOP TX MESSAGE WITHOUT CRC>
1117                    ;*
1118                    ;*  THE USYRT IS INITIALIZED FOR BOP MODE WITH NO ERROR DETECTION. TXC IS THEN
1119                    ;*  MANUALLY CONTROLLED UNTIL TWO FLAG CHARACTERS HAVE BEEN TRANSMITTED. THEN
1120                    ;*  A 5 CHARACTER MESSAGE IS TRANSMITTED, RECEIVED, CHECKED, AND TERMINATED
1121                    ;*  (WITH TEOM). A CHECK IS MADE TO ASCERTAIN THAT NO CRC OR VRC IS GENERATED
1122                    ;*  -- FLAG CHARACTERS SHOULD FOLLOW THE DATA.
1123                    ;*       (NOTE: NO BIT STUFFING OCCURS IN THIS TEST)
1124                    ;*
1125                    ;*      TEST MESSAGE: FLAG FLAG 000 307 125 252 201 FLAG
1126                    ;*****************************************************************************
1127
1128
1129                    ;*****************************************************************************
1130                    ;*      TEST 25 <BOP RX CHARACTER LENGTH TEST>
```

```
1131
1132                    ;*
1133                    ;*  THE  USYRT  IS INITIALIZED FOR BOP WITH CRC-CCITT PRESET TO 1'S. TXC
1134                    ;*  IS   MANUALLY   CONTROLLED  UNTIL  TWO  FLAG  CHARACTERS  HAVE  BEEN
1135                    ;*  TRANSMITTED.  THEN  6  SUBTESTS  FOLLOW,  EACH ONE USING A DIFFERENT
1136                    ;*  TRANSMIT  CHARACTER LENGTH STARTING AT TWO (2) AND ENDING WITH SEVEN
1137                    ;*  (7).  IN  EACH  SUBTEST,  TWO  8  BIT CHARACTERS WILL BE TRANSMITTED
1138                    ;*  BEFORE  TXCL  IS  CHANGED TO THE CHARACTER LENGTH BEING TESTED. THIS
1139                    ;*  CORRESPONDS TO NORMAL USAGE WHERE EITHER:
1140                    ;*
1141                    ;*        1 -- A  MESSAGE  OF  CHARACTERS  WHICH ARE LESS THEN 8 BITS IS
1142                    ;*             SENT  AS  A  STREAM OF 8 BIT CHARACTERS AND THE REMAINING
1143                    ;*             BITS ARE SENT AS A CHARACTER OF LESS THEN 8 BITS OR
1144                    ;*
1145                    ;*        2 -- A  HEADER  OF  TWO 8 BIT CHARACTERS IS SENT FOLLOWED BY A
1146                    ;*             DATA  STREAM  OF DATA CHARACTERS WHICH MAY BE LESS THEN 8
1147                    ;*             BITS  IN  LENGTH  (I.E.  2,  3,  4,  5,  6,  OR 7 BIT
1148                    ;*             CHARACTERS).
1149                    ;*
1150                    ;*        THE TEST PATTERN IS: 123 321 111 222 333 044 155 266 377
1151                    ;*****************************************************************************
1152
1153
1154                    ;*****************************************************************************
1155                    ;*      TEST 26 <TX 'SPACING SEQUENCE'>
1156                    ;*
1157                    ;*  THE  TRANSMITTER IS INITIALIZED AND THE 'SPACING SEQUENCE' IS FORCED
1158                    ;*  BY  ASSERTING  BOTH  TSOM  &  TEOM AT THE SAME TIME -- CHECK THE BIT
1159                    ;*  STREAM FOR ACCURACY (SPACES) AND COMPLETNESS (16 OF THEM). WHEN TXBE
1160                    ;*  GOES  HIGH  (=  1)  A  SMALL  MESSAGE  IS SENT.
1161                    ;*****************************************************************************
1162
1163
1164                    ;*****************************************************************************
1165                    ;*      TEST 27 <FIFO OVERRUN INTEGRITY TEST>
1166                    ;*
1167                    ;* THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS TO FILL
1168                    ;* THE 8 CHARACTER RECEIVER FIFO UNTIL RXOR WITH THE CHARACTERS:
1169                    ;*      (SYNCH),000,377,125,252,347,030,303,074,125.
1170                    ;* THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. OF IMPORTANCE
1171                    ;* IS THE INTEGRITY OF THE LAST OVERRUN-CAUSING FIFO CHARACTER (IT SHOULD
1172                    ;* REMAIN INTACT).
1173                    ;* NOTE THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE  CHARACTERS SINCE THEY
1174                    ;* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
1175                    ;* RDA = 0).
1176                    ;*
1177                    ;*****************************************************************************
1178
1179
1180                    ;*****************************************************************************
1181                    ;*      TEST 28 <BCP RX OVERRUN SET AND CLEAR TEST>
1182                    ;*
1183                    ;*  THE USYRT IS INITIALIZED AND THREE SUBTESTS ARE PERFORMED.
1184                    ;*
1185                    ;*     1 -- AN OVERRUN CONDITION IS FORCED. RECEIVER STATUS REGISTER IS
1186                    ;*          READ TWICE: ONCE TO VERIFY ROR BIT = 1, AND AGAIN TO VERIFY
                        ;*          THAT THE FIRST READ CLEARED ROR .
```

```
1187                    ;*
1188                    ;*     2 -- AN OVERRUN CONDITION IS FORCED (BY THE SAME TECHNIQUE USED IN
1189                    ;*          (2), THE USYRT IS RESET AND THE PROPER STATE OF ALL REGISTERS
1190                    ;*          IS VERIFIED.
1191                    ;*
1192                    ;*     3 -- AN OVERRUN CONDITION IS FORCED (AS ABOVE). RXE IS THEN DROPPED
1193                    ;*          AND A DELAY IS PROVIDED TO ALLOW TIME FOR THE FIFO TO FLUSH
1194                    ;*          (CAUSED BY RDA GOING LOW). RXE IS THEN RE-INITIALIZED AND ROR
1195                    ;*          IS CHECKED = 0. THE RECEIVER IS THEN RE-SYNCHED AND THE TEST IS
1196                    ;*          TERMINATED.
1197                    ;*
1198                    ;*****************************************************************************
1199
1200
1201                    ;*****************************************************************************
1202                    ;*      TEST 29 <BCP RX SYNC CHARACTER RECOGNITION>
1203                    ;*
1204                    ;*  THE  FOLLOWING  MESSAGE  IS INITIATED WITHOUT ASSERTING RXE AND ONCE
1205                    ;*  THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
1206                    ;*
1207                    ;*     SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
1208                    ;*     SYNC SYNC DATA DATA DATA SYNC SYNC
1209                    ;*
1210                    ;*  THE  RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
1211                    ;*  THE  NEXT  TWO SYNC CHARACTERS FOR SYNCRONIZATION, THEN PASS THE REST
1212                    ;*  OF THE MESSAGE (7 SYNC AND 6 DATA CHARACTERS) THROUGH RXDB REGISTER.
1213                    ;*****************************************************************************
1214
1215
1216                    ;*****************************************************************************
1217                    ;*      TEST 30 <BCP RX STRIP-SYNC TEST>
1218                    ;*
1219                    ;*  THE  USYRT  IS INITIALIZED WITH THE STRIP-SYNC CONTROL BIT ASSERTED.
1220                    ;*  THE  FOLLOWING  MESSAGE  IS THEN INITIATED WITHOUT ASSERTING RXE AND
1221                    ;*  ONCE THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
1222                    ;*
1223                    ;*     SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
1224                    ;*     SYNC SYNC DATA DATA DATA SYNC SYNC
1225                    ;*
1226                    ;*  THE  RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
1227                    ;*  THE  NEXT  TWO  SYNC  CHARACTERS FOR SYNCRONIZATION, IGNORE THE NEXT
1228                    ;*  THREE  SYNC CHARACTERS, AND PASS THE REST OF THE MESSAGE (4 SYNC AND
1229                    ;*  6 DATA CHARACTERS) THROUGH RXDB REGISTER.
1230                    ;*****************************************************************************
1231
1232
1233                    ;*****************************************************************************
1234                    ;*      TEST 31 <BCP RX LOST RXE TEST>
1235                    ;*
1236                    ;*  THE USYRT IS INITIALIZED (CRC16,STRIPS,BCP MODE) AND A MESSAGE IS STARTED.
1237                    ;*  WHILE IN THE MIDDLE  OF  TEXT, RXE IS DROPPED AND THE REACTION OF THE
1238                    ;*  RECEIVER IS MONITORED.
1239                    ;*
1240                    ;*****************************************************************************
1241
1242
```

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.
```

CVDMCA.P11 10-DEC-80 09:14 PROGRAM HEADER

```
1314
1315
1316
1317
1318
1319
1320  002000                .SBTTL  PROGRAM HEADER
1321                        ;++
1322                        ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1323  002000                ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1324  002000                ;--
1325  002000    103
1326  002001    126               POINTER BGNAU,BGNDU,ERRTBL
1327  002002    104
1328  002003    115
1329  002004    103               HEADER  CVDMC,A,0,10.,0
1330  002005    000
1331  002006    000
1332  002007    000
1333  002010                LSNAME::
1334  002010    101               .ASCII /C/
1335  002011                        .ASCII /V/
1336  002011    060                 .ASCII /D/
1337  002012                        .ASCII /M/
1338  002012  000000                .ASCII /C/
1339  002014                        .BYTE   0
1340  002014  000012                .BYTE   0
1341  002016                        .BYTE   0
1342  002016  036640        LSREV::
1343  002020                        .ASCII  /A/
1344  002020  000000        LSDEPO::
1345  002022                        .ASCII  /0/
1346  002022  002224        LSUNIT::
1347  002024                        .WORD   0
1348  002024  000000        LSTIML::
1349  002026                        .WORD   10.
1350  002026  037426        LSHPCP::
1351  002030                        .WORD   LSHARD
1352  002030  000000        LSSPCP::
1353  002032                        .WORD   0
1354  002032  000000        LSHPTP::
1355  002034                        .WORD   LSHW
1356  002034  000000        LSSPTP::
1357  002036                        .WORD   0
1358  002036  000000        LSLADP::
1359  002040                        .WORD   LSLAST
1360  002040  002124        LSSTA::
1361  002042                        .WORD   0
1362  002042  000000        LSCO::
1363  002044                        .WORD   0
1364  002044  000000        LSDTYP::
1365  002046                        .WORD   0
1366  002046  000000        LSAPT::
1367  002050                        .WORD   0
1368  002050    003         LSDTP::
1369  002051    003                 .WORD   LSDISPATCH
                            LSPRIO::
                                    .WORD   0
                            LSENVI::
                                    .WORD   0
                            LSEXP1::
                                    .WORD   0
                            LSMREV::
                                    .BYTE   C$REVISION
                                    .BYTE   C$EDIT
```

```
1370  002052
1371  002052  000000
1372  002054  000000
1373  002056
1374  002056  000000
1375  002060
1376  002060  003306
1377  002062
1378  002062  000000
1379  002064
1380  002064  000000
1381  002066
1382  002066  000000
1383  002070
1384  002070  021546
1385  002072
1386  002072  021542
1387  002074
1388  002074  000000
1389  002076
1390  002076  003326
1391  002100
1392  002100  104035
1393  002102
1394  002102  002246
1395  002104
1396  002104  021100
1397  002106
1398  002106  021540
1399  002110
1400  002110  021414
1401  002112
1402  002112  021072
1403  002114
1404  002114  000000
1405  002116
1406  002116  000000
1407  002120
1408  002120  000000
1409
1410
1411                    .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

002122                    SLASH
                          ;;//////////////////////////////////////////////////////////////////
                          ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
                          ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
002122                    SLASH
                          ;;//////////////////////////////////////////////////////////////////

002122                            DISPATCH 31.
002122  000037                                            .WORD   31
002124                                             L$DISPATCH::
002124  021550                                            .WORD   T1
002126  021730                                            .WORD   T2
002130  022730                                            .WORD   T3
002132  023114                                            .WORD   T4
002134  023200                                            .WORD   T5
002136  023424                                            .WORD   T6
002140  023516                                            .WORD   T7
002142  023610                                            .WORD   T8
002144  023702                                            .WORD   T9
002146  024032                                            .WORD   T10
002150  024124                                            .WORD   T11
002152  024200                                            .WORD   T12
002154  024254                                            .WORD   T13
002156  024576                                            .WORD   T14
002160  024730                                            .WORD   T15
002162  025112                                            .WORD   T16
002164  025620                                            .WORD   T17
002166  026220                                            .WORD   T18
002170  026426                                            .WORD   T19
002172  026704                                            .WORD   T20
002174  0274 .                                            .WORD   T21
002176  027610                                            .WORD   T22
002200  030004                                            .WORD   T23
002202  030200                                            .WORD   T24
002204  030556                                            .WORD   T25
002206  033500                                            .WORD   T26
002210  034056                                            .WORD   T27
002212  034442                                            .WORD   T28
002214  035044                                            .WORD   T29
002216  035522                                            .WORD   T30
002220  036256                                            .WORD   T31
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

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

002222                                  BGNHW   DFPTBL
002222  000010                                                                    .WORD   L10000-L$HW/2
002224                                                                  L$HW::
002224                                                                  DFPTBL::

002224  160020                          .WORD   160020          ;DMV11 CSR UNIBUS ADDRESS
002226  000300                          .WORD      300          ;DMV11 INTERRUPT VECTOR
002230  004000                          .WORD     4000          ;DMV11 INTERRUPT PRIORITY LEVEL = 4
002232  000000                          .WORD      000          ;SWITCH REG. #1 (BOOT ADDRESS)
002234  000000                          .WORD      000          ;SWITCH REG. #2 (DDCMP ADDRESS)
002236  000000                          .WORD        0          ;MODULE IS M8064
002240  000000                          .WORD        0          ;H3254&H3255 USED
002242  000001                          .WORD        1          ;BAUD RATE = 56 K
                                                                ;       0 = 19.2 K
                                                                ;       1 = 56 K

002244                                  ENDHW
002244                                                                  L10000:
```

```
CVDMCA.P11    10-DEC-80 09:14          SOFTWARE P-TABLE

  1482                                 .SBTTL  SOFTWARE P-TABLE
  1483
  1484                                 ;////////////////////////////////////////////////////////////////////////
  1485                                 ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
  1486                                 ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
  1487                                 ;////////////////////////////////////////////////////////////////////////
  1488
  1489  002244                             BGNSW   SFPTBL
  1490  002244  000000                                                        .WORD   L10001-L$SW/2
  1491  002246                                                         L$SW::
  1492  002246                                                         SFPTBL::
  1493
  1494  002246                             ENDSW
  1495  002246                                                         L10001:
```

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


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

002246                          EQUALS
                        ;
                        ; BIT DIFINITIONS
                        ;
100000                  BIT15== 100000
040000                  BIT14== 40000
020000                  BIT13== 20000
010000                  BIT12== 10000
004000                  BIT11== 4000
002000                  BIT10== 2000
001000                  BIT09== 1000
000400                  BIT08== 400
000200                  BIT07== 200
000100                  BIT06== 100
000040                  BIT05== 40
000020                  BIT04== 20
000010                  BIT03== 10
000004                  BIT02== 4
000002                  BIT01== 2
000001                  BIT00== 1
                        ;
001000                  BIT9==  BIT09
000400                  BIT8==  BIT08
000200                  BIT7==  BIT07
000100                  BIT6==  BIT06
000040                  BIT5==  BIT05
000020                  BIT4==  BIT04
000010                  BIT3==  BIT03
000004                  BIT2==  BIT02
000002                  BIT1==  BIT01
000001                  BIT0==  BIT00
                        ;
                        ; EVENT FLAG DEFINITIONS
                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                        ;
000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
```

CVDMCA.P11 10-DEC-80 09:14 GLOBAL EQUATES SECTION -- BASIC EQUATES

```
1552    000140          PRI03== 140
1553    000100          PRI02== 100
1554    000040          PRI01== 40
1555    000000          PRI00== 0
1556                    ;
1557                    ;OPERATOR FLAG BITS
1558                    ;
1559    000004          EVL==     4
1560    000010          LOT==    10
1561    000020          ADR==    20
1562    000040          IDU==    40
1563    000100          ISR==   100
1564    000200          UAM==   200
1565    000400          BOE==   400
1566    001000          PNT==  1000
1567    002000          PRI==  2000
1568    004000          IXE==  4000
1569    010000          IBE== 10000
1570    020000          IER== 20000
1571    040000          LOE== 40000
1572    100000          HOE==100000
```

```
CVDMCA.P11    10-DEC-80 09:14        REGISTER DEFINITIONS -- MAINTENANCE REGISTERS -- SELN & BSELN

                                     .SBTTL  REGISTER DEFINITIONS -- MAINTENANCE REGISTERS -- SELN & BSELN
                                     ;;*****************************************************************************
                                     ;* MAINTENANCE REGISTER # 0 - BSEL0
                                     ;;*****************************************************************************
000020                               IEO     = BIT4          ;"INTERRUPT ENABLE OUT"
000001                               IEI     = BIT0          ;"INTERRUPT ENABLE IN"

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

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

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

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

```
CVDMCA.P11    10-DEC-80 09:14        REGISTER DEFINITIONS -- USYRT

                                     .SBTTL  REGISTER DEFINITIONS -- USYRT


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

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

                 120400              RDSRL   = 120400        ;ADDRESS OF THIS REG

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

                 120401              RDSRH   = 120401        ;ADDRESS OF THIS REG

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

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

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

                                     ;;*************************************************************************
                                     ;* USYRT 'TRANSMITTER DATA BUFFER' REGISTER
                                     ;;*************************************************************************

                 120402              TDSRL   = 120402        ;ADDRESS OF THIS REG

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

                 120403              TDSRH   = 120403        ;ADDRESS OF THIS REG

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

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

CVDMCA.P11 10-DEC-80 09:14 REGISTER DEFINITIONS -- USYRT

```
1666    002000          TXAB    = BIT10         ;TRANSMIT ABORT
1667    001000          TXEOM   = BIT9          ;TRANSMIT END-OF-MESSAGE
1668    000400          TXSOM   = BIT8          ;TRANSMIT START-OF-MESSAGE
1669
1670                    ;;*************************************************************************
1671                    ;* USYRT 'SYNC/SECONDARY ADDRESS' REGISTER
1672                    ;;*************************************************************************
1673
1674    120404          PCSARL  = 120404        ;ADDRESS OF THIS REG
1675    000226          SYNCH   = 226           ;STANDARD SYNCH CHARACTER
1676
1677                    ;;*************************************************************************
1678                    ;* USYRT 'MODE CONTROL'
1679                    ;;*************************************************************************
1680
1681    120405          PCSARH  = 120405        ;ADDRESS OF THIS REG
1682
1683                            ;BIT DEFINITIONS ON BYTE BASIS:
1684
1685    000200          APA     = BIT7          ;'ALL PARTIES ADDRESS' ENABLE
1686    000100          PROTO   = BIT6          ;SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
1687    000040          STRIP   = BIT5          ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1688    000020          SECAD   = BIT4          ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1689    000010          IDLE    = BIT3          ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1690    000007          XYZ     = BIT2!BIT1!BIT0 ;CRC/PARITY SELECTION CONTROL
1691
1692                            ;BIT DEFINITIONS ON WORD BASIS:
1693
1694    100000          APAD    = BIT15         ;'ALL PARTIES ADDRESS' ENABLE
1695    040000          DDCMP   = BIT14         ;CODE FOR DDCMP MODE
1696    020000          STRIPS  = BIT13         ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1697    010000          SECADR  = BIT12         ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1698    004000          IDLES   = BIT11         ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1699    001400          CRC16   = BIT9!BIT8     ;CODE FOR CRC-16 SELECTION
1700    003400          NOCHK   = BIT10!BIT9!BIT8 ;CODE FOR NO ERROR CHECKING
1701    002400          EVRC    = BIT10!BIT8    ;CODE FOR VRC EVEN CHECK
1702    002000          OVRC    = BIT10         ;CODE FOR VRC ODD CHECK
1703
1704                    ;;*************************************************************************
1705                    ;* USYRT 'DATA LENGTH SELECT' REGISTER
1706                    ;;*************************************************************************
1707
1708    120407          PCR     = 120407        ;ADDRESS OF THIS REG
1709
1710                            ;BIT DEFINITIONS:
1711
1712    000340          TXDL    = BIT7!BIT6!BIT5 ;TRANSMIT DATA LENGTH SELECTION
1713    000020          EXADD   = BIT4          ;EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
1714    000010          EXCON   = BIT3          ;EXTENDED CONTROL FIELD -- NOT USED OR TESTED
1715    000007          RXDL    = BIT2!BIT1!BIT0 ;RECEIVER DATA LENGTH SELECTION
1716
1717                    ;;*************************************************************************
1718                    ;* USYRT STATUS REGISTER (ADDR. A400)
1719                    ;;*************************************************************************
1720    122000          USTATR = 122000         ;USYRT STATUS REGISTER ADDRESS = A400 (HEX)
1721
```

```
1722
1723                                    ;BIT DEFINITIONS:
1724        000200      RDA     = BIT7      ;RECEIVER DATA AVAILABLE
1725        000100      TBMT    = BIT6      ;TRANSMITTER BUFFER EMPTY
1726        000040      RXACT   = BIT5      ;RECEIVER ACTIVE
1727        000020      RSA     = BIT4      ;RECEIVER STATUS AVAILABLE
1728        000010      TSO     = BIT3      ;TRANSMITTER SERIAL OUTPUT
1729        000004      TXACT   = BIT2      ;TRANSMITTER ACTIVE
1730        000002      TXU     = BIT1      ;TRANSMITTER UNDERRUN
1731        000001      SFR     = BIT0      ;SYNC/FLAG RECEIVED
```

CVDMCA.P11 10-DEC-80 09:14 REGISTER DEFINITIONS -- 6522 VIA CHIP

```
1732                                .SBTTL  REGISTER DEFINITIONS -- 6522 VIA CHIP
1733
1734    120000                      VIA     = 120000                ;VIA BASE ADDRESS = A000 (HEX)
1735
1736                                ;;*************************************************************************
1737                                ;* MODEM & MAINTENANCE CONTROL -- "ORB" 8 BIT PORT B -- WRITE ONLY
1738                                ;;*************************************************************************
1739
1740    120000                      VIAORB  = 120000                ;ADDRESS OF THIS REGISTER -- HEX = A0X0
1741
1742    000200                      NULCLK  = BIT7                  ;"NULL CLK L" -- NULL CLOCK
1743    000100                      RXEN    = BIT6                  ;"RXENL" -- USYRT RECEIVER ENABLE
1744    000040                      TXEN    = BIT5                  ;"TXENL" -- USYRT TRANSMITTER ENABLE
1745    000020                      DTR     = BIT4                  ;"DTR" -- DATA TERMINAL READY
1746    000010                      RTSND   = BIT3                  ;"RTSND" -- REQUEST TO SEND
1747    000004                      HDX     = BIT2                  ;"HDX" -- HALF DUPLEX
1748    000002                      TTLOOP  = BIT1                  ;"SELECT TTL LEVEL LOOPBACK"
1749    000001                      PRESET  = BIT0                  ;"PRESET H" --
1750    000000                      DTRL    = 0                     ;DTR IS ASSERTED LOW
1751
1752                                ;;*************************************************************************
1753                                ;* MODEM STATUS REGISTER -- "ORA" 8 BIT PORT A -- READ ONLY
1754                                ;;*************************************************************************
1755
1756    120001                      VIAMS   = 120001                ;ADDRESS OF THIS REGISTER -- HEX = A0X1
1757
1758    000200                      RING    = BIT7                  ;"RING H" --
1759    000100                      CARIER  = BIT6                  ;"CARRIER H" --
1760    000040                      MDMRDY  = BIT5                  ;"MODEM RDY H" --
1761    000020                      SPEED   = BIT4                  ;"BAUD RATE SWITCH -- (19.2K/56K)
1762    000010                      CTS     = BIT3                  ;"CTS H -- CLEAR TO SEND
1763    000004                      TM      = BIT2                  ;"TEST MODE H" --
1764    000002                      RCVDAT  = BIT1                  ;"RCV DATA H" --
1765    000001                      UMAINT  = BIT0                  ; SELECT USYRT INT LOOPBACK **SELECT BIT**
1766
1767
1768                                ;;*************************************************************************
1769                                ;* DATA DIRECTION FOR PORT B -- "DDRB" -- READ/WRITE
1770                                ;;*************************************************************************
1771
1772    120002                      VIADPB  = 120002                ;ADDRESS OF THIS REGISTER -- HEX = A0X2
1773
1774                                ;  ALL BITS ARE DEFINED THE SAME:
1775                                ;       THE BIT SETTING DEFINED THE DIRECTION OF ITS RELATED BIT IN BIT PORT B
1776
1777                                ;       INITIALIZED TO 377 (HEX = FF) -- PORT B IS READ/WRITE
1778
1779
1780                                ;;*************************************************************************
1781                                ;* DATA DIRECTION FOR PORT A -- "DDRA" -- READ/WRITE
1782                                ;;*************************************************************************
1783
1784    120003                      VIADPA  = 120003                ;ADDRESS OF THIS REGISTER -- HEX = A0X3
1785
1786                                ;  ALL BITS ARE DEFINED THE SAME:
1787                                ;       THE BIT SETTING DEFINED THE DIRECTION OF ITS RELATED BIT IN BIT PORT A
```

```
1788
1789                            ;       INITIALIZED TO 001 (HEX = 01) -- PORT A IS READ ONLY (EXCEPT FOR
1790                            ;                                       BIT0 WHICH ENABLES USYRT INTERNAL LOOPBACK).
1791
1792
1793
1794                            ;;*****************************************************************************
1795                            ;* TIMER 1 LOW ORDER (LATCH & COUNTER) -- 'T1L-L' & 'T1C-L' -- WRITE & READ
1796                            ;;*****************************************************************************
1797
1798    120004                  VIAT1A  = 120004                ;ADDRESS OF THIS REGISTER -- HEX = A0X4
1799
1800                            ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1801                            ;  WHEN READING, LOW ORDER COUNTER IS READ.
1802
1803
1804
1805                            ;;*****************************************************************************
1806                            ;* TIMER 1 HIGH ORDER COUNTER & TRIGGER -- 'T1L-H AND TRIGGER' & 'T1C-H'
1807                            ;*       -- WRITE & READ
1808                            ;;*****************************************************************************
1809
1810    120005                  VIAT1B  = 120005                ;ADDRESS OF THIS REGISTER -- HEX = A0X5
1811
1812                            ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1813                            ;                 ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1814
1815                            ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1816
1817
1818
1819                            ;;*****************************************************************************
1820                            ;* TIMER 1 LOW ORDER LATCH -- 'T1L-L' -- READ/WRITE
1821                            ;;*****************************************************************************
1822
1823    120006                  VIAT1C  = 120006                ;ADDRESS OF THIS REGISTER -- HEX = A0X6
1824
1825                            ;  THE LOW ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1826                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1827
1828
1829
1830                            ;;*****************************************************************************
1831                            ;* TIMER 1 HIGH ORDER LATCH -- 'T1L-H' -- READ/WRITE
1832                            ;;*****************************************************************************
1833
1834    120007                  VIAT1D  = 120007                ;ADDRESS OF THIS REGISTER -- HEX = A0X7
1835
1836                            ;  THE HIGH ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1837                            ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1838
1839
1840
1841                            ;;*****************************************************************************
1842                            ;* TIMER 2 LOW ORDER (LATCH & COUNTER) -- 'T2L-L' & 'T2C-L' -- WRITE & READ
1843                            ;;*****************************************************************************
```

```
1844
1845    120010          VIAT2A  = 120010                ;ADDRESS OF THIS REGISTFR -- HEX = A0X8
1846
1847                    ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1848                    ;  WHEN READING, LOW ORDER COUNTER IS READ.
1849
1850
1851
1852                    ;;*****************************************************************************
1853                    ;* TIMER 2 HIGH ORDER COUNTER & TRIGGER -- 'T2L-H AND TRIGGER' & 'T2C-H'
1854                    ;*      -- WRITE & READ
1855                    ;;*****************************************************************************
1856
1857    120011          VIAT2B  = 120011                ;ADDRESS OF THIS REGISTER -- HEX = A0X9
1858
1859                    ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1860                    ;                ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1861
1862                    ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1863
1864                    ;;*****************************************************************************
1865                    ;* SHIFT REGISTER -- 'SR' -- READ/WRITE
1866                    ;;*****************************************************************************
1867
1868    120012          VIASR   = 120012                ;ADDRESS OF THIS REGISTER -- HEX = A0XA
1869
1870                    ;  SHIFTING IS CONTROLLED BY THE SETTING OF VIASRC (ACR2 ---> ACR4) IN VIAACR
1871
1872
1873
1874                    ;;*****************************************************************************
1875                    ;* AUXILIARY CONTROL REGISTER -- 'ACR' -- READ/WRITE
1876                    ;;*****************************************************************************
1877
1878    120013          VIAACR  = 120013                ;ADDRESS OF THIS REGISTER -- HEX = A0XB
1879
1880    000300          T1MODE  = BIT7!BIT6             ;CONTROL THE MODE OF TIMER # 1
1881
1882                                            ;BIT 7:
1883                                            ; 0     PB7 DISABLED -- ONLY T1TO IN VIAIFR REFLECTS TIMEOUT
1884                                            ; 1     PB7 & T1TO REFLECT TIMEOUT
1885
1886                                            ;BIT 6:
1887                                            ; 0     TIMER 1 IN ONE-SHOT MODE
1888                                            ; 1     TIMER 1 IN CONTINUOUS SQUARE WAVE MODE
1889
1890    000040          T2MODE  = BIT5                  ;CONTROLS THE MODE OF TIMER # 1
1891
1892                                            ; 0     PULSE COUNTING MODE
1893                                            ; 1     INTERVAL TIMER MODE
1894
1895    000034          SRMODE  = BIT4!BIT3!BIT2        ;CONTROLS THE MODE OF THE SHIFT REGISTER
1896
1897                                            ; 0     SR DISABLED
1898                                            ; 1     SHIFT IN UNDER CONTROL OF T2, SHFT PULSES GEN'D ON CB1
1899                                            ; 2     SHIFT IN AT SYS. CLOCK RATE, SHFT PULSES GEN'D ON CB1
```

```
                                        ; 3     SHIFT IN UNDER CONTROL OF EXTERNAL INPUT PULSES
                                        ; 4     SHIFT OUT -- FREE RUNNING -- RATE CONTROLLED BY T2
                                        ; 5     SHIFT OUT -- RATE CONTROLLED BY T2 -- PULSES ON CB1
                                        ; 6     SHIFT OUT -- SYS. CLOCK RATE -- PULSES ON CB1
                                        ; 7     SHIFT OUT -- UNDER CONTROL OF PULSES APPLIED TO CB1

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



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

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

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



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

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

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

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



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

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

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

```
1956                                            ;BITS IN THE INT. ENAB. REG..  IF = 1, THE
1957                                            ;RESPECTIVE BITS WILL BE SET.
1958
1959                    ;  WHEN WRITING THIS REG., THE COMMENT ABOVE HOLDS.
1960
1961                    ;  WHEN READING THIS REG., THE CURRENT STATE OF THE INT. ENABLE REG. IS RETURNED.
1962
1963                    ;  THE BIT ASSIGNMENTS ARE THE SAME AS FOR VIAIFR AS DEFINED ABOVE.
1964
1965
1966
1967                    ;;*************************************************************************
1968                    ;* OUTPUT REGISTER A -- 'ORA' -- READ ONLY (OR READ/WRITE UNDER CONTROL OF 'DDPA')
1969                    ;;*************************************************************************
1970
1971        120017      VIAORA  = 120017                ;ADDRESS OF THIS REGISTER -- HEX = A0XF
1972
1973                    ;  THIS ADDRESS ACCESSES THE SAME DATA AS 'VIAMS' EXCEPT THAT NO 'HANDSHAKING'
1974                    ;  WILL TAKE PLACE (I.E. THERE IS NO CHANGE IN IRQ OR CA2 AS A RESULT OF
1975                    ;  READING ORA THROUGH THIS ADDRESS)
1976
1977                    ;THE BIT ASSIGNMENTS ARE THE SAME AS FOR 'VIAMS' ABOVE.
1978
1979
1980
```

```
                        .SBTTL  REGISTER DEFINITIONS -- MISC

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

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

                        ;MISCELLANEOUS EQUATES

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

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

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

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

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

```
2007                                    .SBTTL  GLOBAL DATA SECTION
2008
2009                                    ;////////////////////////////////////////////////////////////////////////////
2010                                    ;/      THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
2011                                    ;/      IN MORE THAN ONE TEST.
2012                                    ;////////////////////////////////////////////////////////////////////////////
2013
2014                                    ;++*************************************************************************
2015                                    ; CONTROL BLOCK FOR STACKED ERROR MESSAGES
2016                                    ;--*************************************************************************
2017
2018  002246                                    ERRTBL
2019  002246                                                                          L$ERRTBL::
2020  002246  000000                    ERRTYP::        .WORD   0
2021  002250  000000                    ERRNBR::        .WORD   0
2022  002252  000000                    ERRMSG::        .WORD   0
2023  002254  000000                    ERRBLK::        .WORD   0
2024
2025                                    ;;*************************************************************************
2026                                    ;* STORAGE FOR DEVICE REGISTERS
2027                                    ;;*************************************************************************
2028  002256                            WSR0:                                 ;STORAGE FOR DEVICE CSR REGISTERS
2029  002256  000000                    BSR0:   .WORD   0
2030  002260                            WSR2:
2031  002260  000000                    BSR1:   .WORD   0
2032  002262                            WSR4:
2033  002262  000000                    BSR2:   .WORD   0
2034  002264                            WSR6:
2035  002264  000000                    BSR3:   .WORD   0
2036  002266                            WSR10:
2037  002266  000000                    BSR4:   .WORD   0
2038  002270                            WSR12:
2039  002270  000000                    BSR5:   .WORD   0
2040  002272                            WSR14:
2041  002272  000000                    BSR6:   .WORD   0
2042  002274                            WSR16:
2043  002274  000000                    BSR7:   .WORD   0
2044  002276  000000                    BSR10:  .WORD   0
2045  002300  000000                    BSR11:  .WORD   0
2046  002302  000000                    BSR12:  .WORD   0
2047  002304  000000                    BSR13:  .WORD   0
2048  002306  000000                    BSR14:  .WORD   0
2049  002310  000000                    BSR15:  .WORD   0
2050  002312  000000                    BSR16:  .WORD   0
2051  002314  000000                    BSR17:  .WORD   0
2052
2053  002316  000010                    UREGS:  .BLKW   8.              ;THE FIRST 7 ARE FOR THE USYRT'S ACTUAL
2054                                                                    ;REGISTERS.  THE LAST ONE IS FOR THE STATUS
2055                                                                    ;REG. (USTATR).
2056  002336  000020                    VREGS:  .BLKW   16.             ;STORAGE FOR VIA REGISTERS FOR PRINTOUT
```

```
2057
2058                    ;*********************************************************************
2059                    ;* MISCELLANEOUS STORAGE
2060                    ;*********************************************************************
2061  002376  000000    TDATA:  .WORD   0                ;TEST DATA
2062  002400  000000    GDATA:  .WORD   0                ;GOOD DATA
2063  002402  000000    BDATA:  .WORD   0                ;BAD DATA
2064  002404  000000    XDATA:  .WORD   0                ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
2065  002406  000000    SCRACH: .WORD   0                ;GEN'L PURPOSE SCRATCH WORD
2066  002410  000000    LOGDEV: .WORD   0                ;LOGICAL DEVICE NUMBER
2067  002412  000000    REGNUM: .WORD   0                ;CONTAINS A DEVICE REGISTER NUMBER
2068  002414  000000    PSTACK: .WORD   0                ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
2069  002416  000000    PRIOR:  .WORD   0                ;CPU PRIORITY FOR PRINTOUT
2070  002420  000000    SUBRPC: .WORD   0                ;PC OF SUBR CALL FOR ERROR REPORTS
2071  002422  000000    INTFLG: .WORD   0                ;INTERRUPT RECEIVED FLAGS
                                                         ;  BIT 0 FOR TX, BIT 1 FOR RCV
2072  002424  000000    ERRFLG: .WORD   0                ;SUBROUTINE ERROR FLAG
2073  002426  000000    TIMFLG: .WORD   0                ;EVENT TIME-OUT FLAG
2074  002430  000000    RETADR: .WORD   0                ;SUBR ERROR RETURN ADDRESS
2075  002432  000000    REDBYT: .WORD   0                ;LO BYTE CONTAINS BYTE READ FROM LU REG
2076  002434  000000    WRIBYT: .WORD   0                ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
2077  002436  000000    LOADAT: .WORD   0                ;CONTAINS TEST DATA LOADED INTO REG
2078  002440  000000    GOODAT: .WORD   0                ;STORAGE FOR EXPECTED DATA
2079  002442  000000    BADDAT: .WORD   0                ;STORAGE FOR ACTUAL DATA
2080  002444  000000    FRSTIM: .WORD   0                ;FLAG=0 IF PROGRAM JUST LOADED
2081  002446  000000    SAVE4:  .WORD   0                ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
2082  002450  000000    SAVE6:  .WORD   0                ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
2083  002452  000000    ERROR1: .WORD   0                ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
2084  002454  000000    CHPTYP: .WORD   0                ;USYRT CHIP TYPE, =0 FOR SMC, ELSE =1
2085  002456  000000    SAVLEN: .WORD   0                ;SAVED TX AND RCV CHAR LENGTHS
2086  002460  000000    DEVMAP: .WORD   0                ;BIT MAP OF ACTIVE DEVICES
2087  002462  000000    DEVPTR: .WORD   0                ;DEVICE MAP BIT POINTER
2088  002464  000000    UNIT:   .WORD   0                ;CONTAINS UNIT NO. (1 TO N)
2089  002466  000000    STARES: .WORD   0                ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
2090  002470  000000    TSTNUM: .WORD   0                ;NO. OF CURRENT TEST (FOR SOME TESTS)
2091
```

```
2092
2093  002472
2094  002472
2095  002472  160020          ;;********* CURRENT DEVICE PARAMETERS *************
                              BSEL0:
                              SEL0:
                              MPCSR:  .WORD   160020          ;POINTER TO DMV11 CSR'S
2096  002474  160021          BSEL1:  .WORD   160021          ;POINTER TO BSEL1
2097  002476
                              BSEL2:
2098  002476  160022          SEL2:   .WORD   160022          ;POINTER TO SEL2
2099  002500  160023          BSEL3:  .WORD   160023          ;POINTER TO BSEL3
2100  002502
                              BSEL4:
2101  002502  160024          SEL4:   .WORD   160024          ;POINTER TO SEL4
2102  002504  160025          BSEL5:  .WORD   160025          ;POINTER TO BSEL5
2103  002506
                              BSEL6:
2104  002506  160026          SEL6:   .WORD   160026          ;POINTER TO SEL6
2105  002510  160027          BSEL7:  .WORD   160027          ;POINTER TO BSEL7
2106  002512
                              BSEL10:
2107  002512  160030          SEL10:  .WORD   160030          ;POINTER TO SEL10
2108  002514  160031          BSEL11: .WORD   160031          ;POINTER TO BSEL11
2109  002516
                              BSEL12:
2110  002516  160032          SEL12:  .WORD   160032          ;POINTER TO SEL12
2111  002520  160033          BSEL13: .WORD   160033          ;POINTER TO BSEL13
2112  002522
                              BSEL14:
2113  002522  160034          SEL14:  .WORD   160034          ;POINTER TO SEL14
2114  002524  160035          BSEL15: .WORD   160035          ;POINTER TO BSEL15
2115  002526
                              BSEL16:
2116  002526  160036          SEL16:  .WORD   160036          ;POINTER TO SEL16
2117  002530  160037          BSEL17: .WORD   160037          ;POINTER TO BSEL17
2118
2119  002532  000300          MPIVEC: .WORD      300          ;DMV11 INPUT INTERRUPT VECTOR
2120  002534  000304          MPOVEC: .WORD      304          ;DMV11 OUTPUT INTERRUPT VECTOR
2121  002536  000240          MPRIOR: .WORD      240          ;DMV11 DEVICE PRIORITY
2122  002540  000000          LUSWI1: .WORD        0          ;LINE UNIT SWITCH PACK #1
2123  002542  000000          LUSWI2: .WORD        0          ;LINE UNIT SWITCH PACK #2
2124  002544  000000          BRDTYP: .WORD        0          ;0=M8064, 1=M8053/V.35,2=M8053/EIA
2125  002546  000000          TSTCON: .WORD        0          ;TEST CONNECTOR INDICATOR
2126  002550  000001          BDRATE: .WORD        1          ;BAUD RATE = 56 K
2127                                                          ;         0 = 19.2 K
2128                                                          ;         1 = 56 K
```

```
2129
2130  002552  120400          USYREG: .WORD   120400          ;ADDRESS OF RDSRL
2131  002554  120401                  .WORD   120401          ;ADDRESS OF RDSRH
2132  002556  120402                  .WORD   120402          ;ADDRESS OF TDSRL
2133  002560  120403                  .WORD   120403          ;ADDRESS OF TDSRH
2134  002562  120404                  .WORD   120404          ;ADDRESS OF PCSARL
2135  002564  120405                  .WORD   120405          ;ADDRESS OF PCSARH
2136  002566  120407                  .WORD   120407          ;ADDRESS OF PCR
2137  002570  122000                  .WORD   122000          ;ADDRESS OF USYRT STATUS REG
2138
2139                                  ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
2140  002572  000010          REDDAT: .BLKB   8.
2141
2142                                  ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
2143  002602  000000          REG0:   .WORD   0
2144  002604  000000          REG1:   .WORD   0
2145  002606  000000          REG2:   .WORD   0
2146  002610  000000          REG3:   .WORD   0
2147  002612  000000          REG4:   .WORD   0
2148  002614  000000          REG5:   .WORD   0
2149  002616  000000          REG6:   .WORD   0
2150  002620  000000          REG7:   .WORD   0
2151
2152                                  ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
2153  002622  000000          TMP0:   .WORD   0
2154  002624  000000          TMP1:   .WORD   0
2155  002626  000000          TMP2:   .WORD   0
2156  002630  000000          TMP3:   .WORD   0
2157  002632  000000          TMP4:   .WORD   0
2158  002634  000000          TMP5:   .WORD   0
2159  002636  000000          TMP6:   .WORD   0
2160  002640  000000          TMP7:   .WORD   0
2161
2162                                  ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
2163  002642                  UPBITS:
2164  002642      377                 .BYTE   377             ;MASK FOR RDBR
2165  002643      000                 .BYTE   000             ;MASK FOR RDSR
2166  002644      000                 .BYTE   000             ;MASK FOR TDBR
2167  002645      360                 .BYTE   360             ;MASK FOR TDSR
2168  002646      000                 .BYTE   000             ;MASK FOR SSAR
2169  002647      000                 .BYTE   000             ;MASK FOR PCSAR
2170  002650      347                 .BYTE   347             ;MASK FOR PCR
2171
2172  002651      200         TDSRNRW: .BYTE  200             ;TDSR NON-R/W BITS
```

```
2173
2174
2175  002652
2176  002652  377          .SBTTL  DATA TEST PATTERNS
2177  002653  377          ;***** DATA PATTERN E *****
2178  002654  377          PATE:
2179  002655  377                  .BYTE   377
2180  002656  377                  .BYTE   377
2181  002657  377                  .BYTE   377
2182  002660  377                  .BYTE   377
2183  002661  366                  .BYTE   377
2184                               .BYTE   377
2185                               .BYTE   377
2186  002662                       .BYTE   377
2187  002662  000                  .BYTE   366
2188  002663  000
2189  002664  000          ;***** DATA PATTERN F *****
2190  002665  000          PATF:
2191  002666  000                  .BYTE   000
2192  002667  000                  .BYTE   000
2193  002670  000                  .BYTE   000
2194  002671  110                  .BYTE   000
2195                               .BYTE   000
2196                               .BYTE   000
2197  002672                       .BYTE   000
2198  002672  000                  .BYTE   110
2199  002673  001
2200  002674  003          ;***** DATA PATTERN G *****
2201  002675  004          PATG:
2202  002676  005                  .BYTE   000
2203  002677  007                  .BYTE   001
2204  002700  100                  .BYTE   003
2205  002701  101                  .BYTE   004
2206  002702  103                  .BYTE   005
2207  002703  104                  .BYTE   007
2208  002704  105                  .BYTE   100
2209  002705  107                  .BYTE   101
2210  002706  000                  .BYTE   103
2211  002707  017                  .BYTE   104
2212  002710  027                  .BYTE   105
2213  002711  041                  .BYTE   107
2214  002712  200                  .BYTE   000
2215  002713  277                  .BYTE   017
2216  002714  103                  .BYTE   027
2217  002715  144                  .BYTE   041
2218  002716  115                  .BYTE   200
2219  002717  157                  .BYTE   277
2220  002720  000                  .BYTE   103
2221                               .BYTE   144
2222                               .BYTE   115
2223  002721                       .BYTE   157
2224  002721  125                  .BYTE   000
2225  002722  252
2226  002723  000          ;***** DATA PATTERN H *****
2227  002724  377          PATH:
2228  002725  000                  .BYTE   125
                                   .BYTE   252
                                   .BYTE   000
                                   .BYTE   377
                                   .BYTE   000
```

```
2229  002726  001                     .BYTE   001
2230  002727  002                     .BYTE   002
2231  002730  004                     .BYTE   004
2232  002731  010                     .BYTE   010
2233  002732  020                     .BYTE   020
2234  002733  040                     .BYTE   040
2235  002734  100                     .BYTE   100
2236  002735  200                     .BYTE   200
2237  002736  000                     .BYTE   000
2238  002737  377                     .BYTE   377
2239  002740  376                     .BYTE   376
2240  002741  375                     .BYTE   375
2241  002742  373                     .BYTE   373
2242  002743  367                     .BYTE   367
2243  002744  357                     .BYTE   357
2244  002745  337                     .BYTE   337
2245  002746  277                     .BYTE   277
2246  002747  177                     .BYTE   177
2247  002750  377                     .BYTE   377
2248  002751  000                     .BYTE   000
2249
2250                                  ;***** DATA PATTERN I *****
2251  002752                          PATI:
2252  002752  000                     .BYTE   000
2253  002753  041                     .BYTE   041
2254  002754  102                     .BYTE   102
2255  002755  143                     .BYTE   143
2256  002756  204                     .BYTE   204
2257  002757  245                     .BYTE   245
2258  002760  306                     .BYTE   306
2259  002761  347                     .BYTE   347
2260  002762  000                     .BYTE   000
2261  002763  001                     .BYTE   001
2262  002764  002                     .BYTE   002
2263  002765  004                     .BYTE   004
2264  002766  040                     .BYTE   040
2265  002767  100                     .BYTE   100
2266  002770  200                     .BYTE   200
2267  002771  000                     .BYTE   000
2268  002772  346                     .BYTE   346
2269  002773  345                     .BYTE   345
2270  002774  343                     .BYTE   343
2271  002775  307                     .BYTE   307
2272  002776  247                     .BYTE   247
2273  002777  147                     .BYTE   147
2274  003000  347                     .BYTE   347
2275  003001  242                     .BYTE   242
2276  003002  105                     .BYTE   105
2277  003003  347                     .BYTE   347
2278  003004  010                     .BYTE   010
2279  003005  020                     .BYTE   020
2280  003006  367                     .BYTE   367
2281  003007  357                     .BYTE   357
2282  003010  030                     .BYTE   030
2283  003011  027                     .BYTE   027
2284  003012  377                     .BYTE   377
```

```
2285
2286
2287  003013
2288  003013  000          ;***** DATA PATTERN J *****
2289  003014  000          PATJ:
2290  003015  001                  .BYTE   000
2291  003016  002                  .BYTE   000
2292  003017  004                  .BYTE   001
2293  003020  020                  .BYTE   002
2294  003021  040                  .BYTE   004
2295  003022  010                  .BYTE   020
2296                               .BYTE   040
2297                               .BYTE   010
2298  003023
2299  003023  000          ;***** DATA PATTERN K *****
2300  003024  377          PATK:
2301  003025  376                  .BYTE   000
2302  003026  375                  .BYTE   377
2303  003027  373                  .BYTE   376
2304  003030  376                  .BYTE   375
2305  003031  177                  .BYTE   373
2306  003032  377                  .BYTE   376
2307  003033  000                  .BYTE   177
2308  003034  001                  .BYTE   377
2309  003035  002                  .BYTE   000
2310  003036  004                  .BYTE   001
2311  003037  010                  .BYTE   002
2312  003040  200                  .BYTE   004
2313  003041  125                  .BYTE   010
2314  003042  252                  .BYTE   200
2315  003043  000                  .BYTE   125
2316                               .BYTE   252
2317                               .BYTE   000
2318  003044
2319  003044  000          ;***** DATA PATTERN L *****
2320  003045  017          PATL:
2321  003046  016                  .BYTE   000
2322  003047  015                  .BYTE   017
2323  003050  013                  .BYTE   016
2324  003051  016                  .BYTE   015
2325  003052  017                  .BYTE   013
2326  003053  017                  .BYTE   016
2327  003054  000                  .BYTE   017
2328  003055  001                  .BYTE   017
2329  003056  002                  .BYTE   000
2330  003057  004                  .BYTE   001
2331  003060  010                  .BYTE   002
2332  003061  000                  .BYTE   004
2333  003062  005                  .BYTE   010
2334  003063  012                  .BYTE   000
2335  003064  000                  .BYTE   005
                                   .BYTE   012
                                   .BYTE   000
```

```
2336
2337
2338  003065      000          ;***** DATA PATTERN Q *****
2339  003066      003     PATQ:   .BYTE   000
2340  003067      014             .BYTE   003
2341  003070      060             .BYTE   014
2342  003071      001             .BYTE   060
2343  003072      007             .BYTE   001
2344  003073      037             .BYTE   007
2345  003074      177             .BYTE   037
2346                              .BYTE   177
2347
2348  003075      000          ;***** DATA PATTERN INVERTED Q *****
2349  003076      140     PATQB:  .BYTE   000     ;INVERTED 000 (7 BIT)
2350  003077      030             .BYTE   140     ;INVERTED 003 (7 BIT)
2351  003100      006             .BYTE   030     ;INVERTED 014 (7 BIT)
2352  003101      100             .BYTE   006     ;INVERTED 060 (7 BIT)
2353  003102      160             .BYTE   100     ;INVERTED 001 (7 BIT)
2354  003103      174             .BYTE   160     ;INVERTED 007 (7 BIT)
2355  003104      177             .BYTE   174     ;INVERTED 037 (7 BIT)
2356  003105                      .BYTE   177     ;INVERTED 177 (7 BIT)
2357        003106          ENDPAT:
                            .EVEN
```

CVDMCA.P11 10-DEC-80 09:14 DATA TEST PATTERNS

```
2358
2359
2360
2361                                    ;*** RECEIVED DATA BUFFER (64. WORDS) ***
2362  003106  000100                    RCVBUF: .BLKW   64.
2363
2364
2365
2366
```

```
                                         .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
                                         ;;*********************************************************************
003306                                           DEVTYP  <M8053 OR M8064>
003306                                                                         L$DVTYP::
003306  034115  032460  020063                                                           .ASCIZ  /M8053 OR M8064/
003314  051117  046440  030070
003322  032066     000
        003326                                                                           .EVEN

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

        000012                           .RADIX  10.
003326                                           DESCRIPT        <DMV-11 LINE UNIT TESTS - PART 1 OF 3>
003326                                                                         L$DESC::
003326  046504  026526  030461                                                           .ASCIZ  /DMV-11 LINE UNI
003334  046040  047111  020105
003342  047125  052111  052040
003350  051505  051524  026440
003356  050040  051101  020124
003364  020061  043117  031440
003372     000
        003374                                                                           .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>
                                ;
                                ;--------------------------------------------------------------------------------------------------
                                
003374  112777  000301  177072  MSTCLR: MOVB    #RUN!MCLR!MREQ,@BSEL1   ;INITIATE M-LOOP
                                
003402  010346                          MOV     R3,-(SP)
003404  012703  000030                  MOV     #24.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003410  077301                  1$:     SOB     R3,1$
003412  012603                          MOV     (SP)+,R3
                                
003414  132777  000200  177054          BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
003422  001023                          BNE     5$              ;YES.  GOOD.  RETURN
003424  004737  004210                  JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003430  012737  000301  002400          MOV     #RUN!MCLR!MREQ,GDATA    ;IDENTIFY REQUESTED FUNCTION
003436                                  GTDF    EM3,ERR4        ;'MRDY' TIMEOUT
                                                                ;       QUEUE 'DEVICE FATAL' ERROR # 1
003436  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
003444  012737  000001  002250                                                  MOV     #1,ERRNBR
003452  012737  013211  002252                                                  MOV     #EM3,ERRMSG
003460  012737  017030  002254                                                  MOV     #ERR4,ERRBLK
003466  000261                          SEC                     ;SET CARRY TO INDICATE ERROR
003470  000401                          BR      9$              ;EXIT WITH THE 'ERROR' FLAG (CARRY BIT) SET
003472  000241                  5$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
003474  000207                  9$:     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>
                                ;
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

003476  012577  177000          READ:   MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
003502  112777  000001  176766          MOVB    #REDLOC,@BSEL2  ;TELL M-LOOP TO GIVE US THE REQUESTED DATA

003510  010346                          MOV     R3,-(SP)
003512  012703  000050                  MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003516  077301                  1$:     SOB     R3,1$
003520  012603                          MOV     (SP)+,R3

003522  132777  000200  176746          BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
003530  001023                          BNE     5$              ;YES,  GOOD.  RETURN

003532  004737  004210                  JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
003536  012737  000001  002400          MOV     #REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
003544                                  GTDF    EM4,ERR4        ;'MRDY" TIMEOUT
                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 2
003544  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
003552  012737  000002  002250                                                  MOV     #2,ERRNBR
003560  012737  013235  002252                                                  MOV     #EM4,ERRMSG
003566  012737  017030  002254                                                  MOV     #ERR4,ERRBLK
003574  000261                          SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
003576  000401                          BR      6$              ;RETURN WITH THAT INDICATION

003600  000241                  5$:     CLC                     ;INDICATE 'NO ERROR''
003602  117735  176700          6$:     MOVB    @BSEL6,@(R5)+   ;PUT DATA WHERE CALLER WANTS IT
003606  000205                          RTS     R5              ;RETURN
```

```
2495                                    .SBTTL ....M-LOOP -- READ IMMEDIATE
2496                                    ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2497                                    ; READI - READ IMMEDIATE THE SPECIFIED ADDRESS WITHIN THE DMV-11 (M8053)
2498                                    ;
2499                                    ; CALLING SEQUENCE:
2500                                    ;
2501                                    ;       JSR     R5,READI
2502                                    ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
2503                                    ;       .WORD   <DESTINATION -- CONTENTS OF REG. IS PUT HERE>
2504                                    ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2505                                    ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
2506                                    ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
2507                                    ;
2508                                    ; N$:   <RESUMPTION OF NORMAL PROCESSING>
2509                                    ;
2510                                    ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
2511
2512  003610                            READI:
2513  003610  012577  176666                  MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
2514  003614  112777  000001  176654          MOVB    #REDLOC,@BSEL2  ;TELL M-LOOP TO GIVE US THE REQUESTED DATA
2515
2516  003622  010346                          MOV     R3,-(SP)
2517  003624  012703  000050                  MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2518  003630  077301                  1$:     SOB     R3,1$
2519  003632  012603                          MOV     (SP)+,R3
2520
2521  003634  132777  000200  176634          BITB    #MRDY,@BSEL2    ;DID THE M-LOOP FINISH
2522  003642  001023                          BNE     5$              ;YES,  GOOD.  RETURN
2523
2524  003644  004737  004210                  JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
2525  003650  012737  000001  002400          MOV     #REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
2526  003656                                  GTDF    EM4,ERR4        ;''MRDY'' TIMEOUT
2527                                          ;       QUEUE ''DEVICE FATAL'' ERROR # 3
2528  003656  012737  000001  002246                                          MOV     #T.EDF,ERRTYP
2529  003664  012737  000003  002250                                          MOV     #3,ERRNBR
2530  003672  012737  013235  002252                                          MOV     #EM4,ERRMSG
2531  003700  012737  017030  002254                                          MOV     #ERR4,ERRBLK
2532  003706  000261                          SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
2533  003710  000401                          BR      6$              ;RETURN WITH THAT INDICATION
2534
2535  003712  000241                  5$:     CLC                     ;INDICATE ''NO ERROR''
2536  003714  017725  176566          6$:     MOV     @SEL6,(R5)+     ;PUT DATA WHERE CALLER WANTS IT
2537  003720  000205                          RTS     R5              ;RETURN
2538
2539
2540
2541
```

```
                                .SBTTL ....M-LOOP -- WRITE
                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ; WRITE - WRITE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
                                ;
                                ;  CALLING SEQUENCE:
                                ;
                                ;       JSR     R5,WRITE
                                ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
                                ;       .WORD   <ADDRESS OF DATA BYTE>
                                ;       BCC     NS              ;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)>
                                ;
                                ; NS:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

2559  003722  012577  176554    WRITE:  MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
2560  003726  113577  176554            MOVB    @(R5)+,@SEL6    ;MAKE DATA AVAILABLE TO M-LOOP
2561  003732  000404                    BR      MLWRI           ;THE REST OF THIS ROUTINE IS THE SAME AS 'WRITEI'
```

```
                                .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     NS              ;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)>
                                ;
                                ; NS:   <RESUMPTION OF NORMAL PROCESSING>
                                ;
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

 003734                         WRITEI:
 003734  012577  176542                 MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
 003740  012577  176542                 MOV     (R5)+,@SEL6     ;MAKE DATA AVAILABLE TO M-LOOP
 003744  112777  000002  176524 1$:WRI: MOVB    #WRILOC,@BSEL2  ;TELL M-LOOP TO WRITE THE DATA
 003752  010346                 MOV     R3,-(SP)
 003754  012703  000050                 MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
 003760  077301                 1$:     SOB     R3,1$
 003762  012603                 MOV     (SP)+,R3
 003764  132777  000200  176504         BITB    #RDY,@BSEL2     ;DID THE M-LOOP FINISH
 003772  001023                 BNE     5$              ;YES,  GOOD.  RETURN
 003774  004737  004210         JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
 004000  012737  000002  002400 MOV     #WRILOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
 004006                         GTDF    EM4,ERR4        ;"RDY" TIMEOUT
                                ;               QUEUE 'DEVICE FATAL" ERROR # 4
 004006  012737  000001  002246                                 MOV     #T.EDF,ERRTYP
 004014  012737  000004  002250                                 MOV     #4,ERRNBR
 004022  012737  013235  002252                                 MOV     #EM4,ERRMSG
 004030  012737  017030  002254                                 MOV     #ERR4,ERRBLK
 004036  000261                 SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
 004040  000401                 BR      6$              ;RETURN WITH THAT INDICATION
                                
 004042  000241         5$:     CLC                     ;INDICATE "NO ERROR"
 004044  000205         6$:     RTS     R5              ;RETURN
```

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

004046  117737  176420  002256  GETBSR: MOVB    @BSEL0,BSR0     ;PUT THE CURRENT CSR VALUES INTO THE PRINT-OUT
004054  117737  176414  002260          MOVB    @BSEL1,BSR1     ;TABLE
004062  117737  176410  002262          MOVB    @BSEL2,BSR2
004070  117737  176404  002264          MOVB    @BSEL3,BSR3
004076  117737  176400  002266          MOVB    @BSEL4,BSR4
004104  117737  176374  002270          MOVB    @BSEL5,BSR5
004112  117737  176370  002272          MOVB    @BSEL6,BSR6
004120  117737  176364  002274          MOVB    @BSEL7,BSR7
004126  117737  176360  002276          MOVB    @BSEL10,BSR10
004134  117737  176354  002300          MOVB    @BSEL11,BSR11
004142  117737  176350  002302          MOVB    @BSEL12,BSR12
004150  117737  176344  002304          MOVB    @BSEL13,BSR13
004156  117737  176340  002306          MOVB    @BSEL14,BSR14
004164  117737  176334  002310          MOVB    @BSEL15,BSR15
004172  1 7737  176330  002312          MOVB    @BSEL16,BSR16
004200  117737  176324  002314          MOVB    @BSEL17,BSR17
004206  000207                          RTS     PC              ;RETURN TO CALLER

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

004210  017737  176256  002256  GETWSR: MOV     @SEL0,WSR0      ;MOVE THE 4 WORD REGISTERS TO THE OTHERWISE
004216  017737  176254  002260          MOV     @SEL2,WSR2      ;BYTE TABLE
004224  017737  176252  002262          MOV     @SEL4,WSR4
004232  017737  176250  002264          MOV     @SEL6,WSR6
004240  017737  176246  002266          MOV     @SEL10,WSR10
004246  017737  176244  002270          MOV     @SEL12,WSR12
004254  017737  176242  002272          MOV     @SEL14,WSR14
004262  017737  176240  002274          MOV     @SEL16,WSR16
004270  000207                          RTS     PC              ;RETURN TO CALLER
```

```
                        ....STUREG -- STATIC TEST OF SPECIFIED USYRT REGISTER

                        .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>
                        ;
                        ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

004272  010037  004306  STUREG: MOV     R0,2$           ;PUT SPECIFIED REGISTER'S ADDRESS IN I/O CALLS
004276  010037  004324           MOV     R0,4$

004302  004537  003722           JSR     R5,WRITE        ;WRITE IT
004306  000000           2$:     .WORD   0               ;*** MODIFIED FROM ABOVE ***
004310  002376                   .WORD   TDATA           ;
004312  103431                   BCS     10$             ;ON ERROR, EXIT

004314  005037  002402           CLR     BDATA           ;CLEAR BOTH BYTES -- JUST IN CASE....
004320  004537  003476           JSR     R5,RLAD         ;READ IT BACK AGAIN
004324  000000           4$:     .WORD   0               ;*** MODIFIED FROM ABOVE ***
004326  002402                   .WORD   BDATA
004330  103422                   BCS     10$             ;ON ERROR, EXIT

004332  123737  002400  002402   CMPB    GDATA,BDATA     ;DID WE READ WHAT WE WROTE?
004340  000241                   CLC                     ; (THIS ISN'T NEEDED FOR THE ERROR TEST BUT
                                                         ; MUST BE CLEARED ON EXIT IF NO ERROR OCCURED)
004342  001415                   BEQ     10$             ;YES, EXIT FROM SUBTEST
004344                           GTDF    EM25,ERR7A      ;REPORT READ/WRITE ERROR
                                                         ;       QUEUE 'DEVICE FATAL" ERROR # 5
004344  012737  000001  002246                                   MOV     #T.EDF,ERRTYP
004352  012737  000005  002250                                   MOV     #5,ERRNBR
004360  012737  013366  002252                                   MOV     #EM25,ERRMSG
004366  012737  017252  002254                                   MOV     #ERR7A,ERRBLK
004374  000261                   SEC                     ;INDICATE THAT AN ERROR WAS DETECTED
004376  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
                        ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

004400  000207           STALL:  RTS     PC
```

```
2714
2715
```

```
2716                                    .SBTTL
2717
2718                                    ;*******************************************************************************
2719                                    ;* GETURS - LOAD INTO THE 8 WORD STORAGE AREA (UREGS) THE CONTENTS OF THE
2720                                    ;*      VARIOUS USYRT REGISTERS
2721                                    ;*
2722                                    ;*        CALLING SEQUENCE:
2723                                    ;*
2724                                    ;*******************************************************************************
2725  004402  012737  002316  004444  GETURS: MOV     #UREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
2726  004410  012737  120400  004442          MOV     #USYRT,4$       ;INIT POINTER TO REGISTER ADDRESSES
2727
2728  004416  005037  002334                  CLR     UREGS+14.       ;CLEAR STORAGE WORD
2729  004422  004537  003476                  JSR     R5,READ         ;READ THE USYRT STATUS REGISTER
2730  004426  122000                          .WORD   USTATR          ;STATUS REGISTER'S ADDRESS WITHIN DMV-11
2731  004430  002334                          .WORD   UREGS+14.       ;ADDRESS ALLOCATED TO THAT REG. W/IN ''UREGS''
2732
2733  004432  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
2734  004436  004537  003476                  JSR     R5,READ         ;READ A LINE UNIT REG
2735  004442  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
2736  004444  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
2737
2738  004446  005237  004442          6$:     INC     4$              ;INCREMENT REG NO.
2739  004452  023727  004442  120406          CMP     4$,#USYRT+6     ;THIS IS NOT A VALID REGISTER ADDRESS
2740  004460  001772                          BEQ     6$              ;SO IT MUST BE BYPASSED
2741
2742  004462  062737  000002  004444          ADD     #2,5$           ;ADVANCE ADDRESS OF STORAGE AREA POINTER
2743  004470  023727  004442  120410          CMP     4$,#USYRT+10    ;SEE IF ALL REGS READ YET
2744  004476  001355                          BNE     3$              ;BR IF NOT
2745
2746  004500  000207                          RTS     PC              ;RETURN
2747
2748
2749
2750                                    ;*******************************************************************************
2751                                    ;* GETVRS: - LOAD INTO THE 16 WORD STORAGE AREA (VREGS) THE CONTENTS OF THE
2752                                    ;*      VARIOUS VIA REGISTERS.
2753                                    ;*
2754                                    ;*      CALLING SEQUENCE :
2755                                    ;*******************************************************************************
2756  004502  012737  002336  004530  GETVRS: MOV     #VREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
2757  004510  012737  120000  004526          MOV     #VIA,4$         ;INIT POINTER TO REGISTER ADDRESSES
2758  004516  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
2759  004522  004537  003476                  JSR     R5,READ         ;READ A VIA REG
2760  004526  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
2761  004530  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
2762  004532  005237  004526          6$:     INC     4$              ;INCREMENT REG NO.
2763  004536  062737  000002  004530          ADD     #2,5$           ;INCREMENT STORAGE ADRS
2764  004544  023727  004526  120020          CMP     4$,#VIA+16.     ;SEE IF ALL VIA REGS READ YET
2765  004552  001361                          BNE     3$              ;BR IF NOT
2766  004554  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''.)
                        ;*
                        ;*************************************************************************

004556  010146          INITT1: MOV     R1,-(SP)        ;SAVE THE REGISTER WE WILL BE USING
004560  012537  004702          MOV     (R5)+,7$        ;SETUP VALUE TO BE WRITTEN IN LATCH
004564  012537  004730          MOV     (R5)+,10$       ;SETUP VALUE TO BE WRITTEN IN COUNTER
004570  111501                  MOVB    (R5),R1         ;GET & PROCESS BITS FOR ACR 6 & 7
004572  143701  000077          BICB    077,R1
004576  010137  004672          MOV     R1,4$           ;SETUP CALL SET ACR'S BITS 6 & 7
004602  112501                  MOVB    (R5)+,R1        ;NOW, GET THE BIT TO BE USED IN SETTING OR
                                                        ;CLEARING BIT 6 OF ''IER''
004604  106301                  ASLB    R1              ;THE PASSED BIT IS IN THE WRONG POSITION
004606  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).
004610  143701  000177          BICB    177,R1          ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
004614  153701  000100          BISB    100,R1          ;THEN SET BIT 6
004620  010137  004632          MOV     R1,2$           ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE

004624  004537  003734          JSR     R5,WRITEI       ;WRITE TO
004630  120016                  VIAIER                  ;THE VIA'S IER
004632  000000          2$:     .WORD   0               ;INTERRUPT ENABLE/DISABLE INFORMATION

004634  004537  003610          JSR     R5,READI        ;READ THE CURRENT SETTING OF
004640  120013                  VIAACR                  ;THE VIA'S ACR
004642  000000          3$:     .WORD   0               ;INTO '3$'

004644  013701  004642          MOV     3$,R1           ;GET THAT VALUE
004650  143701  000300          BICB    300,R1          ;CLEAR THE CURRENT SETTING OF BITS 6 & 7
004654  053701  004672          BIS     4$,R1           ;SET THEM ACCORDING TO THE PASSED VALUES
```

```
                        ....INITT1 -- INITIALIZE TIMER #1

2823  004660  010137  004672                MOV     R1,4$           ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
2824
2825  004664  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2826  004670  120013                        VIAACR                  ;THE VIA'S ACR
2827  004672  000000              4$:       .WORD   0               ;THE NEW REGISTER SETTING
2828
2829  004674  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2830  004700  120006                        VIAT1C                  ;LOW ORDER LATCH REGISTER (T1L-L)
2831  004702  000000              7$:       .WORD   0               ;THE VALUE PASSED
2832
2833  004704  113737  004703  004720        MOVB    7$+1,8$         ;SETUP FOR AND
2834  004712  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2835  004716  120007                        VIAT1D                  ;HIGH ORDER LATCH REGESTER (T1L-H)
2836  004720  000000              8$:       .WORD   0               ;THE VALUE PASSED
2837
2838  004722  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2839  004726  120004                        VIAT1A                  ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2840  004730  000000               $:       .WORD   0               ;THE VALUE PASSED
2841
2842  004732  113737  004731  004746        MOVB    10$+1,11$       ;SETUP FOR AND
2843  004740  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2844  004744  120005                        VIAT1B                  ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2845  004746  000000              11$:      .WORD   0               ;THE VALUE PASSED
2846
2847                                ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST RETURN!
2848
2849  004750  012601                        MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2850  004752  005205                        INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2851                                                                ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2852
2853  004754  000205                        RTS     R5              ;NOW, RETURN
2854
2855
```

```
                                .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".)
                                ;*
                                ;*****************************************************************************

004756  010146          INITT2: MOV     R1,-(SP)        ;SAVE THE REGISTER WE WILL BE USING
004760  012537  005100          MOV     (R5)+,10$       ;SETUP VALUE TO BE WRITTEN IN COUNTER
004764  111501                  MOVB    (R5),R1         ;GET & PROCESS BIT FOR ACR 5
004766  143701  000337          BICB    337,R1
004772  010137  005070          MOV     R1,4$           ;SETUP CALL TO SET OR CLEAR ACR'S BIT 5
004776  112501                  MOVB    (R5)+,R1        ;NOW, GET THE BIT TO BE USED IN SETTING OR
                                                        ;CLEARING BIT 5 OF "IER"
005000  106301                  ASLB    R1              ;THE PASSED BIT IS IN THE WRONG POSITION
005002  106301                  ASLB    R1              ;BUT, THE PASSED BIT SHOULD CONTROL THE
005004  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).
005006  143701  000177          BICB    177,R1          ;FIRST, MAKE SURE ALL UNWANTED BITS ARE CLEARED
005012  153701  000040          BISB    040,R1          ;THEN SET BIT 5
005016  010137  005030          MOV     R1,2$           ;THE CALL WILL NOW WRITE THE APPROPRIATE VALUE

005022  004537  003734          JSR     R5,WRITEI       ;WRITE TO
005026  120016                  VIAIER                  ;THE VIA'S IER
005030  000000          2$:     .WORD   0               ;INTERRUPT ENABLE/DISABLE INFORMATION

005032  004537  003610          JSR     R5,READI        ;READ THE CURRENT SETTING OF
005036  120013                  VIAACR                  ;THE VIA'S ACR
005040  000000          3$:     .WORD   0               ;INTO "3$"

005042  013701  005040          MOV     3$,R1           ;GET THAT VALUE
005046  143701  000040          BICB    040,R1          ;CLEAR THE CURRENT SETTING OF BIT 5
005052  053701  005070          BIS     4$,R1           ;SET IT ACCORDING TO THE PASSED VALUE
005056  010137  005070          MOV     R1,4$           ;PASS NEW REG. SETTING TO APPROPRIATE CALL
```

```
2912  005062  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2913  005066  120013                        VIAACR                  ;THE VIA'S ACR
2914  005070  000000              4$:       .WORD   0               ;THE NEW REGISTER SETTING
2915
2916  005072  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2917  005076  120010                        VIAT2A                  ;LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
2918  005100  000000              10$:      .WORD   0               ;THE VALUE PASSED
2919
2920  005102  113737  005101  0u5116        MOVB    10$+1,11$       ;SETUP FOR AND
2921  005110  004537  003734                JSR     R5,WRITEI       ;WRITE TO
2922  005114  120011                        VIAT2B                  ;HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
2923  005116  000000              11$       .WORD   0               ;THE VALUE PASSED
2924
2925                                        ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
2926
2927  005120  012601                        MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2928  005122  005205                        INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2929                                                                ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2930
2931  005124  000205                        RTS     R5              ;THEN RETURN
2932
```

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

005126                  RSTCHK:
005126  010146                  MOV     R1,-(SP)        ;SAVE R1
005130  010246                  MOV     R2,-(SP)        ;SAVE R2

005132  004537  003734          JSR     R5,WRITEI       ;SET PROGRAM RESET BIT IN VIA ORB REG
005136  120000                  VIAORB
005140  000031                  DTR!RTSND!PRESET
005142  004537  003734          JSR     R5,WRITEI       ;CLEAR PROGRAM RESET BIT IN VIA ORB REG
005146  120000                  VIAORB
005150  000030                  DTR!RTSND

005152  005001                  CLR     R1              ;INIT USYRT REG ADRS PTR
005154  012702  002662          MOV     #PATF,R2        ;INIT DATA PATTERN POINTER
005160  016137  002552  005172  6$:     MOV     USYREG(R1),7$   ;SET USYRT READ ADDRESS
005166  004537  003610          JSR     R5,READI        ;READ A USYRT REG
005172  000000                  7$:     .WORD   0       ;USYRT REG ADRS GOES HERE
005174  000000                  8$:     .WORD   0       ;DATA READ IS RETURNED HERE
005176  123722  005174          CMPB    8$,(R2)+        ;SEE IF REG CONTAINS EXPECTED DATA
005202  001432                  BEQ     9$              ;BR IF MATCH

005204  010137  002412          MOV     R1,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
005210  006237  002412          ASR     REGNUM          ;GET WORD OFFSET
005214  005037  002400          CLR     GDATA           ;GET EXPECTED DATA
005220  116237  177777  002400  MOVB    -1(R2),GDATA
005226  013737  005174  002402  MOV     8$,BDATA        ;GET ACTUAL DATA
                        ;STACK 'USYRT NOT CLEARED BY PROGRAM RESET' MSG -
005234                  GTDF    EM2,ERR10
                                ;       QUEUE 'DEVICE FATAL' ERROR # 6
005234  012737  000001  002246                  MOV     #T.EDF,ERRTYP
005242  012737  000006  002250                  MOV     #6,ERRNBR
005250  012737  013142  002252                  MOV     #EM2,ERRMSG
005256  012737  017372  002254                  MOV     #ERR10,ERRBLK
005264  000261                  SEC             ;SET C BIT TO FLAG ERROR
005266  000406                  BR      10$     ;TAKE ERROR EXIT

005270  062701  000002  9$:     ADD     #2,R1           ;INCR USYRT REG ADRS PTR
005274  020127  000020          CMP     R1,#16.         ;SEE IF ALL REGS READ YET
005300  002727                  BLT     6$              ;BR IF NOT
005302  000241                  CLC                     ;** CLEAR C BIT FOR NO ERRORS
005304  012602          10$:    MOV     (SP)+,R2        ;RESTORE R2
005306  012601                  MOV     (SP)+,R1        ;RESTORE R1
005310  000205                  RTS     R5              ;** RETURN
```

```
                                ;*****************************************************************************
                                ;* WAIT50 - THIS SUBROUTINE STALLS FOR AT LEAST 50 MICRO-SEC, AND THEN RETURNS.
                                ;*****************************************************************************
005312  010146                  WAIT50: MOV     R1,-(SP)        ;SAVE R1
005314  012701  000005                  MOV     #5.,R1          ;INIT COUNTER
005320  077101                  3$:     SOB     R1,3$           ;DELAY HERE FOR 23.8 MICRO-SEC'S
005322  012601                          MOV     (SP)+,R1        ;RESTORE R1
005324  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
                                ;*****************************************************************************
005326                          SETVIA:
005326  004537  003734                  JSR     R5,WRITEI       ;SET PORT B FOR OUTPUT MODE
005332  120002                          VIADPB
005334  000377                          377
005336  004537  003734                  JSR     R5,WRITEI       ;SET PORT A FOR INPUT MODE
005342  120003                          VIADPA                  ; (BIT0 IS ONLY OUTPUT BIT)
005344  000001                          001
005346  004537  003734                  JSR     R5,WRITEI       ;DISABLE USYRT INTERNAL LOOPBACK
005352  120017                          VIAORA
005354  000000                          000
005356  004537  003734                  JSR     R5,WRITEI       ;INIT PORT B
005362  120000                          VIAORB
005364  000030                          DTR!RTSND
005366  004537  003734                  JSR     R5,WRITEI       ;SET ACR FOR :  T1 SQUARE WAVE OUTPUT MODE,
005372  120013                          VIAACR                  ;               T2 ONE-SHOT OUTPUT MODE,
005374  000350                          350                     ;               SR AT SYS CLOCK RATE ON CB1
005376  004537  003734                  JSR     R5,WRITEI       ;SET PCR FOR :  CB1 NEG TRANS INPUT MODE,
005402  120014                          VIAPCR                  ;               CA2 NEG TRANS INPUT MODE,
005404  000022                          022                     ;               CA1 NEG TRANS INPUT MODE
005406  004537  003734                  JSR     R5,WRITEI       ;DISABLE ALL MICRO-INTRPTS
005412  120016                          VIAIER
005414  000177                          177
005416  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
                                ;*********************************************************************
005420  004737  003374  INIDMV: JSR     PC,MSTCLR       ;MASTER CLR, M-LOOP
005424  004737  005326          JSR     PC,SETVIA       ;PROGRAM VIA
005430  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.
                                ;*********************************************************************
005432                  CKUSTS:
005432  004537  003610          JSR     R5,READI        ;READ USYRT STATUS REGISTER
005436  122000                  USTATR
005440  000000          1$:     .WORD   0
005442  122537  005440          CMPB    (R5)+,1$        ;SEE IF STATUS MATCHES EXPECTED
005446  000241                  CLC                     ;CLEAR C BIT
005450  001430                  BEQ     2$              ;BR IF STATUS OK
005452  012737  000007  002412  MOV     #7,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
005460  016537  177777  002400  MOV     -1(R5),GDATA    ;GET EXPECTED DATA
005466  005037  002402          CLR     BDATA           ;GET ACTUAL DATA
005472  113737  005440  002402  MOVB    1$,BDATA
                                ;STACK 'USYRT STATUS INCORRECT'' ERROR
005500                          GTDF    EM68,ERR10
                                ;       QUEUE 'DEVICE FATAL'' ERROR # 7
005500  012737  000001  002246                          MOV     #T.EDF,ERRTYP
005506  012737  000007  002250                          MOV     #7,ERRNBR
005514  012737  013760  002252                          MOV     #EM68,ERRMSG
005522  012737  017372  002254                          MOV     #ERR10,ERRBLK
005530  000261                  SEC                     ;SET C BIT FOR ERROR
005532  005205          2$:     INC     R5              ;INCREMENT R5 PAST ARGUMENT
005534  000205                  RTS     R5              ;RETURN
```

```
3086                            .SBTTL ....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)
3087                            ;*****************************************************************************
3088                            ;* CKTACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TXACT IN THE USYRT
3089                            ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
3090                            ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
3091                            ;*
3092                            ;*      CALLING SEQUENCE :
3093                            ;*      JSR     R5,CKTACT
3094                            ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TXACT>
3095                            ;*****************************************************************************
3096  005536                    CKTACT:
3097  005536 012737 000007 002412       MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
3098  005544 004537 003610              JSR     R5,READI        ;READ USYRT STATUS
3099  005550 122000                     USTATR
3100  005552 000000             1$:     .WORD   0
3101  005554 032725 000001              BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TXACT
3102  005560 001422                     BEQ     2$              ;BR IF EXPECTED TXACT = 0
3103  005562 132737 000004 005552       BITB    #TXACT,1$       ;SEE IF TXACT = 1
3104  005570 001040                     BNE     3$              ;BR IF TXACT = 1
3105                            ;STACK 'TXACT NOT SET'' MSG
3106  005572                            GTDF    EM69,ERR12
3107                                                            ;       QUEUE 'DEVICE FATAL'' ERROR # 8
3108  005572 012737 000001 002246                                                       MOV     #T.EDF,ERRTYP
3109  005600 012737 000010 002250                                                       MOV     #8,ERRNBR
3110  005606 012737 014007 002252                                                       MOV     #EM69,ERRMSG
3111  005614 012737 017722 002254                                                       MOV     #ERR12,ERRBLK
3112  005622 000261                     SEC                     ;SET C BIT TO FLAG ERROR
3113  005624 000423                     BR      4$              ;TAKE ERROR EXIT
3114  005626 132737 000004 005552 2$:   BITB    #TXACT,1$       ;SEE IF TXACT = 0
3115  005634 001416                     BEQ     3$              ;BR IF TXACT = 0
3116                            ;STACK 'TXACT NOT CLEARED'' MSG
3117  005636                            GTDF    EM70,ERR12
3118                                                            ;       QUEUE 'DEVICE FATAL'' ERROR # 9
3119  005636 012737 000001 002246                                                       MOV     #T.EDF,ERRTYP
3120  005644 012737 000011 002250                                                       MOV     #9,ERRNBR
3121  005652 012737 014025 002252                                                       MOV     #EM70,ERRMSG
3122  005660 012737 017722 002254                                                       MOV     #ERR12,ERRBLK
3123  005666 000261                     SEC                     ;SET C BIT TO FLAG ERROR
3124  005670 000401                     BR      4$              ;TAKE ERROR EXIT
3125  005672 000241             3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
3126  005674 000205             4$:     RTS     R5              ;RETURN
3127
3128
3129
3130
```

```
                                .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>
                                ;*****************************************************************************
005676                          CKRACT:
005676  012737  000007  002412          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
005704  004537  003610                  JSR     R5,READI        ;READ USYRT STATUS
005710  122000                          USTATR
005712  000000                  1$:     .WORD   0
005714  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RXACT
005720  001422                          BEQ     2$              ;BR IF EXPECTED RXACT = 0
005722  132737  000040  005712          BITB    #RXACT,1$       ;SEE IF RXACT = 1
005730  001040                          BNE     3$              ;BR IF RXACT = 1
                                ;STACK 'RXACT NOT SET'' MSG
005732                                  GTDF    EM71,ERR12
                                ;       QUEUE 'DEVICE FATAL'' ERROR # 10
005732  012737  000001  002246                                          MOV     #T.EDF,ERRTYP
005740  012737  000012  002250                                          MOV     #10,ERRNBR
005746  012737  014047  002252                                          MOV     #EM71,ERRMSG
005754  012737  017722  002254                                          MOV     #ERR12,ERRBLK
005762  000261                          SEC                     ;SET C BIT TO FLAG ERROR
005764  000423                          BR      4$              ;TAKE ERROR EXIT
005766  132737  000040  005712  2$:     BITB    #RXACT,1$       ;SEE IF RXACT = 0
005774  001416                          BEQ     3$              ;BR IF RXACT = 0
                                ;STACK 'RXACT NOT CLEARED'' MSG
005776                                  GTDF    EM72,ERR12
                                ;       QUEUE 'DEVICE FATAL'' ERROR # 11
005776  012737  000001  002246                                          MOV     #T.EDF,ERRTYP
006004  012737  000013  002250                                          MOV     #11,ERRNBR
006012  012737  014065  002252                                          MOV     #EM72,ERRMSG
006020  012737  017722  002254                                          MOV     #ERR12,ERRBLK
006026  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006030  000401                          BR      4$              ;TAKE ERROR EXIT
006032  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006034  000205                  4$:     RTS     R5              ;RETURN
```

```
3176                                    .SBTTL ....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY
3177                                    ;*********************************************************************
3178                                    ;* CKTBMT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TBMT IN THE USYRT
3179                                    ;*       STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
3180                                    ;*       STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
3181                                    ;*
3182                                    ;*       CALLING SEQUENCE :
3183                                    ;*       JSR     R5,CKTBMT
3184                                    ;*       .WORD   <BIT 0 IS EXPECTED VALUE OF TBMT>
3185                                    ;*********************************************************************
3186  006036                            CKTBMT:
3187  006036  012737  000007  002412            MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
3188  006044  004537  003610                    JSR     R5,READI        ;READ USYRT STATUS
3189  006050  122000                            USTATR
3190  006052  000000                    1$:     .WORD   0
3191  006054  032725  000001                    BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TBMT
3192  006060  001422                            BEQ     2$              ;BR IF EXPECTED TBMT = 0
3193  006062  132737  000100  006052            BITB    #TBMT,1$        ;SEE IF TBMT = 1
3194  006070  001040                            BNE     3$              ;BR IF TBMT = 1
3195                                    ;STACK 'TBMT NOT SET'' MSG
3196  006072                                    GTDF    EM73,ERR12
3197                                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 12
3198  006072  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
3199  006100  012737  000014  002250                                                    MOV     #12,ERRNBR
3200  006106  012737  014107  002252                                                    MOV     #EM73,ERRMSG
3201  006114  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
3202  006122  000261                            SEC                     ;SET C BIT TO FLAG ERROR
3203  006124  000423                            BR      4$              ;TAKE ERROR EXIT
3204  006126  132737  000100  006052    2$:     BITB    #TBMT,1$        ;SEE IF TBMT = 0
3205  006134  001416                            BEQ     3$              ;BR IF TBMT = 0
3206                                    ;STACK 'TBMT NOT CLEARED'' MSG
3207  006136                                    GTDF    EM74,ERR12
3208                                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 13
3209  006136  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
3210  006144  012737  000015  002250                                                    MOV     #13,ERRNBR
3211  006152  012737  014124  002252                                                    MOV     #EM74,ERRMSG
3212  006160  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
3213  006166  000261                            SEC                     ;SET C BIT TO FLAG ERROR
3214  006170  000401                            BR      4$              ;TAKE ERROR EXIT
3215  006172  000241                    3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
3216  006174  000205                    4$:     RTS     R5              ;RETURN
3217
3218
3219
3220
```

```
                                .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>
                                ;*****************************************************************************
006176                          CKRDA:
006176  012737  000007  002412          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006204  004537  003610                  JSR     R5,READI        ;READ USYRT STATUS
006210  122000                          USTATR
006212  000000                  1$:     .WORD   0
006214  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RDA
006220  001422                          BEQ     2$              ;BR IF EXPECTED RDA = 0
006222  132737  000200  006212          BITB    #RDA,1$         ;SEE IF RDA = 1
006230  001040                          BNE     3$              ;BR IF RDA = 1
                                ;STACK 'RDA NOT SET'' MSG
006232                                  GTDF    EM75,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 14
006232  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006240  012737  000016  002250                                                  MOV     #14,ERRNBR
006246  012737  014145  002252                                                  MOV     #EM75,ERRMSG
006254  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006262  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006264  000423                          BR      4$              ;TAKE ERROR EXIT
006266  132737  000200  006212  2$:     BITB    #RDA,1$         ;SEE IF RDA = 0
006274  001416                          BEQ     3$              ;BR IF RDA = 0
                                ;STACK 'RDA NOT CLEARED'' MSG
006276                                  GTDF    EM76,ERR12
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 15
006276  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006304  012737  000017  002250                                                  MOV     #15,ERRNBR
006312  012737  014161  002252                                                  MOV     #EM76,ERRMSG
006320  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006326  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006330  000401                          BR      4$              ;TAKE ERROR EXIT
006332  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006334  000205                  4$:     RTS     R5              ;RETURN
```

CVDMCA.P11 10-DEC-80 09:14 ....CKRSA -- CHECK RECEIVER STATUS AVAILABLE

```
                                .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>
                                ;*****************************************************************************
006336                          CKRSA:
006336  012737  000007  002412          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006344  004537  003610                  JSR     R5,READI        ;READ USYRT STATUS
006350  122000                          USTATR
006352  000000                  1$:     .WORD   0
006354  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RSA
006360  001422                          BEQ     2$              ;BR IF EXPECTED RSA = 0
006362  132737  000020  006352          BITB    #RSA,1$         ;SEE IF RSA = 1
006370  001040                          BNE     3$              ;BR IF RSA = 1
                                ;STACK 'RSA NOT SET'' MSG
006372                                  GTDF    EM77,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 16
006372  012737  000001  002246                                          MOV     #T.EDF,ERRTYP
006400  012737  000020  002250                                          MOV     #16,ERRNBR
006406  012737  014201  002252                                          MOV     #EM77,ERRMSG
006414  012737  017722  002254                                          MOV     #ERR12,ERRBLK
006422  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006424  000423                          BR      4$              ;TAKE ERROR EXIT
006426  132737  000020  006352  2$:     BITB    #RSA,1$         ;SEE IF RSA = 0
006434  001416                          BEQ     3$              ;BR IF RSA = 0
                                ;STACK 'RSA NOT CLEARED'' MSG
006436                                  GTDF    EM78,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 17
006436  012737  000001  002246                                          MOV     #T.EDF,ERRTYP
006444  012737  000021  002250                                          MOV     #17,ERRNBR
006452  012737  014215  002252                                          MOV     #EM78,ERRMSG
006460  012737  017722  002254                                          MOV     #ERR12,ERRBLK
006466  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006470  000401                          BR      4$              ;TAKE ERROR EXIT
006472  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006474  000205                  4$:     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>
                                ;*****************************************************************************
006476                          CKROR:
006476  012737  000001  002412          MOV     #1,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006504  004537  003610                  JSR     R5,READI        ;READ RECEIVER STATUS
006510  120401                          RDSRH
006512  000000                  1$:     .WORD   0
006514  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF ROR
006520  001422                          BEQ     2$              ;BR IF EXPECTED ROR = 0
006522  132737  000010  006512          BITB    #ROR,1$         ;SEE IF ROR = 1
006530  001040                          BNE     3$              ;BR IF ROR = 1
                                ;STACK 'RECEIVER OVRN NOT SET'' MSG
006532                                  GTDF    EM90,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 18
006532  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006540  012737  000022  002250                                                  MOV     #18,ERRNBR
006546  012737  014235  002252                                                  MOV     #EM90,ERRMSG
006554  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006562  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006564  000423                          BR      4$              ;TAKE ERROR EXIT
006566  132737  000010  006512  2$:     BITB    #ROR,1$         ;SEE IF ROR = 0
006574  001416                          BEQ     3$              ;BR IF ROR = 0
                                ;STACK 'ROR NOT CLEARED'' MSG
006576                                  GTDF    EM91,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 19
006576  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006604  012737  000023  002250                                                  MOV     #19,ERRNBR
006612  012737  014266  002252                                                  MOV     #EM91,ERRMSG
006620  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006626  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006630  000401                          BR      4$              ;TAKE ERROR EXIT
006632  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006634  000205                  4$:     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>
                                ;*****************************************************************************
006636                          CKSEOM:
006636  012737  000007  002412          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006644  004537  003610                  JSR     R5,READI        ;READ USYRT RECEIVER STATUS
006650  120401                          RDSRH
006652  000000                  1$:     .WORD   0
006654  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RSOM
006660  001422                          BEQ     2$              ;BR IF EXPECTED RSOM = 0
006662  132737  000001  006652          BITB    #RSOM,1$        ;SEE IF RSOM = 1
006670  001040                          BNE     3$              ;BR IF RSOM = 1
                                ;STACK 'RSOM NOT SET'' MSG
006672                                  GTDF    EM29,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 20
006672  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006700  012737  000024  002250                                                  MOV     #20,ERRNBR
006706  012737  013520  002252                                                  MOV     #EM29,ERRMSG
006714  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006722  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006724  000473                          BR      6$              ;TAKE ERROR EXIT
006726  132737  000001  006652  2$:     BITB    #RSOM,1$        ;SEE IF RSOM = 0
006734  001416                          BEQ     3$              ;BR IF RSOM = 0
                                ;STACK 'RSOM NOT CLEARED'' MSG
006736                                  GTDF    EM28,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 21
006736  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006744  012737  000025  002250                                                  MOV     #21,ERRNBR
006752  012737  013477  002252                                                  MOV     #EM28,ERRMSG
006760  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006766  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006770  000451                          BR      6$              ;TAKE ERROR EXIT
006772  032765  000002  177776  3$:     BIT     #BIT1,-2(R5)    ;GET EXPECTED STATE OF REOM
007000  001422                          BEQ     4$              ;BR IF EXPECTED REOM = 0
007002  132737  000002  006652          BITB    #REOM,1$        ;SEE IF REOM = 1
007010  001040                          BNE     5$              ;BR IF REOM = 1
                                ;STACK 'REOM NOT SET'' MSG
007012                                  GTDF    EM31,ERR12
                                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 22
007012  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
007020  012737  000026  002250                                                  MOV     #22,ERRNBR
007026  012737  013556  002252                                                  MOV     #EM31,ERRMSG
007034  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
007042  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007044  000423                          BR      6$              ;TAKE ERROR EXIT
007046  132737  000002  006652  4$:     BITB    #REOM,1$        ;SEE IF REOM = 0
```

```
3409  007054  001416                        BEQ     5$              ;BR IF REOM = 0
3410                                  ;STACK 'REOM NOT CLEARED'' MSG
3411  007056                                GTDF    EM30,ERR12
3412                                        ;       QUEUE 'DEVICE FATAL'' ERROR # 23
3413  007056  012737  000001  002246                                  MOV     #T.EDF,ERRTYP
3414  007064  012737  000027  002250                                  MOV     #23,ERRNBR
3415  007072  012737  013535  002252                                  MOV     #EM30,ERRMSG
3416  007100  012737  017722  002254                                  MOV     #ERR12,ERRBLK
3417  007106  000261                        SEC                     ;SET C BIT TO FLAG ERROR
3418  007110  000401                        BR      6$              ;TAKE FRROR EXIT
3419  007112  000241                  5$:   CLC                     ;CLEAR C BIT FOR NO ERRORS
3420  007114  000205                  6$:   RTS     R5              ;RETURN
3421
3422
```

```
                                .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>
                                ;*****************************************************************************
007116                          CHKTSO:
007116  012737  000007  002412          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
007124  004537  003610                  JSR     R5,READI        ;READ USYRT STATUS
007130  122000                          USTATR
007132  000000                  1$:     .WORD   0
007134  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TSO
007140  001422                          BEQ     2$              ;BR IF EXPECTED TSO = 0
007142  132737  000010  007132          BITB    #TSO,1$         ;SEE IF TSO = 1
007150  001040                          BNE     3$              ;BR IF TSO = 1
                                        ;*** STACK 'TSO NOT SET" ERROR ***
007152                                  GTDF    EM100,ERR12
                                                        ;       QUEUE 'DEVICE FATAL" ERROR # 24
007152  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
007160  012737  000030  002250                                                  MOV     #24,ERRNBR
007166  012737  014425  002252                                                  MOV     #EM100,ERRMSG
007174  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
007202  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007204  000423                          BR      4$              ;TAKE ERROR EXIT

007206  132737  000010  007132  2$:     BITB    #TSO,1$         ;SEE IF TSO = 0
007214  001416                          BEQ     3$              ;BR IF TSO = 0
                                        ;*** STACK 'TSO NOT CLEARED" ERROR ***
007216                                  GTDF    EM101,ERR12
                                                        ;       QUEUE 'DEVICE FATAL" ERROR # 25
007216  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
007224  012737  000031  002250                                                  MOV     #25,ERRNBR
007232  012737  014445  002252                                                  MOV     #EM101,ERRMSG
007240  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
007246  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007250  000401                          BR      4$              ;TAKE ERROR EXIT
007252  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
007254  000205                  4$:     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>
                                ;*************************************************************************
007256                          SERIAL:
007256  010146                          MOV     R1,-(SP)        ;SAVE R1
007260  010246                          MOV     R2,-(SP)        ;SAVE R2 (TICKS)
007262  010346                          MOV     R3,-(SP)        ;SAVE R3 (EXPECTED_WORD)

007264  005001                          CLR     R1              ;CLEAR ASSEMBLED_WORD
007266  012502                          MOV     (R5)+,R2        ;GET # OF TICKS

007270  006301                  1$:     ASL     R1              ;SHIFT ASSEMBLED_WORD
007272  004537  011614                  JSR     R5,STEPLU       ;CLOCK USYRT ONCE
007276  000001                          1

007300  004537  007116                  JSR     R5,CHKTSO       ;CHECK FOR TSO=1
007304  000001                          1
007306  103401                          BCS     2$              ;BR IF TSO=0
007310  005201                          INC     R1              ;TSO=1: SET LSB OF ASSEMBLED_WORD
007312  077212                  2$:     SOB     R2,1$           ;LOOP UNTIL NO MORE TICKS

007314  012503                          MOV     (R5)+,R3        ;GET EXPECTED_WORD
007316  020103                          CMP     R1,R3           ;COMPARE EXPECTED  AND ASSEMBLED_WORD
007320  001422                          BEQ     3$              ;BR IF CORRECT VALUE READ

007322  010337  002400                  MOV     R3,GDATA        ;EXPECTED_WORD => GDATA
007326  010137  002402                  MOV     R1,BDATA        ;ASSEMBLED_WORD => BDATA
                                        ;*** STACK 'TRANSMISSION ERROR'' MSG ***
007332                                  GTDF    EM102,ERR13
                                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 26
007332  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
007340  012737  000032  002250                                                  MOV     #26,ERRNBR
007346  012737  014471  002252                                                  MOV     #EM102,ERRMSG
007354  012737  020036  002254                                                  MOV     #ERR13,ERRBLK
007362  000261                          SEC                     ;SET C BIT TO FLAG ERROR
007364  000401                          BR      .+4             ;TAKE ERROR EXIT

007366  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
007370  012603                          MOV     (SP)+,R3        ;RESTORE REGISTERS
007372  012602                          MOV     (SP)+,R2
007374  012601                          MOV     (SP)+,R1
007376  000205                  4$:     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)>
                                ;*****************************************************************************
3537  007400                    INITRN:
3538  007400  010146                   MOV      R1,-(SP)          ;SAVE R1
3539  007402  004537  003734           JSR      R5,WRITEI         ;RESET THE USYRT
3540  007406  120000                   VIAORB
3541  007410  000031                   RTSND!DTR!PRESET
3542  007412  004537  003734           JSR      R5,WRITEI         ;CLEAR USYRT RESET BIT
3543  007416  120000                   VIAORB
3544  007420  000030                   RTSND!DTR
3545  007422  112537  007434           MOVB     (R5)+,1$          ;GET VALUE TO LOAD INTO USYRT PCSARL
3546  007426  004537  003734           JSR      R5,WRITEI         ;LOAD USYRT PCSARL
3547  007432  120404                   PCSARL
3548  007434  000000            1$:    .WORD    0
3549  007436  112537  007450           MOVB     (R5)+,2$          ;GET VALUE TO LOAD INTO PCSARH
3550  007442  004537  003734           JSR      R5,WRITEI         ;LOAD USYRT PCSARH
3551  007446  120405                   PCSARH
3552  007450  000000            2$:    .WORD    0
3553  007452  112537  007476           MOVB     (R5)+,3$          ;GET VALUE TO LOAD INTO PCR
3554  007456  005037  002456           CLR      SAVLEN
3555  007462  113737  007476  002456   MOVB     3$,SAVLEN         ;SAVE CHAR LENGTH BITS
3556  007470  004537  003734           JSR      R5,WRITEI         ;LOAD USYRT PCR
3557  007474  120407                   PCR
3558  007476  000000            3$:    .WORD    0
3559  007500  004537  003734           JSR      R5,WRITEI         ;SET ACR FOR T1 ONE-SHOT MODE
3560  007504  120013                   VIAACR
3561  007506  000200                   200
3562  007510  004537  003734           JSR      R5,WRITEI         ;LOAD VIA T1L-L
3563  007514  120006                   VIAT1C
3564  007516  000300                   300
3565  007520  004537  003734           JSR      R5,WRITEI         ;LOAD VIA T1L-H
3566  007524  120007                   VIAT1D
3567  007526  000000                   000
3568  007530  004537  005432           JSR      R5,CKUSTS         ;CHK USYRT STATUS FOR INIT'D STATE
3569  007534  000110                   110                        ; TBMT = 1, TSO = 1
3570  007536  103454                   BCS      7$                ;IF ERROR, EXIT SUBROUTINE
3571
3572  007540  013737  007674  007560   MOV      20$,13$           ;* SET UP DEFAULT VIAORB PARAMETERS
3573  007546  142537  007560           BICB     (R5)+,13$         ;* CLEAR ANY SPECIFIED VIAORB BITS.
3574
3575  007552  004537  003734           JSR      R5,WRITEI         ;SET UP USYRT
```

```
3576  007556  120000                    VIAORB
3577  007560  000142            13$:    TXEN!RXEN!TTLOOP        ;* THIS VALUE MIGHT BE MODIFIED ABOVE
3578
3579  007562  004537  003734            JSR     R5,WRITEI       ;SET TSOM IN USYRT
3580  007566  120403                    TDSRH
3581  007570  000001                    TSOM
3582  007572  004537  003734            JSR     R5,WRITEI       ;LOAD SYNCH CHAR INTO TX BUF
3583  007576  120402                    TDSRL
3584  007600  000226                    SYNCH
3585  007602  004537  006036            JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
3586  007606  000000                    0
3587  007610  103427                    BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3588  007612  005001                    CLR     R1              ;INIT CYCLE COUNTER
3589  007614  004537  011614    4$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3590  007620  000001                    1
3591  007622  004537  003610            JSR     R5,READI        ;READ USYRT STATUS REG
3592  007626  122000                    USTATR
3593  007630  000000            5$:     .WORD   0
3594  007632  132737  000100  007630    BITB    #TBMT,5$        ;SEE IF TBMT IS SET YET
3595  007640  001010                    BNE     6$              ;BR IF YES
3596  007642  005201                    INC     R1              ;INCR CYCLE COUNTER
3597  007644  020127  000003            CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
3598  007650  002761                    BLT     4$              ;BR IF LESS THAN 3 CYCLES
3599  007652  004537  006036            JSR     R5,CKTBMT       ;GO STACK 'TBMT NOT SET'' MSG
3600  007656  000001                    1
3601  007660  103403                    BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3602  007662  004537  005536    6$:     JSR     R5,CKTACT       ;CHK FOR TXACT = 1
3603  007666  000001                    1
3604  007670  012601            7$:     MOV     (SP)+,R1        ;RESTORE R1
3605  007672  000205                    RTS     R5              ;RETURN (IF C = 1, WE HAD AN ERROR)
3606
3607  007674  000142            20$:    TXEN!RXEN!TTLOOP        ;DEFAULT VALUE FOR VIAORB: ENABLE
3608                                                            ;TX AND RX ON USYRT, ASSERT RTS, DTR
3609
```

```
                                .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)>
                                ;*******************************************************************************
007676                          TXCHAR:
007676  010146                          MOV     R1,-(SP)        ;SAVE R1
007700  010246                          MOV     R2,-(SP)        ;SAVE R2
007702  012537  007714                  MOV     (R5)+,1$        ;GET DATA FOR TDSRL
007706  004537  003734                  JSR     R5,WRITEI       ;LOAD DATA INTO TDSRL
007712  120402                          TDSRL
007714  000000                  1$:     .WORD   0
007716  005001                          CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
007720  005002                          CLR     R2              ;CLEAR REQ'D CYCLE COUNT
007722  112502                          MOVB    (R5)+,R2        ;GET DESIRED NO. OF CYCLES
007724  001425                          BEQ     6$              ;BR IF NO CLOCKING DONE
007726  004537  005536          3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
007732  000001                          1
007734  103421                          BCS     6$              ;BR TO EXIT IF ERROR
007736  020102                          CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
007740  001414                          BEQ     5$              ;BR IF YES
                                
007742  131527  000200                  BITB    (R5),#NCTBMT    ;* CHECK FOR 'TBMT=0 CHECK' DISABLE
007746  001004                          BNE     7$              ;* BR IF MSB IS NOT SET
                                
007750  004537  006036                  JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
007754  000000                          0
007756  103410                          BCS     6$              ;BR TO EXIT IF ERROR
007760  004537  011614          7$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
007764  000001                          1
007766  005201                          INC     R1              ;INCR CYCLE COUNT
007770  000756                          BR      3$              ;KEEP CLOCKING
007772  004537  006036          5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
007776  000001                          1
010000  012602                  6$:     MOV     (SP)+,R2        ;RESTORE R2
010002  012601                          MOV     (SP)+,R1        ;RESTORE R1
010004  005205                          INC     R5              ;ADJUST R5 FOR SANE RETURN
010006  000205                          RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

```
                                        .SBTTL ....TXCTRL -- CONTROL MESSAGE TRANSMISSION (TDSRH)
                                        ;*************************************************************************
                                        ;* TXCTRL - THIS SUBROUTINE ALLOWS CONTROL OF MESSAGE TRANSMISSION BY LOADING
                                        ;*       THE USYRT TDSRH WITH THE DATA PASSED IN THE LO BYTE OF THE WORD
                                        ;*       FOLLOWING THE CALL, AND CLOCKS THE LINE UNIT WITH THE NUMBER OF CYCLES
                                        ;*       PASSED IN THE SECOND WORD FOLLOWING THE CALL. THE PROGRAM CONTINUALLY
                                        ;*       MONITORS TBMT AND TXACT THROUGHOUT THE PROCESS.
                                        ;*       IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                        ;*       IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                        ;*       DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                        ;*
                                        ;*       CALLING SEQUENCE :
                                        ;*       JSR     R5,TXCTRL
                                        ;*       .WORD   <DATA FOR TDSRH IN LO BYTE>
                                        ;*       .WORD   <NUMBER OF CYCLES TO CLOCK>
                                        ;*************************************************************************
3680  010010                            TXCTRL:
3681  010010  010146                           MOV     R1,-(SP)        ;SAVE R1
3682  010012  010246                           MOV     R2,-(SP)        ;SAVE R2
3683  010014  012537  010026                   MOV     (R5)+,2$        ;GET DATA FOR TDSRH
3684  010020  004537  003734                   JSR     R5,WRITEI       ;LOAD DATA INTO TDSRH
3685  010024  120403                           TDSRH
3686  010026  000000                   2$:     .WORD   0
3687  010030  005001                           CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
3688  010032  012502                           MOV     (R5)+,R2        ;GET DESIRED NO. OF CYCLES
3689  010034  001422                           BEQ     6$              ;BR IF NO CLOCKING DONE
3690  010036  004537  005536           3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
3691  010042  000001                           1
3692  010044  103416                           BCS     6$              ;BR TO EXIT IF ERROR
3693  010046  020102                           CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3694  010050  001411                           BEQ     5$              ;BR IF YES
3695  010052  004537  006036                   JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
3696  010056  000000                           0
3697  010060  103410                           BCS     6$              ;BR TO EXIT IF ERROR
3698  010062  004537  011614                   JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3699  010066  000001                           1
3700  010070  005201                           INC     R1              ;INCR CYCLE COUNT
3701  010072  000761                           BR      3$              ;KEEP CLOCKING
3702  010074  004537  006036           5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
3703  010100  000001                           1
3704  010102  012602                   6$:     MOV     (SP)+,R2        ;RESTORE R2
3705  010104  012601                           MOV     (SP)+,R1        ;RESTORE R1
3706  010106  000205                           RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
3707
```

```
                                .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)>
                                ;*****************************************************************************
010110                          RXCHAR:
010110  010146                          MOV     R1,-(SP)        ;SAVE R1
010112  010246                          MOV     R2,-(SP)        ;SAVE R2
010114  004537  003610                  JSR     R5,READI        ;READ RDSRH
010120  120401                          RDSRH
010122  000000                  2$:     .WORD   0
010124  004537  003610                  JSR     R5,READI        ;READ RDSRL
010130  120400                          RDSRL
010132  000000                  1$:     .WORD   0
010134  111501                          MOVB    (R5),R1         ;GET EXPECTED RDSRL
010136  042701  177400                  BIC     #177400,R1      ;MASK OFF UNUSED BITS
010142  023727  002456  000347          CMP     SAVLEN,#TXDL!RXDL ;SEE IF 7-BIT CHARS BEING USED
010150  001005                          BNE     3$              ;BR IF NOT 7-BIT CHARS
010152  142737  000200  010132          BICB    #BIT7,1$        ;CLEAR 8TH BIT FOR COMPARE
010160  142701  000200                  BICB    #BIT7,R1
010164  123701  010132          3$:     CMPB    1$,R1           ;COMPARE RCV'D CHAR TO EXPECTED
010170  001462                          BEQ     6$              ;BR IF MATCH
010172  004537  003610                  JSR     R5,READI        ;READ USYRT STATUS REG
010176  122000                          USTATR
010200  000000                  4$:     .WORD   0
010202  132737  000002  010200          BITB    #TXU,4$         ;SEE IF TX UNDERRUN OCCURRED
010210  001421                          BEQ     5$              ;BR IF NOT
010212  012737  000007  002412          MOV     #7,REGNUM       ;SET USYRT REG NO. FOR STATUS REG
                                ;STACK 'TX UNDERRUN' ERROR
010220                                  GTDF    EM54,ERR12
                                ;               QUEUE 'DEVICE FATAL'' ERROR # 27
010220  012737  000001  002246                                          MOV     #T.EDF,ERRTYP
010226  012737  000033  002250                                          MOV     #27,ERRNBR
010234  012737  013717  002252                                          MOV     #EM54,ERRMSG
010242  012737  017722  002254                                          MOV     #ERR12,ERRBLK
010250  000137  011350                  JMP     20$             ;TAKE ERROR EXIT
010254  005037  002412          5$:     CLR     REGNUM          ;SET USYRT REG NO. FOR RDSRL
010260  005037  002400                  CLR     GDATA           ;SET EXPECTED DATA
010264  110137  002400                  MOVB    R1,GDATA
010270  005037  002402                  CLR     BDATA           ;SET ACTUAL DATA
010274  113737  010132  002402          MOVB    1$,BDATA
                                ;STACK 'RCV'D DATA MISCOMPARE'' ERROR
```

```
                        GTDF    EM34,ERR10
                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 28
                                                        MOV     #T.EDF,ERRTYP
                                                        MOV     #28,ERRNBR
                                                        MOV     #EM34,ERRMSG
                                                        MOV     #ERR10,ERRBLK
                        JMP     20$             ;TAKE ERROR EXIT
6$:     MOVB    1(R5),R1        ;GET RDSRH
        BIC     #177400,R1      ;MASK OFF UNUSED BITS
        CMPB    2$,R1           ;COMPARE RCV'D STATUS TO EXPECTED
        BNE     7$              ;BR IF MISMATCH
        JMP     17$             ;CONTINUE
        MOV     #1,REGNUM       ;SET USYRT REG NO. FOR RDSRH
        CLR     GDATA           ;SET EXPECTED DATA
        MOVB    R1,GDATA
        CLR     BDATA           ;SET ACTUAL DATA
        MOVB    2$,BDATA
7$:     MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
        BIT     #RERCHK,2(R5)   ;SEE IF RCV ERROR BIT SHOULD BE IGNORED
        BEQ     9$              ;BR IF YES
;CHECK RERR BIT
        BITB    #RERR,R1        ;SEE IF EXPECTED BIT = 1
        BNE     8$              ;BR IF YES
        BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 0
        BEQ     9$              ;BR IF YES
;STACK 'RERR NOT CLEARED'' MSG
        GTDF    EM35,ERR12
                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 29
                                                        MOV     #T.EDF,ERRTYP
                                                        MOV     #29,ERRNBR
                                                        MOV     #EM35,ERRMSG
                                                        MOV     #ERR12,ERRBLK
                        JMP     20$             ;TAKE ERROR EXIT
8$:     BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 1
        BNE     9$              ;BR IF YES
;STACK 'RERR NOT SET'' MSG
        GTDF    EM36,ERR12
                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 30
                                                        MOV     #T.EDF,ERRTYP
                                                        MOV     #30,ERRNBR
                                                        MOV     #EM36,ERRMSG
                                                        MOV     #ERR12,ERRBLK
                        JMP     20$             ;TAKE ERROR EXIT
;CHECK ROR BIT
9$:     BITB    #ROR,R1         ;SEE IF EXPECTED BIT = 1
        BNE     10$             ;BR IF YES
        BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 0
        BEQ     11$             ;BR IF YES
;STACK 'ROR NOT CLEARED'' MSG
        GTDF    EM16,ERR12
                                                ;       QUEUE 'DEVICE FATAL'' ERROR # 31
                                                        MOV     #T.EDF,ERRTYP
                                                        MOV     #31,ERRNBR
                                                        MOV     #EM16,ERRMSG
                                                        MOV     #ERR12,ERRBLK
                        JMP     20$             ;TAKE ERROR EXIT
```

```
3820  010616  132737  000010  010122  10$:    BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 1
3821  010624  001016                           BNE     11$             ;BR IF YES
3822                                   ;STACK 'ROR NOT SET' MSG
3823  010626                                   GTDF    EM14,ERR12
3824                                                                   ;       QUEUE 'DEVICE FATAL' ERROR # 32
3825  010626  012737  000001  002246                                                   MOV     #T.EDF,ERRTYP
3826  010634  012737  000040  002250                                                   MOV     #32,ERRNBR
3827  010642  012737  013332  002252                                                   MOV     #EM14,ERRMSG
3828  010650  012737  017722  002254                                                   MOV     #ERR12,ERRBLK
3829  010656  000137  011350                   JMP     20$             ;TAKE ERROR EXIT
3830                                   ;CHECK RABGA BIT
3831  010662  132701  000004           11$:    BITB    #RABGA,R1       ;SEE IF EXPECTED BIT = 1
3832  010666  001022                           BNE     12$             ;BR IF YES
3833  010670  132737  000004  010122           BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 0
3834  010676  001440                           BEQ     13$             ;BR IF YES
3835                                   ;STACK 'RABGA NOT CLEARED' MSG
3836  010700                                   GTDF    EM39,ERR12
3837                                                                   ;       QUEUE 'DEVICE FATAL' ERROR # 33
3838  010700  012737  000001  002246                                                   MOV     #T.EDF,ERRTYP
3839  010706  012737  000041  002250                                                   MOV     #33,ERRNBR
3840  010714  012737  013657  002252                                                   MOV     #EM39,ERRMSG
3841  010722  012737  017722  002254                                                   MOV     #ERR12,ERRBLK
3842  010730  000137  011350                   JMP     20$             ;TAKE ERROR EXIT
3843  010734  132737  000004  010122  12$:    BITB    #RABGA,2$       ;SEE IF ACTUAL BIT = 1
3844  010742  001016                           BNE     13$             ;BR IF YES
3845                                   ;STACK 'RABGA NOT SET' MSG
3846  010744                                   GTDF    EM40,ERR12
3847                                                                   ;       QUEUE 'DEVICE FATAL' ERROR # 34
3848  010744  012737  000001  002246                                                   MOV     #T.EDF,ERRTYP
3849  010752  012737  000042  002250                                                   MOV     #34,ERRNBR
3850  010760  012737  013701  002252                                                   MOV     #EM40,ERRMSG
3851  010766  012737  017722  002254                                                   MOV     #ERR12,ERRBLK
3852  010774  000137  011350                   JMP     20$             ;TAKE ERROR EXIT
3853                                   ;CHECK REOM BIT
3854  011000  132701  000002           13$:    BITB    #REOM,R1        ;SEE IF EXPECTED BIT = 1
3855  011004  001022                           BNE     14$             ;BR IF YES
3856  011006  132737  000002  010122           BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 0
3857  011014  001440                           BEQ     15$             ;BR IF YES
3858                                   ;STACK 'REOM NOT CLEARED' MSG
3859  011016                                   GTDF    EM30,ERR12
3860                                                                   ;       QUEUE 'DEVICE FATAL' ERROR # 35
3861  011016  012737  000001  002246                                                   MOV     #T.EDF,ERRTYP
3862  011024  012737  000043  002250                                                   MOV     #35,ERRNBR
3863  011032  012737  013535  002252                                                   MOV     #EM30,ERRMSG
3864  011040  012737  017722  002254                                                   MOV     #ERR12,ERRBLK
3865  011046  000137  011350                   JMP     20$             ;TAKE ERROR EXIT
3866  011052  132737  000002  010122  14$:    BITB    #REOM,2$        ;SEE IF ACTUAL BIT = 1
3867  011060  001016                           BNE     15$             ;BR IF YES
3868                                   ;STACK 'REOM NOT SET' MSG
3869  011062                                   GTDF    EM31,ERR12
3870                                                                   ;       QUEUE 'DEVICE FATAL' ERROR # 36
3871  011062  012737  000001  002246                                                   MOV     #T.EDF,ERRTYP
3872  011070  012737  000044  002250                                                   MOV     #36,ERRNBR
3873  011076  012737  013556  002252                                                   MOV     #EM31,ERRMSG
3874  011104  012737  017722  002254                                                   MOV     #ERR12,ERRBLK
3875  011112  000137  011350                   JMP     20$             ;TAKE ERROR EXIT
```

```
3876                                    ;CHECK RSOM BIT
3877  011116  132701  000001            15$:    BITB    #RSOM,R1        ;SEE IF EXPECTED BIT = 1
3878  011122  001022                            BNE     16$             ;BR IF YES
3879  011124  132737  000001  010122            BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 0
3880  011132  001440                            BEQ     17$             ;BR IF YES
3881                                    ;STACK 'RSOM NOT CLEARED'' MSG
3882  011134                                    GTDF    EM28,ERR12
3883                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 37
3884  011134  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
3885  011142  012737  000045  002250                                                    MOV     #37,ERRNBR
3886  011150  012737  013477  002252                                                    MOV     #EM28,ERRMSG
3887  011156  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
3888  011164  000137  011350                    JMP     20$             ;TAKE ERROR EXIT
3889  011170  132737  000001  010122  16$:     BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 1
3890  011176  001016                            BNE     17$             ;BR IF YES
3891                                    ;STACK 'RSOM NOT SET'' MSG
3892  011200                                    GTDF    EM29,ERR12
3893                                                    ;       QUEUE 'DEVICE FATAL'' ERROR # 38
3894  011200  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
3895  011206  012737  000046  002250                                                    MOV     #38,ERRNBR
3896  011214  012737  013520  002252                                                    MOV     #EM29,ERRMSG
3897  011222  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
3898  011230  000137  011350                    JMP     20$             ;TAKE ERROR EXIT
3899
3900  011234  116502  000004            17$:    MOVB    4(R5),R2        ;GET DESIRED NO. OF CYCLES
3901  011240  005001                            CLR     R1              ;INIT CYCLE COUNT
3902
3903  011242  136527  000005  000040    18$:    BITB    5(R5),#BIT5     ;* IS RXACT CHECK TO BE DISABLED ?
3904  011250  001004                            BNE     31$             ;* BR IF YES
3905  011252  004537  005676                    JSR     R5,CKRACT       ;CHK FOR RACT = 1
3906  011256  000001                            1
3907  011260  103433                            BCS     20$             ;BR TO EXIT IF ERROR
3908
3909  011262  020102                    31$:    CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3910  011264  001415                            BEQ     19$             ;BR IF YES
3911
3912  011266  136527  000005  000200            BITB    5(R5),#BIT7     ;* SEE IF INITIAL RDA CHECK DESIRED
3913  011274  001004                            BNE     22$             ;* BR IF NO
3914  011276  004537  006176                    JSR     R5,CKRDA        ;CHK FOR RDA = 0
3915  011302  000000                            0
3916  011304  103421                            BCS     20$             ;BR TO EXIT IF ERROR
3917
3918  011306  004537  011614            22$:    JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3919  011312  000001                            1
3920  011314  005201                            INC     R1              ;INCR CYCLE COUNT
3921  011316  000751                            BR      18$             ;CONTINUE CLOCKING
3922
3923  011320  136527  000005  000100    19$:    BITB    5(R5),#BIT6     ;* IS FINAL RDA CHECK TO BE SKIPPED ?
3924  011326  001004                            BNE     30$             ;* BR IF YES
3925  011330  004537  006176                    JSR     R5,CKRDA        ;CHK RDA = 1
3926  011334  000001                            1
3927  011336  103404                            BCS     20$             ;BR IF ERROR
3928
3929  011340  062705  000006            30$:    ADD     #6,R5           ;FIX UP RETURN ADRS
3930  011344  000241                            CLC                     ;SET C = 0 FOR NO ERROR
3931  011346  000403                            BR      21$             ;TAKE ERROR-FREE EXIT
```

```
 3932  011350  062705  000006      20$:    ADD     #6,R5           ;FIX UP RETURN ADDRESS
 3933  011354  000261                      SEC                     ;SET C BIT FOR ERROR
 3934  011356  012602              21$:    MOV     (SP)+,R2        ;RESTORE R2
 3935  011360  012601                      MOV     (SP)+,R1        ;RESTORE R1
 3936  011362  000205                      RTS     R5              ;RETURN
```

```
                                .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>
                                ;*********************************************************************
011364                          RCV1ST:
011364  010146                          MOV     R1,-(SP)        ;SAVE R1
011366  010246                          MOV     R2,-(SP)        ;SAVE R2
011370  005001                          CLR     R1              ;INIT CYCLE COUNT
011372  012502                          MOV     (R5)+,R2                ;GET CYCLE COUNT LIMIT
011374  062702  000003                  ADD     #3,R2
011400  004537  005676                  JSR     R5,CKRACT       ;CHK FOR RXACT = 0
011404  000000                          0
011406  103446                          BCS     6$              ;BR TO EXIT IF ERROR
011410  004537  006176                  JSR     R5,CKRDA        ;CHK FOR RDA = 0
011414  000000                          0
011416  103442                          BCS     6$              ;BR TO EXIT IF ERROR
011420  004537  006636                  JSR     R5,CKSEOM       ;CHK FOR RSOM = 0, REOM = 0
011424  000000                          0
011426  103436                          BCS     6$              ;BR TO EXIT IF ERROR
011430  004537  011614          1$:     JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
011434  000001                          1
011436  005201                          INC     R1              ;INCREMENT CYCLE COUNT
011440  004537  003610                  JSR     R5,READI        ;READ USYRT STATUS REG
011444  122000                          USTATR
011446  000000                  2$:     .WORD   0
011450  132737  000200  011446          BITB    #RDA,2$         ;SEE IF RDA SET YET
011456  001006                          BNE     3$              ;BR IF YES
011460  020102                          CMP     R1,R2           ;SEE IF LIMIT EXCEEDED
011462  002762                          BLT     1$              ;BR IF NOT YET
011464  004537  006176                  JSR     R5,CKRDA        ;GO STACK 'RDA NOT SET' MSG
011470  000001                          1
011472  103414                          BCS     6$              ;BR TO EXIT IF ERROR
011474  020165  177776          3$:     CMP     R1,-2(R5)       ;SEE IF LESS THAN REQUIRED CYCLES
011500  002004                          BGE     4$              ;BR IF NOT
011502  004537  006176                  JSR     R5,CKRDA        ;GO STACK 'RDA NOT CLEARED' MSG
011506  000000                          0
011510  103405                          BCS     6$              ;BR TO EXIT IF ERROR
011512  004537  005676          4$:     JSR     R5,CKRACT       ;CHK FOR RXACT = 1
011516  000001                          1
011520  103401                          BCS     6$              ;BR TO EXIT IF ERROR
011522  000241                  5$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
011524  012602                  6$:     MOV     (SP)+,R2        ;RESTORE R2
011526  012601                          MOV     (SP)+,R1        ;RESTORE R1
011530  000205                          RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

CVDMCA.P11 10-DEC-80 09:14 ....ENDTRN -- SHUT DOWN TRANSMITTER/RECEIVER

```
3992
3993
3994
3995
3996
3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007  011532
4008  011532  012537  011572
4009  011536  004537  005536
4010  011542  000001
4011  011544  103422
4012  011546  004537  005676
4013  011552  000001
4014  011554  103416
4015  011556  004537  003734
4016  011562  120000
4017  011564  000002
4018  011566  004537  011614
4019  011572  000000
4020  011574  004537  005536
4021  011600  000000
4022  011602  103403
4023  011604  004537  005676
4024  011610  000000
4025  011612  000205
4026
4027
```

```
.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 MODE SHOULD BE ENABLED.
;*
;*      CALLING SEQUENCE :
;*        JSR   R5,ENDTRN
;*        <NO. OF CYCLES TO CLOCK>
;*********************************************************************
ENDTRN:
        MOV     (R5)+,2$        ;GET DESIRED NO. OF CYCLES TO CLOCK
        JSR     R5,CKTACT       ;CHK FOR TXACT = 1
        1
        BCS     6$              ;BR IF ERROR
        JSR     R5,CKRACT       ;CHK FOR RXACT = 1
        1
        BCS     6$
        JSR     R5,WRITEI       ;CLEAR TXEN AND RXEN IN USYRT
        VIAORB                  ;** AND KEEP TTLOOP ENABLED **
        TTLOOP                  ;
        JSR     R5,STEPLU       ;CLOCK LU FOR DESIRED NO. OF CYCLES
2$:     .WORD   0
        JSR     R5,CKTACT       ;CHK FOR TXACT = 0
        0
        BCS     6$              ;BR IF ERROR
        JSR     R5,CKRACT       ;CHK FOR RXACT = 0
        0
6$:     RTS     R5
```

```
                                .SBTTL ....STEPLU -- CLOCK THE USYRT N TIMES
                                ;*********************************************************************
                                ;* STEPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NUMBER OF CYCLES
                                ;*      PASSED IN THE WORD FOLLOWING THE CALL. THE VIA ACR MUST BE PREVIOUSLY
                                ;*      SET UP FOR T1 ONE-SHOT MODE, AND THE T1 LATCHES MUST BE PREVIOUSLY SET
                                ;*      TO CONTROL THE WIDTH OF THE CLOCK PULSE. ALL THAT THIS SUBROUTINE
                                ;*      DOES IS TO LOAD 000 INTO THE HI BYTE OF THE T1 COUNTER, FOR THE
                                ;*      DESIRED NUMBER OF TIMES.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,STEPLU
                                ;*      .WORD   <NUMBER OF CYCLES TO CLOCK>
                                ;*********************************************************************
011614                          STEPLU:
011614  010146                          MOV     R1,-(SP)        ;SAVE R1
011616  012501                          MOV     (R5)+,R1        ;INIT CYCLE COUNTER
011620  004537  003734          1$:     JSR     R5,WRITEI       ;LOAD T1C-H, START COUNTER, CLOCK 1 CYCLE
011624  120005                          VIAT1B
011626  000000                          000
011630  005301                          DEC     R1              ;DECR CYCLE COUNTER
011632  001372                          BNE     1$              ;BR IF ALL CYCLES NOT DONE YET
011634  012601                          MOV     (SP)+,R1        ;RESTORE R1
011636  000205                          RTS     R5              ;RETURN
```

CVDMCA.P11 10-DEC-80 09:14 GLOBAL ERROR REPORT SECTION

```
4057                            .SBTTL  GLOBAL ERROR REPORT SECTION
4058
4059                            ;////////////////////////////////////////////////////////////////
4060                            ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
4061                            ;/      THAT ARE USED IN MORE THAN ONE TEST.
4062                            ;////////////////////////////////////////////////////////////////
4063
4064                            .NLIST  BEX
011640 047045 047045 000        ENDEMB: .ASCIZ  /%N%N/
011645 045 000116               NEWLIN: .ASCIZ  /%N/            ;USED TO TERMINATE ERROR MESSAGES

011650 047045 040445 040506     FMT2:   .ASCIZ  /%N%AFAILING REG = %T%ASEL%O1/
011705 045 022516 020101        FMT3:   .ASCIZ  /%N%A  EXPECTED:  %O3%A   ACTUAL:  %O3%A   XOR:  %O3/
011771 045 022516 052101        FMT4:   .ASCIZ  /%N%ATHE CONTENTS OF ALL%T%N%T/
012027 045 022516 030523        FMT4A:  .ASCIZ  /%N%S1%O3%S5%O3%S5%O3%S5%O3/
012062 047045 052045 000        FMT4B:  .ASCIZ  /%N%T/
012067 045 022516 032523        FMT4C:  .ASCIZ  /%N%S5%O3%S5%O3%S5%O3%S5%O3/
012122 047045 040445 020040     FMT5:   .ASCIZ  /%N%A  WHEN %O3%A LOADED INTO BSEL1/
012165 045 022516 020101        FMT5A:  .ASCIZ  /%N%A  ATTEMPTING 'M-LOOP' FUNCTION CODE %O2%A (%T%A)/

012252 047045 040445 020040     FMT10:  .ASCIZ  /%N%A  EXPECTED:%O8%A  ACTUAL:%O8%A  XOR:%O8/
012326 040445 020040 051514     FMT10A: .ASCIZ  /%A  LSI ADDR:%O8/
012347 045 022516 034117        FMT11:  .ASCIZ  /%N%O8%O8%O8%O8/
012366 040445 020040 042504     FMT15:  .ASCIZ  /%A  DETECTED IN %T%T%A --/
012420 040445 020040 042504     FMT15A: .ASCIZ  /%A  DETECTED @ TEST PATTERN ELEMENT # %D2/

012472 047045 040445 042524     FMT19:  .ASCIZ  /%N%ATEST %D2%A NOT RUN%N/
012523 045 022524 033117        FMT21:  .ASCIZ  /%T%O6%N/
012533 045 022516 043101        FMT22:  .ASCIZ  /%N%AFAILING REG: /
012555 045 042501 050130        FMT23:  .ASCIZ  /%AEXPECTED: %O3%S5%AACTUAL: %O3%S5%AXOR: %O3%N/
012634 047045 052045 047045     FMT24:  .ASCIZ  /%N%T%N%T%N/
012647 045 031517 051445        FMT25:  .ASCIZ  /%O3%S5%O3%S5%O3%S5%O3%N/
012677 045 032123 047445        FMT26:  .ASCIZ  /%S4%O3%S5%O3%S5%O3%S5%O3%N/
012732 052045 052045 047045     FMT27:  .ASCIZ  /%T%T%N/
012741 045 022524 000116        FMT29:  .ASCIZ  /%T%N/
012746 047045 040445 047506     FMT30:  .ASCIZ  /%N%AFOR BAUD RATE SPECIFIED./
013003 045 022516 044501        FMT31:  .ASCIZ  /%N%AIMPROPER CONNECTOR TYPE SPECIFIED/
013051 045 022516 043101        FMT32:  .ASCIZ  /%N%AFOR OPTION SPECIFIED./

013103 122 043505 047040        EM1:    .ASCIZ  /REG NOT INITIALIZED BY MST CLR/
013142 051525 051131 020124     EM2:    .ASCIZ  /USYRT NOT INITIALIZED BY PROGRAM RESET/
013211 115 041511 047522        EM3:    .ASCIZ  /MICRO-DIAG. FAILURE/
013235 115 042122 020131        EM4:    .ASCIZ  /MRDY TIMEOUT/
013252 052516 046114 041440     EM5:    .ASCIZ  /NULL CLK BIT STUCK AT 0/
013302 052516 046114 041440     EM6:    .ASCIZ  /NULL CLK BIT STUCK AT 1/
013332 047522 020122 047516     EM14:   .ASCIZ  /ROR NOT SET/
013346 047522 020122 047516     EM16:   .ASCIZ  /ROR NOT CLEARED/
013366 042522 042101 053457     EM25:   .ASCIZ  'READ/WRITE DATA ERROR'
013414 047111 047503 051122     EM26:   .ASCIZ  /INCORRECT DATA CHAR RCV'D/
013446 047111 047503 051122     EM27:   .ASCIZ  /INCORRECT CRC BYTE RCV'D/
013477 122 047523 020115        EM28:   .ASCIZ  /RSOM NOT CLEARED/
013520 051522 046517 047040     EM29:   .ASCIZ  /RSOM NOT SET/
013535 122 047505 020115        EM30:   .ASCIZ  /REOM NOT CLEARED/
013556 042522 046517 047040     EM31:   .ASCIZ  /REOM NOT SET/
013573 122 053103 042047        EM34:   .ASCIZ  /RCV'D DATA MISCOMPARE/
013621 122 051105 020122        EM35:   .ASCIZ  /RERR NOT CLEARED/
```

```
013642 042522 051122 047040 EM36:   .ASCIZ  /RERR NOT SET/
013657    122 041101 040507 EM39:   .ASCIZ  /RABGA NOT CLEARED/
013701    122 041101 040507 EM40:   .ASCIZ  /RABGA NOT SET/
013717    124 020130 047125 EM54:   .ASCIZ  /TX UNDERRUN ERROR/
013741    122 043505 046440 EM66:   .ASCIZ  /REG MISCOMPARE/
013760 051525 051131 020124 EM68:   .ASCIZ  /USYRT STATUS INCORRECT/
014007    124 040530 052103 EM69:   .ASCIZ  /TXACT NOT SET/
014025    124 040530 052103 EM70:   .ASCIZ  /TXACT NOT CLEARED/
014047    122 040530 052103 EM71:   .ASCIZ  /RXACT NOT SET/
014065    122 040530 052103 EM72:   .ASCIZ  /RXACT NOT CLEARED/
014107    124 046502 020124 EM73:   .ASCIZ  /TBMT NOT SET/
014124 041124 052115 047040 EM74:   .ASCIZ  /TBMT NOT CLEARED/
014145    122 040504 047040 EM75:   .ASCIZ  /RDA NOT SET/
014161    122 040504 047040 EM76:   .ASCIZ  /RDA NOT CLEARED/
014201    122 040523 047040 EM77:   .ASCIZ  /RSA NOT SET/
014215    122 040523 047040 EM78:   .ASCIZ  /RSA NOT CLEARED/
014235    122 041505 044505 EM90:   .ASCIZ  /RECEIVER OVERRUN NOT SET/
014266 042522 042503 053111 EM91:   .ASCIZ  /RECEIVER OVERRUN NOT CLEARED/
014323    124 046502 020124 EM92:   .ASCIZ  /TBMT INTERRUPT TEST FAILURE/
014357    124 051105 020122 EM98:   .ASCIZ  /TERR BIT NOT CLEARED/
014404 042524 051122 041040 EM99:   .ASCIZ  /TERR BIT NOT SET/
014425    124 047523 041040 EM100:  .ASCIZ  /TSO BIT NOT SET/
014445    124 047523 041040 EM101:  .ASCIZ  /TSO BIT NOT CLEARED/
014471    124 040522 051516 EM102:  .ASCIZ  /TRANSMISSION ERROR (AS READ BY TSO BIT)/


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

014541    102 042523 030114 TXT1:   .ASCIZ  /BSEL0   BSEL1   BSEL2   BSEL3/
014577    040 020040 041040 TXT2:   .ASCIZ  /    BSEL4   BSEL5   BSEL6   BSEL7/
014641    102 042523 030514 TXT2A:  .ASCIZ  /BSEL10  BSEL11  BSEL12  BSEL13/
014700 020040 020040 051502 TXT2B:  .ASCIZ  /    BSEL14  BSEL15  BSEL16  BSEL17/
014743    040 054502 042524 TXT3:   .ASCIZ  / BYTE SELECT REG'S ARE:/
014773    040 020040 042523 TXT4:   .ASCIZ  /   SEL0    SEL2    SEL4    SEL6/
015033    040 020040 042523 TXT4A:  .ASCIZ  /   SEL10   SEL12   SEL14   SEL16/
015074 000102               TXT5:   .ASCIZ  /B/
015076 051440 046105 041505 TXT6:   .ASCIZ  / SELECT REG'S ARE:/
015121    040 042522 044507 TXT7:   .ASCIZ  / REGISTERS     ORB  ORA  DDRB DDRA T1CL T1CH T1LL T1LH /
015211    040 020040 020040 TXT7A:  .ASCIZ  /                T2CL T2CH SR   ACR  PCR  IFR  IER  ORA  /
015301    040 054105 042520 TXT8:   .ASCIZ  / EXPECTED:     /
015321    040 041501 052524 TXT9:   .ASCIZ  / ACTUAL:       /
015341    040 047530 035122 TXT10:  .ASCIZ  / XOR:          /
015361    040 047040 020040 TXT11:  .ASCIZ  /  N  P  R      R  E  G  I  S  T  E  R  S:/
015433    040 020040 020040 TXT11A: .ASCIZ  /              CONTROL  D A T A/
015471    040 020040 020040 TXT11B: .ASCIZ  /                 OUT ADDR.      IN ADDR./
015541    104 053105 041511 TXT12:  .ASCIZ  /DEVICE CSR ADDRESS : /
015567    125 054523 052122 TXT13:  .ASCIZ  /USYRT REGS :/
015604 042122 051123 020114 TXT14:  .ASCIZ  /RDSRL   RDSRH   TDSRL   TDSRH/
015642 020040 020040 041520 TXT15:  .ASCIZ  /    PCSARL  PCSARH  PCR     USTAT/
015704 044526 020101 042522 TXT16:  .ASCIZ  /VIA REGS :/
015717    117 041122 020040 TXT17:  .ASCIZ  /ORB     ORA     DDRB    DDRA/
015754 020040 020040 030524 TXT18:  .ASCIZ  /    T1CL    T1CH    T1LL    T1LH/
```

```
016015     124  041462  020114  TXT19:  .ASCIZ  /T2CL    T2CH    SR      ACR/
016051     040  020040  050040  TXT20:  .ASCIZ  /    PCR     IFR     IER     ORA/

016111     021     000          TXTNUL: .BYTE   21,0                    ;CTL-Q -- THIS (WE HOPE) IS HARMLESS

016113     116  050117     000  TXTML0: .ASCIZ  /NOP/
016117     122  040505  020104  TXTML1: .ASCIZ  /READ 1 BYTE/
016133     127  044522  042524  TXTML2: .ASCIZ  /WRITE 1 BYTE/
016150  050116  026522  052517  TXTML3: .ASCIZ  /NPR-OUT 256 BYTES/
016172  050116  026522  047111  TXTML4: .ASCIZ  /NPR-IN 256 BYTES/
016213     123  052105  046440  TXTML5: .ASCIZ  /SET MICROPROCESSOR'S PC/
016243     125  042116  043105  TXTML6: .ASCIZ  /UNDEFINED/
016255     101  046114  053517  TXTML7: .ASCIZ  /ALLOW U-PROCESSOR INTERRUPTS/

016312  044526  020101  042522  TXTVR:  .ASCIZ  /VIA REGISTER /
016330  051117  000102          TXTVR0: .ASCIZ  /ORB/
016334  051117  000101          TXTVR1: .ASCIZ  /ORA/
016340  042104  041122     000  TXTVR2: .ASCIZ  /DDRB/
016345     104  051104  000101  TXTVR3: .ASCIZ  /DDRA/
016352  030524  046103     000  TXTVR4: .ASCIZ  /T1CL/
016357     124  041461  000110  TXTVR5: .ASCIZ  /T1CH/
016364  030524  046114     000  TXTVR6: .ASCIZ  /T1LL/
016371     124  046061  000110  TXTVR7: .ASCIZ  /T1LH/
016376  031124  046103     000  TXTVR8: .ASCIZ  /T2CL/
016403     124  041462  000110  TXTVR9: .ASCIZ  /T2CH/
016410  051123     000          TXTVRA: .ASCIZ  /SR/
016413     101  051103     000  TXTVRB: .ASCIZ  /ACR/
016417     120  051103     000  TXTVRC: .ASCIZ  /PCR/
016423     111  051106     000  TXTVRD: .ASCIZ  /IFR/
016427     111  051105     000  TXTVRE: .ASCIZ  /IER/
016433     117  040522     000  TXTVRF: .ASCIZ  /ORA/

016437     116  051120  000040  TXTNP:  .ASCIZ  /NPR /
016444  047503  052116  047522  TXTNP0: .ASCIZ  /CONTROL/
016454  040504  040524  044040  TXTNP1: .ASCIZ  /DATA HI/
016464  040504  040524  046040  TXTNP2: .ASCIZ  /DATA LO/
016474  042101  051104  020056  TXTNP3: .ASCIZ  /ADDR. OUT EX/
016511     101  042104  027122  TXTNP4: .ASCIZ  /ADDR. OUT HI/
016526  042101  051104  020056  TXTNP5: .ASCIZ  /ADDR. OUT LO/
016543     101  042104  027122  TXTNP6: .ASCIZ  /ADDR. IN EX/
016557     101  042104  027122  TXTNP7: .ASCIZ  /ADDR. IN HI/
016573     101  042104  027122  TXTNP8: .ASCIZ  /ADDR. IN LO/

016607     125  054523  052122  TXTUR:  .ASCIZ  /USYRT REG /
016622  042122  051123  000114  TXTUR0: .ASCIZ  /RDSRL/
016630  042122  051123  000110  TXTUR1: .ASCIZ  /RDSRH/
016636  042124  051123  000114  TXTUR2: .ASCIZ  /TDSRL/
016644  042124  051123  000110  TXTUR3: .ASCIZ  /TDSRH/
016652  041520  040523  046122  TXTUR4: .ASCIZ  /PCSARL/
016661     120  051503  051101  TXTUR5: .ASCIZ  /PCSARH/
016670  041520  000122          TXTUR6: .ASCIZ  /PCR/
016674  051525  040524  000124  TXTUR7: .ASCIZ  /USTAT/
                                .LIST   BEX
4065                            .EVEN
4066
4067
```

```
4068                                    .SBTTL ....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- 'TXT__T'
4069                                    ;------------------------------------------------------------
4070                                    ;---------- TEXT ADDRESS TABLES USED BY ERROR HANDLERS -------
4071                                    ;------------------------------------------------------------
4072
4073  016702  016113  016117  016133  TXTMLT: .WORD   TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7
4074  016710  016150  016172  016213
4075  016716  016243  016255
4076
4077  016722  016312                          .WORD   TXTVR
4078  016724  016330  016334  016340  TXTVRT: .WORD   TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7
4079  016732  016345  016352  016357
4080  016740  016364  016371
4081  016744  016376  016403  016410          .WORD   TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF
4082  016752  016413  016417  016423
4083  016760  016427  016433
4084
4085  016764  016437                          .WORD   TXTNP
4086  016766  016444  016454  016464  TXTNPT: .WORD   TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8
4087  016774  016474  016511  016526
4088  017002  016543  016557  016573
4089  017010  016622  016630  016636  TXTURT: .WORD   TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7
4090  017016  016644  016652  016661
4091  017024  016670  016674
4092
4093
```

```
4094
4095
4096
4097  017030                        ;-----------------------------------------------------------------------
4098  017030                        .SBTTL ....ERROR HANDLER -- ERR4 -- M-LOOP TIMEOUT ERROR HANDLING
4099  017030  105037  002401        ;-----------------------------------------------------------------------
4100  017034  010146                        BGNMSG  ERR4
4101  017036  013701  002400                                                          ERR4::
4102  017042  022701  000017        CLRB    GDATA+1                 ;MAKE SURE BIT 8 DOESN'T PRINT!
4103  017046  002012                MOV     R1,-(SP)                ;SAVE THE WORKING REGISTER
4104  017050                        MOV     GDATA,R1                ;SAVE THIS FOR LATER
4105  017050  010146                CMP     #17,R1                  ;WAS THIS AN M-LOOP REQUEST?
4106  017052  012746  012122        BGE     5$                      ;YES, THEN REPORT THE FUNCTION CODE
4107  017056  012746  000002        PRINTX  #FMT5,R1                ;NO,  THEN IT MUST BE A BSEL1 SETTING
4108  017062  010600                                                                  MOV     R1,-(SP)
4109  017064  104415                                                                  MOV     #FMT5,-(SP)
4110  017066  062706  000006                                                          MOV     #2,-(SP)
4111  017072  000424                                                                  MOV     SP,R0
4112                                                                                  TRAP    C$PNTX
4113  017074  001001                                                                  ADD     #6,SP
4114  017076  005001        5$:     BR      20$
4115  017100  022701  000007
4116  017104  002002        6$:     BNE     6$                      ;IF IT WAS A 17, THIS IS A 'NOP' AND
4117  017106  012701  000006        CLR     R1                      ;  THE TEXT POINTER MUST SO REFLECT.
4118  017112  006301        7$:     CMP     #7,R1                   ;IS FUNCTION CODE > 7?
4119  017114                        BGE     7$                      ;NO,  THAN WE CAN HANDLE IT
4120  017114  016146  016702        MOV     #6,R1                   ;YES, THAN IT'S UNDEFINED -- SAY SO
4121  017120  013746  002400        ASL     R1                      ;CONVERT TO A WORD OFFSET
4122  017124  012746  012165        PRINTX  #FMT5A,GDATA,TXTMLT(R1) ;REPORT THE FAILING FUNCTION
4123  017130  012746  000003                                                          MOV     TXTMLT(R1),-(SP)
4124  017134  010600                                                                  MOV     GDATA,-(SP)
4125  017136  104415                                                                  MOV     #FMT5A,-(SP)
4126  017140  062706  000010                                                          MOV     #3,-(SP)
4127                                                                                  MOV     SP,R0
4128  017144  012601        20$:    MOV     (SP)+,R1                ;RESTORE THE WORKING REGISTER
4129  017146  004737  020160        JSR     PC,ERR5$                ;DUMP THE SELECT REGISTERS
4130  017152                        ENDMSG
4131  017152                                                                  L10002:
4132  017152  104423                                                                  TRAP    C$MSG
4133                                ;-----------------------------------------------------------------------
4134                                .SBTTL ....ERROR HANDLER -- ERR5 -- WORD SELECT REG. ERRORS
4135                                ;-----------------------------------------------------------------------
4136  017154                                BGNMSG  ERR5
4137  017154                                                                  ERR5::
4138  017154                                PRINTB  #FMT2,#TXT5,REGNUM
4139  017154  013746  002412                                                          MOV     REGNUM,-(SP)
4140  017160  012746  015074                                                          MOV     #TXT5,-(SP)
4141  017164  012746  011650                                                          MOV     #FMT2,-(SP)
4142  017170  012746  000003                                                          MOV     #3,-(SP)
4143  017174  010600                                                                  MOV     SP,R0
4144  017176  104414                                                                  TRAP    C$PNTB
4145  017200  062706  000010                                                          ADD     #10,SP
4146  017204  004737  020134        JSR     PC,XORGB
4147  017210                        PRINTB  #FMT10,GDATA,BDATA,XDATA
4148  017210  013746  002404                                                          MOV     XDATA,-(SP)
4149  017214  013746  002402                                                          MOV     BDATA,-(SP)
```

```
4150  017220  013746  002400
4151  017224  012746  012252
4152  017230  012746  000004
4153  017234  010600
4154  017236  104414
4155  017240  062706  000012
4156  017244  004737  020160
4157  017250
4158  017250
4159  017250  104423
4160
4161
4162
4163  017252
4164  017252
4165  017252  113701  002412
4166  017256  006301
4167  017260
4168  017260  016146  017010
4169  017264  012746  016607
4170  017270  012746  012366
4171  017274  012746  000003
4172  017300  010600
4173  017302  104414
4174  017304  062706  000010
4175  017310  004737  020134
4176  017314
4177  017314  013746  002404
4178  017320  013746  002402
4179  017324  013746  002400
4180  017330  012746  011705
4181  017334  012746  000004
4182  017340  010600
4183  017342  104414
4184  017344  062706  000012
4185  017350
4186  017350  012746  011640
4187  017354  012746  000001
4188  017360  010600
4189  017362  104414
4190  017364  062706  000004
4191  017370
4192  017370
4193  017370  104423
4194
4195
4196
4197
4198  017372
4199  017372
4200  017372
4201  017372  013746  002472
4202  017376  012746  015541
4203  017402  012746  012523
4204  017406  012746  000003
4205  017412  010600
```

```
....ERROR HANDLER -- ERR5 -- WORD SELECT REG. ERRORS
                                                        MOV     GDATA,-(SP)
                                                        MOV     #FMT10,-(SP)
                                                        MOV     #4,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #12,SP
        JSR     PC,ERR5$                ;DUMP THE SELECT REGISTERS
        ENDMSG
                                                L10003:
                                                        TRAP    C$MSG
;-------------------------------------------------------------------------------
.SBTTL ....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS
;-------------------------------------------------------------------------------
        BGNMSG  ERR7A
                                                ERR7A::
        MOVB    REGNUM,R1
        ASL     R1              ;AS PASSED, THIS WAS A BYTE OFFSET
        PRINTB  #FMT15,#TXTUR,TXTURT(R1)
                                                        MOV     TXTURT(R1),-(SP)
                                                        MOV     #TXTUR,-(SP)
                                                        MOV     #FMT15,-(SP)
                                                        MOV     #3,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #10,SP
        JSR     PC,XORGB
        PRINTB  #FMT3,GDATA,BDATA,XDATA
                                                        MOV     XDATA,-(SP)
                                                        MOV     BDATA,-(SP)
                                                        MOV     GDATA,-(SP)
                                                        MOV     #FMT3,-(SP)
                                                        MOV     #4,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #12,SP
        PRINTB  #ENDEMB
                                                        MOV     #ENDEMB,-(SP)
                                                        MOV     #1,-(SP)
                                                        MOV     SP,R0
                                                        TRAP    C$PNTB
                                                        ADD     #4,SP
        ENDMSG
                                                L10004:
                                                        TRAP    C$MSG
;-------------------------------------------------------------------------------
.SBTTL ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)
;-------------------------------------------------------------------------------
        BGNMSG  ERR10
                                                ERR10::
        PRINTB  #FMT21,#TXT12,MPCSR
                                                        MOV     MPCSR,-(SP)
                                                        MOV     #TXT12,-(SP)
                                                        MOV     #FMT21,-(SP)
                                                        MOV     #3,-(SP)
                                                        MOV     SP,R0
```

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

```
4206  017414  104414
4207  017416  062706  000010                                                    TRAP    C$PNTB
4208  017422                                                                    ADD     #10,SP
                      PRINTB  #FMT22
4209  017422  012746  012533                                                    MOV     #FMT22,-(SP)
4210  017426  012746  000001                                                    MOV     #1,-(SP)
4211  017432  010600                                                           MOV     SP,R0
4212  017434  104414                                                           TRAP    C$PNTB
4213  017436  062706  000004                                                    ADD     #4,SP
4214  017442  013701  002412           MOV     REGNUM,R1
4215  017446  006301                   ASL     R1                      ;GET PTR TO USYRT REG ASCII
4216  017450                           PRINTB  #FMT27,#TXTUR,TXTURT(R1)
4217  017450  016146  017010                                                    MOV     TXTURT(R1),-(SP)
4218  017454  012746  016607                                                    MOV     #TXTUR,-(SP)
4219  017460  012746  012732                                                    MOV     #FMT27,-(SP)
4220  017464  012746  000003                                                    MOV     #3,-(SP)
4221  017470  010600                                                           MOV     SP,R0
4222  017472  104414                                                           TRAP    C$PNTB
4223  017474  062706  000010                                                    ADD     #10,SP
4224  017500  004737  020134           JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
4225  017504                           PRINTB  #FMT23,GDATA,BDATA,XDATA
4226  017504  013746  002404                                                    MOV     XDATA,-(SP)
4227  017510  013746  002402                                                    MOV     BDATA,-(SP)
4228  017514  013746  002400                                                    MOV     GDATA,-(SP)
4229  017520  012746  012555                                                    MOV     #FMT23,-(SP)
4230  017524  012746  000004                                                    MOV     #4,-(SP)
4231  017530  010600                                                           MOV     SP,R0
4232  017532  104414                                                           TRAP    C$PNTB
4233  017534  062706  000012                                                    ADD     #12,SP
4234  017540  004737  020710           JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
4235  017544                           ENDMSG
4236  017544                                                            L10005:
4237  017544  104423                                                           TRAP    C$MSG
4238
4239
4240
4241
4242
```

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

                      BGNMSG  ERR11
4243  017546                                                            ERR11::
                      PRINTB  #FMT21,#TXT12,MPCSR
4244  017546                                                                    MOV     MPCSR,-(SP)
4245  017546                                                                    MOV     #TXT12,-(SP)
4246  017546  013746  002472                                                    MOV     #FMT21,-(SP)
4247  017552  012746  015541                                                    MOV     #3,-(SP)
4248  017556  012746  012523                                                    MOV     SP,R0
4249  017562  012746  000003                                                    TRAP    C$PNTB
4250  017566  010600                                                           ADD     #10,SP
4251  017570  104414
4252  017572  062706  000010
4253  017576                           PRINTB  #FMT22
4254  017576  012746  012533                                                    MOV     #FMT22,-(SP)
4255  017602  012746  000001                                                    MOV     #1,-(SP)
4256  017606  010600                                                           MOV     SP,R0
4257  017610  104414                                                           TRAP    C$PNTB
4258  017612  062706  000004                                                    ADD     #4,SP
4259  017616  013701  002412           MOV     REGNUM,R1
4260  017622  006301                   ASL     R1                      ;GET PTR TO VIA REG ASCII
4261  017624                           PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
```

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

4262  017624  016146  016724                                                MOV     TXTVRT(R1),-(SP)
4263  017630  012746  016312                                                MOV     #TXTVR,-(SP)
4264  017634  012746  012732                                                MOV     #FMT27,-(SP)
4265  017640  012746  000003                                                MOV     #3,-(SP)
4266  017644  010600                                                        MOV     SP,R0
4267  017646  104414                                                        TRAP    C$PNTB
4268  017650  062706  000010                                                ADD     #10,SP
4269  017654  004737  020134          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
4270  017660                          PRINTB  #FMT23,GDATA,BDATA,XDATA
4271  017660  013746  002404                                                MOV     XDATA,-(SP)
4272  017664  013746  002402                                                MOV     BDATA,-(SP)
4273  017670  013746  002400                                                MOV     GDATA,-(SP)
4274  017674  012746  012555                                                MOV     #FMT23,-(SP)
4275  017700  012746  000004                                                MOV     #4,-(SP)
4276  017704  010600                                                        MOV     SP,R0
4277  017706  104414                                                        TRAP    C$PNTB
4278  017710  062706  000012                                                ADD     #12,SP
4279  017714  004737  020356          JSR     PC,ERR11$               ;GET & PRINT VIA REGISTERS
4280  017720                          ENDMSG
4281  017720                                                          L10006:
4282  017720  104423                                                        TRAP    C$MSG
4283
4284
4285
4286
4287
                        ;-------------------------------------------------------------------------
                        ;SBTTL ....ERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)
                        ;-------------------------------------------------------------------------
4288  017722                          BGNMSG  ERR12
4289  017722                                                          ERR12::
4290  017722                          PRINTB  #FMT21,#TXT12,MPCSR
4291  017722  013746  002472                                                MOV     MPCSR,-(SP)
4292  017726  012746  015541                                                MOV     #TXT12,-(SP)
4293  017732  012746  012523                                                MOV     #FMT21,-(SP)
4294  017736  012746  000003                                                MOV     #3,-(SP)
4295  017742  010600                                                        MOV     SP,R0
4296  017744  104414                                                        TRAP    C$PNTB
4297  017746  062706  000010                                                ADD     #10,SP
4298  017752                          PRINTB  #FMT22
4299  017752  012746  012533                                                MOV     #FMT22,-(SP)
4300  017756  012746  000001                                                MOV     #1,-(SP)
4301  017762  010600                                                        MOV     SP,R0
4302  017764  104414                                                        TRAP    C$PNTB
4303  017766  062706  000004                                                ADD     #4,SP
4304  017772  013701  002412          MOV     REGNUM,R1
4305  017776  006301                  ASL     R1                      ;GET PTR TO USYRT REG ASCII
4306  020000                          PRINTB  #FMT27,#TXTUR,TXTURT(R1)
4307  020000  016146  017010                                                MOV     TXTURT(R1),-(SP)
4308  020004  012746  016607                                                MOV     #TXTUR,-(SP)
4309  020010  012746  012732                                                MOV     #FMT27,-(SP)
4310  020014  012746  000003                                                MOV     #3,-(SP)
4311  020020  010600                                                        MOV     SP,R0
4312  020022  104414                                                        TRAP    C$PNTB
4313  020024  062706  000010                                                ADD     #10,SP
4314  020030  004737  020710          JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
4315  020034                          ENDMSG
4316  020034                                                          L10007:
4317  020034  104423                                                        TRAP    C$MSG
```

```
                                ;-------------------------------------------------------------------------
                                .SBTTL ....ERROR HANDLER -- ERR13 -- TRANSMISSION/TSO ERROR (XOR, REG PRINTOUT)
                                ;-------------------------------------------------------------------------

                                        BGNMSG  ERR13
                                                                                        ERR13::
                                        PRINTB  #FMT21,#TXT12,MPCSR
                                                                                                MOV     MPCSR,-(SP)
                                                                                                MOV     #TXT12,-(SP)
                                                                                                MOV     #FMT21,-(SP)
                                                                                                MOV     #3,-(SP)
                                                                                                MOV     SP,R0
                                                                                                TRAP    C$PNTB
                                                                                                ADD     #10,SP
                                        JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
                                        PRINTB  #FMT23,GDATA,BDATA,XDATA
                                                                                                MOV     XDATA,-(SP)
                                                                                                MOV     BDATA,-(SP)
                                                                                                MOV     GDATA,-(SP)
                                                                                                MOV     #FMT23,-(SP)
                                                                                                MOV     #4,-(SP)
                                                                                                MOV     SP,R0
                                                                                                TRAP    C$PNTB
                                                                                                ADD     #12,SP
                                        JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
                                        ENDMSG
                                                                                        L10010:
                                                                                                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''

020134  010146                  XORGB:  MOV     R1,-(SP)        ;PRESERVE WORKING REGISTER
020136  013701  002400                  MOV     GDATA,R1        ;GET 'GOOD'' DATA
020142  013737  002402  002404          MOV     BDATA,XDATA     ;AND 'BAD'' DATA
020150  074137  002404                  XOR     R1,XDATA        ;PERFORM EXCLUSIVE OR
020154  012601                          MOV     (SP)+,R1        ;RESTORE R1
020156  000207                          RTS     PC              ;RETURN


                                ;-------------------------------------------------------------------------
                                .SBTTL ........ERROR HANDLER SUBROUTINE -- ERR5$
                                ;-------------------------------------------------------------------------
                                ;       COMMON ERROR SUBROUTINE TO PRINT SELECT REGISTERS
020160                          ERR5$:
                                        PRINTX  #FMT4,#TXT6,#TXT4
020160  012746  014773                                                                 MOV     #TXT4,-(SP)
020164  012746  015076                                                                 MOV     #TXT6,-(SP)
```

```
4374  020170  012746  011771                                        MOV     #FMT4,-(SP)
4375  020174  012746  000003                                        MOV     #3,-(SP)
4376  020200  010600                                              MOV     SP,R0
4377  020202  104415                                              TRAP    C$PNTX
4378  020204  062706  000010                                      ADD     #10,SP
4379  020210                  PRINTX  #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS
4380  020210  013746  002264                                        MOV     WSR6,-(SP)
4381  020214  013746  002262                                        MOV     WSR4,-(SP)
4382  020220  013746  002260                                        MOV     WSR2,-(SP)
4383  020224  013746  002256                                        MOV     WSR0,-(SP)
4384  020230  012746  012347                                        MOV     #FMT11,-(SP)
4385  020234  012746  000005                                        MOV     #5,-(SP)
4386  020240  010600                                              MOV     SP,R0
4387  020242  104415                                              TRAP    C$PNTX
4388  020244  062706  000014                                      ADD     #14,SP
4389  020250                  PRINTX  #FMT4B,#TXT4A
4390  020250  012746  015033                                        MOV     #TXT4A,-(SP)
4391  020254  012746  012062                                        MOV     #FMT4B,-(SP)
4392  020260  012746  000002                                        MOV     #2,-(SP)
4393  020264  010600                                              MOV     SP,R0
4394  020266  104415                                              TRAP    C$PNTX
4395  020270  062706  000006                                      ADD     #6,SP
4396  020274                  PRINTX  #FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS
4397  020274  013746  002274                                        MOV     WSR16,-(SP)
4398  020300  013746  002272                                        MOV     WSR14,-(SP)
4399  020304  013746  002270                                        MOV     WSR12,-(SP)
4400  020310  013746  002266                                        MOV     WSR10,-(SP)
4401  020314  012746  012347                                        MOV     #FMT11,-(SP)
4402  020320  012746  000005                                        MOV     #5,-(SP)
4403  020324  010600                                              MOV     SP,R0
4404  020326  104415                                              TRAP    C$PNTX
4405  020330  062706  000014                                      ADD     #14,SP
4406  020334                  PRINTB  #ENDEMB
4407  020334  012746  011640                                        MOV     #ENDEMB,-(SP)
4408  020340  012746  000001                                        MOV     #1,-(SP)
4409  020344  010600                                              MOV     SP,R0
4410  020346  104414                                              TRAP    C$PNTB
4411  020350  062706  000004                                      ADD     #4,SP
4412  020354  000207                  RTS     PC
4413
4414                    ;-------------------------------------------------------------------
4415                    .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR11$
4416                    ;-------------------------------------------------------------------
4417                    ;       COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS
4418
4419  020356  004737  004502  ERR11$: JSR     PC,GETVRS               ;GET VIA REGS FOR PRINTOUT
4420  020362                  PRINTX  #FMT24,#TXT16,#TXT17
4421  020362  012746  015717                                        MOV     #TXT17,-(SP)
4422  020366  012746  015704                                        MOV     #TXT16,-(SP)
4423  020372  012746  012634                                        MOV     #FMT24,-(SP)
4424  020376  012746  000003                                        MOV     #3,-(SP)
4425  020402  010600                                              MOV     SP,R0
4426  020404  104415                                              TRAP    C$PNTX
4427  020406  062706  000010                                      ADD     #10,SP
4428  020412                  PRINTX  #FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6
4429  020412  013746  002344                                        MOV     VREGS+6,-(SP)
```

```
4430  020416  013746  002342                                        MOV     VREGS+4,-(SP)
4431  020422  013746  002340                                        MOV     VREGS+2,-(SP)
4432  020426  013746  002336                                        MOV     VREGS+0,-(SP)
4433  020432  012746  012647                                        MOV     #FMT25,-(SP)
4434  020436  012746  000005                                        MOV     #5,-(SP)
4435  020442  010600                                                MOV     SP,R0
4436  020444  104415                                                TRAP    C$PNTX
4437  020446  062706  000014                                        ADD     #14,SP
4438  020452                      PRINTX  #FMT29  #TXT18
4439  020452  012746  015754                                        MOV     #TXT18,-(SP)
4440  020456  012746  012741                                        MOV     #FMT29,-(SP)
4441  020462  012746  000002                                        MOV     #2,-(SP)
4442  020466  010600                                                MOV     SP,R0
4443  020470  104415                                                TRAP    C$PNTX
4444  020472  062706  000006                                        ADD     #6,SP
4445  020476                      PRINTX  #FMT26,VREGS+8.,VREGS+10.,VREGS+12.,VREGS+14.
4446  020476  013746  002354                                        MOV     VREGS+14.,-(SP)
4447  020502  013746  002352                                        MOV     VREGS+12.,-(SP)
4448  020506  013746  002350                                        MOV     VREGS+10.,-(SP)
4449  020512  013746  002346                                        MOV     VREGS+8.,-(SP)
4450  020516  012746  012677                                        MOV     #FMT26,-(SP)
4451  020522  012746  000005                                        MOV     #5,-(SP)
4452  020526  010600                                                MOV     SP,R0
4453  020530  104415                                                TRAP    C$PNTX
4454  020532  062706  000014                                        ADD     #14,SP
4455  020536                      PRINTX  #FMT29,#TXT19
4456  020536  012746  016015                                        MOV     #TXT19,-(SP)
4457  020542  012746  012741                                        MOV     #FMT29,-(SP)
4458  020546  012746  000002                                        MOV     #2,-(SP)
4459  020552  010600                                                MOV     SP,R0
4460  020554  104415                                                TRAP    C$PNTX
4461  020556  062706  000006                                        ADD     #6,SP
4462  020562                      PRINTX  #FMT25,VREGS+16.,VREGS+18.,VREGS+20.,VREGS+22.
4463  020562  013746  002364                                        MOV     VREGS+22.,-(SP)
4464  020566  013746  002362                                        MOV     VREGS+20.,-(SP)
4465  020572  013746  002360                                        MOV     VREGS+18.,-(SP)
4466  020576  013746  002356                                        MOV     VREGS+16.,-(SP)
4467  020602  012746  012647                                        MOV     #FMT25,-(SP)
4468  020606  012746  000005                                        MOV     #5,-(SP)
4469  020612  010600                                                MOV     SP,R0
4470  020614  104415                                                TRAP    C$PNTX
4471  020616  062706  000014                                        ADD     #14,SP
4472  020622                      PRINTX  #FMT29,#TXT20
4473  020622  012746  016051                                        MOV     #TXT20,-(SP)
4474  020626  012746  012741                                        MOV     #FMT29,-(SP)
4475  020632  012746  000002                                        MOV     #2,-(SP)
4476  020636  010600                                                MOV     SP,R0
4477  020640  104415                                                TRAP    C$PNTX
4478  020642  062706  000006                                        ADD     #6,SP
4479  020646                      PRINTX  #FMT26,VREGS+24.,VREGS+26.,VREGS+28.,VREGS+30.
4480  020646  013746  002374                                        MOV     VREGS+30.,-(SP)
4481  020652  013746  002372                                        MOV     VREGS+28.,-(SP)
4482  020656  013746  002370                                        MOV     VREGS+26.,-(SP)
4483  020662  013746  002366                                        MOV     VREGS+24.,-(SP)
4484  020666  012746  012677                                        MOV     #FMT26,-(SP)
4485  020672  012746  000005                                        MOV     #5,-(SP)
```

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

                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #14,SP
                        RTS     PC

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

                        ERR12$: JSR     PC,GETURS               ;GET USYRT REGS FOR PRINTOUT
                                PRINTX  #FMT24,#TXT13,#TXT14
                                                                        MOV     #TXT14,-(SP)
                                                                        MOV     #TXT13,-(SP)
                                                                        MOV     #FMT24,-(SP)
                                                                        MOV     #3,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #10,SP
                                PRINTX  #FMT25,UREGS+0,UREGS+2,UREGS+4,UREGS+6
                                                                        MOV     UREGS+6,-(SP)
                                                                        MOV     UREGS+4,-(SP)
                                                                        MOV     UREGS+2,-(SP)
                                                                        MOV     UREGS+0,-(SP)
                                                                        MOV     #FMT25,-(SP)
                                                                        MOV     #5,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #14,SP
                                PRINTX  #FMT29,#TXT15
                                                                        MOV     #TXT15,-(SP)
                                                                        MOV     #FMT29,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #6,SP
                                PRINTX  #FMT26,UREGS+10,UREGS+12,UREGS+14,UREGS+16
                                                                        MOV     UREGS+16,-(SP)
                                                                        MOV     UREGS+14,-(SP)
                                                                        MOV     UREGS+12,-(SP)
                                                                        MOV     UREGS+10,-(SP)
                                                                        MOV     #FMT26,-(SP)
                                                                        MOV     #5,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #14,SP
                        RTS     PC
```

CVDMCA.P11 10-DEC-80 09:14 LOAD DEVICE PROTECTION TABLE

```
4535                                    .SBTTL  LOAD DEVICE PROTECTION TABLE
4536
4537                                    ;////////////////////////////////////////////////////////////////////////////////
4538                                    ;/ THIS TABLE IDENTIFIES THE LOAD DEVICE TO THE SUPERVISOR, SO THAT IT CAN BE
4539                                    ;/ PROTECTED FROM TESTING, IF DESIRED.
4540                                    ;////////////////////////////////////////////////////////////////////////////////
4541
4542  021072                                    BGNPROT
4543  021072                                                                    L$PROT::
4544  021072  177777                           .WORD   -1      ;DON'T CHK CSR ADRS
4545  021074  177777                           .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
4546  021076  177777                           .WORD   -1      ;DON'T CHK DRIVE NO.
4547  021100                                    ENDPROT
```

```
                                .SBTTL  INITIALIZE SECTION

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

021100                                  BGNINIT
021100                                                                          L$INIT::

021100  010637  002414                  MOV     SP,PSTACK       ;SAVE BASE-LEVEL STACK POINTER
021104  005037  002420                  CLR     SUBRPC          ;CLEAR SUBR CALL PC
021110  005037  002454                  CLR     CHPTYP          ;CLEAR USYRT CHIP TYPE INDICATOR
021114  005037  002452                  CLR     ERROR1          ;CLEAR ERROR FLAG
021120  005037  002456                  CLR     SAVLEN          ;CLEAR CHAR LENGTH FROM SETUP
021124  005737  002444                  TST     FRSTIM          ;SEE IF FIRST TIME THROUGH AFTER LOAD
021130  001007                          BNE     6$              ;BR IF NOT
021132  013737  000004  002446          MOV     @#4,SAVE4       ;SAVE ERROR TRAP VECTOR
021140  013737  000006  002450          MOV     @#6,SAVE6
021146  000406                          BR      9$

021150  013737  002446  000004  6$:     MOV     SAVE4,@#4       ;RESTORE ERROR TRAP VECTOR
021156  013737  002450  000006          MOV     SAVE6,@#6

021164  012737  000001  002444  9$:     MOV     #1,FRSTIM       ;MARK FLAG FOR NEXT TIME THROUGH

                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
021172                                  READEF  #EF.START
021172  012700  000040                                                  MOV     #EF.START,R0
021176  104447                                                          TRAP    C$REFG
021200                                  BCOMPLETE       STARST
021200  103415                                                          BCS     STARST

                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
021202                                  READEF  #EF.RESTART
021202  012700  000037                                                  MOV     #EF.RESTART,R0
021206  104447                                                          TRAP    C$REFG
021210                                  BCOMPLETE       STARST
021210  103411                                                          BCS     STARST

                                ;SEE IF THIS IS A NEW PASS, BR IF YES
021212                                  READEF  #EF.NEW
021212  012700  000035                                                  MOV     #EF.NEW,R0
021216  104447                                                          TRAP    C$REFG
021220                                  BCOMPLETE       NEWST
021220  103411                                                          BCS     NEWST

                                ;SEE IF PROGRAM WAS JUST CONTINUED
021222                                  READEF  #EF.CONTINUE
021222  012700  000036                                                  MOV     #EF.CONTINUE,R0
021226  104447                                                          TRAP    C$REFG
021230                                  BCOMPLETE       ENDIT
021230  103470                                                          BCS     ENDIT
021232  000414                          BR      GETPRM

021234                          STARST.
```

```
4604  021234  005037  002466          CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
4605
4606                                  ;CLEAR DEVICE MAP
4607  021240  005037  002460          CLR     DEVMAP
4608  021244                        NEWST:
4609  021244  012737  177777  002410  MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
4610  021252  005237  002466          INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
4611  021256  012737  000001  002462  MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
4612
4613                                  ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
4614                                  ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
4615  021264                        GETPRM:
4616  021264  005237  002410          INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
4617  021270                          GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
4618  021270  013700  002410                                                          MOV     LOGDEV,R0
4619  021274  104442                                                                  TRAP    C$GPHRD
4620  021276  010001                                                                  MOV     R0,R1
4621  021300                          BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
4622  021300  103403                                                                  BCS     10$
4623  021302  006337  002462          ASL     DEVPTR          ;SHIFT DEVICE POINTER
4624  021306  000766          BR      GETPRM          ;SKIP THIS DEVICE
4625  021310  053737  002462  002460  10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
4626  021316  006337  002462          ASL     DEVPTR          ;SHIFT BIT POINTER
4627
4628  021322  012102          MOV     (R1)+,R2        ;R2=CSR ADDR VALUE
4629  021324  012703  002472          MOV     #MPCSR,R3       ;R3=POINTER TO CSR ADDR STORAGE AREA
4630
4631  021330  010223          11$:    MOV     R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
4632  021332  005202          INC     R2              ;BUMP BSEL ADDR
4633  021334  022703  002532          CMP     #BSEL17+2,R3    ;ALL 16 ADDRESSES MOVED ?
4634  021340  001373          BNE     11$             ;NO: DO ANOTHER ADDRESS
4635                                                  ;YES: CONTINUE
4636
4637  021342  011137  002532          MOV     (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
4638  021346  012137  002534          MOV     (R1)+,MPOVEC
4639  021352  062737  000004  002534  ADD     #4,MPOVEC       ;GET DMV11 OUTPUT INTRPT VECTOR
4640  021360  012137  002536          MOV     (R1)+,MPRIOR    ;GET DMV11 DEVICE PRIORITY
4641  021364  012137  002540          MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
4642  021370  012137  002542          MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
4643  021374  012137  002544          MOV     (R1)+,BRDTYP    ;GET DMV-11 BOARD TYPE
4644  021400  012137  002546          MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
4645  021404  011137  002550          MOV     (R1),BDRATE     ;GET BAUD RATE FOR THIS DEVICE
4646                                ;ISSUE LSI BUS RESET, TO INIT DMV11
4647  021410                          BRESET
4648  021410  104433                                                                  TRAP    C$RESET
4649  021412                        ENDIT:
4650  021412                          ENDINIT
4651  021412                                                                L10012:
4652  021412  104411                                                                  TRAP    C$INIT
```

```
                        .SBTTL AUTO DROP UNIT SECTION

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

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

021414                          BGNAUTO
021414                                                                  L$AUTO::
                        ;
021414                          SETVEC  #4,#AD.HIT,#0   ;SETUP INVALID-ADDRESS TRAP VECTOR
021414  012746  000000                                                  MOV     #0,-(SP)
021420  012746  021532                                                  MOV     #AD.HIT,-(SP)
021424  012746  000004                                                  MOV     #4,-(SP)
021430  012746  000003                                                  MOV     #3,-(SP)
021434  104437                                                          TRAP    C$SVEC
021436  062706  000010                                                  ADD     #10,SP
021442  005037  002622                  CLR     TMP0            ;INITIALIZE TRAP FLAG REGISTER
021446  012702  000001                  MOV     #1,R2           ;FLAG BIT
021452  013703  002472                  MOV     BSEL0,R3        ;INIT ADDRESS POINTER
                        ;
021456  105723          1$:             TSTB    (R3)+           ;ACCESS THE CSR'S BY BYTES.
021460  006302                          ASL     R2
021462  103375                          BCC     1$
                        ;
021464  013703  002472                  MOV     BSEL0,R3        ;RE-INIT ADDRESS POINTER
021470  012702  000001                  MOV     #1,R2           ;RE-INIT FLAG BIT
021474  005723          2$:             TST     (R3)+           ;ACCESS THE CSR'S BY WORDS.
021476  006302                          ASL     R2
021500  006302                          ASL     R2
021502  103374                          BCC     2$
                        ;
021504                          CLRVEC  #4              ;RESTORE THE VECTOR TO DS
021504  012700  000004                                                  MOV     #4,R0
021510  104436                                                          TRAP    C$CVEC
021512  005737  002622                  TST     TMP0            ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP?
021516  001403                          BEQ     AD.OK           ;NO,  EXIT TEST
021520                                  DODU    LOGDEV          ;YES, DROP THIS LOGICAL DEV.
021520  013700  002410                                                  MOV     LOGDEV,R0
021524  104451                                                          TRAP    C$DODU
                        ;
021526  000240          AD.OK:          NOP                     ;(FOR PATCHING IN A HALT IF NECESSARY)
                        ;
021530                          ENDAUTO
021530                                                                  L10013:
021530  104461                                                          TRAP    C$AUTO
```

```
 4709  021532  050237  002622        AD.HIT: BIS     R2,TMP0         ;FLAG THE HIT IF WE GET IT!
 4710  021536  000002                        RTI                     ;RETURN
 4711
```

```
4712
4713
4714                    .SBTTL  CLEANUP CODING SECTION
4715
4716                    ;////////////////////////////////////////////////////////////////////////
4717                    ;// THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
4718                    ;// AT THE END OF THE TEST SEQUENCE ON A PARTICULAR UNIT.
4719                    ;////////////////////////////////////////////////////////////////////////
4720
4721 021540                       BGNCLN
4722 021540                                                          L$CLEAN::
4723
4724
4725 021540                       ENDCLN
4726 021540                                                          L10014:
4727 021540 104412                                                          TRAP    C$CLEAN
```

```
 4726
 4727
 4728                    .SBTTL  DROP UNIT SECTION
 4729
 4730                    ;////////////////////////////////////////////////////////////////////////////
 4731                    ;/ THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
 4732                    ;/ TO NO LONGER BE TESTED.
 4733                    ;////////////////////////////////////////////////////////////////////////////
 4734
 4735                            BGNDU
 4736                                                                    L$DU::
 4737                    ;ISSUE UNIBUS RESET TO CLEAN UP
 4738                            BRESET
 4739                                                                            TRAP    C$RESET
 4740                            ENDDU
                                                                        L10015:
                                                                            TRAP    C$DU
```

CVDMCA.P11 10-DEC-80 09:14 ADD UNIT SECTION

```
4741                                .SBTTL  ADD UNIT SECTION
4742
4743                                ;////////////////////////////////////////////////////////////////////////////
4744                                ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
4745                                ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
4746                                ;/  'EF.AUNIT' IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
4747                                ;////////////////////////////////////////////////////////////////////////////
4748
4749  021546                            BGNAU
4750  021546                                                          LSAU::
4751  021546                            ENDAU
4752  021546                                                          L10016:
4753  021546  104452                                                       TRAP    CSAU
4754
```

```
                        .SBTTL  TEST 1 -- TBMT MICROCODE INTERRUPT TEST

                        ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        ;*
                        ;*      TEST 1 -- TBMT MICROCODE INTERRUPT TEST
                        ;*
                        ;*  THIS TEST CHECKS THE OPERATION OF THE TBMT (IRQ) INTERRUPT.
                        ;*  THIS IS DONE BY ISSUEING THE 'SET MAINTENANCE INTERRUPT FLAG AND CLEAR
                        ;*  INTERRUPT DISABLE IN PROCESSOR STATUS'' COMMAND WHILE IN THE MAINTENANCE
                        ;*  LOOP AND THEN CHECKING FOR BIT 7 OF BSEL3 TO BE SET (THE BIT IS SET
                        ;*  BY THE MICROCODE WHEN THE TBMT INTERRUPT OCCURS).
                        ;*
                        ;---------------------------------------------------------------------------------------
                        ;
021550                  ;       BGNTST
                                                                                T1::
021550  004737  003374          ;       JSR     PC,@MSTCLR      ;PUT THE MICROPROCESSOR IN THE MAINTENANCE LOOP
021554  103003                  ;       BCC     .+8.            ;IF NO ERROR, PROCEED
021556                          ;       ERROR                   ;ELSE, REPORT IT AND
021556  104460                                                                  TRAP    C$ERROR
021560                          ;       ESCAPE  TST             ;       EXIT THIS TEST
021560  104410                                                                  TRAP    C$ESCAPE
021562  000144                                                                  .WORD   L10017-.

021564  004537  005432          ;       JSR     R5,@CKUSTS      ;CHK USYRT STATUS FOR INIT STATE
021570  000110                  ;       110                     ;TBMT=1, TSO=1
021572  103003                  ;       BCC     .+8.            ;IF ERROR, PRINT REPORT
021574                          ;       ERROR
021574  104460                                                                  TRAP    C$ERROR
021576                          ;       ESCAPE  TST             ; AND SKIP REMAINDER OF TEST
021576  104410                                                                  TRAP    C$ESCAPE
021600  000126                                                                  .WORD   L10017-.

021602  012777  000007  160666  ;       MOV     #DOTBMT,@SEL2   ;ISSUE 'TBMT TEST'' COMMAND

021610  010346                  ;       MOV     R3,-(SP)
021612  012703  000050          ;       MOV     #40.,R3         ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
021616  077301          1$:     ;       SOB     R3,1$
021620  012603                  ;       MOV     (SP)+,R3

021622  132777  000200  160646  ;       BITB    #MRDY,@BSEL2    ;'MRDY'' SHOULD BE HIGH BY NOW.
021630  001013                  ;       BNE     5$              ;BR IF NO ERROR

021632  004737  004210          ;       JSR     PC,@GETWSR      ;GET SELECT REGISTERS
021636  012737  000007  002400  ;       MOV     #DOTBMT,@DATA   ;IDENTIFY REQUESTED FUNCTION
021644                          ;       GEDF    EM4,ERR4        ;REPORT 'MRDY'' TIMEOUT ERROR...
                                ;                               ;       'DEVICE FATAL'' ERROR # 39
021644  104455                                                                  TRAP    C$ERDF
021646  000047                                                                  .WORD   39
021650  013235                                                                  .WORD   EM4
021652  017030                                                                  .WORD   ERR4

021654                          ;       ESCAPE  TST             ;AND EXIT TEST
021654  104410                                                                  TRAP    C$ESCAPE
021656  000050                                                                  .WORD   L10017-.
```

```
4811  021660  004737  004210          5$:     JSR     PC,GETWSR       ;GET CURRENT REGISTER CONTENTS
4812  021664  023727  002260  100200          CMP     WSR2,#100200    ;COMPARE BYTE SELECT REGISTERS 2 AND 3
4813                                                                  ;REG 2 = 200 -- 'MRDY' SET/COMMAND CLEARED
4814                                                                  ;REG 3 = 200 -- 'TBMT' INTERRUPT BIT SET
4815  021672  001415                          BEQ     7$              ;BR IF NO ERROR.
4816
4817  021674  012737  100200  002400          MOV     #100200,GDATA   ;GET THE GOOD DATA
4818  021702  013737  002260  002402          MOV     WSR2,BDATA      ;GET THE BAD DATA
4819  021710  012737  000002  002412          MOV     #2,REGNUM       ;GET THE REGISTER NUMBER
4820  021716                                  GEDF    EM92,ERR5       ;REPORT 'TBMT INTERRUPT FAILURE' ERROR.
4821                                                                  ;       'DEVICE FATAL' ERROR # 40
4822  021716  104455                                                                  TRAP    C$ERDF
4823  021720  000050                                                                  .WORD   40
4824  021722  014323                                                                  .WORD   EM92
4825  021724  017154                                                                  .WORD   ERR5
4826
4827  021726                          7$:     ENDTST
4828  021726                                                                  L10017:
4829  021726  104401                                                                  TRAP    C$ETST
4830
```

```
                                .SBTTL  TEST 2 -- SWITCH SETTING TEST

                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 2 -- SWITCH SETTING TEST
                                ;*
                                ;* SUBTEST #1:
                                ;* THE TWO READABLE SWITCH PACKS WILL BE SAMPLED AND COMPARED AGAINST THE 2
                                ;* VALUES IN THE P-TABLE. AN ERROR IS REPORTED ON A MISMATCH.
                                ;*
                                ;* SUBTEST #2:
                                ;* THE SPEED SELECT SWITCH (SPDSEL) IS READ VIA THE VIAORA REGISTER (BIT PA4)
                                ;* AND COMPARED AGAINST THE BAUD RATE VALUE IN THE P-TABLE. IF A MISMATCH
                                ;* OCCURS IT WILL BE REPORTED. NOTE: THIS SUBROUTINE IS NOT RUN IF AN M8064
                                ;* BOARD IS BEING TESTED (IT ONLY RUNS @56K... MAKING A SPEED SWITCH USELESS).
                                ;*
                                ;*
021730                          ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;
                                ;       BGNTST
                                                                                T2::
021730  004737  005420          ;       JSR     PC,INIDMV       ;INIT DMV-11 (MAINT LOOP)
                                ;------------------------------ ------------------------------
021734                          ;       BGNSUB
021734                                                                          T2.1:
104402                                                                                  TRAP    C$BSUB
003610                          ;       JSR     R5,READI        ;GET 'DDCMP ADDRESS''
121400                          ;       .WORD   SWPDDCMP
000000                  T3.SW1: .WORD   0               ;  (IT WILL BE PUT HERE)
003610                          ;       JSR     R5,READI        ;GET 'BOOT ADDRESS''
121000                          ;       .WORD   SWPBOT
000000                  T3.SW2: .WORD   0               ;  (IT WILL BE PUT HERE)

021744  002540                  ;       CMPB    T3.SW1,LUSWI1   ;DOES 'DDCMP ADDRESS'' MATCH P-TABLE VALUE
                                ;       BNE     T3.ERR          ;NO,  REPORT ERROR

021754  002542                  ;       CMPB    T3.SW2,LUSWI2   ;DOES BOOT ADDRESS'' MATCH P-TABLE VALUE
                                ;       BEQ     T3.OK           ;NO,  REPORT ERROR

021776                  T3.ERR: GEDF    T3.EHD,T3.EM1   ;REPORT SWITCH SETTINGS DON'T MATCH P-TABLE
                                ;                       ;       'DEVICE FATAL'' ERROR # 41
104455                                                                                  TRAP    C$ERDF
000051                                                                                  .WORD   41
022346                                                                                  .WORD   T3.EHD
022116                                                                                  .WORD   T3.EM1
                                ;       CKLOOP
104406                                                                                  TRAP    C$CLP1

000240                  T3.OK:  NOP
                                ;       ENDSUB
                                                                                L10021:
104403                                                                                  TRAP    C$ESUB
                                ;------------------------------------------------------------------------------
                                ;       BGNSUB
                                                                                T2.2:
104402                                                                                  TRAP    C$BSUB
```

```
4887  022016  005737  002544          TST     BRDTYP          ;IS THIS AN M8064 ?
4888  022022  001433                  BEQ     10$             ; IF YES: THEN SKIP THIS SUBROUTINE
4889  022024  004537  003610          JSR     R5,READI        ;GET VIAORA REGISTER
4890  022030  120017                  VIAORA
4891  022032  000000          1$:     000                     ;STATUS WORD GOES HERE
4892  022034  103003                  BCC     .+8.
4893  022036                          ERROR
4894  022036  104460                                                  TRAP    C$ERROR
4895  022040                          ESCAPE  SUB
4896  022040  104410                                                  TRAP    C$ESCAPE
4897  022042  000050                                                  .WORD   L10022-.
4898  022044  142737  000357  022032          BICB    #357,1$         ;CLEAR ALL BUT SPEED SELECT BIT(PA4)
4899  022052  106237  022032                  ASRB    1$              ;RIGHT JUSTIFY SPDSEL SWITCH BIT FOR
4900  022056  106237  022032                  ASRB    1$              ; COMPARISON WITH OPERATOR'S REPLY
4901  022062  106237  022032                  ASRB    1$
4902  022066  106237  022032                  ASRB    1$
4903  022072  123737  022032  002550          CMPB    1$,BDRATE       ;IS THE SWITCH IN THE DESIRED POSITION?
4904  022100  001404                  BEQ     10$             ; IF YES: END THE SUBROUTINE/TEST
4905
4906  022102                          GEDF    T3.6,T3.EM2     ;REPORT ERROR
4907                                                          ;       'DEVICE FATAL'' ERROR # 42
4908  022102  104455                                                  TRAP    C$ERDF
4909  022104  000052                                                  .WORD   42
4910  022106  022644                                                  .WORD   T3.6
4911  022110  022310                                                  .WORD   T3.EM2
4912  022112                  10$:    ENDSUB
4913  022112                                                  L10022:
4914  022112  104403                                                  TRAP    C$ESUB
4915                          ;-------------------------------------------------------------
4916  022114                  T3.END: ENDTST
4917  022114                                                  L10020:
4918  022114  104401                                                  TRAP    C$ETST
4919
4920  022116                          BGNMSG  T3.EM1
4921  022116                                                  T3.EM1::
4922  022116                          PRINTB  #T3.1           ;PRINT ERROR MESSAGE
4923  022116  012746  022372                                          MOV     #T3.1,-(SP)
4924  022122  012746  000001                                          MOV     #1,-(SP)
4925  022126  010600                                                  MOV     SP,R0
4926  022130  104414                                                  TRAP    C$PNTB
4927  022132  062706  000004                                          ADD     #4,SP
4928  022136                          PRINTX  #T3.2
4929  022136  012746  022457                                          MOV     #T3.2,-(SP)
4930  022142  012746  000001                                          MOV     #1,-(SP)
4931  022146  010600                                                  MOV     SP,R0
4932  022150  104415                                                  TRAP    C$PNTX
4933  022152  062706  000004                                          ADD     #4,SP
4934  022156                          PRINTX  #T3.3
4935  022156  012746  022506                                          MOV     #T3.3,-(SP)
4936  022162  012746  000001                                          MOV     #1,-(SP)
4937  022166  010600                                                  MOV     SP,R0
4938  022170  104415                                                  TRAP    C$PNTX
4939  022172  062706  000004                                          ADD     #4,SP
4940  022176                          PRINTX  #T3.4,LUSWI2,LUSWI1
4941  022176  013746  002540                                          MOV     LUSWI1,-(SP)
4942  022202  013746  002542                                          MOV     LUSWI2,-(SP)
```

```
4943  022206  012746  022537                                                  MOV     #T3.4,-(SP)
4944  022212  012746  000003                                                  MOV     #3,-(SP)
4945  022216  010600                                                          MOV     SP,R0
4946  022220  104415                                                          TRAP    C$PNTX
4947  022222  062706  000010                                                  ADD     #10,SP
4948  022226  105037  021745          CLRB    T3.SW1+1        ;MAKE SURE BITS 8 THROUGH 15 AREN'T REPORTED
4949  022232  105037  021755          CLRB    T3.SW2+1        ; -- ESPECIALLY BIT 8!!
4950  022236                          PRINTX  #T3.5,T3.SW1,T3.SW2
4951  022236  013746  021754                                                  MOV     T3.SW2,-(SP)
4952  022242  013746  021744                                                  MOV     T3.SW1,-(SP)
4953  022246  012746  022602                                                  MOV     #T3.5,-(SP)
4954  022252  012746  000003                                                  MOV     #3,-(SP)
4955  022256  010600                                                          MOV     SP,R0
4956  022260  104415                                                          TRAP    C$PNTX
4957  022262  062706  000010                                                  ADD     #10,SP
4958  022266                          PRINTB  #NEWLIN
4959  022266  012746  011645                                                  MOV     #NEWLIN,-(SP)
4960  022272  012746  000001                                                  MOV     #1,-(SP)
4961  022276  010600                                                          MOV     SP,R0
4962  022300  104414                                                          TRAP    C$PNTB
4963  022302  062706  000004                                                  ADD     #4,SP
4964  022306                          ENDMSG
4965  022306                                                          L10023:
4966  022306  104423                                                          TRAP    C$MSG
4967
4968  022310                          BGNMSG  T3.EM2
4969  022310                                                          T3.EM2::
4970  022310                          PRINTB  #FMT21,#TXT12,MPCSR
4971  022310  013746  002472                                                  MOV     MPCSR,-(SP)
4972  022314  012746  015541                                                  MOV     #TXT12,-(SP)
4973  022320  012746  012523                                                  MOV     #FMT21,-(SP)
4974  022324  012746  000003                                                  MOV     #3,-(SP)
4975  022330  010600                                                          MOV     SP,R0
4976  022332  104414                                                          TRAP    C$PNTB
4977  022334  062706  000010                                                  ADD     #10,SP
4978  022340  004737  020356          JSR     PC,ERR11$       ;GET AND PRINT VIA REGISTERS
4979  022344                          ENDMSG
4980  022344                                                          L10024:
4981  022344  104423                                                          TRAP    C$MSG
4982
4983                                  .NLIST  BEX
      022346  053523  052111  044103  T3.EHD: .ASCIZ  'SWITCH SETTING TEST'
      022372  047045  040445  053523  T3.1:   .ASCIZ  /%N%ASWITCH PACK SETTING DOES NOT MATCH P-TABLE ENTRY/
      022457     045  022516  031123  T3.2:   .ASCIZ  '%N%S22%A DDCMP    BOOT'
      022506  047045  051445  031062  T3.3:   .ASCIZ  '%N%S22%AADDRESS  ADDRESS'
      022537     045  022516  033123  T3.4:   .ASCIZ  '%N%S6%AP-TABLE VALUES:%S1%03%S6%03'
      022602  047045  051445  022466  T3.5:   .ASCIZ  '%N%S6%AREAD FROM DMV:%S2%03%S6%03'
      022644  050123  051504  046105  T3.6:   .ASCIZ /SPDSEL SWITCH DOESN'T MATCH P-TABLE BAUD RATE ENTRY/
                                      .LIST   BEX
4984                                  .EVEN
```

```
                                .SBTTL  TEST 3 -- USYRT MASTER CLEAR TEST

                                ;++*************************************************************************
                                ;*
                                ;*      TEST 3 -- USYRT MASTER CLEAR TEST
                                ;*
                                ;*  ALL  REGISTERS  ARE  LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR
                                ;*  PATTERN  F  BELOW.  THE USYRT IS THEN CLEARED BY A MASTER CLEAR
                                ;*  (BIT 6 OF BSEL 1).   ALL REGISTERS ARE THEN CHECKED FOR
                                ;*  THE PROPER CONTENTS. THE INITIALIZED STATE OF THE REGISTERS IS CHECKED
                                ;*  AGAINST DATA PATTERN F.
                                ;*
                                ;*      PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
                                ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.
                                ;*
                                ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:
                                ;*
                                ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
                                ;*
                                ;*
                                ;--*************************************************************************
                                ;
                                ;       BGNTST
                                                                                        T3::
022730                                  JSR     PC,INIDMV       ;INIT DMV-11
022730  004737  005420                  CLR     R1              ;INIT USYRT REG ADRS POINTER
022734  005001                          MOV     #PATE,R2        ;INIT PATTERN E POINTER
022736  012702  002652                  MOV     USYREG(R1),2$   ;GET USYRT REG ADRS
022742  016137  002552  022760  1$:     MOVB    (R2)+,3$        ;GET A PATTERN BYTE
022750  112237  022762                  JSR     R5,WRITEI       ;WRITE A USYRT REG
022754  004537  003734                  .WORD   0               ;USYRT REG ADRS GOES HERE
022760  000000                  2$:     .WORD   0               ;DATA BYTE GOES HERE
022762  000000                  3$:     ADD     #2,R1           ;INCR REG ADRS PTR
022764  062701  000002                  CMP     R1,#16.         ;SEE IF ALL REGS WRITTEN YET
022770  020127  000020                  BLT     1$              ;BR IF NOT
022774  002762                          JSR     PC,INIDMV       ;ISSUE MASTER CLEAR
022776  004737  005420                  CLR     R1              ;INIT USYRT REG ADRS PTR
023002  005001                          MOV     #PATF,R2        ;INIT DATA PATTERN POINTER
023004  012702  002662                  MOV     USYREG(R1),7$   ;SET USYRT READ ADDRESS
023010  016137  002552  023022  6$:     JSR     R5,READI        ;READ A USYRT REG
023016  004537  003610                  .WORD   0               ;USYRT REG ADRS GOES HERE
023022  000000                  7$:     .WORD   0               ;DATA READ IS RETURNED HERE
023024  000000                  8$:     CMPB    8$,(R2)+        ;SEE IF REG CONTAINS EXPECTED DATA
023026  123722  023024                  BEQ     9$              ;BR IF MATCH
023032  001422                          MOV     R1,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
023034  010137  002412                  ASR     REGNUM          ;GET WORD OFFSET
023040  006237  002412                  CLR     GDATA           ;GET EXPECTED DATA
023044  005037  002400                  MOVB    -1(R2),GDATA
023050  116237  177777  002400          MOV     8$,BDATA        ;GET ACTUAL DATA
023056  013737  023024  002402  ;REPORT REG NOT CLEARED BY MASTER CLEAR
                                        GEDF    EM1,ERR10
023064                                                  ;       'DEVICE FATAL'' ERROR # 43
                                                                TRAP    C$ERDF
023064  104455                                            .WORD   43
023066  000053                                            .WORD   EM1
023070  013103                                            .WORD   ERR10
023072  017372
```

```
5041  023074
5042  023074  104410                    ESCAPE  TST                                                     TRAP    C$ESCAPE
5043  023076  000014                                                                                    .WORD   L10025-.
5044  023100  062701  000002  9$:       ADD     #2,R1           ;INCR USYRT REG ADRS PTR
5045  023104  020127  000020            CMP     R1,#16.         ;SEE IF ALL REGS READ YET
5046  023110  002737                    BLT     6$              ;BR IF NOT
5047  023112                          ENDTST
5048  023112                                                                          L10025:
5049  023112  104401                                                                 TRAP    C$ETST
```

```
                                .SBTTL  TEST 4 -- USYRT PROGRAM RESET TEST

                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 4 -- USYRT PROGRAM RESET TEST
                                ;*
                                ;*  ALL  REGISTERS  ARE  LOADED WITH PATTERN E IN THE SAME SEQUENCE AS FOR
                                ;*  PATTERN  F  BELOW.  THE USYRT IS THEN RESET BY ASSERTING PROGRAM RESET
                                ;*  (BIT 0 @ A000) IN THE 6522 VIA. ALL REGISTERS ARE THEN CHECKED FOR
                                ;*  THE PROPER CONTENTS. THE INITIALIZED STATE OF THE REGISTERS IS CHECKED
                                ;*  AGAINST DATA PATTERN F.
                                ;*
                                ;*      PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
                                ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.
                                ;*
                                ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:
                                ;*
                                ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
                                ;*
                                ;*
                                ;-----------------------------------------------------------------------------------------------
                                ;
                                ;       BGNTST
023114                                                                          T4::
023114  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11
023120  005001                          CLR     R1              ;INIT USYRT REG ADRS POINTER
023122  012702  002652                  MOV     #PATE,R2        ;INIT PATTERN E POINTER
023126  016137  002552  023144  1$:     MOV     USYREG(R1),2$   ;GET USYRT REG ADRS
023134  112237  023146                  MOVB    (R2)+,3$        ;GET A PATTERN BYTE
023140  004537  003734                  JSR     R5,WRITEI       ;WRITE A USYRT REG
023144  000000                  2$:     .WORD   0               ;USYRT REG ADRS GOES HERE
023146  000000                  3$:     .WORD   0               ;DATA BYTE GOES HERE
023150  062701  000002                  ADD     #2,R1           ;INCR REG ADRS PTR
023154  020127  000020                  CMP     R1,#16.         ;SEE IF ALL REGS WRITTEN YET
023160  002762                          BLT     1$              ;BR IF NOT

023162  004537  005126                  JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
023166  103003                          BCC     .+8.            ;BR IF NO ERROR
023170                                  ERROR                   ;REPORT STACKED ERROR
023170  104460                                                                  TRAP    C$ERROR
023172                                  ESCAPE TST              ;SKIP TO END OF TEST
023172  104410                                                                  TRAP    C$ESCAPE
023174  000002                                                                  .WORD   L10026-.

023176                  ENDTST
023176                                                                  L10026:
023176  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 5 -- USYRT REGISTER ADDRESSING TEST

                                ;++*************************************************************************
                                ;*
                                ;*      TEST 5 -- USYRT REGISTER ADDRESSING TEST
                                ;*
                                ;* FIRST, A MASTER CLEAR IS ISSUED, TO INITIALIZE THE USYRT REGS TO
                                ;* PATTERN F. THEN, EACH REGISTER IS WRITTEN WITH A BYTE OF PATTERN J,
                                ;* AND AFTER EACH IS WRITTEN, ALL ARE READ AND COMPARED TO THE CURRENT
                                ;* EXPECTED VALUES. THIS IS PERFORMED FOR ALL REGISTERS -- INCLUDING THE
                                ;* READ ONLY REGS -- IN ORDER TO MAKE SURE THAT EACH REGISTER ONLY RESPONDS
                                ;* TO ITS OWN ADDRESS.
                                ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110
                                ;*      PATTERN J: 000, 000, 001, 002, 004, 020, 040, 010
                                ;*
                                ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS F & J:
                                ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
                                ;*
                                ;--*************************************************************************
                                ;
                                ;       BGNTST
                                                                                        T5::
5119  023200
5120  023200  004737  005420           JSR     PC,INIDMV       ;ISSUE MASTER CLEAR TO INIT DMV
5121  023204  012702  002662           MOV     #PATF,R2        ;INIT PATTERN I POINTER
5122  023210  012703  002572           MOV     #REDDAT,R3      ;INIT PTR TO EXPECTED DATA AREA
5123  023214  112223     1$:   MOVB    (R2)+,(R3)+     ;MOV PATTERN F INTO REDDAT TABLE
5124  023216  020227  002672           CMP     R2,#PATG
5125  023222  103774                   BLO     1$
5126  023224  005001                   CLR     R1              ;INIT USYRT REG ADRS PTR FOR WRITING
5127  023226  005002                   CLR     R2              ;INIT INDEX FOR WRITING
5128  023230  016137  002552  023274 2$:    MOV     USYREG(R1),3$   ;SET USYRT REG ADRS
5129  023236  116237  003013  023276         MOVB    PATJ(R2),4$     ;SET DATA FOR WRITE
5130  023244  113762  023276  002572         MOVB    4$,REDDAT(R2)   ;SET EXPECTED DATA BYTE
5131  023252  023727  023274  120402         CMP     3$,#TDSRL       ;SEE IF WRITING TDSRL
5132  023260  001003                   BNE     10$             ;BR IF NOT
5133  023262  142737  000100  002601         BICB    #100,REDDAT+7   ;FIX EXPECTED USTAT VALUE
5134  023270  004537  003734    10$:  JSR     R5,WRITEI       ;WRITE BYTE INTO A USYRT REG
5135  023274  000000    3$:   .WORD   0               ;REG ADRS GOES HERE
5136  023276  000000    4$:   .WORD   0               ;DATA BYTE GOES HERE
5137  023300  005003                   CLR     R3              ;INIT USYRT REG ADRS PTR FOR READING
5138  023302  005004                   CLR     R4              ;INIT INDEX FOR READING
5139  023304  016337  002552  023316 5$:    MOV     USYREG(R3),6$   ;SET USYRT REG ADRS
5140  023312  004537  003610           JSR     R5,READI        ;READ A USYRT REG
5141  023316  000000    6$:   .WORD   0               ;REG ADRS GOES HERE
5142  023320  000000    7$:   .WORD   0               ;DATA BYTE READ IS RETURNED HERE
5143  023322  123764  023320  002572         CMPB    7$,REDDAT(R4)   ;SEE IF BYTE READ MATCHES EXPECTED BYTE
5144  023330  001420                   BEQ     8$              ;BR IF MATCH
5145  023332  010437  002412           MOV     R4,REGNUM       ;SET FAILING REG NO. FOR ERROR REPORT
5146  023336  005037  002400           CLR     GDATA           ;GET EXPECTED DATA
5147  023342  116437  002572  002400         MOVB    REDDAT(R4),GDATA
5148  023350  013737  023320  002402         MOV     7$,BDATA        ;GET ACTUAL DATA
5149                           ;REPORT REGISTER MISCOMPARE ERROR
5150  023356                           GEDF    EM66,ERR10
5151                                                   ;       'DEVICE FATAL'' ERROR # 44
5152  023356  104455                                                   TRAP    C$ERDF
5153  023360  000054                                                   .WORD   44
```

```
5154  023362  013741                                                    .WORD   EM66
5155  023364  017372                                                    .WORD   ERR10
5156  023366
5157  023366  104410                                                    TRAP    C$ESCAPE
5158  023370  000032                                                    .WORD   L10027-.
5159  023372  062703  000002  8$:     ADD     #2,R3       ;INCR USYRT REG ADRS PTR FOR READING
5160  023376  005204            INC     R4          ;INCR READ INDEX
5161  023400  020427  000010    CMP     R4,#10      ;SEE IF ALL REGS READ YET
5162  023404  002737            BLT     5$          ;BR IF NOT ALL READ YET
5163  023406  062701  000002    ADD     #2,R1       ;INCR USYRT REG ADRS PTR FOR WRITING
5164  023412  005202            INC     R2          ;INCR WRITE INDEX
5165  023414  020227  000010    CMP     R2,#10      ;SEE IF ALL REGS WRITTEN YET
5166  023420  002703            BLT     2$          ;BR IF NOT ALL WRITTEN YET
5167  023422                    ENDTST
5168  023422                                                    L10027:
5169  023422  104401                                                    TRAP    C$ETST
```

```
                                        .SBTTL  TEST 6 -- R/W BIT TEST OF PCSAR HIGH BYTE

                                        ;+*************************************************************************
                                        ;*
                                        ;*      TEST 6 -- R/W BIT TEST OF PCSAR HIGH BYTE
                                        ;*
                                        ;*
                                        ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN G.
                                        ;*
                                        ;*  PATTERN  G:  000,  001, 003, 004, 005, 007, 100, 101, 103, 104, 105,
                                        ;*  107, 000, 017, 027, 041, 200, 277, 103, 144, 115, 157, 000.
                                        ;*
                                        ;--************************************************************************
                                        ;
                                        ;       BGNTST
023424                                                                                  T6::
023424  004737  005420                          JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
023430  103003                                  BCC     30$             ;IF AN ERROR OCCURED,
023432                                          ERROR                   ;REPORT IT &
023432  104460                                                                          TRAP    C$ERROR
023434                                          ESCAPE  TST             ;  EXIT
023434  104410                                                                          TRAP    C$ESCAPE
023436  000056                                                                          .WORD   L10030-.

023440  012701  002672          30$:    MOV     #PATG,R1                ;POINT TO PATTERN TABLE
023444  012703  000025                  MOV     #<PATH-PATG-2>,R3       ;GET # OF ENTRIES IN TABLE
023450  012737  000005  002412          MOV     #5,REGNUM               ;ERROR INDEX FOR PCSARH

023456                          T7.LP:  BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
023456                                                                                  T6.1:
023456  104402                                                                          TRAP    C$BSUB

023460  111137  002376                  MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR 'STUREG'
023464  112137  002400                  MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
023470  012700  120405                  MOV     #PCSARH,R0      ;SPECIFY THE REGISTER BEING TESTED
023474  004737  004272                  JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
023500  103003                          BCC     10$             ;WAS AN ERROR FOUND?
023502                                  ERROR                   ;YES, REPORT IT AND
023502  104460                                                                          TRAP    C$ERROR
023504                                  ESCAPE  TST             ;     EXIT FROM THE TEST.  'CKLOOP' IS IMPLIED
023504  104410                                                                          TRAP    C$ESCAPE
023506  000006                                                                          .WORD   L10030-.

023510                          10$:    ENDSUB
023510                                                                          L10031:
023510  104403                                                                          TRAP    C$ESUB

023512  077317                          SOB     R3,T7.LP        ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
                                                                ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
023514                          ENDTST
023514                                                                          L10030:
023514  104401                                                                          TRAP    C$ETST
```

```
                                TEST 7 -- R/W BIT TEST OF S/AR REGISTER

                                .SBTTL  TEST 7 -- R/W BIT TEST OF S/AR REGISTER

                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 7 -- R/W BIT TEST OF S/AR REGISTER
                                ;*
                                ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
                                ;*
                                ;*  PATTERN  H:  125, 252, 000, 377, 000, 001, 002, 004, 010, 020, 040, 100,
                                ;*               200, 000, 377, 376, 375, 373, 367, 357, 337, 277, 177, 377,
                                ;*               000
                                ;*
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;
                                ;       BGNTST
                                                                                        T7::
023516
023516  004737  005420                  JSR     PC,INIDMV       ;INIT DMV & START UP THE MAINT. LOOP
023522  103003                          BCC     30$             ;IF AN ERROR OCCURED,
023524                                  ERROR                   ;REPORT IT &
023524  104460                                                                  TRAP    C$ERROR
023526                                  ESCAPE  TST             ;  EXIT
023526  104410                                                                  TRAP    C$ESCAPE
023530  000056                                                                  .WORD   L10032-.

023532  012701  002721          30$:    MOV     #PATH,R1            ;POINT TO PATTERN TABLE
023536  012703  000027                  MOV     #<PATI-PATH-2>,R3   ;GET # OF ENTRIES IN TABLE
023542  012737  000004  002412          MOV     #4,REGNUM           ;ERROR INDEX FOR S/AR BYTE

023550                          T8.LP:  BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
023550                                                                  T7.1:
023550  104402                                                                  TRAP    C$BSUB

023552  111137  002376                  MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR 'STUREG'
023556  112137  002400                  MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
023562  012700  120404                  MOV     #PCSARL,R0      ;SPECIFY THE REGISTER BEING TESTED
023566  004737  004272                  JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
023572  103003                          BCC     10$             ;WAS AN ERROR FOUND?
023574                                  ERROR                   ;YES, REPORT IT AND
023574  104460                                                                  TRAP    C$ERROR
023576                                  ESCAPE  TST             ;     EXIT FROM THE TEST.  'CKLOOP' IS IMPLIED
023576  104410                                                                  TRAP    C$ESCAPE
023600  000006                                                                  .WORD   L10032-.

023602                          10$:    ENDSUB
023602                                                                  L10033:
023602  104403                                                                  TRAP    C$ESUB

023604  077317                          SOB     R3,T8.LP        ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
                                                                ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
023606                          ENDTST
023606                                                                  L10032:
023606  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 8 -- R/W BIT TEST OF PCR REGISTER

                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 8 -- R/W BIT TEST OF PCR REGISTER
                                ;*
                                ;*  PATTERN  I  IS  LOADED  INTO  PCR  (HIGH)  AND THE DATA READ BACK AND
                                ;*  CHECKED.
                                ;*
                                ;*  PATTERN  I:  000, 041, 102, 143, 204, 245, 306, 347, 000, 001, 002,
                                ;*  004,  040,  100,  200,  000, 346, 345, 343, 307, 247, 147, 347, 242,
                                ;*  105, 347, 010, 020, 357, 357, 030, 027, 377.
                                ;*
                                ;*
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;
                                ;       BGNTST
023610                                                                          T8::
023610  004737  005420                JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
023614  103003                        BCC     30$             ;IF AN ERROR OCCURED,
023616                                ERROR                   ;REPORT IT &
023616  104460                                                                    TRAP    C$ERROR
023620                                ESCAPE  TST             ;  EXIT
023620  104410                                                                    TRAP    C$ESCAPE
023622  000056                                                                    .WORD   L10034-.

023624  012701  002752        30$:    MOV     #PATI,R1            ;POINT TO PATTERN TABLE
023630  012703  000037                MOV     #<PATJ-PATI-2>,R3   ;GET # OF ENTRIES IN TABLE
023634  012737  000006  002412        MOV     #6,REGNUM           ;ERROR INDEX FOR PCR REGISTER

023642                        T9.LP:  BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
023642                                                                  T8.1:
023642  104402                                                                  TRAP    C$BSUB

023644  111137  002376                MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR 'STUREG'
023650  112137  002400                MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
023654  012700  120407                MOV     #PCR,R0         ;SPECIFY THE REGISTER BEING TESTED
023660  004737  004272                JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
023664  103003                        BCC     10$             ;WAS AN ERROR FOUND?
023666                                ERROR                   ;YES, REPORT IT AND
023666  104460                                                                    TRAP    C$ERROR
023670                                ESCAPE  TST             ;     EXIT FROM THE TEST.  'CKLOOP' IS IMPLIED
023670  104410                                                                    TRAP    C$ESCAPE
023672  000006                                                                    .WORD   L10034-.

023674                        10$:    ENDSUB
023674                                                                  L10035:
023674  104403                                                                  TRAP    C$ESUB

023676  077317                        SOB     R3,T9.LP        ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
                                                              ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
023700                        ENDTST
023700                                                                  L10034:
023700  104401                                                                  TRAP    C$ETST
```

```
                                        .SBTTL  TEST 9 -- R/W BIT TEST OF TDSR REGISTER'S HIGH BYTE

                                        ;++*********************************************************************
                                        ;*
                                        ;*       TEST 9 -- R/W BIT TEST OF TDSR REGISTER'S HIGH BYTE
                                        ;*
                                        ;*  PATTERN  K  IS  LOADED  INTO  TDSR  (HIGH) AND THE DATA READ BACK IS
                                        ;*  COMPARED  AGAINST PATTERN L. (UNPREDICTABLE BITS ARE MASKED OFF TO 0
                                        ;*  WHEN READING FOR COMPARISON.)
                                        ;*
                                        ;*  PATTERN  K:  000,  377, 376, 375, 373, 376, 177, 377, 000, 001, 002,
                                        ;*  004, 010, 200, 125, 252, 000.
                                        ;*
                                        ;*  PATTERN  L:  000,  017, 016, 015, 013, 016, 017, 017, 000, 001, 002,
                                        ;*  004, 010, 000, 005, 012, 000.
                                        ;*
                                        ;*  NOTE  THAT  THE  UNDEFINED BITS (12, 13, & 14) ARE MASKED OFF TO 0'S
                                        ;*  FOR THE COMPARISON. ALSO THAT BIT 15 IS A READ/ONLY BIT AND CAN'T BE
                                        ;*  SET -- THEREFORE SHOULD ALWAYS BE READ AS A 0 BY THIS TEST.
                                        ;*
                                        ;--*********************************************************************
                                        ;
                                        ;       BGNTST
 023702                                                                                  T9::
 023702  004737  005420                         JSR     PC,INIDMV          ;INIT DMV/VIA & START UP MAINT. LOOP
 023706  103003                                 BCC     30$                ;IF AN ERROR OCCURED,
 023710                                         ERROR                      ;REPORT IT &
 023710  104460                                                                                  TRAP    C$ERROR
 023712                                         ESCAPE  TST                ;  EXIT
 023712  104410                                                                                  TRAP    C$ESCAPE
 023714  000114                                                                                  .WORD   L10036-.

 023716  012701  003023                 30$:    MOV     #PATK,R1              ;POINT TO PATTERN TABLE
 023722  012703  000017                         MOV     #<PATL-PATK-2>,R3     ;GET # OF ENTRIES IN TABLE
 023726  012702  003044                         MOV     #PATL,R2              ;POINT TO 'EXPECTED'' DATA PATTERN TABLE

 023732                                 T10.LP: BGNSUB                     ;THE SUBTEST ONLY TESTS THE ONE PATTERN
 023732                                                                                  T9.1:
 023732  104402                                                                                  TRAP    C$BSUB

 023734  112137  002376                         MOVB    (R1)+,TDATA        ;SETUP TEST DATA BYTE
 023740  112237  002400                         MOVB    (R2)+,GDATA        ;SETUP EXPECTED DATA BYTE

 023744  004537  003722                         JSR     R5,WRITE           ;WRITE TO DMV-11
 023750  120403                                 .WORD   TDSRH              ; DMV-11 ADDRESS WRITTEN TO
 023752  002376                                 .WORD   TDATA              ; LOCATION OF DATA WRITTEN

 023754  004537  003476                         JSR     R5,READ            ;READ FROM DMV-11
 023760  120403                                 .WORD   TDSRH              ; DMV-11 ADDRESS READ FROM
 023762  002402                                 .WORD   BDATA              ; LOCATION WHERE READ DATA IS PUT

 023764  042737  177760  002402                 BIC     #177760,BDATA      ;MASK OUT 'DON'T CARE'' BITS
 023772  023737  002400  002402                 CMP     GDATA,BDATA        ;READ DATA = EXPECTED DATA ?
 024000  001411                                 BEQ     10$                ;WAS AN ERROR FOUND?

 024002  012737  000003  002412                 MOV     #3,REGNUM          ;YES: SET UP REGISTER NUMBER
```

```
5384  024010                  GEDF    EM25,ERR7A      ;REPORT IT AND
5385                                                  ;       'DEVICE FATAL'' ERROR # 45
5386  024010  104455                                                  TRAP    C$ERDF
5387  024012  000055                                                  .WORD   45
5388  024014  013366                                                  .WORD   EM25
5389  024016  017252                                                  .WORD   ERR7A
5390  024020                  ESCAPE  TST             ;  EXIT FROM THE TEST.  ''CKLOOP'' IS IMPLIED
5391  024020  104410                                                  TRAP    C$ESCAPE
5392  024022  000006                                                  .WORD   L10036-.
5393
5394  024024          10$:    ENDSUB
5395  024024                                                          L10037:
5396  024024  104403                                                  TRAP    C$ESUB
5397
5398  024026  077337          SOB     R3,T10.LP       ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
5399                                                  ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
5400  024030          ENDTST
5401  024030                                                          L10036:
5402  024030  104401                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 10 -- R/W BIT TEST OF TXDB REGISTER

                                ;++*************************************************************************
                                ;*
                                ;*      TEST 10 -- R/W BIT TEST OF TXDB REGISTER
                                ;*
                                ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
                                ;*
                                ;*  PATTERN  H:  000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
                                ;*  376, 375, 373, 367, 357, 337, 277, 177, 377, 000
                                ;*
                                ;--*************************************************************************
                                ;
                                ;       BGNTST
                                                                                        T10::
024032
024032  004737  005420          ;       JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
024036  103003                  ;       BCC     30$             ;IF AN ERROR OCCURED,
024040                          ;       ERROR                   ;REPORT IT &
024040  104460                                                                  TRAP    C$ERROR
024042                          ;       ESCAPE  TST             ;  EXIT
024042  104410                                                                  TRAP    C$ESCAPE
024044  000056                                                                  .WORD   L10040-.

024046  012701  002721          30$:    MOV     #PATH,R1            ;POINT TO PATTERN TABLE
024052  012703  000027                  MOV     #<PATI-PATH-2>,R3   ;GET # OF ENTRIES IN TABLE
024056  012737  000002  002412          MOV     #2,REGNUM           ;ERROR INDEX FOR TXDB REGISTER

024064                          T11.LP: BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
024064                                                                      T10.1:
024064  104402                                                              TRAP    C$BSUB

024066  111137  002376                  MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR 'STUREG'
024072  112137  002400                  MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR 'STUREG'
024076  012700  120402                  MOV     #TDSRL,R0       ;SPECIFY THE REGISTER BEING TESTED
024102  004737  004272                  JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
024106  103003                  BCC     10$             ;WAS AN ERROR FOUND?
024110                          ERROR                   ;YES, REPORT IT AND
024110  104460                                                              TRAP    C$ERROR
024112                          ESCAPE  TST             ;       EXIT FROM THE TEST.  'CKLOOP' IS IMPLIED
024112  104410                                                              TRAP    C$ESCAPE
024114  000006                                                              .WORD   L10040-.

024116                          10$:    ENDSUB
024116                                                                  L10041:
024116  104403                                                          TRAP    C$ESUB

024120  077317                  SOB     R3,T11.LP       ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
                                                        ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
024122                          ENDTST
024122                                                                  L10040:
024122  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  TEST 11 -- PSEUDO R/W BIT TEST OF RXDB

                                ;++*************************************************************************
                                ;*
                                ;*      TEST 11 -- PSEUDO R/W BIT TEST OF RXDB
                                ;*
                                ;*  WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
                                ;*  PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
                                ;*
                                ;*  PATTERN H:  000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
                                ;*  376, 375, 373, 367, 357, 337, 277, 177, 377, 000
                                ;*
                                ;--*************************************************************************
                                ;
                                ;       BGNTST
                                                                                        T11::
024124                          
024124  004737  005420          
024130  103003                  
024132                          
024132  104460                  
024134                          
024134  104410                  
024136  000040                  
                                
024140  012701  002721          
024144  012703  000027          
                                
024150  012137  024162          
                                
024154  004537  003734          
024160  120400                  
024162  000000                  
                                
024164  004537  003610          
024170  120400                  
024172  000000                  
                                
024174  077313                  
                                
024176                          
024176                          
024176  104401                  
```

```
                                        JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
                                        BCC     30$             ;IF AN ERROR OCCURED,
                                        ERROR                   ;REPORT IT &
                                                                                TRAP    CSERROR
                                        ESCAPE  TST             ;  EXIT
                                                                                TRAP    CSESCAPE
                                                                                .WORD   L10042-.

                                30$:    MOV     #PATH,R1            ;POINT TO PATTERN TABLE
                                        MOV     #<PATI-PATH-2>,R3   ;GET # OF ENTRIES IN TABLE

                                20$:    MOV     (R1)+,2$

                                        JSR     R5,WRITEI       ;WRITE TO DMV-11
                                        .WORD   RDSRL           ; DMV-11 ADDRESS WRITTEN TO
                                2$:     .WORD   0               ; ACTUAL DATA WRITTEN

                                        JSR     R5,READI        ;READ FROM DMV-11
                                        .WORD   RDSRL           ; DMV-11 ADDRESS READ FROM
                                        .WORD   0               ; READ DATA IS PUT HERE

                                        SOB     R3,20$          ;IF MORE DATA, LOOP BACK TO WRITE/READ IT,
                                                                ;  ELSE, FALL OUT OF LOOP AND TEST
                                ENDTST
                                                                                L10042:
                                                                                TRAP    CSETST
```

```
5496                            .SBTTL  TEST 12 -- PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE
5497
5498                            ;++*************************************************************************
5499                            ;*
5500                            ;*      TEST 12 -- PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE
5501                            ;*
5502                            ;*  WRITE, READ (BUT NO COMPARE) OF EACH WORD IN DATA PATTERN H. THIS IS
5503                            ;*  PRIMARILY TO PROVIDE A SCOPE LOOP FUNCTION ON THIS REGISTER.
5504                            ;*
5505                            ;*  PATTERN H:  000, 001, 002, 004, 010, 020, 040, 100, 200, 000, 377,
5506                            ;*  376, 375, 373, 367, 357, 337, 277, 177, 377, 000
5507                            ;*
5508                            ;--*************************************************************************
5509                            ;
5510                            ;       BGNTST
5511  024200                                                                    T12::
5512  024200  004737  005420            JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
5513  024204  103003                    BCC     30$             ;IF AN ERROR OCCURED,
5514  024206                            ERROR                   ;REPORT IT &
5515  024206  104460                                                            TRAP    C$ERROR
5516  024210                            ESCAPE  TST             ;  EXIT
5517  024210  104410                                                            TRAP    C$ESCAPE
5518  024212  000040                                                            .WORD   L10043-.
5519
5520  024214  012701  002721    30$:    MOV     #PATH,R1            ;POINT TO PATTERN TABLE
5521  024220  012703  000027            MOV     #<PATI-PATH-2>,R3   ;GET # OF ENTRIES IN TABLE
5522
5523  024224  012137  024236    20$:    MOV     (R1)+,2$
5524
5525  024230  004537  003734            JSR     R5,WRITEI       ;WRITE TO DMV-11
5526  024234  120401                    .WORD   RDSRH           ; DMV-11 ADDRESS WRITTEN TO
5527  024236  000000            2$:     .WORD   0               ; ACTUAL DATA WRITTEN
5528
5529  024240  004537  003610            JSR     R5,READI        ;READ FROM DMV-11
5530  024244  120401                    .WORD   RDSRH           ; DMV-11 ADDRESS READ FROM
5531  024246  000000                    .WORD   0               ; READ DATA IS PUT HERE
5532
5533  024250  077313                    SOB     R3,20$          ;IF MORE DATA, LOOP BACK TO WRITE/READ IT,
5534                                                            ;  ELSE, FALL OUT OF LOOP AND TEST
5535  024252                    ENDTST
5536  024252                                                                  L10043:
5537  024252  104401                                                                   TRAP    C$ETST
```

```
                                        .SBTTL  TEST 13 -- NULL CLOCK TEST

                                        ;++*******************************************************************
                                        ;*
                                        ;*      TEST 13 -- NULL CLOCK TEST
                                        ;*
                                        ;* FIRST, A MASTER CLEAR IS DONE TO INIT THE DMV. THEN, THE T1 TIMER ON THE
                                        ;* VIA CHIP IS PROGRAMMED FOR SQUARE WAVE CLOCK GENERATION ON PB7 (BIT 7
                                        ;* OF VIA OUTPUT REG B), WITH A BAUD RATE = 56 KBAUD. THIS IS THE MODE OF
                                        ;* VIA OPERATION WHICH IS USED TO GENERATE THE NULL CLOCK. THEN, THE PROGRAM
                                        ;* SCANS ORB REPEATEDLY TO MONITOR THE NULL CLOCK BIT, IN THE FOLLOWING
                                        ;* SEQUENCE :
                                        ;* - THE PROGRAM REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE, AND
                                        ;*   IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC (A GROSS TIMEOUT
                                        ;*   INTERVAL), AN ERROR IS REPORTED. (AT 56 KBAUD, THE CLOCK SHOULD
                                        ;*   HAVE A PERIOD OF ABOUT 18 MICRO-SEC.)
                                        ;* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 0 STATE,
                                        ;*   AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC, AN ERROR IS
                                        ;*   REPORTED.
                                        ;* - THE PROGRAM NEXT REPEATEDLY CHECKS THE NULL CLOCK BIT FOR THE 1 STATE
                                        ;*   AGAIN, AND IF IT IS NOT FOUND WITHIN SEVERAL HUNDRED MICRO-SEC,
                                        ;*   AN ERROR IS REPORTED.
                                        ;*
                                        ;--*******************************************************************
                                        ;
                                        ;       BGNTST
024254                                                                          T13::
024254  004737  005420                          JSR     PC,INIDMV       ;INIT DMV11
024260  004537  003734                          JSR     R5,WRITEI       ;LOAD T1C-L, T1L-L FOR 56K BAUD
024264  120004                                  VIAT1A
024266  000017                                  15.
024270  004537  003734                          JSR     R5,WRITEI       ;LOAD T1C-H, T1L-H, START CLOCK
024274  120005                                  VIAT1B
024276  000000                                  000
                                        ;--------------------------------------------------------------
                                        ;WAIT FOR NULCLK BIT TO BE SET TO 1
                                        ;--------------------------------------------------------------
024300  012737  170000  002602                  MOV     #170000,REG0    ;INIT PROGRAM TIMER
024306  004537  003610                  1$:     JSR     R5,READI        ;READ VIA ORB
024312  120000                                  VIAORB
024314  000000                          2$:     .WORD   0
024316  132737  000200  024314                  BITB    #NULCLK,2$      ;SEE IF CLOCK BIT SET
024324  001025                                  BNE     3$              ;BR IF SET
024326  005237  002602                          INC     REG0            ;INCR TIMER
024332  001365                                  BNE     1$              ;BR IF TIMER DID NOT TIME OUT
                                                                        ; (TIME OUT = SEVERAL HUNDRED MICRO-SEC)
024334  012737  000000  002412                  MOV     #0,REGNUM       ;SET VIA REG NO. FOR PRINTOUT
024342  012737  000200  002400                  MOV     #NULCLK,GDATA   ;GET EXPECTED DATA
024350  013737  024314  002402                  MOV     2$,BDATA        ;GET ACTUAL DATA
024356  042737  000177  002402                  BIC     #177,BDATA      ;CLEAR UNUSED BITS
                                        ;REPORT NULL CLK STUCK AT 0
024364                                          GEDF    EM5,ERR11
                                                                        ;       "DEVICE FATAL" ERROR # 46
024364  104455                                                                          TRAP    C$ERDF
024366  000056                                                                          .WORD   46
024370  013252                                                                          .WORD   EM5
```

```
5594  024372  017546                                                          .WORD   ERR11
5595  024374
5596  024374  104410                                                          TRAP    C$ESCAPE
5597  024376  000176                                                          .WORD   L10044-.
5598                          ;-------------------------------------------------------------
5599                          ;WAIT FOR NULCLK BIT TO BE CLEARED TO 0
5600                          ;-------------------------------------------------------------
5601  024400  012737  170000  002602  3$:     MOV     #170000,REG0    ;INIT PROGRAM TIMER
5602  024406  004537  003610          4$:     JSR     R5,READI        ;READ VIA ORB
5603  024412  120000                          VIAORB
5604  024414  000000                  5$:     .WORD   0
5605  024416  132737  000200  024414          BITB    #NULCLK,5$      ;SEE IF CLOCK BIT CLEARED
5606  024424  001425                          BEQ     6$              ;BR IF CLEARED
5607  024426  005237  002602                  INC     REG0            ;INCR TIMER
5608  024432  001365                          BNE     4$              ;BR IF TIMER DID NOT TIME OUT
5609                                                                  ; (TIME OUT = SEVERAL HUNDRED MICRO-SEC)
5610  024434  012737  000000  002412          MOV     #0,REGNUM       ;SET VIA REG NO. FOR PRINTOUT
5611  024442  012737  000000  002400          MOV     #000,GDATA      ;GET EXPECTED DATA
5612  024450  013737  024414  002402          MOV     5$,BDATA        ;GET ACTUAL DATA
5613  024456  042737  000177  002402          BIC     #177,BDATA      ;CLEAR UNUSED BITS
5614                          ;REPORT NULL CLK STUCK AT 1
5615  024464                                  GEDF    EM6,ERR11
5616                                                  ;       'DEVICE FATAL'' ERROR # 47
5617  024464  104455                                                          TRAP    C$ERDF
5618  024466  000057                                                          .WORD   47
5619  024470  013302                                                          .WORD   EM6
5620  024472  017546                                                          .WORD   ERR11
5621  024474
5622  024474  104410                                                          TRAP    C$ESCAPE
5623  024476  000076                                                          .WORD   L10044-.
5624                          ;-------------------------------------------------------------
5625                          ;WAIT FOR NULCLK BIT TO BE SET TO 1 AGAIN
5626                          ;-------------------------------------------------------------
5627  024500  012737  170000  002602  6$:     MOV     #170000,REG0    ;INIT PROGRAM TIMER
5628  024506  004537  003610          7$:     JSR     R5,READI        ;READ VIA ORB
5629  024512  120000                          VIAORB
5630  024514  000000                  8$:     .WORD   0
5631  024516  132737  000200  024514          BITB    #NULCLK,8$      ;SEE IF CLOCK BIT SET
5632  024524  001023                          BNE     9$              ;BR IF SET
5633  024526  005237  002602                  INC     REG0            ;INCR TIMER
5634  024532  001365                          BNE     7$              ;BR IF TIMER DID NOT TIME OUT
5635                                                                  ; (TIME OUT = SEVERAL HUNDRED MICRO-SEC)
5636  024534  012737  000000  002412          MOV     #0,REGNUM       ;SET VIA REG NO. FOR PRINTOUT
5637  024542  012737  000200  002400          MOV     #NULCLK,GDATA   ;GET EXPECTED DATA
5638  024550  013737  024514  002402          MOV     8$,BDATA        ;GET ACTUAL DATA
5639  024556  042737  000177  002402          BIC     #177,BDATA      ;CLEAR UNUSED BITS
5640                          ;REPORT NULL CLK STUCK AT 0
5641  024564                                  GEDF    EM5,ERR11
5642                                                  ;       'DEVICE FATAL'' ERROR # 48
5643  024564  104455                                                          TRAP    C$ERDF
5644  024566  000060                                                          .WORD   48
5645  024570  013252                                                          .WORD   EM5
5646  024572  017546                                                          .WORD   ERR11
5647  024574                  9$:
5648  024574                  ENDTST
5649  024574                                                          L10044:
```

```
  5650  024574  104401                                    TRAP    C$ETST
```

```
                        .SBTTL  TEST 14 -- BCP TX RESET W/IDLE = 0

                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        ;*
                        ;*      TEST 14 -- BCP TX RESET W/IDLE = 0
                        ;*
                        ;*  THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL'' (BCP) WITH IDLE
                        ;*  SET TO ZERO AND A 125 SYNC CHARACTER IS LOADED INTO S/AR. A 226 SYNC
                        ;*  CHARACTER IS LOADED INTO TXDB SO THAT THE SOURCE OF SYNC CHARACTERS
                        ;*  CAN BE LATER DETERMINED. THE VALID STATE OF THE USYRT REGISTERS IS
                        ;*  READ AND CHECKED. TXE IS ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
                        ;*   THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
                        ;*  TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
                        ;*   (TXBE SHOULD GO HIGH; AT THIS TIME THE S/AR'S SYNC CHARACTER SHOULD
                        ;*  BE LOADED INTO TXSO AND TSOM IS AGAIN SET -- DRIVING TXBE LOW.)
                        ;*   THREE SYNC CHARACTERS ARE SENT/RECEIVED: THE FIRST TWO SYNCHRONIZE
                        ;*  THE RECEIVER, THE THIRD IS DIRECTLY READ (STRIP SYNC IS OFF) AND
                        ;*  COMPARED AGAINST 125 (THE S/AR SYNC CHARACTER).
                        ;*   IF VALUE READ IS 226, THEN TXDB PROVIDED THE SYNC (IE: ERROR).
                        ;*   THE USYRT IS THEN RESET AND REGISTERS ARE AGAIN READ AND CHECKED.
                        ;*   THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
                        ;*  ONE MARK AND THREE SYNC CHARACTERS (FROM THE S/AR) IS TRANSMITTED.
                        ;*   ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
                        ;*  SEQUENCE.
                        ;*
                        ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        ;
                        ;       BGNTST
024576                                                                  T14::

024576  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

024602  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
024606  043525                  DDCMP!NOCHK!125         ;SET DDCMP,NO CHECK,S/AR(SYNC)=125
024610  000000                  0                       ;USE 8 BIT CHARS
024612  103003                  BCC     .+8.            ;BR IF NO ERROR
024614                          ERROR                   ;REPORT STACKED ERROR
024614  104460                                                          TRAP    C$ERROR
024616                          ESCAPE  TST             ;SKIP TO END OF TEST
024616  104410                                                          TRAP    C$ESCAPE
024620  000106                                                          .WORD   L10045-.

024622  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
024626  000001                  TSOM                    ;AND KNOCK DOWN TBMT
024630  000007                  7.

024632  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
024636  000001                  TSOM                    ;AND KNOCK DOWN TBMT
024640  000010                  8.

024642  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 3RD SYNC CHARACTER (125)
024646  000001                  TSOM                    ;AND KNOCK DOWN TBMT
024650  000010                  8.

024652  004537  011364          JSR     R5,RCV1ST       ;CLOCK AND RCV TSOM
024656  000000                  0
```

```
5707  024660  103003          BCC     .+8.            ;BR IF NO ERROR
5708  024662                  ERROR                   ;REPORT STACKED ERROR
5709  024662  104460                                                  TRAP    C$ERROR
5710  024664                  ESCAPE  TST             ;SKIP TO END OF TEST
5711  024664  104410                                                  TRAP    C$ESCAPE
5712  024666  000040                                                  .WORD   L10045-.
5713
5714  024670  004537  010110  JSR     R5,RXCHAR       ;READ AND CHECK FOR S/AR 125 SYNC
5715  024674  000125          125                     ; AND CLOCK IN NEXT ONE.
5716  024676  000000          0
5717  024700  000010          8.
5718  024702  103003          BCC     .+8.            ;BR IF NO ERROR
5719  024704                  ERROR                   ;REPORT STACKED ERROR
5720  024704  104460                                                  TRAP    C$ERROR
5721  024706                  ESCAPE  TST             ;SKIP TO END OF TEST
5722  024706  104410                                                  TRAP    C$ESCAPE
5723  024710  000016                                                  .WORD   L10045-.
5724
5725  024712  004537  005126  JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
5726  024716  103003          BCC     .+8.            ;BR IF NO ERROR
5727  024720                  ERROR                   ;REPORT STACKED ERROR
5728  024720  104460                                                  TRAP    C$ERROR
5729  024722                  ESCAPE  TST
5730  024722  104410                                                  TRAP    C$ESCAPE
5731  024724  000002                                                  .WORD   L10045-.
5732
5733  024726                  ENDTST
5734  024726                                                  L10045:
5735  024726  104401                                                  TRAP    C$ETST
```

```
                        .SBTTL  TEST 15 -- BCP TX RESET W/IDLE = 1

                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        ;*
                        ;*      TEST 15 -- BCP TX RESET W/IDLE = 1
                        ;*
                        ;*  THE USYRT IS INITIALIZED FOR 'BYTE-CONTROL PROTOCOL'' (BCP) WITH IDLE
                        ;*  SET TO ONE AND A 226 SYNC CHARACTERS LOADED INTO S/AR AND TXDB.
                        ;*  THE VALID STATE OF THE USYRT REGISTERS IS READ AND CHECKED. TXE IS
                        ;*  ASSERTED TO ENABLE THE TRANSMITTER LOGIC.
                        ;*   THEN, TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE OBSERVING
                        ;*  TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
                        ;*   (TXBE SHOULD GO HIGH; AT THIS TIME THE TXDB SYNC CHARACTER SHOULD
                        ;*  BE LOADED INTO TXSO AND TSOM IS AGAIN SET -- DRIVING TXBE LOW.)
                        ;*   AFTER THE RECEIVER IS SYNCHRONIZED (TWO SYNC CHARACTERS), TXDB IS
                        ;*  LOADED WITH A 125 AND, WITH TSOM STILL = 1 (SYNC SOURCE = TXDB), THE
                        ;*  USYRT IS AGAIN CLOCKED.
                        ;*   AT THIS POINT, IF THE IDLE BIT WORKED, THE VALUE 125 WILL BE READ
                        ;*  BY THE RECEIVER. OTHERWISE A 226 WILL BE READ, INDICATING TXDB WASN'T
                        ;*  PROVIDING THE SYNC CHARACTERS.
                        ;*   WHEN TXBE GOES HIGH AGAIN, THE USYRT IS RESET.
                        ;*   ALL REGISTERS ARE AGAIN READ AND CHECKED.
                        ;*   THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY
                        ;*  ONE MARK AND THREE SYNCS (226,226,125 FROM TXDB) ARE TRANSMITTED.
                        ;*   ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN THE
                        ;*  SEQUENCE.
                        ;*
                        ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        ;
                            BGNTST
024730                                                          T15::

024730  004737  005420      JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

024734  004537  007400      JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
024740  047626              DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
024742  000000              0                       ;USE 8 BIT CHARS
024744  103003              BCC     .+8.            ;BR IF NO ERROR
024746                      ERROR                   ;REPORT STACKED ERROR
024746  104460                                                      TRAP    C$ERROR
024750                      ESCAPE  TST             ;SKIP TO END OF TEST
024750  104410                                                      TRAP    C$ESCAPE
024752  000136                                                      .WORD   L10046-.

024754  004537  007676      JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
024760  000226              SYNCH
024762  000007              7.
024764  103003              BCC     .+8.            ;BR IF NO ERROR
024766                      ERROR                   ;REPORT STACKED ERROR
024766  104460                                                      TRAP    C$ERROR
024770                      ESCAPE  TST             ;SKIP TO END OF TEST
024770  104410                                                      TRAP    C$ESCAPE
024772  000116                                                      .WORD   L10046-.

024774  004537  007676      JSR     R5,TXCHAR       ;LOAD 125, TX 2ND SYNCH
025000  000125              125
```

```
5792  025002  000010                    8.
5793  025004  103003                    BCC     .+8.            ;BR IF NO ERROR
5794  025006                            ERROR                   ;REPORT STACKED ERROR
5795  025006  104460                                                    TRAP    C$ERROR
5796  025010                            ESCAPE  TST             ;SKIP TO END OF TEST
5797  025010  104410                                                    TRAP    C$ESCAPE
5798  025012  000076                                                    .WORD   L10046-.
5799
5800  025014  004537  007676            JSR     R5,TXCHAR       ;LOAD 377, TX 125
5801  025020  000377                    377
5802  025022  000010                    8.
5803  025024  103003                    BCC     .+8.            ;BR IF NO ERROR
5804  025026                            ERROR                   ;REPORT STACKED ERROR
5805  025026  104460                                                    TRAP    C$ERROR
5806  025030                            ESCAPE  TST             ;SKIP TO END OF TEST
5807  025030  104410                                                    TRAP    C$ESCAPE
5808  025032  000056                                                    .WORD   L10046-.
5809
5810  025034  004537  011364            JSR     R5,RCV1ST       ;CLOCK AND RCV 125
5811  025040  000000                    0
5812  025042  103003                    BCC     .+8.            ;BR IF NO ERROR
5813  025044                            ERROR                   ;REPORT STACKED ERROR
5814  025044  104460                                                    TRAP    C$ERROR
5815  025046                            ESCAPE  TST             ;SKIP TO END OF TEST
5816  025046  104410                                                    TRAP    C$ESCAPE
5817  025050  000040                                                    .WORD   L10046-.
5818
5819  025052  004537  010110            JSR     R5,RXCHAR       ;READ AND CHECK FOR TXDB 125 SYNC
5820  025056  000125                    125
5821  025060  000000                    0
5822  025062  000010                    8.
5823  025064  103003                    BCC     .+8.            ;BR IF NO ERROR
5824  025066                            ERROR                   ;REPORT STACKED ERROR
5825  025066  104460                                                    TRAP    C$ERROR
5826  025070                            ESCAPE  TST             ;SKIP TO END OF TEST
5827  025070  104410                                                    TRAP    C$ESCAPE
5828  025072  000016                                                    .WORD   L10046-.
5829
5830  025074  004537  005126            JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
5831  025100  103003                    BCC     .+8.            ;BR IF NO ERROR
5832  025102                            ERROR                   ;REPORT STACKED ERROR
5833  025102  104460                                                    TRAP    C$ERROR
5834  025104                            ESCAPE TST              ;SKIP TO END OF TEST
5835  025104  104410                                                    TRAP    C$ESCAPE
5836  025106  000002                                                    .WORD   L10046-.
5837  025110                    ENDTST
5838  025110                                                    L10046:
5839  025110  104401                                                    TRAP    C$ETST
5840
```

```
                        .SBTTL  TEST 16 -- BCP TX UNDERRUN W/TSOM TERMINATION

                        ;++*************************************************************************
                        ;*
                        ;*      TEST 16 -- BCP TX UNDERRUN W/TSOM TERMINATION
                        ;*
                        ;*  THE  USYRT  IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY
                        ;*  CONTROLLED  UNTIL  TWO  SYNC CHARACTERS AND ONE DATA CHARACTER (000)
                        ;*  HAVE  BEEN  TRANSMITTED.  AT  THIS TIME WHEN TXBE IS ASSERTED BY THE
                        ;*  USYRT,  NO  DATA  IS  LOADED  INTO  TXDB  --  FORCING  AN  UNDER RUN
                        ;*  CONDITION.  TXU  AND  TERR  ARE  CHECKED BOTH BEFORE AND AFTER THEIR
                        ;*  EXPECTED  ASSERTIONS.  AFTER  THE  FIRST  NON-DATA  CHARACTER (WHICH
                        ;*  SHOULD  BE  THE  MARK CHARACTER) HAS BEEN STARTED, IDLE IS SET TO 0.
                        ;*  THIS SHOULD FORCE THE NEXT NON-DATA CHARACTER TO BE A SYNC CHARACTER
                        ;*  FROM  S/AR.  WHILE THIS SYNC CHARACTER IS BEING TRANSMITTED, TSOM IS
                        ;*  ASSERTED  (CLEARING TXU AND TERR) -- IDLE IS LEFT AT 0. TXBE IS THEN
                        ;*  CYCLED  THROUGH AT LEAST ONE MORE SYNC CHARACTER AND THE TEST IS
                        ;*  ABORTED. ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS
                        ;*  WITHIN THE SEQUENCE.
                        ;*      NOTE: BITS SHIFT OUT OF TX LSB FIRST.
                        ;*
                        ;--*************************************************************************
                        ;
                        ;       BGNTST
                                                                                        T16::
025112                  ;
025112  004737  005420  ;       JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
                        ;
025116  004537  007400  ;       JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
025122  047626          ;DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
025124  040000          ;NORXEN                 ;USE 8 BIT CHARS, RECEIVER DISABLED
025126  103003          ;       BCC     .+8.            ;BR IF NO ERROR
025130                  ;       ERROR                   ;REPORT STACKED ERROR
025130  104460                                                          TRAP    C$ERROR
025132                  ;       ESCAPE  TST             ;SKIP TO END OF TEST
025132  104410                                                          TRAP    C$ESCAPE
025134  000462                                                          .WORD   L10047-.
                        ;
025136  004537  007116  ;       JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'SYNCH'
025142  000000          ;0                      ; CHARACTER (SHOULD BE 0)
025144  103003          ;       BCC     .+8.            ;BR IF NO ERROR
025146                  ;       ERROR                   ;REPORT STACKED ERROR
025146  104460                                                          TRAP    C$ERROR
025150                  ;       ESCAPE  TST             ;AND EXIT TEST
025150  104410                                                          TRAP    C$ESCAPE
025152  000444                                                          .WORD   L10047-.
                        ;
025154  004537  007256  ;       JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF 'SYNCH' CHARACTER
025160  000007          ;7.                     ; (OFF OF TSO BIT)
025162  000151          ;151                    ; EXPECTED BIT SEQUENCE (0010110)
025164  103003          ;       BCC     .+8.            ;BR IF NO ERROR
025166                  ;       ERROR                   ;REPORT STACKED ERROR
025166  104460                                                          TRAP    C$ERROR
025170                  ;       ESCAPE  TST             ;AND EXIT TEST
025170  104410                                                          TRAP    C$ESCAPE
025172  000424                                                          .WORD   L10047-.
                        ;
```

```
5897  025174  004537  003734          JSR     R5,WRITEI       ;LOAD 1ST DATA CHARACTER (000)
5898  025200  120402                  TDSRL
5899  025202  000000                  000
5900
5901  025204  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
5902  025210  000000                  000
5903  025212  000000                  0
5904
5905  025214  004537  007256          JSR     R5,SERIAL       ;READ 2ND SYNCH CHARACTER VIA TSO
5906  025220  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
5907  025222  000151                  151                     ; EXPECTED BIT SEQUENCE (010010110)
5908  025224  103003                  BCC     .+8.            ;BR IF NO ERROR
5909  025226                          ERROR                   ;REPORT STACKED ERROR
5910  025226  104460                                                          TRAP    C$ERROR
5911  025230                          ESCAPE  TST             ;AND EXIT TEST
5912  025230  104410                                                          TRAP    C$ESCAPE
5913  025232  000364                                                          .WORD   L10047-.
5914                          ;-------------------------------------------------------------
5915                          ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
5916                          ;-------------------------------------------------------------
5917  025234  004537  005432          JSR     R5,CKUSTS       ;CHECK USYRT STATUS
5918  025240  000114                  TXACT!TBMT!TSO          ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
5919  025242  103003                  BCC     .+8.            ;BR IF NO ERROR
5920  025244                          ERROR                   ;REPORT STACKED ERROR
5921  025244  104460                                                          TRAP    C$ERROR
5922  025246                          ESCAPE  TST             ;SKIP TO END OF TEST
5923  025246  104410                                                          TRAP    C$ESCAPE
5924  025250  000346                                                          .WORD   L10047-.
5925
5926  025252  004537  003610          JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
5927  025256  120403                  TDSRH                   ; TERR BIT IS IN TDSRH
5928  025260  000000          10$:    000                     ; TDSRH STORED HERE
5929  025262  133727  025260  000200  BITB    10$,#BIT7       ;CHECK TERR BIT
5930  025270  001406                  BEQ     15$             ; BR IF TERR NOT SET
5931  025272                          GEDF    EM98,ERR12      ;REPORT ERROR: 'TERR NOT CLEAR'
5932                                                          ;       'DEVICE FATAL' ERROR # 49
5933  025272  104455                                                          TRAP    C$ERDF
5934  025274  000061                                                          .WORD   49
5935  025276  014357                                                          .WORD   EM98
5936  025300  017722                                                          .WORD   ERR12
5937  025302                          ESCAPE  TST             ;AND EXIT TEST
5938  025302  104410                                                          TRAP    C$ESCAPE
5939  025304  000312                                                          .WORD   L10047-.
5940                          ;-------------------------------------------------------------
5941                          ; NOW READ 1ST DATA CHARACTER (AND CHECK TERR=1,TXU=1)
5942                          ;-------------------------------------------------------------
5943  025306  004537  007256  15$:    JSR     R5,SERIAL       ;READ DATA CHARACTER (000) VIA TSO
5944  025312  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
5945  025314  000000                  000                     ; EXPECTED BIT SEQUENCE (00000000)
5946  025316  103003                  BCC     .+8.            ;BR IF NO ERROR
5947  025320                          ERROR                   ;REPORT STACKED ERROR
5948  025320  104460                                                          TRAP    C$ERROR
5949  025322                          ESCAPE  TST             ;AND EXIT TEST
5950  025322  104410                                                          TRAP    C$ESCAPE
5951  025324  000272                                                          .WORD   L10047-.
5952
```

```
CVDMCA.P11    10-DEC-80 09:14          TEST 16 -- BCP TX UNDERRUN W/TSOM TERMINATION

 5953  025326  004537  011614                JSR     R5,STEPLU       ;GENERATE 1 TICK TO UNDERRUN TX
 5954  025332  000001                        1
 5955
 5956  025334  004537  005432                JSR     R5,CKUSTS       ;CHECK USYRT STATUS
 5957  025340  000116                        TBMT!TSO!TXACT!TXU      ; FOR TBMT AND TX UNDERRUN
 5958  025342  103003                        BCC     .+8.            ;BR IF NO ERROR
 5959  025344                                ERROR                   ;REPORT STACKED ERROR
 5960  025344  104460                                                                TRAP    C$ERROR
 5961  025346                                ESCAPE  TST             ;SKIP TO END OF TEST
 5962  025346  104410                                                                TRAP    C$ESCAPE
 5963  025350  000246                                                                .WORD   L10047-.
 5964
 5965  025352  004537  003610                JSR     R5.READI        ;GET TX ERROR BIT (TO SEE IF SET)
 5966  025356  120403                        TDSRH                   ; TERR BIT IS IN TDSRH
 5967  025360  000000              20$:      000                     ; TDSRH STORED HERE
 5968  025362  133727  025360  000200       BITB    20$,#B:T7       ;CHECK TERR BIT
 5969  025370  001006                        BNE     25$             ; BR IF TERR IS SET
 5970  025372                                GEDF    EM99,ERR12      ;REPORT ERROR: 'TERR NOT SET''
 5971                                                                  ;       'DEVICE FATAL'' ERROR # 50
 5972  025372  104455                                                                TRAP    C$ERDF
 5973  025374  000062                                                                .WORD   50
 5974  025376  014404                                                                .WORD   EM99
 5975  025400  017722                                                                .WORD   ERR12
 5976  025402                                ESCAPE  TST             ;AND EXIT TEST
 5977  025402  104410                                                                TRAP    C$ESCAPE
 5978  025404  000212                                                                .WORD   L10047-.
 5979
 5980                                  ;-----------------------------------------------------------
 5981                                  ; NOW CHECK MARK CHARACTER (RESULT OF UNDERRUNNING TRANSMITTER)
 5982  025406  004537  007116        ;-----------------------------------------------------------
 5983  025412  000001              25$:      JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'MARK''
 5984  025414  103003                        1                       ; CHARACTER (BIT SHOULD=1)
 5985  025416                                BCC     .+8.            ;BR IF NO ERROR
 5986  025416  104460                        ERROR                   ;REPORT STACKED ERROR
 5987  025420                                                                TRAP    C$ERROR
 5988  025420  104410                        ESCAPE  TST             ;AND EXIT TEST
 5989  025422  000174                                                                TRAP    C$ESCAPE
 5990                                                                  .WORD   L10047-.
 5991  02542?  004537  007256
 5992  025430  000007                        JSR     R5,SERIAL       ;READ MARK CHARACTER VIA TSO
 5993  025432  000177                        7                       ; 7 BIT CHAR/CLOCK TICKS
 5994  025434  103003                        177                     ; EXPECTED BIT SEQUENCE (1111111)
 5995  025436                                BCC     .+8.            ;BR IF NO ERROR
 5996  025436  104460                        ERROR                   ;REPORT STACKED ERROR
 5997  025440                                                                TRAP    C$ERROR
 5998  025440  104410                        ESCAPE  TST             ;AND EXIT TEST
 5999  025442  000154                                                                TRAP    C$ESCAPE
 6000                                                                  .WORD   L10047-.
 6001                                  ;-----------------------------------------------------------
 6002                                  ; SET IDLE=0, VERIFY SYNCH CHARACTER IS OUTPUT
 6003  025444  004537  003734        ;-----------------------------------------------------------
 6004  025450  120405                        JSR     R5,WRITEI       ;SET IDLE BIT=0
 6005  025452  000107                        PCSARH
 6006                                        PROTO!XYZ
 6007  025454  004537  007256
 6008  025460  000010                        JSR     R5,SERIAL       ;READ MARK CHARACTER VIA TSO
                                              8.                      ; 8 BIT CHAR/CLOCK TICKS
```

```
6009  025462  000377                    377                     ; EXPECTED BIT SEQUENCE (11111111)
6010  025464  103003                    BCC     .+8.            ;BR IF NO ERROR
6011  025466                            ERROR                   ;REPORT STACKED ERROR
6012  025466  104460                                                            TRAP    C$ERROR
6013  025470                            ESCAPE  TST             ;AND EXIT TEST
6014  025470  104410                                                            TRAP    C$ESCAPE
6015  025472  000124                                                            .WORD   L10047-.
6016
6017  025474  004537  007256            JSR     R5,SERIAL       ;READ SYNCH CHARACTER VIA TSO
6018  025500  000010                    8.                      ; 8 BIT CHAR/CLOCK TICKS
6019  025502  000151                    151                     ; EXPECTED BIT SEQUENCE (010010110)
6020  025504  103003                    BCC     .+8.            ;BR IF NO ERROR
6021  025506                            ERROR                   ;REPORT STACKED ERROR
6022  025506  104460                                                            TRAP    C$ERROR
6023  025510                            ESCAPE  TST             ;AND EXIT TEST
6024  025510  104410                                                            TRAP    C$ESCAPE
6025  025512  000104                                                            .WORD   L10047-.
6026
6027  025514  004537  010010            JSR     R5,TXCTRL       ;* ASSERT TSOM (SHOULD CLEAR TXU,TERR)
6028  025520  000001                    TSOM                    ;*
6029  025522  000000                    0                       ;*
6030
6031                                    ;-----------------------------------------------------------
6032                                    ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
6033  025524  004537  005432            ;-----------------------------------------------------------
6034  025530  000014                    JSR     R5,CKUSTS       ;CHECK USYRT STATUS
6035  025532  103003                    TXACT!TSO               ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
6036  025534                            BCC     .+8.            ;BR IF NO ERROR
6037  025534  104460                    ERROR                   ;REPORT STACKED ERROR
6038  025536                                                            TRAP    C$ERROR
6039  025536  104410                    ESCAPE  TST             ;SKIP TO END OF TEST
6040  025540  000056                                                            TRAP    C$ESCAPE
6041                                                              .WORD   L10047-.
6042  025542  004537  003610            JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
6043  025546  120403                    TDSRH                   ; TERR BIT IS IN TDSRH
6044  025550  000000            17$:    000                     ; TDSRH STORED HERE
6045  025552  133727  025550  000200    BITB    17$,#BIT7       ;CHECK TERR BIT
6046  025560  001406                    BEQ     18$             ; BR IF TERR NOT SET
6047  025562                            GEDF    EM98,ERR12      ;REPORT ERROR: "TERR NOT CLEAR"
6048                                                            ;       "DEVICE FATAL" ERROR # 51
6049  025562  104455                                                            TRAP    C$ERDF
6050  025564  000063                                                            .WORD   51
6051  025566  014357                                                            .WORD   EM98
6052  025570  017722                                                            .WORD   ERR12
6053  025572                            ESCAPE  TST             ;AND EXIT TEST
6054  025572  104410                                                            TRAP    C$ESCAPE
6055  025574  000022                                                            .WORD   L10047-.
6056                                    ;-----------------------------------------------------------
6057                                    ;READ/CHECK FOR SYNCH CHARACTER
6058                                    ;-----------------------------------------------------------
6059  025576  004537  007256    18$:    JSR     R5,SERIAL       ;READ !SYNCH! CHARACTER VIA TSO
6060  025602  000010                    8.                      ; 8 BIT CHAR/CLOCK TICKS
6061  025604  000151                    151                     ; EXPECTED BIT SEQUENCE (10010110)
6062  025606  103003                    BCC     .+8.            ;BR IF NO ERROR
6063  025610                            ERROR                   ;REPORT STACKED ERROR
6064  025610  104460                                                            TRAP    C$ERROR
```

CVDMCA.P11 10-DEC-80 09:14 TEST 16 -- BCP TX UNDERRUN W/TSOM TERMINATION

```
6065  025612
6066  025612  104410          ESCAPE  TST             ;AND EXIT TEST
6067  025614  000002                                                  TRAP    C$ESCAPE
6068  025616                                                          .WORD   L10047-.
6069  025616          ENDTST                                  L10047:
6070  025616  104401                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION

                                ;++*****************************************************************************
                                ;*
                                ;*      TEST 17 -- BCP TX UNDERRUN W/RESET TERMINATION
                                ;*
                                ;*  THE  USYRT  IS INITIALIZED FOR BCP WITH IDLE = 1 AND TXC IS MANUALLY
                                ;*  CONTROLLED  UNTIL  TWO  SYNC  CHARACTERS AND ONE DATA CHARACTER HAVE
                                ;*  BEEN  TRANSMITTED.  AT THIS TIME WHEN TXBE IS ASSERTED BY THE USYRT,
                                ;*  NO  DATA  IS LOADED INTO TXDB -- FORCING AN UNDER RUN CONDITION. TXU
                                ;*  AND   TERR   ARE  CHECKED  BOTH  BEFORE  AND  AFTER  THEIR  EXPECTED
                                ;*  ASSERTIONS.  AFTER THE FIRST NON-DATA CHARACTER (WHICH SHOULD BE THE
                                ;*  MARK  CHARACTER)  HAS  BEEN  STARTED,  IDLE IS SET TO 0. THIS SHOULD
                                ;*  FORCE   THE   NEXT  NON-DATA  CHARACTER  TO  BE  A  SYNC  CHARACTER.
                                ;*  IMMEDIATELY  AFTER  THIS  SYNC  CHARACTER  HAS  BEING TRANSMITTED, A
                                ;*  PROGRAM RESET IS ISSUED AND ALL REGISTERS ARE CHECKED. ERROR
                                ;*  LOOPING  WILL  DEPEND  ON  WHERE  THE FIRST ERROR OCCURS WITHIN THE
                                ;*  SEQUENCE.
                                ;*
                                ;--*****************************************************************************
                                ;
                                        BGNTST
                                                                                        T17::
025620
025620  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

025624  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
025630  047626                          DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
025632  040000                          NORXEN                  ;USE 8 BIT CHARS, RECEIVER DISABLED
025634  103003                          BCC     .+8.            ;BR IF NO ERROR
025636                                  ERROR                   ;REPORT STACKED ERROR
025636  104460                                                                  TRAP    C$ERROR
025640                                  ESCAPE  TST             ;SKIP TO END OF TEST
025640  104410                                                                  TRAP    C$ESCAPE
025642  000354                                                                  .WORD   L10050-.

025644  004537  007116                  JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'SYNCH'
025650  000000                          0                       ; CHARACTER (SHOULD BE 0)
025652  103003                          BCC     .+8.            ;BR IF NO ERROR
025654                                  ERROR                   ;REPORT STACKED ERROR
025654  104460                                                                  TRAP    C$ERROR
025656                                  ESCAPE  TST             ;AND EXIT TEST
025656  104410                                                                  TRAP    C$ESCAPE
025660  000336                                                                  .WORD   L10050-.

025662  004537  007256                  JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF 'SYNCH' CHARACTER
025666  000007                          7.                      ; (OFF OF TSO BIT)
025670  000151                          151                     ; EXPECTED BIT SEQUENCE (0010110)
025672  103003                          BCC     .+8.            ;BR IF NO ERROR
025674                                  ERROR                   ;REPORT STACKED ERROR
025674  104460                                                                  TRAP    C$ERROR
025676                                  ESCAPE  TST             ;AND EXIT TEST
025676  104410                                                                  TRAP    C$ESCAPE
025700  000316                                                                  .WORD   L10050-.

025702  004537  003734                  JSR     R5,WRITEI       ;LOAD 1ST DATA CHARACTER (000)
025706  120402                          TDSRL
```

```
6127  025710  000000                  000
6128
6129  025712  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
6130  025716  000000                  000
6131  025720  000000                  0
6132
6133  025722  004537  007256          JSR     R5,SERIAL       ;READ 2ND SYNCH CHARACTER VIA TSO
6134  025726  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6135  025730  000151                  151                     ; EXPECTED BIT SEQUENCE (010010110)
6136  025732  103003                  BCC     .+8.            ;BR IF NO ERROR
6137  025734                          ERROR                   ;REPORT STACKED ERROR
6138  025734  104460                                                  TRAP    C$ERROR
6139  025736                          ESCAPE  TST             ;AND EXIT TEST
6140  025736  104410                                                  TRAP    C$ESCAPE
6141  025740  000256                                                  .WORD   L10050-.
6142                          ;-----------------------------------------------------------
6143                          ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
6144                          ;-----------------------------------------------------------
6145  025742  004537  005432          JSR     R5,CKUSTS       ;CHECK USYRT STATUS
6146  025746  000114                  TXACT!TBMT!TSO          ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
6147  025750  103003                  BCC     .+8.            ;BR IF NO ERROR
6148  025752                          ERROR                   ;REPORT STACKED ERROR
6149  025752  104460                                                  TRAP    C$ERROR
6150  025754                          ESCAPE  TST             ;SKIP TO END OF TEST
6151  025754  104410                                                  TRAP    C$ESCAPE
6152  025756  000240                                                  .WORD   L10050-.
6153
6154  025760  004537  003610          JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
6155  025764  120403                  TDSRH                   ; TERR BIT IS IN TDSRH
6156  025766  000000          10$:    J00                     ; TDSRH STORED HERE
6157  025770  133727  025766  000200  BITB    10$,#BIT7       ;CHECK TERR BIT
6158  025776  001406                  BEQ     15$             ; BR IF TERR NOT SET
6159  026000                          GEDF    EM98,ERR12      ;REPORT ERROR: 'TERR NOT CLEAR''
6160                                                          ;       'DEVICE FATAL'' ERROR # 52
6161  026000  104455                                                  TRAP    C$ERDF
6162  026002  000064                                                  .WORD   52
6163  026004  014357                                                  .WORD   EM98
6164  026006  017722                                                  .WORD   ERR12
6165  026010                          ESCAPE  TST             ;AND EXIT TEST
6166  026010  104410                                                  TRAP    C$ESCAPE
6167  026012  000204                                                  .WORD   L10050-.
6168                          ;-----------------------------------------------------------
6169                          ; NOW READ 1ST DATA CHARACTER (AND CHECK TERR=1,TXU=1)
6170                          ;-----------------------------------------------------------
6171  026014  004537  007256  15$:     JSR     R5,SERIAL       ;READ DATA CHARACTER (000) VIA TSO
6172  026020  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6173  026022  000000                  000                     ; EXPECTED BIT SEQUENCE (00000000)
6174  026024  103003                  BCC     .+8.            ;BR IF NO ERROR
6175  026026                          ERROR                   ;REPORT STACKED ERROR
6176  026026  104460                                                  TRAP    C$ERROR
6177  026030                          ESCAPE  TST             ;AND EXIT TEST
6178  026030  104410                                                  TRAP    C$ESCAPE
6179  026032  000164                                                  .WORD   L10050-.
6180
6181  026034  004537  011614          JSR     R5,STEPLU       ;GENERATE 1 TICK TO UNDERRUN TX
6182  026040  000001                  1
```

```
6183
6184  026042  004537  005432          JSR     R5,CKUSTS       ;CHECK USYRT STATUS
6185  026046  000116                  TBMT!TSO!TXACT!TXU      ; FOR TBMT AND TX UNDERRUN
6186  026050  103003                  BCC     .+8.            ;BR IF NO ERROR
6187  026052                          ERROR                   ;REPORT STACKED ERROR
6188  026052  104460                                                          TRAP    C$ERROR
6189  026054                          ESCAPE  TST             ;SKIP TO END OF TEST
6190  026054  104410                                                          TRAP    C$ESCAPE
6191  026056  000140                                                          .WORD   L10050-.
6192
6193  026060  004537  003610          JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
6194  026064  120403                  TDSRH                   ; TERR BIT IS IN TDSRH
6195  026066  000000          20$:    000                     ; TDSRH STORED HERE
6196  026070  133727  026066  000200  BITB    20$,#BIT7       ;CHECK TERR BIT
6197  026076  001006                  BNE     25$             ; BR IF TERR IS SET
6198  026100                          GEDF    EM99,ERR12      ;REPORT ERROR: 'TERR NOT SET''
6199                                                          ;       'DEVICE FATAL'' ERROR # 53
6200  026100  104455                                                          TRAP    C$ERDF
6201  026102  000065                                                          .WORD   53
6202  026104  014404                                                          .WORD   EM99
6203  026106  017722                                                          .WORD   ERR12
6204  026110                          ESCAPE  TST             ;AND EXIT TEST
6205  026110  104410                                                          TRAP    C$ESCAPE
6206  026112  000104                                                          .WORD   L10050-.
6207                                  ;-------------------------------------------------------
6208                                  ; NOW CHECK MARK CHARACTER (RESULT OF UNDERRUNNING TRANSMITTER)
6209                                  ;-------------------------------------------------------
6210  026114  004537  003734  25$:    JSR     R5,WRITEI       ;CLEAR IDLE BIT
6211  026120  120405                  PCSARH
6212  026122  000107                  BIT6!BIT2!BIT1!BIT0     ;DDCMP FORMAT, NO ERROR CHECKING
6213
6214  026124  004537  007116          JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'MARK''
6215  026130  000001                  1                       ; CHARACTER (BIT SHOULD=1)
6216  026132  103003                  BCC     .+8.            ;BR IF NO ERROR
6217  026134                          ERROR                   ;REPORT STACKED ERROR
6218  026134  104460                                                          TRAP    C$ERROR
6219  026136                          ESCAPE  TST             ;AND EXIT TEST
6220  026136  104410                                                          TRAP    C$ESCAPE
6221  026140  000056                                                          .WORD   L10050-.
6222
6223  026142  004537  007256          JSR     R5,SERIAL       ;READ MARK CHARACTER VIA TSO
6224  026146  000007                  7                       ; 7 BIT CHAR/CLOCK TICKS
6225  026150  000177                  177                     ; EXPECTED BIT SEQUENCE (1111111)
6226  026152  103003                  BCC     .+8.            ;BR IF NO ERROR
6227  026154                          ERROR                   ;REPORT STACKED ERROR
6228  026154  104460                                                          TRAP    C$ERROR
6229  026156                          ESCAPE  TST             ;AND EXIT TEST
6230  026156  104410                                                          TRAP    C$ESCAPE
6231  026160  000036                                                          .WORD   L10050-.
6232
6233                                  ;-------------------------------------------------------
6234                                  ; READ/CHECK FOR SYNCH, RESET/CHECK USYRT REGISTERS
6235                                  ;-------------------------------------------------------
6236  026162  004537  007256          JSR     R5,SERIAL       ;READ SYNCH CHARACTER VIA TSO
6237  026166  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6238  026170  000151                  151                     ; EXPECTED BIT SEQUENCE (00000000)
```

```
6239  026172  103003          BCC     .+8.            ;BR IF NO ERROR
6240  026174                  ERROR                   ;REPORT STACKED ERROR
6241  026174  104460                                                  TRAP    C$ERROR
6242  026176                  ESCAPE  TST             ;AND EXIT TEST
6243  026176  104410                                                  TRAP    C$ESCAPE
6244  026200  000016                                                  .WORD   L10050-.
6245
6246  026202  004537  005126  JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
6247  026206  103003          BCC     .+8.            ;BR IF NO ERROR
6248  026210                  ERROR                   ;REPORT STACKED ERROR
6249  026210  104460                                                  TRAP    C$ERROR
6250  026212                  ESCAPE TST              ;SKIP TO END OF TEST
6251  026212  104410                                                  TRAP    C$ESCAPE
6252  026214  000002                                                  .WORD   L10050-.
6253  026216          ENDTST
6254  026216                                                  L10050:
6255  026216  104401                                                  TRAP    C$ETST
```

```
                        .SBTTL  TEST 18 -- BCP TX DISABLE TEST

                        ;++*************************************************************************
                        ;*
                        ;*      TEST 18 -- BCP TX DISABLE TEST
                        ;*
                        ;*  THE  USYRT IS INITIALIZED FOR BCP AND A MESSAGE IS STARTED. ONCE THE
                        ;*  SECOND  DATA  CHARACTER  IS LOADED INTO TXDB TXE IS DROPPED. TXSO IS
                        ;*  WATCHED TO ASSURE THAT THE CHARACTER BEING TRANSMITTED IS COMPLETED.
                        ;*  WHEN  IT  IS, THE USYRT SHOULD DROP TXA AND STOP TRANSMITTING -- THE
                        ;*  LAST CHARACTER LOADED INTO TXDB SHOULD BE LOST.
                        ;*
                        ;*      CHARACTERS LOADED: 125 252
                        ;*      CHARACTERS TRANSMITTED: 125
                        ;*
                        ;--*************************************************************************
                        ;
                        ;       BGNTST
026220                                                                  T18::

026220  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

026224  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026230  043626                  DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
026232  000000                  0                       ;USE 8 BIT CHARS
026234  103003                  BCC     .+8.            ;BR IF NO ERROR
026236                          ERROR                   ;REPORT STACKED ERROR
026236  104460                                                          TRAP    C$ERROR
026240                          ESCAPE  TST             ;SKIP TO END OF TEST
026240  104410                                                          TRAP    C$ESCAPE
026242  000162                                                          .WORD   L10051-.

026244  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
026250  000001                  TSOM                    ;AND KNOCK DOWN TBMT
026252  000007                  7.

026254  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM (GET READY TO SEND DATA)
026260  000000                  000
026262  000000                  0

026264  004537  007676          JSR     R5,TXCHAR       ;LOAD 125, TX 2ND SYNCH
026270  000125                  125
026272  000010                  8.
026274  103003                  BCC     .+8.            ;BR IF NO ERROR
026276                          ERROR                   ;REPORT STACKED ERROR
026276  104460                                                          TRAP    C$ERROR
026300                          ESCAPE  TST             ;SKIP TO END OF TEST
026300  104410                                                          TRAP    C$ESCAPE
026302  000122                                                          .WORD   L10051-.

026304  004537  007676          JSR     R5,TXCHAR       ;LOAD 252
026310  000252                  252
026312  000000                  0
026314  103003                  BCC     .+8.            ;BR IF NO ERROR
026316                          ERROR                   ;REPORT STACKED ERROR
026316  104460                                                          TRAP    C$ERROR
```

```
6312  026320
6313  026320  104410                    ESCAPE  TST             ;SKIP TO END OF TEST
6314  026322  000102                                                            TRAP    C$ESCAPE
6315                                                                            .WORD   L10051-.
6316  026324  004537  003734            JSR     R5,WRITEI       ;DROP TRANSMIT ENABLE (TXE)
6317  026330  120000                    VIAORB
6318  026332  000102                    RXEN!TTLOOP
6319
6320  026334  004537  011614            JSR     R5,STEPLU       ;CLOCK IN 125
6321  026340  000007                    7.                      ;***
6322
6323  026342  004537  011364            JSR     R5,RCV1ST       ;CLOCK AND RCV 125
6324  026346  000000                    0
6325  026350  103003                    BCC     .+8.            ;BR IF NO ERROR
6326  026352                            ERROR                   ;REPORT STACKED ERROR
6327  026352  104460                                                            TRAP    C$ERROR
6328  026354                            ESCAPE  TST             ;SKIP TO END OF TEST
6329  026354  104410                                                            TRAP    C$ESCAPE
6330  026356  000046                                                            .WORD   L10051-.
6331
6332  026360  004537  010110            JSR     R5,RXCHAR       ;READ AND CHECK FOR 125
6333  026364  000125                    125
6334  026366  000000                    0
6335  026370  000010                    8.                      ;AND CLOCK IN NEXT CHAR.
6336  026372  103003                    BCC     .+8.            ;BR IF NO ERROR
6337  026374                            ERROR                   ;REPORT STACKED ERROR
6338  026374  104460                                                            TRAP    C$ERROR
6339  026376                            ESCAPE  TST             ;SKIP TO END OF TEST
6340  026376  104410                                                            TRAP    C$ESCAPE
6341  026400  000024                                                            .WORD   L10051-.
6342
6343  026402  004537  010110            JSR     R5,RXCHAR       ;READ AND CHECK FOR 377
6344  026406  000377                    377                     ;** IF 252 IS READ, THEN DROPPING
6345  026410  000000                    0                       ;** TXEN DIDN'T WORK !!!
6346  026412  000010                    8.
6347  026414  103003                    BCC     .+8.            ;BR IF NO ERROR ****
6348  026416                            ERROR                   ;REPORT STACKED ERROR ****
6349  026416  104460                                                            TRAP    C$ERROR
6350  026420                            ESCAPE  TST             ;SKIP TO END OF TEST ****
6351  026420  104410                                                            TRAP    C$ESCAPE
6352  026422  000002                                                            .WORD   L10051-.
6353  026424                    ENDTST
6354  026424                                                            L10051:
6355  026424  104401                                                            TRAP    C$ETST
```

```
                                .SBTTL  TEST 19 -- FIFO STACKING CHARACTERS TEST

                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 19 -- FIFO STACKING CHARACTERS TEST
                                ;*
                                ;*  THE USYRT IS SETUP FOR BCP MODE WITH NO ERROR DETECTION.
                                ;*  THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS
                                ;* TO FILL THE 8 CHARACTER RECEIVER FIFO WITH THE CHARACTERS:
                                ;*      1/2(SYNCH),000,377,125,252,347,030,303,1/2(074).
                                ;*  THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. NOTE
                                ;* THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE  CHARACTERS SINCE THEY
                                ;* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
                                ;* RDA = 0).
                                ;*  ALSO NOTE THAT DUE TO FIFO TIMING, TWO 'HALF CHARACTERS' ARE LOADED
                                ;* INTO THE FIFO (THE 1ST AND LAST CHARACTERS).
                                ;*
                                ;*
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;
026426                          ;       BGNTST                                          T19::
026426  004737  005420          ;       JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
                                ;
026432  004537  007400          ;       JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026436  043626                  ;       DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
026440  000000                  ;       0                       ;USE 8 BIT CHARS
026442  103003                  ;       BCC     .+8.            ;BR IF NO ERROR
026444                          ;       ERROR                   ;REPORT STACKED ERROR
026444  104460                  ;                                                       TRAP    C$ERROR
026446                          ;       ESCAPE  TST             ;SKIP TO END OF TEST
026446  104410                  ;                                                       TRAP    C$ESCAPE
026450  000216                  ;                                                       .WORD   L10052-.
                                ;
026452  004537  010010          ;       JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
026456  000001                  ;       TSOM                    ;
026460  000007                  ;       7.
026462  004537  010010          ;       JSR     R5,TXCTRL       ;CLEAR TSOM
026466  000000                  ;       000
026470  000000                  ;       0
                                ;-----------------------------------------------------------
                                ; FILL THE FIFO WITH CHARACTERS (DATA1 - DATA8)
                                ;-----------------------------------------------------------
                                ;
026472  012702  026670          ;       MOV     #TXTBL3,R2      ;SET UP TABLE POINTER
026476  112237  026506          ;5$:    MOVB    (R2)+,10$       ;SETUP TRANSMIT CHARACTER
                                ;
026502  004537  007676          ;       JSR     R5,TXCHAR       ;TRANSMIT A CHARACTER
026506  000000                  ;10$:   000                     ;** HOLE FOR NEXT TX CHARACTER
026510  100010                  ;       NCTBMT*256.!8.          ;NO CHECK OF INITIAL TBMT=0
026512  103003                  ;       BCC     .+8.            ;BR IF NO ERROR
026514                          ;       ERROR                   ;REPORT STACKED ERROR
026514  104460                  ;                                                       TRAP    C$ERROR
026516                          ;       ESCAPE  TST             ;SKIP TO END OF TEST
026516  104410                  ;                                                       TRAP    C$ESCAPE
026520  000146                  ;                                                       .WORD   L10052-.
```

```
6412  026522  022702  026702          CMP     #TXEND3,R2
6413  026526  001363                  BNE     5$
6414
6415  026530  004537  011614          JSR     R5,STEPLU       ;ADD A FEW TX TICKS TO COMPLETELY
6416  026534  000005                  5                       ; FILL UP FIFO.
6417                                  ;-----------------------------------------------------
6418                                  ; THE FIFO IS NOW FULL OF CHARACTERS.
6419                                  ; NOW READ/VERIFY THEM.
6420                                  ;-----------------------------------------------------
6421  026536  012702  026670          MOV     #TXTBL3,R2      ;SET UP TABLE POINTER
6422  026542  112237  026552  15$:    MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
6423
6424  026546  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
6425  026552  000000  20$:    000             ;** HOLE FOR EXPECTED RECEIVE CHAR.
6426  026554  000000                  0                       ;
6427  026556  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
6428  026560  103003                  BCC     .+8.            ;BR IF NO ERROR
6429  026562                          ERROR                   ;REPORT STACKED ERROR
6430  026562  104460                                                  TRAP    C$ERROR
6431  026564                          ESCAPE  TST             ;SKIP TO END OF TEST
6432  026564  104410                                                  TRAP    C$ESCAPE
6433  026566  000100                                                  .WORD   L10052-.
6434
6435  026570  022702  026700          CMP     #TXEND3-2,R2
6436  026574  001362                  BNE     15$
6437
6438  026576  004537  011614          JSR     R5,STEPLU       ;CLOCK IN LAST FEW CHARACTERS OFF
6439  026602  000015                  15                      ; OF FIFO.
6440
6441  026604  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
6442  026610  000303                  303
6443  026612  000000                  0
6444  026614  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
6445  026616  103003                  BCC     .+8.            ;BR IF NO ERROR
6446  026620                          ERROR                   ;REPORT STACKED ERROR
6447  026620  104460                                                  TRAP    C$ERROR
6448  026622                          ESCAPE  TST             ;SKIP TO END OF TEST
6449  026622  104410                                                  TRAP    C$ESCAPE
6450  026624  000042                                                  .WORD   L10052-.
6451
6452  026626  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
6453  026632  000074                  074
6454  026634  000000                  0
6455  026636  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
6456  026640  103003                  BCC     .+8.            ;BR IF NO ERROR
6457  026642                          ERROR                   ;REPORT STACKED ERROR
6458  026642  104460                                                  TRAP    C$ERROR
6459  026644                          ESCAPE  TST             ;SKIP TO END OF TEST
6460  026644  104410                                                  TRAP    C$ESCAPE
6461  026646  000020                                                  .WORD   L10052-.
6462
6463  026650  004537  011532          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
6464  026654  000010                  8.
6465  026656  103003                  BCC     .+8.            ;BR IF NO ERROR
6466  026660                          ERROR                   ;REPORT STACKED ERROR
6467  026660  104460                                                  TRAP    C$ERROR
```

```
                        TEST 19 -- FIFO STACKING CHARACTERS TEST

                                ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10052-.
                        ENDTST
                                                                L10052:
                                                                        TRAP    C$ETST
                        ;-------------------------------
 6468  026662
 6469  026662  104410  TXTBL3: .BYTE   226     ;SYNCH
 6470  026664  000002          .BYTE   226     ;SYNCH
 6471  026666                  .BYTE   000
 6472  026666                  .BYTE   377
 6473  026666  104401          .BYTE   125
 6474                          .BYTE   252
 6475  026670     226          .BYTE   347
 6476  026671     226          .BYTE   030
 6477  026672     000          .BYTE   303
 6478  026673     377          .BYTE   074
 6479  026674     125  TXEND3: .BYTE   000
 6480  026675     252          .EVEN
 6481  026676     347  ;-------------------------------
 6482  026677     030
 6483  026700     303
 6484  026701     074
 6485  026702     000
 6486          026704
 6487
```

```
                                .SBTTL  TEST 20 -- BCP CHARACTER LENGTH TEST

                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 20 -- BCP CHARACTER LENGTH TEST
                                ;*
                                ;*  THE USYRT IS INITIALIZED FOR BCP WITH NO ERROR CHECKING. TXC IS MANUALLY
                                ;*  CONTROLLED UNTIL TWO SYNC CHARACTERS HAVE BEEN TRANSMITTED.  THEN  3
                                ;*  SUBTESTS FOLLOW, EACH ONE USING A DIFFERENT TRANSMIT CHARACTER LENGTH
                                ;*  STARTING AT FIVE (5) AND ENDING WITH SEVEN (7).
                                ;*
                                ;*        TEST PATTERN: 111 222 333 044 155 266 377
                                ;*
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;
                                ;       BGNTST
                                                                                        T20::
026704                          ;
026704  004737  005420          ;       JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
                                ;-----------------------------------------------------------------
                                ; SUBROUTINE # 1: 5 BIT CHARACTERS
                                ;-----------------------------------------------------------------
                                ;       BGNSUB
026710                          ;
026710                                                                  T20.1:
026710  104402                                                                  TRAP    C$BSUB
026712  004537  007400          ;       JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026716  043626                  ;       DDCMP!NOCHK!SYNCH       ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
026720  000245                  ;       BIT7!BIT5!BIT2!BIT0     ;TXCL=RXCL=5 BITS
026722  103003                  ;       BCC     .+8.            ;BR IF NO ERROR
026724                          ;       ERROR                   ;REPORT STACKED ERROR
026724  104460                                                                  TRAP    C$ERROR
026726                          ;       ESCAPE  SUB             ;SKIP TO END OF TEST
026726  104410                                                                  TRAP    C$ESCAPE
026730  000126                                                                  .WORD   L10054-.
                                ;
026732  004537  010010          ;       JSR     R5,TXCTRL       ;LOAD 2ND SYNCH,TX 1ST SYNCH
026736  000001                  ;       TSOM
026740  000004                  ;       4.
026742  004537  010010          ;       JSR     R5,TXCTRL       ;CLEAR TSOM
026746  000000                  ;       000
026750  000000                  ;       0
                                ;===================================================================
026752  012703  027402          ;       MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
026756  112337  026766  10$:    ;       MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
                                ;
026762  004537  007676          ;       JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
026766  000000          1$:     ;       000                     ;** HOLE FOR NEXT CHARACTER **
026770  000005                  ;       5.
026772  103003                  ;       BCC     .+8.            ;BR IF NO ERROR
026774                          ;       ERROR                   ;REPORT STACKED ERROR
026774  104460                                                                  TRAP    C$ERROR
026776                          ;       ESCAPE  SUB             ;SKIP TO END OF TEST
026776  104410                                                                  TRAP    C$ESCAPE
027000  000056                                                                  .WORD   L10054-.
                                ;
027002  022703  027413          ;       CMP     #T24TBL+9.,R3   ;ALL CHARACTERS TRANSMITTED ?
027006  001363                  ;       BNE     10$             ; IF NOT, TX ANOTHER ONE
```

```
6544
6545  027010  012703  027402                ;================================================================
6546  027014  112337  027032                        MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
6547  027020  142737  000340  027032      40$:      MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
6548                                              BICB    #340,4$         ;MASK OUT UNTRANSMITTED BITS
6549  027026  004537  010110                      JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
6550  027032  000000                      4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
6551  027034  000000                              0
6552  027036  100000                              NOCRDA                  ;NO INITIAL CHECK OF RDA=0
6553  027040  103003                              BCC     .+8.            ;BR IF NO ERROR
6554  027042                                      ERROR                   ;REPORT STACKED ERROR
6555  027042  104460                                                              TRAP    C$ERROR
6556  027044                                      ESCAPE  SUB             ;SKIP TO END OF TEST
6557  027044  104410                                                              TRAP    C$ESCAPE
6558  027046  000010                                                              .WORD   L10054-.
6559
6560  027050  022703  027411                      CMP     #T24TBL+7,R3    ;ALL CHARACTERS CHECKED ?
6561  027054  001357                              BNE     40$             ; IF NOT, CHECK ANOTHER ONE
6562
6563  027056                                ENDSUB
6564  027056                                                          L10054:
6565  027056  104403                                                              TRAP    C$ESUB
6566                                        ;----------------------------------------------------------------
6567                                        ; SUBROUTINE # 2: 6 BIT CHARACTERS
6568                                        ;----------------------------------------------------------------
6569  027060                                      BGNSUB
6570  027060                                                          T20.2:
6571  027060  104402                                                              TRAP    C$BSUB
6572  027062  004537  007400                      JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
6573  027066  043626                              DDCMP!NOCHK!SYNCH       ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
6574  027070  000306                              BIT7!BIT6!BIT2!BIT1     ;TXCL=RXCL=6 BITS
6575  027072  103003                              BCC     .+8.            ;BR IF NO ERROR
6576  027074                                      ERROR                   ;REPORT STACKED ERROR
6577  027074  104460                                                              TRAP    C$ERROR
6578  027076                                      ESCAPE  SUB             ;SKIP TO END OF TEST
6579  027076  104410                                                              TRAP    C$ESCAPE
6580  027100  000126                                                              .WORD   L10055-.
6581
6582  027102  004537  010010                      JSR     R5,TXCTRL       ;LOAD 2ND SYNCH,TX 1ST SYNCH
6583  027106  000001                              TSOM
6584  027110  000005                              5.
6585  027112  004537  010010                      JSR     R5,TXCTRL       ;CLEAR TSOM
6586  027116  000000                              000
6587  027120  000000                              0
6588                                        ;================================================================
6589  027122  012703  027402                      MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
6590  027126  112337  027136      20$:      MOVB    (R3)+,2$        ;INSTALL NEXT TX CHARACTER
6591
6592  027132  004537  007676                      JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
6593  027136  000000                      2$:     000                     ;** HOLE FOR NEXT CHARACTER **
6594  027140  000006                              6.
6595  027142  103003                              BCC     .+8.            ;BR IF NO ERROR
6596  027144                                      ERROR                   ;REPORT STACKED ERROR
6597  027144  104460                                                              TRAP    C$ERROR
6598  027146                                      ESCAPE  SUB             ;SKIP TO END OF TEST
6599  027146  104410                                                              TRAP    C$ESCAPE
```

```
6600  027150  000056                                                          .WORD   L10055-.
6601
6602  027152  022703  027414          CMP     #T24TBL+10.,R3  ;ALL CHARACTERS TRANSMITTED ?
6603  027156  001363                  BNE     20$             ; IF NOT, TX ANOTHER ONE
6604                                ;=================================================================
6605  027160  012703  027402          MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
6606  027164  112337  027202  50$:    MOVB    (R3)+,5$        ;INSTALL NEXT EXPECTED RX CHARACTER
6607  027170  142737  000300  027202  BICB    #300,5$         ;MASK OUT UNTRANSMITTED BITS
6608
6609  027176  004537  010110          JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
6610  027202  000000          5$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
6611  027204  000000                  0
6612  027206  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
6613  027210  103003                  BCC     .+8.            ;BR IF NO ERROR
6614  027212                          ERROR                   ;REPORT STACKED ERROR
6615  027212  104460                                                          TRAP    C$ERROR
6616  027214                          ESCAPE  SUB             ;SKIP TO END OF TEST
6617  027214  104410                                                          TRAP    C$ESCAPE
6618  027216  000010                                                          .WORD   L10055-.
6619
6620  027220  022703  027411          CMP     #T24TBL+7,R3    ;ALL CHARACTERS CHECKED ?
6621  027224  001357                  BNE     50$             ; IF NOT, CHECK ANOTHER ONE
6622
6623  027226                        ENDSUB
6624  027226                                                          L10055:
6625  027226  104403                                                          TRAP    C$ESJB
6626                                ;-----------------------------------------------------------------
6627                                ; SUBROUTINE # 3: 7 BIT CHARACTERS
6628                                ;-----------------------------------------------------------------
6629  027230                          BGNSUB
6630  027230                                                          T20.3:
6631  027230  104402                                                          TRAP    C$BSUB
6632  027232  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
6633  027236  043626                  DDCMP!NOCHK!SYNCH       ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
6634  027240  000347                  TXDL!RXDL               ;TXCL=RXCL=7 BITS
6635  027242  103003                  BCC     .+8.            ;BR IF NO ERROR
6636  027244                          ERROR                   ;REPORT STACKED ERROR
6637  027244  104460                                                          TRAP    C$ERROR
6638  027246                          ESCAPE  SUB             ;SKIP TO END OF TEST
6639  027246  104410                                                          TRAP    C$ESCAPE
6640  027250  000126                                                          .WORD   L10056-.
6641
6642  027252  004537  010010          JSR     R5,TXCTRL       ;LOAD 2ND SYNCH,TX 1ST SYNCH
6643  027256  000001                  TSOM
6644  027260  000006                  6.
6645  027262  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
6646  027266  000000                  000
6647  027270  000000                  0
6648                                ;=================================================================
6649  027272  012703  027402          MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
6650  027276  112337  027306  30$:    MOVB    (R3)+,3$        ;INSTALL NEXT TX CHARACTER
6651
6652  027302  004537  007676          JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
6653  027306  000000          3$:     000                     ;** HOLE FOR NEXT CHARACTER **
6654  027310  000007                  7.
6655  027312  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
6656  027314                          ERROR                   ;REPORT STACKED ERROR
6657  027314  104460                                                          TRAP    C$ERROR
6658  027316                          ESCAPE  SUB             ;SKIP TO END OF TEST
6659  027316  104410                                                          TRAP    C$ESCAPE
6660  027320  000056                                                          .WORD   L10056-.
6661
6662  027322  022703  027413          CMP     #T24TBL+9.,R3   ;ALL CHARACTERS TRANSMITTED ?
6663  027326  001363                  BNE     30$             ; IF NOT, TX ANOTHER ONE
6664                                ;=================================================================
6665  027330  012703  027402          MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
6666  027334  112337  027352  60$:    MOVB    (R3)+,6$        ;INSTALL NEXT EXPECTED RX CHARACTER
6667  027340  142737  000200  027352  BICB    #200,6$         :MASK OUT UNTRANSMITTED BITS
6668
6669  027346  004537  010110          JSR     R5,RXCHAR       ;READ/ .CK NEXT CHARACTER
6670  027352  000000  6$:     000             ;** HOLE FOR NEXT EXPECTED CHARACTER
6671  027354  000000                  0
6672  027356  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
6673  027360  103003                  BCC     .+8.            ;BR IF NO ERROR
6674  027362                          ERROR                   ;REPORT STACKED ERROR
6675  027362  104460                                                          TRAP    C$ERROR
6676  027364                          ESCAPE  SUB             ;SKIP TO END OF TEST
6677  027364  104410                                                          TRAP    C$ESCAPE
6678  027366  000010                                                          .WORD   L10056-.
6679
6680  027370  022703  027411          CMP     #T24TBL+7,R3    ;ALL CHARACTERS CHECKED ?
6681  027374  001357                  BNE     60$             ; IF NOT, CHECK ANOTHER ONE
6682
6683  027376                        ENDSUB
6684  027376                                                          L10056:
6685  027376  104403                                                          TRAP    C$ESUB
6686  027400                        ENDTST
6687  027400                                                          L10053:
6688  027400  104401                                                          TRAP    C$ETST
6689
6690  027402      111   ;----------------------------------
6691  027403      222   T24TBL: .BYTE   111     ;D1
6692  027404      333           .BYTE   222     ;D2
6693  027405      044           .BYTE   333     ;D3
6694  027406      155           .BYTE   044     ;D4
6695  027407      266           .BYTE   155     ;D5
6696  027410      377           .BYTE   266     ;D6
6697  027411      000           .BYTE   377     ;D7
6698  027412      000           .BYTE   000     ;FILLER 1
6699  027413      000           .BYTE   000     ;FILLER 2
6700                            .BYTE   000     ;FILLER 3
6701                            .EVEN
                              ;----------------------------------
```

```
6702                                    .SBTTL  TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST
6703
6704                                    ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
6705                                    ;*
6706                                    ;*      TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST
6707                                    ;*
6708                                    ;*  THE USYRT IS INITIALIZED FOR 'BIT-ORIENTED PROTOCOL'' (BOP) WITH IDLE
6709                                    ;*  SET TO ZERO. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
6710                                    ;*  OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
6711                                    ;*  NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
6712                                    ;*  LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
6713                                    ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
6714                                    ;*  ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
6715                                    ;*  THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
6716                                    ;*  FLAGS, ONE ZERO CHARACTER, AND ONE ABORT CHARACTER IS SENT (INTO THE BIT
6717                                    ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
6718                                    ;*  THE SEQUENCE.
6719                                    ;*
6720                                    ;--*************************************************************************************
6721                                    ;
6722                                            BGNTST
6723  027414                                                                                    T21::
6724  027414  004737  005420                   JSR     PC,INIDMV       ;INIT VIA
6725
6726  027420  004537  007400                   JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
6727  027424  003400                           NOCHK                   ;BOP MODE, NO ERROR CHECK
6728  027426  040000                           NORXEN                  ;NO RECEIVER ENABLE,USE 8 BIT CHARS
6729  027430  103003                           BCC     .+8.            ;BR IF NO ERROR
6730  027432                                   ERROR                   ;REPORT STACKED ERROR
6731  027432  104460                                                                   TRAP    C$ERROR
6732  027434                                   ESCAPE  TST             ;SKIP TO END OF TEST
6733  027434  104410                                                                   TRAP    C$ESCAPE
6734  027436  000150                                                                   .WORD   L10057-.
6735
6736  027440  004537  007116                   JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'FLAG'
6737  027444  000000                           0                       ; CHARACTER (SHOULD BE 0)
6738  027446  103003                           BCC     .+8.            ;BR IF NO ERROR
6739  027450                                   ERROR                   ;REPORT STACKED ERROR
6740  027450  104460                                                                   TRAP    C$ERROR
6741  027452                                   ESCAPE  TST             ;AND EXIT TEST
6742  027452  104410                                                                   TRAP    C$ESCAPE
6743  027454  000132                                                                   .WORD   L10057-.
6744
6745  027456  004537  007256                   JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF 'FLAG' CHARACTER
6746  027462  000007                           7.                      ; (OFF OF TSO BIT)
6747  027464  000176                           176                     ; EXPECTED BIT SEQUENCE (1111110)
6748  027466  103003                           BCC     .+8.            ;BR IF NO ERROR
6749  027470                                   ERROR                   ;REPORT STACKED ERROR
6750  027470  104460                                                                   TRAP    C$ERROR
6751  027472                                   ESCAPE  TST             ;AND EXIT TEST
6752  027472  104410                                                                   TRAP    C$ESCAPE
6753  027474  000112                                                                   .WORD   L10057-.
6754
6755  027476  004537  010010                   JSR     R5,TXCTRL       ;CLEAR TSOM
6756  027502  000000                           000
6757  027504  000000                           0
```

```
6758
6759  027506  004537  003734          JSR     R5,WRITEI       ;LOAD 000 CHARACTER
6760  027512  120402                  TDSRL
6761  027514  000000                  000
6762
6763  027516  004537  007256          JSR     R5,SERIAL       ;READ FLAG CHARACTER VIA TSO
6764  027522  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6765  027524  000176                  176                     ; EXPECTED BIT SEQUENCE (01111110)
6766  027526  103003                  BCC     .+8.            ;BR IF NO ERROR
6767  027530                          ERROR                   ;REPORT STACKED ERROR
6768  027530  104460                                                  TRAP    C$ERROR
6769  027532                          ESCAPE  TST             ;AND EXIT TEST
6770  027532  104410                                                  TRAP    C$ESCAPE
6771  027534  000052                                                  .WORD   L10057-.
6772
6773  027536  004537  010010          JSR     R5,TXCTRL       ;SET TXABT BIT
6774  027542  000004                  TAB
6775  027544  000000                  0
6776
6777  027546  004537  007256          JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
6778  027552  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6779  027554  000000                  000                     ; EXPECTED BIT SEQUENCE (00000000)
6780  027556  103003                  BCC     .+8.            ;BR IF NO ERROR
6781  027560                          ERROR                   ;REPORT STACKED ERROR
6782  027560  104460                                                  TRAP    C$ERROR
6783  027562                          ESCAPE  TST             ;AND EXIT TEST
6784  027562  104410                                                  TRAP    C$ESCAPE
6785  027564  000022                                                  .WORD   L10057-.
6786
6787  027566  004537  007256          JSR     R5,SERIAL       ;READ ABORT CHARACTER VIA TSO
6788  027572  000007                  7                       ; 7 BIT CHAR/CLOCK TICKS
6789  027574  000177                  177                     ; EXPECTED BIT SEQUENCE (1111111)
6790  027576  103003                  BCC     .+8.            ;BR IF NO ERROR
6791  027600                          ERROR                   ;REPORT STACKED ERROR
6792  027600  104460                                                  TRAP    C$ERROR
6793  027602                          ESCAPE  TST             ;AND EXIT TEST
6794  027602  104410                                                  TRAP    C$ESCAPE
6795  027604  000002                                                  .WORD   L10057-.
6796  027606                  ENDTST
6797  027606                                                  L10057:
6798  027606  104401                                                  TRAP    C$ETST
```

CVDMCA.P11 10-DEC-80 09:14 TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST

```
6799                                    .SBTTL  TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST
6800
6801                                    ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
6802                                    ;*
6803                                    ;*      TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST
6804                                    ;*
6805                                    ;*  THE USYRT IS INITIALIZED FOR "BIT-ORIENTED PROTOCOL" (BOP) WITH IDLE
6806                                    ;*  SET TO ONE. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
6807                                    ;*  OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
6808                                    ;*  NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
6809                                    ;*  LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
6810                                    ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
6811                                    ;*  ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
6812                                    ;*  THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
6813                                    ;*  FLAGS, ONE ZERO CHARACTER, AND ONE FLAG CHARACTER IS SENT (INTO THE BIT
6814                                    ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
6815                                    ;*  THE SEQUENCE.
6816                                    ;*
6817                                    ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
6818                                    ;
6819                                    ;       BGNTST
6820  027610                                                                    T22::
6821  027610  004737  005420           ;       JSR     PC,INIDMV       ;INIT VIA
6822
6823  027614  004537  007400           ;       JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
6824  027620  007400                   ;       NOCHK!IDLES             ;BOP MODE, NO ERROR CHECK, IDLE=1
6825  027622  040000                   ;       NORXEN                  ;NO RECEIVER ENABLE,USE 8 BIT CHARS
6826  027624  103003                   ;       BCC     .+8.            ;BR IF NO ERROR
6827  027626                           ;       ERROR                   ;REPORT STACKED ERROR
6828  027626  104460                                                            TRAP    C$ERROR
6829  027630                           ;       ESCAPE  TST             ;SKIP TO END OF TEST
6830  027630  104410                                                            TRAP    C$ESCAPE
6831  027632  000150                                                            .WORD   L10060-.
6832
6833  027634  004537  007116           ;       JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "FLAG"
6834  027640  000000                   ;       0                       ; CHARACTER (SHOULD BE 0)
6835  027642  103003                   ;       BCC     .+8.            ;BR IF NO ERROR
6836  027644                           ;       ERROR                   ;REPORT STACKED ERROR
6837  027644  104460                                                            TRAP    C$ERROR
6838  027646                           ;       ESCAPE  TST             ;AND EXIT TEST
6839  027646  104410                                                            TRAP    C$ESCAPE
6840  027650  000132                                                            .WORD   L10060-.
6841
6842  027652  004537  007256           ;       JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF "FLAG" CHARACTER
6843  027656  000007                   ;       7.                      ; (OFF OF TSO BIT)
6844  027660  000176                   ;       176                     ; EXPECTED BIT SEQUENCE (1111110)
6845  027662  103003                   ;       BCC     .+8.            ;BR IF NO ERROR
6846  027664                           ;       ERROR                   ;REPORT STACKED ERROR
6847  027664  104460                                                            TRAP    C$ERROR
6848  027666                           ;       ESCAPE  TST             ;AND EXIT TEST
6849  027666  104410                                                            TRAP    C$ESCAPE
6850  027670  000112                                                            .WORD   L10060-.
6851
6852  027672  004537  010010           ;       JSR     R5,TXCTRL       ;CLEAR TSOM
6853  027676  000000                   ;       000
6854  027700  000000                   ;       0
```

```
6855
6856  027702  004537  003734          JSR     R5,WRITEI       ;LOAD 000 CHARACTER
6857  027706  120402                  TDSRL
6858  027710  000000                  000
6859
6860  027712  004537  007256          JSR     R5,SERIAL       ;READ FLAG CHARACTER VIA TSO
6861  027716  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6862  027720  000176                  176                     ; EXPECTED BIT SEQUENCE (01111110)
6863  027722  103003                  BCC     .+8.            ;BR IF NO ERROR
6864  027724                          ERROR                   ;REPORT STACKED ERROR
6865  027724  104460                                                  TRAP    C$ERROR
6866  027726                          ESCAPE  TST             ;AND EXIT TEST
6867  027726  104410                                                  TRAP    C$ESCAPE
6868  027730  000052                                                  .WORD   L10060-.
6869
6870  027732  004537  010010          JSR     R5,TXCTRL       ;SET TXABT BIT
6871  027736  000004                  TAB
6872  027740  000000                  0
6873
6874  027742  004537  007256          JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
6875  027746  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6876  027750  000000                  000                     ; EXPECTED BIT SEQUENCE (00000000)
6877  027752  103003                  BCC     .+8.            ;BR IF NO ERROR
6878  027754                          ERROR                   ;REPORT STACKED ERROR
6879  027754  104460                                                  TRAP    C$ERROR
6880  027756                          ESCAPE  TST             ;AND EXIT TEST
6881  027756  104410                                                  TRAP    C$ESCAPE
6882  027760  000022                                                  .WORD   L10060-.
6883
6884  027762  004537  007256          JSR     R5,SERIAL       ;READ ABORT (FLAG) CHARACTER VIA TSO
6885  027766  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6886  027770  000176                  176                     ; EXPECTED BIT SEQUENCE (01111110)
6887  027772  103003                  BCC     .+8.            ;BR IF NO ERROR
6888  027774                          ERROR                   ;REPORT STACKED ERROR
6889  027774  104460                                                  TRAP    C$ERROR
6890  027776                          ESCAPE  TST             ;AND EXIT TEST
6891  027776  104410                                                  TRAP    C$ESCAPE
6892  030000  000002                                                  .WORD   L10060-.
6893  030002                  ENDTST
6894  030002                                                  L10060:
6895  030002  104401                                                  TRAP    C$ETST
```

```
                        .SBTTL  TEST 23 -- BOP TX TXGA (TRANSMIT GO-AHEAD) TEST

                        ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                        ;*
                        ;*      TEST 23 -- BOP TX TXGA (TRANSMIT GO-AHEAD) TEST
                        ;*
                        ;*  THE USYRT IS INITIALIZED FOR BOP AND TXE IS ASSERTED. TSOM IS ASSERTED
                        ;*  AND TXA OBSERVED -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
                        ;*  NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
                        ;*  LOADED  INTO  TXDB DRIVING TXBE LOW. WHEN TXBE GOES HIGH AGAIN, TXGA
                        ;*  IS  ASSERTED CAUSING GA TO BE TRANSMITTED. THE SEQUENCE OF EVENTS IS
                        ;*  CONTINUALLY  MONITORED  WHILE  TXC  IS  MANUALLY  CONTROLLED AND ALL
                        ;*  CHARACTERS  ARE  CHECKED  AT TXSO. THIS TEST WILL GO NO FURTHER INTO
                        ;*  THE  TRANSMIT  SEQUENCE  SO  THAT  ONLY TWO FLAGS, ONE ZERO CHARACTER
                        ;*  AND  ONE GA CHARACTER IS SENT (INTO THE BIT BUCKET).
                        ;*  ERROR  LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
                        ;*  THE SEQUENCE.
                        ;*
                        ;--**********************************************************************************************
                        ;
                                BGNTST
                                                                                        T23::
030004
030004  004737  005420          JSR     PC,INIDMV       ;INIT VIA

030010  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030014  003400                  NOCHK                   ;BOP MODE, NO ERROR CHECK
030016  040000                  NORXEN                  ;NO RECEIVER ENABLE,USE 8 BIT CHARS
030020  103003                  BCC     .+8.            ;BR IF NO ERROR
030022                          ERROR                   ;REPORT STACKED ERROR
030022  104460                                                  TRAP    C$ERROR
030024                          ESCAPE  TST             ;SKIP TO END OF TEST
030024  104410                                                  TRAP    C$ESCAPE
030026  000150                                                  .WORD   L10061-.

030030  004537  007116          JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED 'FLAG'
030034  000000                  0                       ; CHARACTER (SHOULD BE 0)
030036  103003                  BCC     .+8.            ;BR IF NO ERROR
030040                          ERROR                   ;REPORT STACKED ERROR
030040  104460                                                  TRAP    C$ERROR
030042                          ESCAPE  TST             ;AND EXIT TEST
030042  104410                                                  TRAP    C$ESCAPE
030044  000132                                                  .WORD   L10061-.

030046  004537  007256          JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF 'FLAG' CHARACTER
030052  000007                  7.                      ; (OFF OF TSO BIT)
030054  000176                  176                     ; EXPECTED BIT SEQUENCE (1111110)
030056  103003                  BCC     .+8.            ;BR IF NO ERROR
030060                          ERROR                   ;REPORT STACKED ERROR
030060  104460                                                  TRAP    C$ERROR
030062                          ESCAPE  TST             ;AND EXIT TEST
030062  104410                                                  TRAP    C$ESCAPE
030064  000112                                                  .WORD   L10061-.

030066  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
030072  000000                  000
030074  000000                  0
```

```
6952
6953  030076  004537  003734          JSR     R5,WRITEI       ;LOAD 000 CHARACTER
6954  030102  120402                  TDSRL
6955  030104  000000                  000
6956
6957  030106  004537  007256          JSR     R5,SERIAL       ;READ 'FLAG' CHARACTER VIA TSO
6958  030112  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6959  030114  000176                  176                     ; EXPECTED BIT SEQUENCE (01111110)
6960  030116  103003                  BCC     .+8.            ;BR IF NO ERROR
6961  030120                          ERROR                   ;REPORT STACKED ERROR
6962  030120  104460                                                  TRAP    CSERROR
6963  030122                          ESCAPE  TST             ;AND EXIT TEST
6964  030122  104410                                                  TRAP    CSESCAPE
6965  030124  000052                                                  .WORD   L10061-.
6966
6967  030126  004537  010010          JSR     R5,TXCTRL       ;SET TX GO-AHEAD AND TEOM
6968  030132  000012                  TGA!TEOM
6969  030134  000000                  0
6970
6971  030136  004537  007256          JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
6972  030142  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6973  030144  000000                  000                     ; EXPECTED BIT SEQUENCE (00000000)
6974  030146  103003                  BCC     .+8.            ;BR IF NO ERROR
6975  030150                          ERROR                   ;REPORT STACKED ERROR
6976  030150  104460                                                  TRAP    CSERROR
6977  030152                          ESCAPE  TST             ;AND EXIT TEST
6978  030152  104410                                                  TRAP    CSESCAPE
6979  030154  000022                                                  .WORD   L10061-.
6980
6981  030156  004537  007256          JSR     R5,SERIAL       ;READ GO-AHEAD CHARACTER VIA TSO
6982  030162  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6983  030164  000177                  177                     ; EXPECTED BIT SEQUENCE (01111111)
6984  030166  103003                  BCC     .+8.            ;BR IF NO ERROR
6985  030170                          ERROR                   ;REPORT STACKED ERROR
6986  030170  104460                                                  TRAP    CSERROR
6987  030172                          ESCAPE  TST             ;AND EXIT TEST
6988  030172  104410                                                  TRAP    CSESCAPE
6989  030174  000002                                                  .WORD   L10061-.
6990  030176                  ENDTST
6991  030176                                                  L10061:
6992  030176  104401                                                  TRAP    CSETST
```

```
                        .SBTTL  TEST 24 -- BOP TX MESSAGE WITHOUT CRC

                        ;++*****************************************************************************
                        ;*
                        ;*      TEST 24 -- BOP TX MESSAGE WITHOUT CRC
                        ;*
                        ;*  THE USYRT IS INITIALIZED FOR BOP MODE WITH NO ERROR DETECTION. TXC IS THEN
                        ;*  MANUALLY CONTROLLED UNTIL TWO FLAG CHARACTERS HAVE BEEN TRANSMITTED. THEN
                        ;*  A 5 CHARACTER MESSAGE IS TRANSMITTED, RECEIVED, CHECKED, AND TERMINATED
                        ;*  (WITH TEOM). A CHECK IS MADE TO ASCERTAIN THAT NO CRC OR VRC IS GENERATED
                        ;*  -- FLAG CHARACTERS SHOULD FOLLOW THE DATA.
                        ;*       (NOTE: NO BIT STUFFING OCCURS IN THIS TEST)
                        ;*
                        ;*      TEST MESSAGE: FLAG FLAG 000 307 125 252 201 FLAG
                        ;*
                        ;--*****************************************************************************
                        ;
                              BGNTST
                                                                                        T24::
030200
030200  004737  005420        JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
030204  004537  007400        JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030210  003400                NOCHK                   ;SET BOP MODE,NO CHECK
030212  000000                0                       ;USE 8 BIT CHARS
030214  103003                BCC     .+8.            ;BR IF NO ERROR
030216                        ERROR                   ;REPORT STACKED ERROR
030216  104460                                                        TRAP    C$ERROR
030220                        ESCAPE  TST             ;SKIP TO END OF TEST
030220  104410                                                        TRAP    C$ESCAPE
030222  000332                                                        .WORD   L10062-.

030224  004537  010010        JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
030230  000001                TSOM
030232  000007                7.

030234  004537  010010        JSR     R5,TXCTRL       ;CLEAR TSOM
030240  000000                000
030242  000000                0

030244  004537  007676        JSR     R5,TXCHAR       ;LOAD 000, TX 2ND FLAG
030250  000000                000
030252  000010                8.
030254  103003                BCC     .+8.            ;BR IF NO ERROR
030256                        ERROR                   ;REPORT STACKED ERROR
030256  104460                                                        TRAP    C$ERROR
030260                        ESCAPE  TST             ;SKIP TO END OF TEST
030260  104410                                                        TRAP    C$ESCAPE
030262  000272                                                        .WORD   L10062-.

030264  004537  007676        JSR     R5,TXCHAR       ;LOAD 307, TX 000
030270  000307                307
030272  000010                8.
030274  103003                BCC     .+8.            ;BR IF NO ERROR
030276                        ERROR                   ;REPORT STACKED ERROR
030276  104460                                                        TRAP    C$ERROR
030300                        ESCAPE  TST             ;SKIP TO END OF TEST
```

```
7049  030300  104410                                                          TRAP    C$ESCAPE
7050  030302  000252                                                          .WORD   L10062-.
7051
7052  030304  004537  007676          JSR     R5,TXCHAR       ;LOAD 125; TX 307
7053  030310  000125                  125
7054  030312  000010                  8.
7055  030314  103003                  BCC     .+8.            ;BR IF NO ERROR
7056  030316                          ERROR                   ;REPORT STACKED ERROR
7057  030316  104460                                                          TRAP    C$ERROR
7058  030320                          ESCAPE  TST             ;SKIP TO END OF TEST
7059  030320  104410                                                          TRAP    C$ESCAPE
7060  030322  000232                                                          .WORD   L10062-.
7061
7062  030324  004537  007676          JSR     R5,TXCHAR       ;LOAD 252
7063  030330  000252                  252
7064  030332  000000                  0
7065  030334  103003                  BCC     .+8.            ;BR IF NO ERROR
7066  030336                          ERROR                   ;REPORT STACKED ERROR
7067  030336  104460                                                          TRAP    C$ERROR
7068  030340                          ESCAPE  TST             ;SKIP TO END OF TEST
7069  030340  104410                                                          TRAP    C$ESCAPE
7070  030342  000212                                                          .WORD   L10062-.
7071
7072  030344  004537  011364          JSR     R5,RCV1ST       ;CLOCK AND RCV 000
7073  030350  000000                  0
7074  030352  103003                  BCC     .+8.            ;BR IF NO ERROR
7075  030354                          ERROR                   ;REPORT STACKED ERROR
7076  030354  104460                                                          TRAP    C$ERROR
7077  030356                          ESCAPE  TST             ;SKIP TO END OF TEST
7078  030356  104410                                                          TRAP    C$ESCAPE
7079  030360  000174                                                          .WORD   L10062-.
7080
7081  030362  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 000, RCV 307
7082  030366  000400                  RXSOM!000               ; & CHECK RSOM=1
7083  030370  000000                  0
7084  030372  000010                  8.
7085  030374  103003                  BCC     .+8.            ;BR IF NO ERROR
7086  030376                          ERROR                   ;REPORT STACKED ERROR
7087  030376  104460                                                          TRAP    C$ERROR
7088  030400                          ESCAPE  TST             ;SKIP TO END OF TEST
7089  030400  104410                                                          TRAP    C$ESCAPE
7090  030402  000152                                                          .WORD   L10062-.
7091
7092  030404  004537  007676          JSR     R5,TXCHAR       ;LOAD 201
7093  030410  000201                  201
7094  030412  000000                  0
7095  030414  103003                  BCC     .+8.            ;BR IF NO ERROR
7096  030416                          ERROR                   ;REPORT STACKED ERROR
7097  030416  104460                                                          TRAP    C$ERROR
7098  030420                          ESCAPE  TST             ;SKIP TO END OF TEST
7099  030420  104410                                                          TRAP    C$ESCAPE
7100  030422  000132                                                          .WORD   L10062-.
7101
7102  030424  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 307, RCV 125
7103  030430  000307                  307
7104  030432  000000                  0
```

```
7105  030434  000010                  8.
7106  030436  103003                  BCC     .+8.            ;BR IF NO ERROR
7107  030440                          ERROR                   ;REPORT STACKED ERROR
7108  030440  104460                                                  TRAP    C$ERROR
7109  030442                          ESCAPE  TST             ;SKIP TO END OF TEST
7110  030442  104410                                                  TRAP    C$ESCAPE
7111  030444  000110                                                  .WORD   L10062-.
7112
7113  030446  004537  010010          JSR     R5,TXCTRL       ;SET TEOM BIT
7114  030452  000002                  TEOM
7115  030454  000000                  0
7116  030456  103003                  BCC     .+8.            ;BR IF NO ERROR
7117  030460                          ERROR                   ;REPORT STACKED ERROR
7118  030460  104460                                                  TRAP    C$ERROR
7119  030462                          ESCAPE  TST             ;SKIP TO END OF TEST
7120  030462  104410                                                  TRAP    C$ESCAPE
7121  030464  000070                                                  .WORD   L10062-.
7122
7123  030466  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 125, RCV 252
7124  030472  000125                  125
7125  030474  000000                  0
7126  030476  000010                  8.
7127  030500  103003                  BCC     .+8.            ;BR IF NO ERROR
7128  030502                          ERROR                   ;REPORT STACKED ERROR
7129  030502  104460                                                  TRAP    C$ERROR
7130  030504                          ESCAPE  TST             ;SKIP TO END OF TEST
7131  030504  104410                                                  TRAP    C$ESCAPE
7132  030506  000046                                                  .WORD   L10062-.
7133
7134  030510  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 252, RCV 201
7135  030514  000252                  252
7136  030516  000000                  0
7137  030520  020010                  NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
7138  030522  103003                  BCC     .+8.            ;BR IF NO ERROR
7139  030524                          ERROR                   ;REPORT STACKED ERROR
7140  030524  104460                                                  TRAP    C$ERROR
7141  030526                          ESCAPE  TST             ;SKIP TO END OF TEST
7142  030526  104410                                                  TRAP    C$ESCAPE
7143  030530  000024                                                  .WORD   L10062-.
7144
7145  030532  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 201, RCV FIRST FLAG
7146  030536  001201                  RXEOM!201               ; & CHECK REOM
7147  030540  000000                  0
7148  030542  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
7149  030544  103003                  BCC     .+8.            ;BR IF NO ERROR
7150  030546                          ERROR                   ;REPORT STACKED ERROR
7151  030546  104460                                                  TRAP    C$ERROR
7152  030550                          ESCAPE  TST             ;SKIP TO END OF TEST
7153  030550  104410                                                  TRAP    C$ESCAPE
7154  030552  000002                                                  .WORD   L10062-.
7155  030554                  ENDTST
7156  030554                                                  L10062:
7157  030554  104401                                                  TRAP    C$ETST
```

```
                                .SBTTL  TEST 25 -- BOP RX CHARACTER LENGTH TEST

                                ;++*********************************************************************
                                ;*
                                ;*      TEST 25 -- BOP RX CHARACTER LENGTH TEST
                                ;*
                                ;*  THE  USYRT  IS INITIALIZED FOR BOP WITH CRC-CCITT PRESET TO 1'S. TXC
                                ;*  IS   MANUALLY   CONTROLLED  UNTIL  TWO  FLAG  CHARACTERS  HAVE  BEEN
                                ;*  TRANSMITTED.  THEN  6  SUBTESTS  FOLLOW,  EACH ONE USING A DIFFERENT
                                ;*  TRANSMIT  CHARACTER LENGTH STARTING AT TWO (2) AND ENDING WITH SEVEN
                                ;*  (7).  IN  EACH  SUBTEST,  TWO  8  BIT CHARACTERS WILL BE TRANSMITTED
                                ;*  BEFORE  TXCL  IS  CHANGED TO THE CHARACTER LENGTH BEING TESTED. THIS
                                ;*  CORRESPONDS TO NORMAL USAGE WHERE EITHER:
                                ;*
                                ;*       1 -- A  MESSAGE  OF  CHARACTERS  WHICH ARE LESS THEN 8 BITS IS
                                ;*            SENT  AS  A  STREAM OF 8 BIT CHARACTERS AND THE REMAINING
                                ;*            BITS ARE SENT AS A CHARACTER OF LESS THEN 8 BITS OR
                                ;*
                                ;*       2 -- A  HEADER  OF  TWO 8 BIT CHARACTERS IS SENT FOLLOWED BY A
                                ;*            DATA  STREAM  OF DATA CHARACTERS WHICH MAY BE LESS THEN 8
                                ;*            BITS  IN  LENGTH  (I.E.  2,  3,  4,  5,  6,  OR 7 BIT
                                ;*            CHARACTERS).
                                ;*
                                ;*       THE TEST PATTERN IS: 123 321 111 222 333 044 155 266 377
                                ;*
                                ;--*********************************************************************
                                ;
                                ;
                                        BGNTST
030556                                                                  T25::
030556  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
                                ;-----------------------------------------------------------------
                                ; SUBROUTINE # 1: 2 BIT CHARACTERS
                                ;-----------------------------------------------------------------
030562                                  BGNSUB
030562                                                                  T25.1:
030562  104402                                                                  TRAP    C$BSUB
030564  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030570  004226                          IDLES!SYNCH             ;SET BOP MODE,CRC-CCITT-1,S/AR=226
030572  000002                          BIT1                    ;INITIALLY: TXCL=8 BITS / RXCL=2 BITS
030574  103003                          BCC     .+8.            ;BR IF NO ERROR
030576                                  ERROR                   ;REPORT STACKED ERROR
030576  104460                                                                  TRAP    C$ERROR
030600                                  ESCAPE  SUB             ;SKIP TO END OF TEST
030600  104410                                                                  TRAP    C$ESCAPE
030602  000344                                                                  .WORD   L10064-.

030604  004537  010010                  JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
030610  000001                          TSOM
030612  000007                          7.
030614  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
030620  000000                          000
030622  000000                          0
030624  004537  007676                  JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
030630  000123                          123
030632  000010                          8.
030634  103003                          BCC     .+8.            ;BR IF NO ERROR
```

```
7214  030636                        ERROR                   ;REPORT STACKED ERROR
7215  030636  104460                                                          TRAP    C$ERROR
7216  030640                        ESCAPE  SUB             ;SKIP TO END OF TEST
7217  030640  104410                                                          TRAP    C$ESCAPE
7218  030642  000304                                                          .WORD   L10064-.
7219
7220  030644  004537  007676        JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7221  030650  000321                321
7222  030652  000010                8.
7223  030654  103003                BCC     .+8.            ;BR IF NO ERROR
7224  030656                        ERROR                   ;REPORT STACKED ERROR
7225  030656  104460                                                          TRAP    C$ERROR
7226  030660                        ESCAPE  SUB             ;SKIP TO END OF TEST
7227  030660  104410                                                          TRAP    C$ESCAPE
7228  030662  000264                                                          .WORD   L10064-.
7229
7230  030664  004537  003734        JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 2 BITS TOO
7231  030670  120407                PCR
7232  030672  000102                BIT6!BIT1
7233  030674  004537  007676        JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7234  030700  000111                111
7235  030702  000010                8.
7236  030704  103003                BCC     .+8.            ;BR IF NO ERROR
7237  030706                        ERROR                   ;REPORT STACKED ERROR
7238  030706  104460                                                          TRAP    C$ERROR
7239  030710                        ESCAPE  SUB             ;SKIP TO END OF TEST
7240  030710  104410                                                          TRAP    C$ESCAPE
7241  030712  000234                                                          .WORD   L10064-.
7242                        ;=================================================================
7243  030714  012703  033471        MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7244  030720  112337  030730  10$:  MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7245
7246  030724  004537  007676        JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7247  030730  000000        1$:     000                     ;** HOLE FOR NEXT CHARACTER **
7248  030732  000002                2.
7249  030734  103003                BCC     .+8.            ;BR IF NO ERROR
7250  030736                        ERROR                   ;REPORT STACKED ERROR
7251  030736  104460                                                          TRAP    C$ERROR
7252  030740                        ESCAPE  SUB             ;SKIP TO END OF TEST
7253  030740  104410                                                          TRAP    C$ESCAPE
7254  030742  000204                                                          .WORD   L10064-.
7255
7256  030744  022703  033477        CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7257  030750  001363                BNE     10$             ; IF NOT, TX ANOTHER ONE
7258                        ;=================================================================
7259  030752  004537  010010        JSR     R5,TXCTRL       ;SET TEOM
7260  030756  000002                TEOM
7261  030760  000000                0
7262  030762  004537  011614        JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7263  030766  000044                36.
7264
7265  030770  004537  010110        JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7266  030774  000523                RXSOM!123               ; & CHECK FOR RSOM=1
7267  030776  000000                0
7268  031000  100000                NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7269  031002  103003                BCC     .+8.            ;BR IF NO ERROR
```

```
7270  031004                        ERROR                   ;REPORT STACKED ERROR
7271  031004  104460                                                        TRAP    C$ERROR
7272  031006                        ESCAPE  SUB             ;SKIP TO END OF TEST
7273  031006  104410                                                        TRAP    C$ESCAPE
7274  031010  000136                                                        .WORD   L10064-.
7275
7276  031012  004537  010110        JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 001(DATA1)
7277  031016  000321                321
7278  031020  000000                0
7279  031022  100000                NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7280  031024  103003                BCC     .+8.            ;BR IF NO ERROR
7281  031026                        ERROR                   ;REPORT STACKED ERROR
7282  031026  104460                                                        TRAP    C$ERROR
7283  031030                        ESCAPE  SUB             ;SKIP TO END OF TEST
7284  031030  104410                                                        TRAP    C$ESCAPE
7285  031032  000114                                                        .WORD   L10064-.
7286                        ;=================================================================
7287  031034  012703  033470        MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7288  031040  112337  031056  20$:   MOVB    (R3)+,2$        ;INSTALL NEXT EXPECTED RX CHARACTER
7289  031044  142737  000374  031056  BICB  #374,2$          ;MASK OUT UNTRANSMITTED BITS
7290
7291  031052  004537  010110        JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7292  031056  000000        2$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
7293  031060  000000                0
7294  031062  100000                NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7295  031064  103003                BCC     .+8.            ;BR IF NO ERROR
7296  031066                        ERROR                   ;REPORT STACKED ERROR
7297  031066  104460                                                        TRAP    C$ERROR
7298  031070                        ESCAPE  SUB             ;SKIP TO END OF TEST
7299  031070  104410                                                        TRAP    C$ESCAPE
7300  031072  000054                                                        .WORD   L10064-.
7301
7302  031074  022703  033475        CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7303  031100  001357                BNE     20$             ; IF NOT, CHECK ANOTHER ONE
7304                        ;=================================================================
7305  031102  004537  010110        JSR     R5,RXCHAR       ;READ/CHK DATA6(002), RCV DATA7
7306  031106  000002                002
7307  031110  000000                0                       ;NO INITIAL CHECK OF RDA=0
7308  031112  120000                NCRACT!NOCRDA           ;DON'T CHECK FOR FINAL RXACT=1
7309  031114  103003                BCC     .+8.            ;BR IF NO ERROR
7310  031116                        ERROR                   ;REPORT STACKED ERROR
7311  031116  104460                                                        TRAP    C$ERROR
7312  031120                        ESCAPE  SUB             ;SKIP TO END OF TEST
7313  031120  104410                                                        TRAP    C$ESCAPE
7314  031122  000024                                                        .WORD   L10064-.
7315
7316  031124  004537  010110        JSR     R5,RXCHAR       ;READ/CHK DATA7(003), RCV FIRST FLAG
7317  031130  001003                RXEOM!003               ;& CHECK RERR BIT=0 (GOOD CRC) & REOM=1
7318  031132  000001                RERCHK                  ;NO INITIAL CHECK OF RDA=0
7319  031134  160000                NOCRDA!NFCRDA!NCRACT    ;DON'T CHECK FOR FINAL RDA=RXACT=1
7320  031136  103003                BCC     .+8.            ;BR IF NO ERROR
7321  031140                        ERROR                   ;REPORT STACKED ERROR
7322  031140  104460                                                        TRAP    C$ERROR
7323  031142                        ESCAPE  SUB             ;SKIP TO END OF TEST
7324  031142  104410                                                        TRAP    C$ESCAPE
7325  031144  000002                                                        .WORD   L10064-.
```

```
7326  031146
7327  031146
7328  031146  104403                          ENDSUB
7329                                                                  L10064:
7330                                                                          TRAP    C$ESUB
7331          ;-------------------------------------------------------------------------
7332  031150  ; SUBROUTINE # 2: 3 BIT CHARACTERS
7333  031150  ;-------------------------------------------------------------------------
7334  031150  104402                          BGNSUB
7335  031152  004537  007400                                                  T25.2:
7336  031156  000226                                                                  TRAP    C$BSUB
7337  031160  000003          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7338  031162  103003          SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7339  031164                  BIT1!BIT0               ;INITIALLY: TXCL=8 BITS / RXCL=3 BITS
7340  031164  104460          BCC     .+8.            ;BR IF NO ERROR
7341  031166                  ERROR                   ;REPORT STACKED ERROR
7342  031166  104410                                                          TRAP    C$ERROR
7343  031170  000344          ESCAPE  SUB             ;SKIP TO END OF TEST
7344                                                                          TRAP    C$ESCAPE
7345  031172  004537  010010                                                .WORD   L10065-.
7346  031176  000001          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
7347  031200  000007          TSOM
7348  031202  004537  010010  7.
7349  031206  000000          JSR     R5,TXCTRL       ;CLEAR TSOM
7350  031210  000000          000
7351  031212  004537  007676  0
7352  031216  000123          JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
7353  031220  000010          123
7354  031222  103003          8.
7355  031224                  BCC     .+8.            ;BR IF NO ERROR
7356  031224  104460          ERROR                   ;REPORT STACKED ERROR
7357  031226                                                          TRAP    C$ERROR
7358  031226  104410          ESCAPE  SUB             ;SKIP TO END OF TEST
7359  031230  000304                                                  TRAP    C$ESCAPE
7360                                                                  .WORD   L10065-.
7361  031232  004537  007676  JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7362  031236  000321          321
7363  031240  000010          8.
7364  031242  103003          BCC     .+8.            ;BR IF NO ERROR
7365  031244                  ERROR                   ;REPORT STACKED ERROR
7366  031244  104460                                                  TRAP    C$ERROR
7367  031246          ESCAPE  SUB             ;SKIP TO END OF TEST
7368  031246  104410                                                  TRAP    C$ESCAPE
7369  031250  000264                                                  .WORD   L10065-.
7370
7371  031252  004537  003734  JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 3 BITS TOO
7372  031256  120407          PCR
7373  031260  000143          BIT6!BIT5!BIT1!BIT0
7374  031262  004537  007676  JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7375  031266  000111          111
7376  031270  000010          8.
7377  031272  103003          BCC     .+8.            ;BR IF NO ERROR
7378  031274                  ERROR                   ;REPORT STACKED ERROR
7379  031274  104460                                                  TRAP    C$ERROR
7380  031276          ESCAPE  SUB             ;SKIP TO END OF TEST
7381  031276  104410                                                  TRAP    C$ESCAPE
```

```
7382  031300  000234                                                          .WORD   L10065-.
7383
7384  031302  012703  033471          ;=============================================================
7385  031306  112337  031316          10$:    MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7386                                          MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7387  031312  004537  007676                  JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7388  031316  000000                  1$:     000                     ;** HOLE FOR NEXT CHARACTER **
7389  031320  000003                          3.
7390  031322  103003                          BCC     .+8.            ;BR IF NO ERROR
7391  031324                                  ERROR                   ;REPORT STACKED ERROR
7392  031324  104460                                                          TRAP    C$ERROR
7393  031326                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7394  031326  104410                                                          TRAP    C$ESCAPE
7395  031330  000204                                                          .WORD   L10065-.
7396
7397  031332  022703  033477                  CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7398  031336  001363                          BNE     10$             ; IF NOT, TX ANOTHER ONE
7399          ;=============================================================
7400  031340  004537  010010                  JSR     R5,TXCTRL       ;SET TEOM
7401  031344  000002                          TEOM
7402  031346  000000                          0
7403  031350  004537  011614                  JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7404  031354  000044                          36.
7405
7406  031356  004537  010110                  JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7407  031362  000523                          RXSOM!123               ; & CHECK FOR RSOM=1
7408  031364  000000                          0
7409  031366  100000                          NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7410  031370  103003                          BCC     .+8.            ;BR IF NO ERROR
7411  031372                                  ERROR                   ;REPORT STACKED ERROR
7412  031372  104460                                                          TRAP    C$ERROR
7413  031374                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7414  031374  104410                                                          TRAP    C$ESCAPE
7415  031376  000136                                                          .WORD   L10065-.
7416
7417  031400  004537  010110                  JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 001(DATA1)
7418  031404  000321                          321
7419  031406  000000                          0
7420  031410  100000                          NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7421  031412  103003                          BCC     .+8.            ;BR IF NO ERROR
7422  031414                                  ERROR                   ;REPORT STACKED ERROR
7423  031414  104460                                                          TRAP    C$ERROR
7424  031416                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7425  031416  104410                                                          TRAP    C$ESCAPE
7426  031420  000114                                                          .WORD   L10065-.
7427                  ;=============================================================
7428  031422  012703  033470                  MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7429  031426  112337  031444          40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
7430  031432  142737  000370  031444        BICB    #370,4$         ;MASK OUT UNTRANSMITTED BITS
7431
7432  031440  004537  010110                  JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7433  031444  000000                  4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
7434  031446  000000                          0
7435  031450  100000                          NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7436  031452  103003                          BCC     .+8.            ;BR IF NO ERROR
7437  031454                                  ERROR                   ;REPORT STACKED ERROR
```

```
7438  031454  104460                                                          TRAP    C$ERROR
7439  031456
7440  031456  104410                                                          TRAP    C$ESCAPE
7441  031460  000054                                                          .WORD   L10065-.
7442
7443  031462  022703  033475          CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7444  031466  001357                  BNE     40$             ; IF NOT, CHECK ANOTHER ONE
7445                          ;=================================================================
7446  031470  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA6(006), RCV DATA7
7447  031474  000006                  006
7448  031476  000000                  0
7449  031500  020000                  NCRACT                  ;DON'T CHECK FOR FINAL RXACT=1
7450  031502  103003                  BCC     .+8.            ;BR IF NO ERROR
7451  031504                          ERROR                   ;REPORT STACKED ERROR
7452  031504  104460                                                          TRAP    C$ERROR
7453  031506                          ESCAPE  TST             ;SKIP TO END OF TEST
7454  031506  104410                                                          TRAP    C$ESCAPE
7455  031510  001756                                                          .WORD   L10063-.
7456
7457  031512  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA7(007), RCV FIRST FLAG
7458  031516  001007                  RXEOM!007               ;AND CHECK FOR REOM=1
7459  031520  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
7460  031522  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
7461  031524  103003                  BCC     .+8.            ;BR IF NO ERROR
7462  031526                          ERROR                   ;REPORT STACKED ERROR
7463  031526  104460                                                          TRAP    C$ERROR
7464  031530                          ESCAPE  TST             ;SKIP TO END OF TEST
7465  031530  104410                                                          TRAP    C$ESCAPE
7466  031532  001734                                                          .WORD   L10063-.
7467  031534                  ENDSUB
7468  031534                                                          L10065:
7469  031534  104403                                                          TRAP    C$ESUB
7470                          ;-----------------------------------------------------------------
7471                          ; SUBROUTINE # 3: 4 BIT CHARACTERS
7472                          ;-----------------------------------------------------------------
7473  031536                  BGNSUB
7474  031536                                                          T25.3:
7475  031536  104402                                                          TRAP    C$BSUB
7476  031540  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7477  031544  000226                  SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7478  031546  000004                  BIT2                    ;INITIALLY: TXCL=8 BITS / RXCL=4 BITS
7479  031550  103003                  BCC     .+8.            ;BR IF NO ERROR
7480  031552                          ERROR                   ;REPORT STACKED ERROR
7481  031552  104460                                                          TRAP    C$ERROR
7482  031554                          ESCAPE  SUB             ;SKIP TO END OF TEST
7483  031554  104410                                                          TRAP    C$ESCAPE
7484  031556  000344                                                          .WORD   L10066-.
7485
7486  031560  004537  010010          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
7487  031564  000001                  TSOM
7488  031566  000007                  7.
7489  031570  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
7490  031574  000000                  000
7491  031576  000000                  0
7492  031600  004537  007676          JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
7493  031604  000123                  123
```

```
7494  031606  000010                    8.
7495  031610  103003                    BCC     .+8.            ;BR IF NO ERROR
7496  031612                            ERROR                   ;REPORT STACKED ERROR
7497  031612  104460                                                    TRAP    C$ERROR
7498  031614                            ESCAPE  SUB             ;SKIP TO END OF TEST
7499  031614  104410                                                    TRAP    C$ESCAPE
7500  031616  000304                                                    .WORD   L10066-.
7501
7502  031620  004537  007676            JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7503  031624  000321                    321
7504  031626  000010                    8.
7505  031630  103003                    BCC     .+8.            ;BR IF NO ERROR
7506  031632                            ERROR                   ;REPORT STACKED ERROR
7507  031632  104460                                                    TRAP    C$ERROR
7508  031634                            ESCAPE  SUB             ;SKIP TO END OF TEST
7509  031634  104410                                                    TRAP    C$ESCAPE
7510  031636  000264                                                    .WORD   L10066-.
7511
7512  031640  004537  003734            JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 4 BITS TOO
7513  031644  120407                    PCR
7514  031646  000204                    BIT7!BIT2
7515  031650  004537  007676            JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7516  031654  000111                    111
7517  031656  000010                    8.
7518  031660  103003                    BCC     .+8.            ;BR IF NO ERROR
7519  031662                            ERROR                   ;REPORT STACKED ERROR
7520  031662  104460                                                    TRAP    C$ERROR
7521  031664                            ESCAPE  SUB             ;SKIP TO END OF TEST
7522  031664  104410                                                    TRAP    C$ESCAPE
7523  031666  000234                                                    .WORD   L10066-.
7524                            ;=================================================================
7525  031670  012703  033471            MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7526  031674  112337  031704    10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7527
7528  031700  004537  007676            JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7529  031704  000000            1$:     000                     ;** HOLE FOR NEXT CHARACTER **
7530  031706  000004                    4.
7531  031710  103003                    BCC     .+8.            ;BR IF NO ERROR
7532  031712                            ERROR                   ;REPORT STACKED ERROR
7533  031712  104460                                                    TRAP    C$ERROR
7534  031714                            ESCAPE  SUB             ;SKIP TO END OF TEST
7535  031714  104410                                                    TRAP    C$ESCAPE
7536  031716  000204                                                    .WORD   L10066-.
7537
7538  031720  022703  033477            CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7539  031724  001363                    BNE     10$             ; IF NOT, TX ANOTHER ONE
7540                            ;=================================================================
7541  031726  004537  010010            JSR     R5,TXCTRL       ;SET TEOM
7542  031732  000002                    TEOM
7543  031734  000000                    0
7544  031736  004537  011614            JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7545  031742  000044                    36.
7546
7547  031744  004537  010110            JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7548  031750  000523                    RXSOM!123               ; & CHECK FOR RSOM=1
7549  031752  000000                    0
```

```
7550  031754  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7551  031756  103003                  BCC     .+8.            ;BR IF NO ERROR
7552  031760                          ERROR                   ;REPORT STACKED ERROR
7553  031760  104460                                                  TRAP    C$ERROR
7554  031762                          ESCAPE  SUB             ;SKIP TO END OF TEST
7555  031762  104410                                                  TRAP    C$ESCAPE
7556  031764  000136                                                  .WORD   L10066-.
7557
7558  031766  004537  010110          JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7559  031772  000321                  321
7560  031774  000000                  0
7561  031776  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7562  032000  103003                  BCC     .+8.            ;BR IF NO ERROR
7563  032002                          ERROR                   ;REPORT STACKED ERROR
7564  032002  104460                                                  TRAP    C$ERROR
7565  032004                          ESCAPE  SUB             ;SKIP TO END OF TEST
7566  032004  104410                                                  TRAP    C$ESCAPE
7567  032006  000114                                                  .WORD   L10066-.
7568                  ;=================================================================
7569  032010  012703  033470          MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7570  032014  112337  032032  40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
7571  032020  142737  000360  032032  BICB    #360,4$         ;MASK OUT UNTRANSMITTED BITS
7572
7573  032026  004537  010110          JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7574  032032  000000          4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
7575  032034  000000                  0
7576  032036  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7577  032040  103003                  BCC     .+8.            ;BR IF NO ERROR
7578  032042                          ERROR                   ;REPORT STACKED ERROR
7579  032042  104460                                                  TRAP    C$ERROR
7580  032044                          ESCAPE  SUB             ;SKIP TO END OF TEST
7581  032044  104410                                                  TRAP    C$ESCAPE
7582  032046  000054                                                  .WORD   L10066-.
7583
7584  032050  022703  033475          CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7585  032054  001357                  BNE     40$             ; IF NOT, CHECK ANOTHER ONE
7586                  ;=================================================================
7587  032056  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA6(006), RCV DATA7
7588  032062  000006                  006
7589  032064  000000                  0
7590  032066  020000                  NCRACT                  ;DON'T CHECK FOR FINAL RXACT=1
7591  032070  103003                  BCC     .+8.            ;BR IF NO ERROR
7592  032072                          ERROR                   ;REPORT STACKED ERROR
7593  032072  104460                                                  TRAP    C$ERROR
7594  032074                          ESCAPE  TST             ;SKIP TO END OF TEST
7595  032074  104410                                                  TRAP    C$ESCAPE
7596  032076  001370                                                  .WORD   L10063-.
7597
7598  032100  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA7(017), RCV FIRST FLAG
7599  032104  001017                  RXEOM!017               ; AND CHECK FOR REOM=1
7600  032106  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
7601  032110  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
7602  032112  103003                  BCC     .+8.            ;BR IF NO ERROR
7603  032114                          ERROR                   ;REPORT STACKED ERROR
7604  032114  104460                                                  TRAP    C$ERROR
7605  032116                          ESCAPE  TST             ;SKIP TO END OF TEST
```

```
7606  032116  104410                                                    TRAP    C$ESCAPE
7607  032120  001346                                                    .WORD   L10063-.
7608  032122                  ENDSUB
7609  032122                                                          L10066:
7610  032122  104403                                                    TRAP    C$ESUB
7611
7612                    ;-------------------------------------------------------------
7613                    ; SUBROUTINE # 4: 5 BIT CHARACTERS
7614  032124            ;-------------------------------------------------------------
7615  032124                  BGNSUB
7616  032124  104402                                                  T25.4:
                                                                      TRAP    C$BSUB
7617  032126  004537  007400        JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7618  032132  000226                SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7619  032134  000005                BIT2!BIT0               ;INITIALLY: TXCL=8 BITS / RXCL=5 BITS
7620  032136  103003                BCC     .+8.            ;BR IF NO ERROR
7621  032140                        ERROR                   ;REPORT STACKED ERROR
7622  032140  104460                                                    TRAP    C$ERROR
7623  032142                        ESCAPE  SUB             ;SKIP TO END OF TEST
7624  032142  104410                                                    TRAP    C$ESCAPE
7625  032144  000344                                                    .WORD   L10067-.
7626
7627  032146  004537  010010        JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
7628  032152  000001                TSOM
7629  032154  000007                7.
7630  032156  004537  010010        JSR     R5,TXCTRL       ;CLEAR TSOM
7631  032162  000000                000
7632  032164  000000                0
7633  032166  004537  007676        JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
7634  032172  000123                123
7635  032174  000010                8.
7636  032176  103003                BCC     .+8.            ;BR IF NO ERROR
7637  032200                        ERROR                   ;REPORT STACKED ERROR
7638  032200  104460                                                    TRAP    C$ERROR
7639  032202                        ESCAPE  SUB             ;SKIP TO END OF TEST
7640  032202  104410                                                    TRAP    C$ESCAPE
7641  032204  000304                                                    .WORD   L10067-.
7642
7643  032206  004537  007676        JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7644  032212  000321                321
7645  032214  000010                8.
7646  032216  103003                BCC     .+8.            ;BR IF NO ERROR
7647  032220                        ERROR                   ;REPORT STACKED ERROR
7648  032220  104460                                                    TRAP    C$ERROR
7649  032222                        ESCAPE  SUB             ;SKIP TO END OF TEST
7650  032222  104410                                                    TRAP    C$ESCAPE
7651  032224  000264                                                    .WORD   L10067-.
7652
7653  032226  004537  003734        JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 5 BITS TOO
7654  032232  120407                PCR
7655  032234  000245                BIT7!BIT5!BIT2!BIT0
7656  032236  004537  007676        JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7657  032242  000111                111
7658  032244  000010                8.
7659  032246  103003                BCC     .+8.            ;BR IF NO ERROR
7660  032250                        ERROR                   ;REPORT STACKED ERROR
7661  032250  104460                                                    TRAP    C$ERROR
```

```
7662  032252
7663  032252  104410                          ESCAPE  SUB             ;SKIP TO END OF TEST
7664  032254  000234                                                                  TRAP    C$ESCAPE
7665                                                                                  .WORD   L10067-.
7666  032256  012703  033471                  ;=======================================================================
7667  032262  112337  032272                          MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7668                                          10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7669  032266  004537  007676                          JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7670  032272  000000                          1$:     000                     ;** HOLE FOR NEXT CHARACTER **
7671  032274  000005                                  5.
7672  032276  103003                                  BCC     .+8.            ;BR IF NO ERROR
7673  032300                                          ERROR                   ;REPORT STACKED ERROR
7674  032300  104460                                                                  TRAP    C$ERROR
7675  032302                                          ESCAPE  SUB             ;SKIP TO END OF TEST
7676  032302  104410                                                                  TRAP    C$ESCAPE
7677  032304  000204                                                                  .WORD   L10067-.
7678
7679  032306  022703  033477                          CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7680  032312  001363                                  BNE     10$             ; IF NOT, TX ANOTHER ONE
7681                                          ;=======================================================================
7682  032314  004537  010010                          JSR     R5,TXCTRL       ;SET TEOM
7683  032320  000002                                  TEOM
7684  032322  000000                                  0
7685  032324  004537  011614                          JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7686  032330  000044                                  36.
7687
7688  032332  004537  010110                          JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7689  032336  000523                                  RXSOM!123               ; & CHECK FOR RSOM=1
7690  032340  000000                                  0
7691  032342  100000                                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7692  032344  103003                                  BCC     .+8.            ;BR IF NO ERROR
7693  032346                                          ERROR                   ;REPORT STACKED ERROR
7694  032346  104460                                                                  TRAP    C$ERROR
7695  032350                                          ESCAPE  SUB             ;SKIP TO END OF TEST
7696  032350  104410                                                                  TRAP    C$ESCAPE
7697  032352  000136                                                                  .WORD   L10067-.
7698
7699  032354  004537  010110                          JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7700  032360  000321                                  321
7701  032362  000000                                  0
7702  032364  100000                                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7703  032366  103003                                  BCC     .+8.            ;BR IF NO ERROR
7704  032370                                          ERROR                   ;REPORT STACKED ERROR
7705  032370  104460                                                                  TRAP    C$ERROR
7706  032372                                          ESCAPE  SUB             ;SKIP TO END OF TEST
7707  032372  104410                                                                  TRAP    C$ESCAPE
7708  032374  000114                                                                  .WORD   L10067-.
7709                                          ;=======================================================================
7710  032376  012703  033470                          MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7711  032402  112337  032420                  40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
7712  032406  142737  000340  032420                  BICB    #340,4$         ;MASK OUT UNTRANSMITTED BITS
7713
7714  032414  004537  010110                          JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7715  032420  000000                          4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
7716  032422  000000                                  0
7717  032424  100000                                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
```

```
7718  032426  103003          BCC     .+8.            ;BR IF NO ERROR
7719  032430                  ERROR                   ;REPORT STACKED ERROR
7720  032430  104460                                                  TRAP    C$ERROR
7721  032432                  ESCAPE  SUB             ;SKIP TO END OF TEST
7722  032432  104410                                                  TRAP    C$ESCAPE
7723  032434  000054                                                  .WORD   L10067-.
7724
7725  032436  022703  033475  CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7726  032442  001357          BNE     40$             ; IF NOT, CHECK ANOTHER ONE
7727  ;=================================================================
7728  032444  004537  010110  JSR     R5,RXCHAR       ;READ/CHK DATA6(026), RCV DATA7
7729  032450  000026          026
7730  032452  000001          RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
7731  032454  020000          NCRACT                  ;DON'T CHECK FOR INITIAL RDA=0
7732  032456  103003          BCC     .+8.            ;BR IF NO ERROR
7733  032460                  ERROR                   ;REPORT STACKED ERROR
7734  032460  104460                                                  TRAP    C$ERROR
7735  032462                  ESCAPE  TST             ;SKIP TO END OF TEST
7736  032462  104410                                                  TRAP    C$ESCAPE
7737  032464  001002                                                  .WORD   L10063-.
7738
7739  032466  004537  010110  JSR     R5,RXCHAR       ;READ/CHK DATA7(037), RCV FIRST FLAG
7740  032472  001037          RXEOM!037               ; & CHECK FOR REOM=1
7741  032474  000000          0
7742  032476  060000          NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
7743  032500  103003          BCC     .+8.            ;BR IF NO ERROR
7744  032502                  ERROR                   ;REPORT STACKED ERROR
7745  032502  104460                                                  TRAP    C$ERROR
7746  032504                  ESCAPE  TST             ;SKIP TO END OF TEST
7747  032504  104410                                                  TRAP    C$ESCAPE
7748  032506  000760                                                  .WORD   L10063-.
7749  032510          ENDSUB
7750  032510                                                  L10067:
7751  032510  104403                                                  TRAP    C$ESUB
7752
7753                  ;-----------------------------------------------------------------
7754                  ; SUBROUTINE # 5: 6 BIT CHARACTERS
7755  032512          ;-----------------------------------------------------------------
7756  032512          BGNSUB
7757  032512  104402                                          T25.5:  TRAP    C$BSUB
7758  032514  004537  007400  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7759  032520  000226          SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7760  032522  000006          BIT2!BIT1               ;INITIALLY: TXCL=8 BITS / RXCL=6 BITS
7761  032524  103003          BCC     .+8.            ;BR IF NO ERROR
7762  032526                  ERROR                   ;REPORT STACKED ERROR
7763  032526  104460                                                  TRAP    C$ERROR
7764  032530                  ESCAPE  SUB             ;SKIP TO END OF TEST
7765  032530  104410                                                  TRAP    C$ESCAPE
7766  032532  000344                                                  .WORD   L10070-.
7767
7768  032534  004537  010010  JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
7769  032540  000001          TSOM
7770  032542  000007          7.
7771  032544  004537  010010  JSR     R5,TXCTRL       ;CLEAR TSOM
7772  032550  000000          000
7773  032552  000000          0
```

```
7774  032554  004537  007676          JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
7775  032560  000123                  123
7776  032562  000010                  8.
7777  032564  103003                  BCC     .+8.            ;BR IF NO ERROR
7778  032566                          ERROR                   ;REPORT STACKED ERROR
7779  032566  104460                                                  TRAP    C$ERROR
7780  032570                          ESCAPE  SUB             ;SKIP TO END OF TEST
7781  032570  104410                                                  TRAP    C$ESCAPE
7782  032572  000304                                                  .WORD   L10070-.
7783
7784  032574  004537  007676          JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7785  032600  000321                  321
7786  032602  000010                  8.
7787  032604  103003                  BCC     .+8.            ;BR IF NO ERROR
7788  032606                          ERROR                   ;REPORT STACKED ERROR
7789  032606  104460                                                  TRAP    C$ERROR
7790  032610                          ESCAPE  SUB             ;SKIP TO END OF TEST
7791  032610  104410                                                  TRAP    C$ESCAPE
7792  032612  000264                                                  .WORD   L10070-.
7793
7794  032614  004537  003734          JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 6 BITS TOO
7795  032620  120407                  PCR
7796  032622  000306                  BIT7!BIT6!BIT2!BIT1
7797  032624  004537  007676          JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7798  032630  000111                  111
7799  032632  000010                  8.
7800  032634  103003                  BCC     .+8.            ;BR IF NO ERROR
7801  032636                          ERROR                   ;REPORT STACKED ERROR
7802  032636  104460                                                  TRAP    C$ERROR
7803  032640                          ESCAPE  SUB             ;SKIP TO END OF TEST
7804  032640  104410                                                  TRAP    C$ESCAPE
7805  032642  000234                                                  .WORD   L10070-.
7806                          ;=============================================================
7807  032644  012703  033471          MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7808  032650  112337  032660  10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7809
7810  032654  004537  007676          JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7811  032660  000000          1$:     000                     ;** HOLE FOR NEXT CHARACTER **
7812  032662  000006                  6.
7813  032664  103003                  BCC     .+8.            ;BR IF NO ERROR
7814  032666                          ERROR                   ;REPORT STACKED ERROR
7815  032666  104460                                                  TRAP    C$ERROR
7816  032670                          ESCAPE  SUB             ;SKIP TO END OF TEST
7817  032670  104410                                                  TRAP    C$ESCAPE
7818  032672  000204                                                  .WORD   L10070-.
7819
7820  032674  022703  033477          CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7821  032700  001363                  BNE     10$             ; IF NOT, TX ANOTHER ONE
7822                          ;=============================================================
7823  032702  004537  010010          JSR     R5,TXCTRL       ;SET TEOM
7824  032706  000002                  TEOM
7825  032710  000000                  0
7826  032712  004537  011614          JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7827  032716  000044                  36.
7828
7829  032720  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
```

```
7830  032724  000523                  RXSOM!123                ; & CHECK FOR RSOM=1
7831  032726  000000                  0
7832  032730  100000                  NOCRDA                   ;NO INITIAL CHECK OF RDA=0
7833  032732  103003                  BCC     .+8.             ;BR IF NO ERROR
7834  032734                          ERROR                    ;REPORT STACKED ERROR
7835  032734  104460                                                   TRAP    C$ERROR
7836  032736                          ESCAPE  SUB              ;SKIP TO END OF TEST
7837  032736  104410                                                   TRAP    C$ESCAPE
7838  032740  000136                                                   .WORD   L10070-.
7839
7840  032742  004537  010110          JSR     R5,RXCHAR        ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7841  032746  000321                  321
7842  032750  000000                  0
7843  032752  100000                  NOCRDA                   ;NO INITIAL CHECK OF RDA=0
7844  032754  103003                  BCC     .+8.             ;BR IF NO ERROR
7845  032756                          ERROR                    ;REPORT STACKED ERROR
7846  032756  104460                                                   TRAP    C$ERROR
7847  032760                          ESCAPE  SUB              ;SKIP TO END OF TEST
7848  032760  104410                                                   TRAP    C$ESCAPE
7849  032762  000114                                                   .WORD   L10070-.
7850                          ;=================================================================
7851  032764  012703  033470          MOV     #T30TBL,R3       ;SET UP DATA TABLE POINTER
7852  032770  112337  033006  40$:    MOVB    (R3)+,4$         ;INSTALL NEXT EXPECTED RX CHARACTER
7853  032774  142737  000300  033006  BICB    #300,4$          ;MASK OUT UNTRANSMITTED BITS
7854
7855  033002  004537  010110          JSR     R5,RXCHAR        ;READ/CHECK NEXT CHARACTER
7856  033006  000000          4$:     000                      ;** HOLE FOR NEXT EXPECTED CHARACTER
7857  033010  000000                  0
7858  033012  100000                  NOCRDA                   ;NO INITIAL CHECK OF RDA=0
7859  033014  103003                  BCC     .+8.             ;BR IF NO ERROR
7860  033016                          ERROR                    ;REPORT STACKED ERROR
7861  033016  104460                                                   TRAP    C$ERROR
7862  033020                          ESCAPE  SUB              ;SKIP TO END OF TEST
7863  033020  104410                                                   TRAP    C$ESCAPE
7864  033022  000054                                                   .WORD   L10070-.
7865
7866  033024  022703  033475          CMP     #T30TBL+5,R3     ;ALL CHARACTERS CHECKED ?
7867  033030  001357                  BNE     40$              ; IF NOT, CHECK ANOTHER ONE
7868                          ;=================================================================
7869  033032  004537  010110          JSR     R5,RXCHAR        ;READ/CHK DATA6(066), RCV DATA7
7870  033036  000066                  066
7871  033040  000001                  RERCHK                   ; & CHECK RERR BIT=0 (GOOD CRC)
7872  033042  020000                  NCRACT                   ;DON'T CHECK FOR INITIAL RDA=0
7873  033044  103003                  BCC     .+8.             ;BR IF NO ERROR
7874  033046                          ERROR                    ;REPORT STACKED ERROR
7875  033046  104460                                                   TRAP    C$ERROR
7876  033050                          ESCAPE  TST              ;SKIP TO END OF TEST
7877  033050  104410                                                   TRAP    C$ESCAPE
7878  033052  000414                                                   .WORD   L10063-.
7879
7880  033054  004537  010110          JSR     R5,RXCHAR        ;READ/CHK DATA7(077), RCV FIRST FLAG
7881  033060  001077                  RXEOM!077                ; & CHECK FOR REOM=1
7882  033062  000000                  0
7883  033064  060000                  NFCRDA!NCRACT            ;DON'T CHECK FOR FINAL RDA=RXACT=1
7884  033066  103003                  BCC     .+8.             ;BR IF NO ERROR
7885  033070                          ERROR                    ;REPORT STACKED ERROR
```

```
7886  033070  104460                                                    TRAP    C$ERROR
7887  033072
7888  033072  104410                                                    TRAP    C$ESCAPE
7889  033074  000372                                                    .WORD   L10063-.
7890  033076                  ENDSUB
7891  033076                                                          L10070:
7892  033076  104403                                                    TRAP    C$ESUB
7893                  ;-----------------------------------------------------------------
7894                  ; SUBROUTINE # 6: 7 BIT CHARACTERS
7895                  ;-----------------------------------------------------------------
7896  033100                  BGNSUB
7897  033100                                                          T25.6:
7898  033100  104402                                                    TRAP    C$BSUB
7899  033102  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7900  033106  000226                  SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7901  033110  000007                  RXDL                    ;INITIALLY: TXCL=8 BITS / RXCL=7 BITS
7902  033112  103003                  BCC     .+8.            ;BR IF NO ERROR
7903  033114                          ERROR                   ;REPORT STACKED ERROR
7904  033114  104460                                                    TRAP    C$ERROR
7905  033116                  ESCAPE  SUB             ;SKIP TO END OF TEST
7906  033116  104410                                                    TRAP    C$ESCAPE
7907  033120  000344                                                    .WORD   L10071-.
7908
7909  033122  004537  010010          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
7910  033126  000001                  TSOM
7911  033130  000007                  7.
7912  033132  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
7913  033136  000000                  000
7914  033140  000000                  0
7915  033142  004537  007676          JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
7916  033146  000123                  123
7917  033150  000010                  8.
7918  033152  103003                  BCC     .+8.            ;BR IF NO ERROR
7919  033154                          ERROR                   ;REPORT STACKED ERROR
7920  033154  104460                                                    TRAP    C$ERROR
7921  033156                  ESCAPE  SUB             ;SKIP TO END OF TEST
7922  033156  104410                                                    TRAP    C$ESCAPE
7923  033160  000304                                                    .WORD   L10071-.
7924
7925  033162  004537  007676          JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7926  033166  000321                  321
7927  033170  000010                  8.
7928  033172  103003                  BCC     .+8.            ;BR IF NO ERROR
7929  033174                          ERROR                   ;REPORT STACKED ERROR
7930  033174  104460                                                    TRAP    C$ERROR
7931  033176                  ESCAPE  SUB             ;SKIP TO END OF TEST
7932  033176  104410                                                    TRAP    C$ESCAPE
7933  033200  000264                                                    .WORD   L10071-.
7934
7935  033202  004537  003734          JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 6 BITS TOO
7936  033206  120407                  PCR
7937  033210  000347                  TXDL!RXDL
7938  033212  004537  007676          JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7939  033216  000111                  111
7940  033220  000010                  8.
7941  033222  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
7942  033224
7943  033224  104460                          ERROR                   ;REPORT STACKED ERROR
                                                                                TRAP    C$ERROR
7944  033226
7945  033226  104410                          ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
7946  033230  000234                                                            .WORD   L10071-.
7947                          ;=============================================================
7948  033232  012703  033471                  MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7949  033236  112337  033246          10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7950
7951  033242  004537  007676                  JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7952  033246  000000          1$:     000             ;** HOLE FOR NEXT CHARACTER **
7953  033250  000007                          7.
7954  033252  103003                          BCC     .+8.            ;BR IF NO ERROR
7955  033254                                  ERROR                   ;REPORT STACKED ERROR
7956  033254  104460                                                          TRAP    C$ERROR
7957  033256
7958  033256  104410                          ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
7959  033260  000204                                                            .WORD   L10071-.
7960
7961  033262  022703  033477                  CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7962  033266  001363                          BNE     10$             ; IF NOT, TX ANOTHER ONE
                              ;=============================================================
7964  033270  004537  010010                  JSR     R5,TXCTRL       ;SET TEOM
7965  033274  000002                          TEOM
7966  033276  000000                          0
7967  033300  004537  011614                  JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7968  033304  000044                          36.
7969
7970  033306  004537  010110                  JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7971  033312  000523                          RXSOM!123               ; & CHECK FOR RSOM=1
7972  033314  000000                          0
7973  033316  100000                          NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7974  033320  103003                          BCC     .+8.            ;BR IF NO ERROR
7975  033322                                  ERROR                   ;REPORT STACKED ERROR
7976  033322  104460                                                          TRAP    C$ERROR
7977  033324
7978  033324  104410                          ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
7979  033326  000136                                                            .WORD   L10071-.
7980
7981  033330  004537  010110                  JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7982  033334  000321                          321
7983  033336  000000                          0
7984  033340  100000                          NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7985  033342  103003                          BCC     .+8.            ;BR IF NO ERROR
7986  033344                                  ERROR                   ;REPORT STACKED ERROR
7987  033344  104460                                                          TRAP    C$ERROR
7988  033346
7989  033346  104410                          ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
7990  033350  000114                                                            .WORD   L10071-.
7991                          ;=============================================================
7992  033352  012703  033470                  MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7993  033356  112337  033374          40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
7994  033362  142737  000200  033374        BICB    #200,4$         ;MASK OUT UNTRANSMITTED BITS
7995
7996  033370  004537  010110                  JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7997  033374  000000          4$:     000             ;** HOLE FOR NEXT EXPECTED CHARACTER
```

```
7998  033376  000000                  0
7999  033400  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
8000  033402  103003                  BCC     .+8.            ;BR IF NO ERROR
8001  033404                          ERROR                   ;REPORT STACKED ERROR
8002  033404  104460                                                  TRAP    C$ERROR
8003  033406                          ESCAPE  SUB             ;SKIP TO END OF TEST
8004  033406  104410                                                  TRAP    C$ESCAPE
8005  033410  000054                                                  .WORD   L10071-.
8006
8007  033412  022703  033475          CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
8008  033416  001357                  BNE     40$             ; IF NOT, CHECK ANOTHER ONE
8009          ;=======================================================================
8010  033420  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA6(066), RCV DATA7
8011  033424  000066                  066
8012  033426  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
8013  033430  020000                  NCRACT                  ;DON'T CHECK FOR INITIAL RDA=0
8014  033432  103003                  BCC     .+8.            ;BR IF NO ERROR
8015  033434                          ERROR                   ;REPORT STACKED ERROR
8016  033434  104460                                                  TRAP    C$ERROR
8017  033436                          ESCAPE  TST             ;SKIP TO END OF TEST
8018  033436  104410                                                  TRAP    C$ESCAPE
8019  033440  000026                                                  .WORD   L10063-.
8020
8021  033442  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA7(177), RCV FIRST FLAG
8022  033446  001177                  RXEOM!177               ; & CHECK FOR REOM=1
8023  033450  000000                  0
8024  033452  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
8025  033454  103003                  BCC     .+8.            ;BR IF NO ERROR
8026  033456                          ERROR                   ;REPORT STACKED ERROR
8027  033456  104460                                                  TRAP    C$ERROR
8028  033460                          ESCAPE  TST             ;SKIP TO END OF TEST
8029  033460  104410                                                  TRAP    C$ESCAPE
8030  033462  000004                                                  .WORD   L10063-.
8031  033464
8032  033464                  ENDSUB                                  L10071:
8033  033464  104403                                                  TRAP    C$ESUB
8034  033466
8035  033466                  ENDTST                                  L10063:
8036  033466  104401                                                  TRAP    C$ETST
8037
8038  033470     111          ;-------------------------------
8039  033471     222          T30TBL: .BYTE   111     ;D1
8040  033472     333                  .BYTE   222     ;D2
8041  033473     044                  .BYTE   333     ;D3
8042  033474     155                  .BYTE   044     ;D4
8043  033475     266                  .BYTE   155     ;D5
8044  033476     377                  .BYTE   266     ;D6
8045          033500                   .BYTE   377     ;D7
8046                                   .EVEN
                                ;-------------------------------
```

```
8047
8048
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8060
8061  033500
8062  033500  004737  005420
8063
8064
8065
8066  033504  004537  003734
8067  033510  120013
8068  033512  000200
8069  033514  004537  003734
8070  033520  120006
8071  033522  000300
8072  033524  004537  003734
8073  033530  120007
8074  033532  000000
8075
8076  033534  004537  003734
8077  033540  120300
8078  033542  000031
8079  033544  004537  003734
8080  033550  120000
8081  033552  000030
8082
8083  033554  004537  003734
8084  033560  120404
8085  033562  000000
8086  033564  004537  003734
8087  033570  120405
8088  033572  000007
8089
8090  033574  004537  003734
8091  033600  120000
8092  033602  000172
8093
8094  033604  004537  006036
8095  033610  000001
8096  033612  103003
8097  033614
8098  033614  104460
8099  033616
8100  033616  104410
8101  033620  000234
8102
```

```
TEST 26 -- TX 'SPACING SEQUENCE'

.SBTTL  TEST 26 -- TX 'SPACING SEQUENCE'

;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;*
;*      TEST 26 -- TX 'SPACING SEQUENCE'
;*
;*  THE  TRANSMITTER IS INITIALIZED AND THE 'SPACING SEQUENCE' IS FORCED
;*  BY  ASSERTING  BOTH  TSOM  &  TEOM AT THE SAME TIME -- CHECK THE BIT
;*  STREAM FOR ACCURACY (SPACES) AND COMPLETNESS (16 OF THEM). WHEN TXBE
;*  GOES  HIGH  (=  1)  A  SMALL  MESSAGE  IS SENT.
;*
;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
;
;       BGNTST
                                                                T26::
        JSR     PC,INIDMV       ;INIT VIA
;-------------------------------------------------------------------
; SET UP USYRT AND VIA REGISTERS
;-------------------------------------------------------------------
        JSR     R5,WRITEI       ;SET ACR FOR T1 ONE-SHOT MODE
        VIAACR
        200
        JSR     R5,WRITEI       ;LOAD VIA T1L-L
        VIAT1C
        300
        JSR     R5,WRITEI       ;LOAD VIA T1L-H
        VIAT1D
        000

        JSR     R5,WRITEI       ;RESET THE USYRT
        VIAORB
        RTSND!DTR!PRESET
        JSR     R5,WRITEI       ;CLEAR USYRT RESET BIT
        VIAORB
        RTSND!DTR

        JSR     R5,WRITEI       ;LOAD USYRT PCSARL
        PCSARL
        000
        JSR     R5,WRITEI       ;LOAD USYRT PCSARH
        PCSARH
        XYZ                     ;(NO ERROR CHECKING)

        JSR     R5,WRITEI       ;SET UP USYRT
        VIAORB
        RTSND!TXEN!RXEN!DTR!TTLOOP

        JSR     R5,CKTBMT       ;CHK FOR TBMT = 1
        1
        BCC     .+8.            ;IF NO ERROR, PROCEED
        ERROR                   ;ELSE, REPORT IT AND
                                                        TRAP    C$ERROR
        ESCAPE  TST             ;       EXIT THIS TEST
                                                        TRAP    C$ESCAPE
                                                        .WORD   L10072-.
;-------------------------------------------------------------------
```

```
8103
8104                            ; INIT SPACING SEQUENCE BY SETTING TSOM AND TEOM
8105   033622  004537  010010   ;-------------------------------------------------------------
8106   033626  000003   6$:     JSR     R5,TXCTRL       ;SET TSOM AND TEOM
8107   033630  000000           TSOM!TEOM               ; ('SPACING SEQUENCE')
8108                            0
8109   033632  004537  011614           JSR     R5,STEPLU       ;1 TICK TO START SPACE SEQUENCE
8110   033636  000001           1
8111
8112   033640  004537  007256           JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
8113   033644  000010           8.                      ; 8 BIT CHAR/CLOCK TICKS
8114   033646  000000           000                     ; EXPECTED BIT SEQUENCE (0000000)
8115   033650  103003           BCC     .+8.            ;BR IF NO ERROR
8116   033652                   ERROR                   ;REPORT STACKED ERROR
8117   033652  104460                                                   TRAP    C$ERROR
8118   033654                   ESCAPE  TST             ;AND EXIT TEST
8119   033654  104410                                                   TRAP    C$ESCAPE
8120   033656  000176                                                   .WORD   L10072-.
8121
8122   033660  004537  006036           JSR     R5,CKTBMT       ;CHK FOR TBMT = 1
8123   033664  000001           1
8124   033666  103003           BCC     .+8.            ;IF NO ERROR, PROCEED
8125   033670                   ERROR                   ;ELSE, REPORT IT AND
8126   033670  104460                                                   TRAP    C$ERROR
8127   033672                   ESCAPE  TST             ;       EXIT THIS TEST
8128   033672  104410                                                   TRAP    C$ESCAPE
8129   033674  000160                                                   .WORD   L10072-.
8130                            ;-------------------------------------------------------------
8131   033676  004537  010010           JSR     R5,TXCTRL       ;CLEAR TEOM
8132   033702  000001           TSOM
8133   033704  000000           0
8134
8135   033706  004537  007256           JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
8136   033712  000010           8.                      ; 8 BIT CHAR/CLOCK TICKS
8137   033714  000000           000                     ; EXPECTED BIT SEQUENCE (00000000)
8138   033716  103003           BCC     .+8.            ;BR IF NO ERROR
8139   033720                   ERROR                   ;REPORT STACKED ERROR
8140   033720  104460                                                   TRAP    C$ERROR
8141   033722                   ESCAPE  TST             ;AND EXIT TEST
8142   033722  104410                                                   TRAP    C$ESCAPE
8143   033724  000130                                                   .WORD   L10072-.
8144
8145   033726  004537  006036           JSR     R5,CKTBMT       ;CHK FOR TBMT = 1
8146   033732  000001           1
8147   033734  103003           BCC     .+8.            ;IF NO ERROR, PROCEED
8148   033736                   ERROR                   ;ELSE, REPORT IT AND
8149   033736  104460                                                   TRAP    C$ERROR
8150   033740                   ESCAPE  TST             ;       EXIT THIS TEST
8151   033740  104410                                                   TRAP    C$ESCAPE
8152   033742  000112                                                   .WORD   L10072-.
8153                            ;-------------------------------------------------------------
8154                            ; SEND 'SHORT MESSAGE' (125,252)
8155                            ;-------------------------------------------------------------
8156   033744  004537  003734           JSR     R5,WRITEI       ;LOAD 125 CHARACTER
8157   033750  120402           TDSRL
8158   033752  000125           125
```

```
CVDMCA.P11    10-DEC-80 09:14          TEST 26 -- TX 'SPACING SEQUENCE'

 8159  033754  004537  010010                JSR     R5,TXCTRL       ;CLEAR TSOM
 8160  033760  000000                        000
 8161  033762  000000                        0
 8162
 8163  033764  004537  007256                JSR     R5,SERIAL       ;READ FLAG CHARACTER (176) VIA TSO
 8164  033770  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 8165  033772  000176                        176                     ; EXPECTED BIT SEQUENCE (01111110)
 8166  033774  103003                        BCC     .+8.            ;BR IF NO ERROR
 8167  033776                                ERROR                   ;REPORT STACKED ERROR
 8168  033776  104460                                                                TRAP    CSERROR
 8169  034000                                ESCAPE  TST             ;AND EXIT TEST
 8170  034000  104410                                                                TRAP    CSESCAPE
 8171  034002  000052                                                                .WORD   L10072-.
 8172
 8173  034004  004537  003734                JSR     R5,WRITEI       ;LOAD 252 CHARACTER
 8174  034010  120402                        TDSRL
 8175  034012  000252                        252
 8176
 8177  034014  004537  007256                JSR     R5,SERIAL       ;READ 1ST DATA CHARACTER (125) VIA TSO
 8178  034020  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 8179  034022  000252                        252                     ; EXPECTED BIT SEQUENCE (01010110)
 8180  034024  103003                        BCC     .+8.            ;BR IF NO ERROR
 8181  034026                                ERROR                   ;REPORT STACKED ERROR
 8182  034026  104460                                                                TRAP    CSERROR
 8183  034030                                ESCAPE  TST             ;AND EXIT TEST
 8184  034030  104410                                                                TRAP    CSESCAPE
 8185  034032  000022                                                                .WORD   L10072-.
 8186
 8187  034034  004537  007256                JSR     R5,SERIAL       ;READ 2ND DATA CHARACTER (252) VIA TSO
 8188  034040  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 8189  034042  000125                        125                     ; EXPECTED BIT SEQUENCE (10101010)
 8190  034044  103003                        BCC     .+8.            ;BR IF NO ERROR
 8191  034046                                ERROR                   ;REPORT STACKED ERROR
 8192  034046  104460                                                                TRAP    CSERROR
 8193  034050                                ESCAPE  TST             ;AND EXIT TEST
 8194  034050  104410                                                                TRAP    CSESCAPE
 8195  034052  000002                                                                .WORD   L10072-.
 8196  034054                        ENDTST
 8197  034054                                                                L10072:
 8198  034054  104401                                                                TRAP    CSETST
```

```
CVDMCA.P11    10-DEC-80 09:14        TEST 27 -- FIFO OVERRUN INTEGRITY TEST

                                     .SBTTL  TEST 27 -- FIFO OVERRUN INTEGRITY TEST

                                     ;++*************************************************************** ************
                                     ;*
                                     ;*      TEST 27 -- FIFO OVERRUN INTEGRITY TEST
                                     ;*
                                     ;*
                                     ;* THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS TO FILL
                                     ;* THE 8 CHARACTER RECEIVER FIFO UNTIL RXOR WITH THE CHARACTERS:
                                     ;*      (SYNCH),000,377,125,252,347,030,303,074,125.
                                     ;* THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. OF IMPORTANCE
                                     ;* IS THE INTEGRITY OF THE LAST OVERRUN-CAUSING FIFO CHARACTER (IT SHOULD
                                     ;* REMAIN INTACT).
                                     ;* NOTE THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE  CHARACTERS SINCE THEY
                                     ;* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
                                     ;* RDA = 0).
                                     ;*
                                     ;*
                                     ;--*************************************************************************

                                             BGNTST
034056                                                                   T27::
034056  004737  005420                      JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
034062  004537  007400                      JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
034066  043626                            DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
034070  000000                            0                       ;USE 8 BIT CHARS
034072  103003                            BCC     .+8.            ;BR IF NO ERROR
034074                                    ERROR                   ;REPORT STACKED ERROR
034074  104460                                                                   TRAP    C$ERROR
034076                                    ESCAPE  TST             ;SKIP TO END OF TEST
034076  104410                                                                   TRAP    C$ESCAPE
034100  000324                                                                   .WORD   L10073-.
034102  004537  010010                    JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
034106  000001                            TSOM                    ;
034110  000007                            7.
                                     ;
034112  004537  010010                    JSR     R5,TXCTRL       ;CLEAR TSOM
034116  000000                            000
034120  000000                            0
                                     ;-------------------------------------------------------------------
                                     ; FILL THE FIFO WITH CHARACTERS AND FORCE OVERRUN
                                     ;-------------------------------------------------------------------
034122  012702  034426                    MOV     #TXTBL4,R2      ;SET UP TABLE POINTER
034126  112237  034136            5$:     MOVB    (R2)+,10$       ;SETUP TRANSMIT CHARACTER
034132  004537  007676                    JSR     R5,TXCHAR       ;TRANSMIT A CHARACTER
034136  000000                    10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
034140  100010                            NCTBMT*256.!8.          ;NO CHECK OF INITIAL TBMT=0
034142  103003                            BCC     .+8.            ;BR IF NO ERROR
034144                                    ERROR                   ;REPORT STACKED ERROR
034144  104460                                                                   TRAP    C$ERROR
034146                                    ESCAPE  TST             ;SKIP TO END OF TEST
034146  104410                                                                   TRAP    C$ESCAPE
034150  000254                                                                   .WORD   L10073-.
```

```
                        TEST 27 -- FIFO OVERRUN INTEGRITY TEST

8255  034152  022702  034441            CMP     #TXEND4,R2      ;CONTINUE TRANSMITTING UNTIL
8256  034156  001363                    BNE     5$              ;OVERRUN.
8257
8258  034160  004537  006476            JSR     R5,CKROR        ;VERIFY RECEIVER OVERRUN OCCURED
8259  034164  000001                    1
8260  034166  103003                    BCC     .+8.            ;BR IF OK
8261  034170                            ERROR                   ;REPORT STACKED ERROR
8262  034170  104460                                                            TRAP    C$ERROR
8263  034172                            ESCAPE  TST             ;SKIP TO END OF TEST
8264  034172  104410                                                            TRAP    C$ESCAPE
8265  034174  000230                                                            .WORD   L10073-.
8266                                ;-------------------------------------------------------
8267                                ; THE RECEIVER SHOULD HAVE OVERRUN. VERIFY THIS AND READ/VERIFY
8268                                ; THE REMAINING FIFO CHARACTERS (ESPECIALLY THE LAST ONE).
8269                                ;-------------------------------------------------------
8270  034176  004537  010110            JSR     R5,RXCHAR       ;READ & CHK CHARACTER
8271  034202  000226                    SYNCH                   ; (THIS CHARACTER ISN'T AFFECTED)
8272  034204  000000                    0
8273  034206  100000                    NOCRDA          ;NO INITIAL CHECK OF RDA=0
8274  034210  103003                    BCC     .+8.            ;BR IF NO ERROR
8275  034212                            ERROR                   ;REPORT STACKED ERROR
8276  034212  104460                                                            TRAP    C$ERROR
8277  034214                            ESCAPE  TST             ;SKIP TO END OF TEST
8278  034214  104410                                                            TRAP    C$ESCAPE
8279  034216  000206                                                            .WORD   L10073-.
8280
8281  034220  004537  010110            JSR     R5,RXCHAR       ;READ & CHK CHARACTER
8282  034224  000000                    000                     ; (THIS IS THE OVERWRITTEN CHARACTER)
8283  034226  000000                    0                       ; ...WAS A 'SYNCH' BEFORE OVERWRITTEN
8284  034230  100000                    NOCRDA          ;NO INITIAL CHECK OF RDA=0
8285  034232  103003                    BCC     .+8.            ;BR IF NO ERROR
8286  034234                            ERROR                   ;REPORT STACKED ERROR
8287  034234  104460                                                            TRAP    C$ERROR
8288  034236                            ESCAPE  TST             ;SKIP TO END OF TEST
8289  034236  104410                                                            TRAP    C$ESCAPE
8290  034240  000164                                                            .WORD   L10073-.
8291
8292  034242  012702  034431            MOV     #TXTBL4+3,R2    ;SET UP TABLE POINTER
8293  034246  112237  034256    15$:    MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
8294
8295  034252  004537  010110            JSR     R5,RXCHAR       ;READ & CHK CHARACTER
8296  034256  000000    20$:            000                     ;** HOLE FOR EXPECTED RECEIVE CHAR.
8297  034260  000000                    0                       ;
8298  034262  100000                    NOCRDA          ;NO INITIAL CHECK OF RDA=0
8299  034264  103003                    BCC     .+8.            ;BR IF NO ERROR
8300  034266                            ERROR                   ;REPORT STACKED ERROR
8301  034266  104460                                                            TRAP    C$ERROR
8302  034270                            ESCAPE  TST             ;SKIP TO END OF TEST
8303  034270  104410                                                            TRAP    C$ESCAPE
8304  034272  000132                                                            .WORD   L10073-.
8305  034274  022702  034436            CMP     #TXEND4-3,R2
8306  034300  001362                    BNE     15$
8307
8308  034302  012702  000004            MOV     #4,R2           ;TRANSMIT 4 EXTRA CHARACTERS TO AVOID
8309  034306  004537  007676    30$:    JSR     R5,TXCHAR       ;TRANSMITTER UNDERRUN.
8310  034312  000333                    333                     ;FILLER CHARACTER
```

```
8311  034314  100010                  NCTBMT*256.!8.          ;NO CHECK OF INITIAL TBMT=0
8312  034316  077205                  SOB     R2,30$
8313
8314  034320  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
8315  034324  000303                  303
8316  034326  000000                  0
8317  034330  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
8318  034332  103003                  BCC     .+8.            ;BR IF NO ERROR
8319  034334                          ERROR                   ;REPORT STACKED ERROR
8320  034334  104460                                                          TRAP    C$ERROR
8321  034336                          ESCAPE  TST             ;SKIP TO END OF TEST
8322  034336  104410                                                          TRAP    C$ESCAPE
8323  034340  000064                                                          .WORD   L10073-.
8324
8325  034342  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
8326  034346  000074                  074
8327  034350  000000                  0
8328  034352  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
8329  034354  103003                  BCC     .+8.            ;BR IF NO ERROR
8330  034356                          ERROR                   ;REPORT STACKED ERROR
8331  034356  104460                                                          TRAP    C$ERROR
8332  034360                          ESCAPE  TST             ;SKIP TO END OF TEST
8333  034360  104410                                                          TRAP    C$ESCAPE
8334  034362  000042                                                          .WORD   L10073-.
8335
8336  034364  004537  010110          JSR     R5,RXCHAR       ;* READ & CHK FINAL CHARACTER
8337  034370  000125                  125                     ;*
8338  034372  000000                  0                       ;*
8339  034374  100000                  NOCRDA          ;* NO INITIAL CHECK OF RDA=0
8340  034376  103003                  BCC     .+8.            ;* BR IF NO ERROR
8341  034400                          ERROR                   ;* REPORT STACKED ERROR
8342  034400  104460                                                          TRAP    C$ERROR
8343  034402                          ESCAPE  TST             ;* SKIP TO END OF TEST
8344  034402  104410                                                          TRAP    C$ESCAPE
8345  034404  000020                                                          .WORD   L10073-.
8346
8347  034406  004537  011532          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
8348  034412  000011                  9.
8349  034414  103003                  BCC     .+8.            ;BR IF NO ERROR
8350  034416                          ERROR                   ;REPORT STACKED ERROR
8351  034416  104460                                                          TRAP    C$ERROR
8352  034420                          ESCAPE  TST             ;SKIP TO END OF TEST
8353  034420  104410                                                          TRAP    C$ESCAPE
8354  034422  000002                                                          .WORD   L10073-.
8355  034424                  ENDTST
8356  034424                                                          L10073:
8357  034424  104401                                                          TRAP    C$ETST
8358
8359  034426      226         ;-----------------------------
8360  034427      226         TXTBL4: .BYTE   226
8361  034430      000                 .BYTE   226
8362  034431      377                 .BYTE   000
8363  034432      125                 .BYTE   377
8364  034433      252                 .BYTE   125
8365  034434      347                 .BYTE   252
8366  034435      030                 .BYTE   347
                                      .BYTE   030
```

```
8367  034434  303                     .BYTE   303
8368  034435  074                     .BYTE   074
8369  034440  125                     .BYTE   125
8370  034441  000             TXEND4: .BYTE   000
8371                                  .EVEN
8372                          ;-------------------------------
```

```
                                .SBTTL  TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

                                ;+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST
                                ;*
                                ;*  THE USYRT IS INITIALIZED AND THREE SUBTESTS ARE PERFORMED.
                                ;*
                                ;*     1 -- AN OVERRUN CONDITION IS FORCED, RECEIVER STATUS REGISTER IS
                                ;*          READ TWICE: ONCE TO VERIFY ROR BIT = 1, AND AGAIN TO VERIFY
                                ;*          THAT THE FIRST READ CLEARED ROR .
                                ;*
                                ;*     2 -- AN OVERRUN CONDITION IS FORCED (BY THE SAME TECHNIQUE USED IN
                                ;*         (2), THE USYRT IS RESET AND THE PROPER STATE OF ALL REGISTERS
                                ;*         IS VERIFIED.
                                ;*
                                ;*     3 -- AN OVERRUN CONDITION IS FORCED (AS ABOVE). RXE IS THEN DROPPED
                                ;*          AND A DELAY IS PROVIDED TO ALLOW TIME FOR THE FIFO TO FLUSH
                                ;*          (CAUSED BY RDA GOING LOW). RXE IS THEN RE-INITIALIZED AND ROR
                                ;*          IS CHECKED = 0. THE RECEIVER IS THEN RE-SYNCHED AND THE TEST IS
                                ;*          TERMINATED.
                                ;*
                                ;*
                                ;--++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;
                                        BGNTST
                                                                                T28::
                                ;*************** SUBTEST #1 ***************
                                        BGNSUB
                                                                                T28.1:
 8373                                                                                   TRAP    C$BSUB
 8374   034442  104402
 8375   034444  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
 8376
 8377   034450  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
 8378   034454  043626                  DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
 8379   034456  000000                  0                       ;USE 8 BIT CHARS
 8380   034460  103003                  BCC     .+8.            ;BR IF NO ERROR
 8381   034462                          ERROR                   ;REPORT STACKED ERROR
 8382   034462  104460                                                          TRAP    C$ERROR
 8383   034464                          ESCAPE  SUB             ;SKIP TO END OF TEST
 8384   034464  104410                                                          TRAP    C$ESCAPE
 8385   034466  000112                                                          .WORD   L10075-.
 8386
 8387   034470  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
 8388   034474  000001                  TSOM                    ;
 8389   034476  000007                  7.
 8390   034500  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
 8391   034504  000001                  TSOM                    ;
 8392   034506  000010                  8.
 8393   034510  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
 8394   034514  000000                  000
 8395   034516  000000                  0
 8396
 8397   034520  004537  006476          JSR     R5,CKROR        ;CHECK FOR RXOR = 0
 8398   034524  000000                  0
 8399   034526  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
8429  034530                        ERROR                   ;REPORT STACKED ERROR
8430  034530  104460                                                          TRAP    C$ERROR
8431  034532                        ESCAPE  SUB             ;SKIP TO END OF TEST
8432  034532  104410                                                          TRAP    C$ESCAPE
8433  034534  000044                                                          .WORD   L10075-.
8434
8435  034536  004537  011614        JSR     R5,STEPLU       ;FORCE RECEIVER OVERRUN
8436  034542  000116                78.
8437
8438  034544  004537  006476        JSR     R5,CKROR        ;CHECK FOR RXOR = 1
8439  034550  000001                1
8440  034552  103003                BCC     .+8.            ;BR IF NO ERROR
8441  034554                        ERROR                   ;REPORT STACKED ERROR
8442  034554  104460                                                          TRAP    C$ERROR
8443  034556                        ESCAPE  TST             ;SKIP TO END OF TEST
8444  034556  104410                                                          TRAP    C$ESCAPE
8445  034560  000262                                                          .WORD   L10074-.
8446
8447  034562  004537  006476        JSR     R5,CKROR        ;CHECK FOR RXOR = 0
8448  034566  000000                0
8449  034570  103003                BCC     .+8.            ;BR IF NO ERROR
8450  034572                        ERROR                   ;REPORT STACKED ERROR
8451  034572  104460                                                          TRAP    C$ERROR
8452  034574                        ESCAPE  SUB             ;SKIP TO END OF TEST
8453  034574  104410                                                          TRAP    C$ESCAPE
8454  034576  000002                                                          .WORD   L10075-.
8455  034600                        ENDSUB
8456  034600                                                          L10075:
8457  034600  104403                                                          TRAP    C$ESUB
8458                        ;**************** SUBTEST #2 ****************
8459  034602                        BGNSUB
8460  034602                                                          T28.2:
8461  034602  104402                                                          TRAP    C$BSUB
8462  034604  004737  005420        JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
8463
8464  034610  004537  007400        JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
8465  034614  043626                DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
8466  034616  000000                0                       ;USE 8 BIT CHARS
8467  034620  103003                BCC     .+8.            ;BR IF NO ERROR
8468  034622                        ERROR                   ;REPORT STACKED ERROR
8469  034622  104460                                                          TRAP    C$ERROR
8470  034624                        ESCAPE  SUB             ;SKIP TO END OF TEST
8471  034624  104410                                                          TRAP    C$ESCAPE
8472  034626  000054                                                          .WORD   L10076-.
8473
8474  034630  004537  010010        JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
8475  034634  000001                TSOM                    ;
8476  034636  000007                7.
8477  034640  004537  010010        JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
8478  034644  000001                TSOM                    ;
8479  034646  000010                8.
8480  034650  004537  010010        JSR     R5,TXCTRL       ;CLEAR TSOM
8481  034654  000000                000
8482  034656  000000                0
8483  034660  004537  011614        JSR     R5,STEPLU       ;FORCE RECEIVER OVERRUN
8484  034664  000116                78.
```

```
8485
8486  034666  004537  005126          JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
8487  034672  103003                  BCC     .+8.            ;BR IF NO ERROR
8488  034674                          ERROR                   ;REPORT STACKED ERROR
8489  034674  104460                                                  TRAP    C$ERROR
8490  034676                          ESCAPE TST              ;SKIP TO END OF TEST
8491  034676  104410                                                  TRAP    C$ESCAPE
8492  034700  000142                                                  .WORD   L10074-.
8493  034702                          ENDSUB
8494  034702                                                      L10076:
8495  034702  104403                                                  TRAP    C$ESUB
8496                        ;*************** SUBTEST #3 ***************
8497  034704                          BGNSUB
8498  034704                                                      T28.3:
8499  034704  104402                                                  TRAP    C$BSUB
8500  034706  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
8501
8502  034712  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
8503  034716  043626                  DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
8504  034720  000000                  0                       ;USE 8 BIT CHARS
8505  034722  103003                  BCC     .+8.            ;BR IF NO ERROR
8506  034724                          ERROR                   ;REPORT STACKED ERROR
8507  034724  104460                                                  TRAP    C$ERROR
8508  034726                          ESCAPE  SUB             ;SKIP TO END OF TEST
8509  034726  104410                                                  TRAP    C$ESCAPE
8510  034730  000110                                                  .WORD   L10077-.
8511
8512  034732  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
8513  034736  000001                  TSOM                    ;
8514  034740  000007                  7.
8515  034742  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
8516  034746  000001                  TSOM                    ;
8517  034750  000010                  8.
8518  034752  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
8519  034756  000000                  000
8520  034760  000000                  0
8521  034762  004537  011614          JSR     R5,STEPLU       ;FORCE RECEIVER OVERRUN
8522  034766  000116                  78.
8523
8524  034770  004537  003734          JSR     R5,WRITEI       ;DROP RECEIVER ENABLE (RXEN)
8525  034774  120000                  VIAORB                  ; (RDA SHOULD ALSO DROP, WHICH WILL
8526  034776  000042                  TXEN!TTLOOP             ;  CAUSE FIFO TO FLUSH ITSELF).
8527
8528  035000  012701  000050          MOV     #50,R1          ;DELAY FOR NNN SEC. TO ALLOW FIFO TIME
8529  035004  004737  005312  10$:    JSR     PC,WAIT50       ;TO FLUSH ITSELF.
8530  035010  077103                  SOB     R1,10$
8531
8532  035012  004537  003734          JSR     R5,WRITEI       ;TURN ON RECEIVER ENABLE (RXEN)
8533  035016  120000                  VIAORB
8534  035020  000142                  TXEN!RXEN!TTLOOP
8535
8536  035022  004537  006476          JSR     R5,CKROR        ;VERIFY CLEARING OF RECEIVER OVERRUN
8537  035026  000000                  0
8538  035030  103003                  BCC     .+8.            ;BR IF NO ERROR
8539  035032                          ERROR                   ;REPORT STACKED ERROR
8540  035032  104460                                                  TRAP    C$ERROR
```

TEST 28 -- BCP RX OVERRUN SET AND CLEAR TEST

```
8541  035034
8542  035034  104410          ESCAPE  SUB             ;SKIP TO END OF TEST
8543  035036  000002                                                  TRAP    C$ESCAPE
8544  035040                                                          .WORD   L10077-.
8545  035040
8546  035040  104403          ENDSUB                          L10077:
8547  035042                                                          TRAP    C$ESUB
8548  035042
8549  035042  104401  ENDTST                                 L10074:
                                                                      TRAP    C$ETST
```

```
                                TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION

                                .SBTTL  TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION

                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 29 -- BCP RX SYNC CHARACTER RECOGNITION
                                ;*
                                ;*  THE  FOLLOWING  MESSAGE  IS INITIATED WITHOUT ASSERTING RXE AND ONCE
                                ;*  THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
                                ;*
                                ;*     SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
                                ;*     SYNC SYNC DATA DATA DATA SYNC SYNC
                                ;*
                                ;*  THE  RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
                                ;*  THE  NEXT  TWO SYNC CHARACTERS FOR SYNCRONIZATION, THEN PASS THE REST
                                ;*  OF THE MESSAGE (7 SYNC AND 6 DATA CHARACTERS) THROUGH RXDB REGISTER.
                                ;*
                                ;--**********************************************************************************************
                                ;
                                        BGNTST
                                                                                        T29::
8569  035044
8570
8571  035044  004737  005420            JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
8572
8573  035050  004537  007400            JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
8574  035054  043626                    DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
8575  035056  040000                    NORXEN                  ;USE 8 BIT CHARS; LEAVE RXEN=0
8576  035060  103003                    BCC     .+8.            ;BR IF NO ERROR
8577  035062                            ERROR                   ;REPORT STACKED ERROR
8578  035062  104460                                                                    TRAP    C$ERROR
8579  035064                            ESCAPE  TST             ;SKIP TO END OF TEST
8580  035064  104410                                                                    TRAP    C$ESCAPE
8581  035066  000414                                                                    .WORD   L10100-.
8582
8583  035070  004537  010010            JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
8584  035074  000001                    TSOM                    ; (IGNORED BY RECEIVER)
8585  035076  000007                    7.
8586  035100  004537  010010            JSR     R5,TXCTRL       ;CLEAR TSOM
8587  035104  000000                    000
8588  035106  000000                    0
8589  035110  004537  003734            JSR     R5,WRITEI       ;ENABLE RECEIVER (RXEN => 1)
8590  035114  120000                    VIAORB
8591  035116  000142                    RXEN!TXEN!TTLOOP
8592                                  ;-----------------------------------------------------
8593                                  ; THE RECEIVER SHOULD IGNORE THE NEXT STRING OF CHARACTERS
8594                                  ;-----------------------------------------------------
8595  035120  004537  007676            JSR     R5,TXCHAR       ;LOAD 000, TX 2ND SYNCH
8596  035124  000000                    000
8597  035126  000010                    8.
8598  035130  103003                    BCC     .+8.            ;BR IF NO ERROR
8599  035132                            ERROR                   ;REPORT STACKED ERROR
8600  035132  104460                                                                    TRAP    C$ERROR
8601  035134                            ESCAPE  TST             ;SKIP TO END OF TEST
8602  035134  104410                                                                    TRAP    C$ESCAPE
8603  035136  000344                                                                    .WORD   L10100-.
8604
8605  035140  004537  007676            JSR     R5,TXCHAR       ;LOAD 377, TX 000
```

```
8606  035144  000377                  377
8607  035146  000010                  8.
8608  035150  103003                  BCC     .+8.            ;BR IF NO ERROR
8609  035152                          ERROR                   ;REPORT STACKED ERROR
8610  035152  104460                                                          TRAP    C$ERROR
8611  035154                          ESCAPE  TST             ;SKIP TO END OF TEST
8612  035154  104410                                                          TRAP    C$ESCAPE
8613  035156  000324                                                          .WORD   L10100-.
8614
8615  035160  004537  007676          JSR     R5,TXCHAR       ;LOAD 125, TX 377
8616  035164  000125                  125
8617  035166  000010                  8.
8618  035170  103003                  BCC     .+8.            ;BR IF NO ERROR
8619  035172                          ERROR                   ;REPORT STACKED ERROR
8620  035172  104460                                                          TRAP    C$ERROR
8621  035174                          ESCAPE  TST             ;SKIP TO END OF TEST
8622  035174  104410                                                          TRAP    C$ESCAPE
8623  035176  000304                                                          .WORD   L10100-.
8624
8625  035200  004537  007676          JSR     R5,TXCHAR       ;TX 125, LOAD SYNCH
8626  035204  000226                  SYNCH
8627  035206  000010                  8.
8628  035210  103003                  BCC     .+8.            ;BR IF NO ERROR
8629  035212                          ERROR                   ;REPORT STACKED ERROR
8630  035212  104460                                                          TRAP    C$ERROR
8631  035214                          ESCAPE  TST             ;SKIP TO END OF TEST
8632  035214  104410                                                          TRAP    C$ESCAPE
8633  035216  000264                                                          .WORD   L10100-.
8634
8635  035220  004537  006176          JSR     R5,CKRDA        ;CHECK RECEIVE DATA AVAILABLE
8636  035224  000000                  0                       ;   (NO DATA EXPECTED)
8637  035226  103003                  BCC     .+8.            ;BR IF NO ERROR
8638  035230                          ERROR                   ;REPORT STACKED ERROR
8639  035230  104460                                                          TRAP    C$ERROR
8640  035232                          ESCAPE  TST             ;SKIP TO END OF TEST
8641  035232  104410                                                          TRAP    C$ESCAPE
8642  035234  000246                                                          .WORD   L10100-.
8643                          ;-------------------------------------------------------------
8644                          ; THE RECEIVER SHOULD SYNCHRONIZE ON THE NEXT TWO SYNC CHARACTERS
8645                          ; AND THEN READ THE REMAINING ONES.
8646                          ;-------------------------------------------------------------
8647  035236  004537  007676          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
8648  035242  000226                  SYNCH
8649  035244  000010                  8.                      ;
8650  035246  103003                  BCC     .+8.            ;BR IF NO ERROR
8651  035250                          ERROR                   ;REPORT STACKED ERROR
8652  035250  104460                                                          TRAP    C$ERROR
8653  035252                          ESCAPE  TST             ;SKIP TO END OF TEST
8654  035252  104410                                                          TRAP    C$ESCAPE
8655  035254  000226                                                          .WORD   L10100-.
8656
8657  035256  004537  007676          JSR     R5,TXCHAR       ;LOAD 3ND SYNCH, TX 2ND SYNCH
8658  035262  000226                  SYNCH
8659  035264  000010                  8.                      ;
8660  035266  103003                  BCC     .+8.            ;BR IF NO ERROR
8661  035270                          ERROR                   ;REPORT STACKED ERROR
```

```
8662  035270  104460                                                                  TRAP    C$ERROR
8663  035272
8664  035272  104410          ESCAPE  TST             ;SKIP TO END OF TEST            TRAP    C$ESCAPE
8665  035274  000206                                                                  .WORD   L10100-.
8666
8667  035276  004537  007676  JSR     R5,TXCHAR       ;LOAD 4TH SYNCH, TX 3RD SYNCH
8668  035302  000226          SYNCH
8669  035304  000010          8.                      ;
8670  035306  103003          BCC     .+8.            ;BR IF NO ERROR
8671  035310                  ERROR                   ;REPORT STACKED ERROR
8672  035310  104460                                                                  TRAP    C$ERROR
8673  035312
8674  035312  104410          ESCAPE  TST             ;SKIP TO END OF TEST            TRAP    C$ESCAPE
8675  035314  000166                                                                  .WORD   L10100-.
8676
8677  035316  004537  007676  JSR     R5,TXCHAR       ;LOAD 5TH SYNCH
8678  035322  000226          SYNCH
8679  035324  000000          0                       ;
8680  035326  103003          BCC     .+8.            ;BR IF NO ERROR
8681  035330                  ERROR                   ;REPORT STACKED ERROR
8682  035330  104460                                                                  TRAP    C$ERROR
8683  035332
8684  035332  104410          ESCAPE  TST             ;SKIP TO END OF TEST            TRAP    C$ESCAPE
8685  035334  000146                                                                  .WORD   L10100-.
8686
8687  035336  004537  011364  JSR     R5,RCV1ST       ;CLOCK AND RECEIVE 3RD SYNCH
8688  035342  000000          0
8689  035344  103003          BCC     .+8.            ;BR IF NO ERROR
8690  035346                  ERROR                   ;REPORT STACKED ERROR
8691  035346  104460                                                                  TRAP    C$ERROR
8692  035350
8693  035350  104410          ESCAPE  TST             ;SKIP TO END OF TEST            TRAP    C$ESCAPE
8694  035352  000130                                                                  .WORD   L10100-.
8695
8696  035354  004537  010110  JSR     R5,RXCHAR       ;RECEIVE/CHECK SYNCH CHARACTER
8697  035360  000226          SYNCH                   ;
8698  035362  000000          0
8699  035364  000010          8.
8700  035366  103003          BCC     .+8.            ;BR IF NO ERROR
8701  035370                  ERROR                   ;REPORT STACKED ERROR
8702  035370  104460                                                                  TRAP    C$ERROR
8703  035372
8704  035372  104410          ESCAPE  TST             ;SKIP TO END OF TEST            TRAP    C$ESCAPE
8705  035374  000106                                                                  .WORD   L10100-.
8706                  ;-------------------------------------------------------------------------------
8707  035376  012702  035504  MOV     #TXTBL1,R2      ;SET UP TABLE POINTER
8708  035402  112237  035440  5$:     MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
8709  035406  116237  000001  035420  MOVB    1(R2),10$       ;SETUP TRANSMIT CHARACTER
8710
8711  035414  004537  007676  JSR     R5,TXCHAR       ;LOAD A CHARACTER
8712  035420  000000          10$:    000             ;** HOLE FOR NEXT TX CHARACTER
8713  035422  000000                  0
8714  035424  103003          BCC     .+8.            ;BR IF NO ERROR
8715  035426                  ERROR                   ;REPORT STACKED ERROR
8716  035426  104460                                                                  TRAP    C$ERROR
8717  035430          ESCAPE  TST             ;SKIP TO END OF TEST
```

```
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10100-.

        JSR     R5,RXCHAR       ;CLOCK/RECEIVE/CHECK PREVIOUS CHARACTER
20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
        0
        8.
        BCC     .+8.            ;BR IF NO ERROR
        ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10100-.

        CMP     #TXEND1,R2
        BNE     5$
;-------------------------------------------------------------------------
        JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
        8.
        BCC     .+8.            ;BR IF NO ERROR
        ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
        ESCAPE  TST             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10100-.

ENDTST

                                                                L10100:
                                                                        TRAP    C$ETST

;-------------------------------
TXTBL1: .BYTE   226             ;SYNCH
        .BYTE   226             ;SYNCH
        .BYTE   000
        .BYTE   377
        .BYTE   125
        .BYTE   226             ;SYNCH
        .BYTE   226             ;SYNCH
        .BYTE   252
        .BYTE   101
        .BYTE   202
        .BYTE   226             ;SYNCH
        .BYTE   226             ;SYNCH
        .BYTE   000
TXEND1: .BYTE   000
        .EVEN
;-------------------------------
```

```
                                .SBTTL  TEST 30 -- BCP RX STRIP-SYNC TEST

                        ;*************************************************************************
                        ;*
                        ;*      TEST 30 -- BCP RX STRIP-SYNC TEST
                        ;*
                        ;*  THE  USYRT  IS INITIALIZED WITH THE STRIP-SYNC CONTROL BIT ASSERTED.
                        ;*  THE  FOLLOWING  MESSAGE  IS THEN INITIATED WITHOUT ASSERTING RXE AND
                        ;*  ONCE THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
                        ;*
                        ;*    SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
                        ;*    SYNC SYNC DATA DATA DATA SYNC SYNC
                        ;*
                        ;*  THE  RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
                        ;*  THE  NEXT  TWO  SYNC  CHARACTERS FOR SYNCRONIZATION, IGNORE THE NEXT
                        ;*  THREE  SYNC CHARACTERS, AND PASS THE REST OF THE MESSAGE (4 SYNC AND
                        ;*  6 DATA CHARACTERS) THROUGH RXDB REGISTER.
                        ;*
                        ;-------------------------------------------------------------------------
                        ;
                        ;       BGNTST
035522                                                                  T30::
035522  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

035526  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
035532  063626                  DDCMP!STRIPS!NOCHK!SYNCH        ;SET DDCMP,NO CHECK,SYNCH=226
035534  040000                  NORXEN                  ;USE 8 BIT CHARS; LEAVE RXEN=0
035536  103003                  BCC     .+8.            ;BR IF NO ERROR
035540                          ERROR                   ;REPORT STACKED ERROR
035540  104460                                                          TRAP    C$ERROR
035542                          ESCAPE  TST             ;SKIP TO END OF TEST
035542  104410                                                          TRAP    C$ESCAPE
035544  000474                                                          .WORD   L10101-.

035546  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
035552  000001                  TSOM                    ; (IGNORED BY RECEIVER)
035554  000007                  7.
035556  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
035562  000000                  000
035564  000000                  0
035566  004537  003734          JSR     R5,WRITEI       ;ENABLE RECEIVER (RXEN => 1)
035572  120000                  VIAORB
035574  000142                  RXEN!TXEN!TTLOOP
                        ;------------------------------------------------------------
                        ; THE RECEIVER SHOULD IGNORE THE NEXT STRING OF CHARACTERS
                        ;------------------------------------------------------------
035576  004537  007676          JSR     R5,TXCHAR       ;LOAD 000, TX 2ND SYNCH
035602  000000                  000
035604  000010                  8.
035606  103003                  BCC     .+8.            ;BR IF NO ERROR
035610                          ERROR                   ;REPORT STACKED ERROR
035610  104460                                                          TRAP    C$ERROR
035612                          ESCAPE  TST             ;SKIP TO END OF TEST
035612  104410                                                          TRAP    C$ESCAPE
035614  000424                                                          .WORD   L10101-.
```

```
8820  035616  004537  007676          JSR     R5,TXCHAR       ;LOAD 377, TX 000
8821  035622  000377                  377
8822  035624  000010                  8.
8823  035626  103003                  BCC     .+8.            ;BR IF NO ERROR
8824  035630                          ERROR                   ;REPORT STACKED ERROR
8825  035630  104460                                                  TRAP    C$ERROR
8826  035632                          ESCAPE  TST             ;SKIP TO END OF TEST
8827  035632  104410                                                  TRAP    C$ESCAPE
8828  035634  000404                                                  .WORD   L10101-.
8829
8830  035636  004537  007676          JSR     R5,TXCHAR       ;LOAD 125, TX 377
8831  035642  000125                  125
8832  035644  000010                  8.
8833  035646  103003                  BCC     .+8.            ;BR IF NO ERROR
8834  035650                          ERROR                   ;REPORT STACKED ERROR
8835  035650  104460                                                  TRAP    C$ERROR
8836  035652                          ESCAPE  TST             ;SKIP TO END OF TEST
8837  035652  104410                                                  TRAP    C$ESCAPE
8838  035654  000364                                                  .WORD   L10101-.
8839
8840  035656  004537  007676          JSR     R5,TXCHAR       ;LOAD SYNCH, TX 125
8841  035662  000226                  SYNCH
8842  035664  000010                  8.
8843  035666  103003                  BCC     .+8.            ;BR IF NO ERROR
8844  035670                          ERROR                   ;REPORT STACKED ERROR
8845  035670  104460                                                  TRAP    C$ERROR
8846  035672                          ESCAPE  TST             ;SKIP TO END OF TEST
8847  035672  104410                                                  TRAP    C$ESCAPE
8848  035674  000344                                                  .WORD   L10101-.
8849
8850  035676  004537  006176          JSR     R5,CKRDA        ;CHECK RECEIVE DATA AVAILABLE
8851  035702  000000                  0                       ;   (NO DATA EXPECTED)
8852  035704  103003                  BCC     .+8.            ;BR IF NO ERROR
8853  035706                          ERROR                   ;REPORT STACKED ERROR
8854  035706  104460                                                  TRAP    C$ERROR
8855  035710                          ESCAPE  TST             ;SKIP TO END OF TEST
8856  035710  104410                                                  TRAP    C$ESCAPE
8857  035712  000326                                                  .WORD   L10101-.
8858                          ;-----------------------------------------------------------
8859                          ; THE RECEIVER SHOULD SYNCHRONIZE ON THE NEXT TWO SYNC CHARACTERS,
8860                          ; STRIP THE NEXT THREE, AND THEN READ THE REMAINING ONES.
8861                          ;-----------------------------------------------------------
8862  035714  004537  007676          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
8863  035720  000226                  SYNCH
8864  035722  000010                  8.                      ;
8865  035724  103003                  BCC     .+8.            ;BR IF NO ERROR
8866  035726                          ERROR                   ;REPORT STACKED ERROR
8867  035726  104460                                                  TRAP    C$ERROR
8868  035730                          ESCAPE  TST             ;SKIP TO END OF TEST
8869  035730  104410                                                  TRAP    C$ESCAPE
8870  035732  000306                                                  .WORD   L10101-.
8871
8872  035734  004537  007676          JSR     R5,TXCHAR       ;LOAD 3ND SYNCH, TX 2ND SYNCH
8873  035740  000226                  SYNCH
8874  035742  000010                  8.                      ;
8875  035744  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
8876  035746                        ERROR                   ;REPORT STACKED ERROR
8877  035746  104460                                                        TRAP    C$ERROR
8878  035750                        ESCAPE  TST             ;SKIP TO END OF TEST
8879  035750  104410                                                        TRAP    C$ESCAPE
8880  035752  000266                                                        .WORD   L10101-.
8881
8882  035754  004567  007676        JSR     R5,TXCHAR       ;LOAD 4TH SYNCH, TX 3RD SYNCH
8883  035760  000226                SYNCH
8884  035762  000010                8.                      ;
8885  035764  103003                BCC     .+8.            ;BR IF NO ERROR
8886  035766                        ERROR                   ;REPORT STACKED ERROR
8887  035766  104460                                                        TRAP    C$ERROR
8888  035770                        ESCAPE  TST             ;SKIP TO END OF TEST
8889  035770  104410                                                        TRAP    C$ESCAPE
8890  035772  000246                                                        .WORD   L10101-.
8891
8892  035774  004537  007676        JSR     R5,TXCHAR       ;LOAD 5TH SYNCH, TX 4TH SYNCH
8893  036000  000226                SYNCH
8894  036002  000010                8.                      ;
8895  036004  103003                BCC     .+8.            ;BR IF NO ERROR
8896  036006                        ERROR                   ;REPORT STACKED ERROR
8897  036006  104460                                                        TRAP    C$ERROR
8898  036010                        ESCAPE  TST             ;SKIP TO END OF TEST
8899  036010  104410                                                        TRAP    C$ESCAPE
8900  036012  000226                                                        .WORD   L10101-.
8901
8902  036014  004537  007676        JSR     R5,TXCHAR       ;LOAD DATA1 ,TX 5TH SYNCH
8903  036020  000252                252
8904  036022  000010                8.                      ;
8905  036024  103003                BCC     .+8.            ;BR IF NO ERROR
8906  036026                        ERROR                   ;REPORT STACKED ERROR
8907  036026  104460                                                        TRAP    C$ERROR
8908  036030                        ESCAPE  TST             ;SKIP TO END OF TEST
8909  036030  104410                                                        TRAP    C$ESCAPE
8910  036032  000206                                                        .WORD   L10101-.
8911
8912  036034  004537  007676        JSR     R5,TXCHAR       ;LOAD DATA2 ,TX DATA1
8913  036040  000347                347
8914  036042  000010                8.                      ;
8915  036044  103003                BCC     .+8.            ;BR IF NO ERROR
8916  036046                        ERROR                   ;REPORT STACKED ERROR
8917  036046  104460                                                        TRAP    C$ERROR
8918  036050                        ESCAPE  TST             ;SKIP TO END OF TEST
8919  036050  104410                                                        TRAP    C$ESCAPE
8920  036052  000166                                                        .WORD   L10101-.
8921
8922  036054  004537  007676        JSR     R5,TXCHAR       ;LOAD DATA3
8923  036060  000030                030
8924  036062  000000                0                       ;
8925  036064  103003                BCC     .+8.            ;BR IF NO ERROR
8926  036066                        ERROR                   ;REPORT STACKED ERROR
8927  036066  104460                                                        TRAP    C$ERROR
8928  036070                        ESCAPE  TST             ;SKIP TO END OF TEST
8929  036070  104410                                                        TRAP    C$ESCAPE
8930  036072  000146                                                        .WORD   L10101-.
8931
```

```
CVDMCA.P11    10-DEC-80 09:14          TEST 30 -- BCP RX STRIP-SYNC TEST

 8932  036074  004537  011364          JSR     R5,RCV1ST       ;CLOCK AND RECEIVE DATA1
 8933  036100  000000                  0
 8934  036102  103003                  BCC     .+8.            ;BR IF NO ERROR
 8935  036104                          ERROR                   ;REPORT STACKED ERROR
 8936  036104  104460                                                          TRAP    C$ERROR
 8937  036106                          ESCAPE  TST             ;SKIP TO END OF TEST
 8938  036106  104410                                                          TRAP    C$ESCAPE
 8939  036110  000130                                                          .WORD   L10101-.
 8940
 8941  036112  004537  010110          JSR     R5,RXCHAR       ;RECEIVE/CHECK DATA1 (252)
 8942  036116  000252                  252                     ;
 8943  036120  000000                  0
 8944  036122  000010                  8.
 8945  036124  103003                  BCC     .+8.            ;BR IF NO ERROR
 8946  036126                          ERROR                   ;REPORT STACKED ERROR
 8947  036126  104460                                                          TRAP    C$ERROR
 8948  036130                          ESCAPE  TST             ;SKIP TO END OF TEST
 8949  036130  104410                                                          TRAP    C$ESCAPE
 8950  036132  000106                                                          .WORD   L10101-.
 8951                                  ;-----------------------------------------------------
 8952  036134  012702  036242          MOV     #TXTBL2,R2      ;SET UP TABLE POINTER
 8953  036140  112237  036176  5$:     MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
 8954  036144  116237  000001  036156  MOVB    1(R2),10$       ;SETUP TRANSMIT CHARACTER
 8955
 8956  036152  004537  007676          JSR     R5,TXCHAR       ;LOAD A CHARACTER
 8957  036156  000000          10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
 8958  036160  000000                  0
 8959  036162  103003                  BCC     .+8.            ;BR IF NO ERROR
 8960  036164                          ERROR                   ;REPORT STACKED ERROR
 8961  036164  104460                                                          TRAP    C$ERROR
 8962  036166                          ESCAPE  TST             ;SKIP TO END OF TEST
 8963  036166  104410                                                          TRAP    C$ESCAPE
 8964  036170  000050                                                          .WORD   L10101-.
 8965
 8966  036172  004537  010110          JSR     R5,RXCHAR       ;CLOCK/RECEIVE/CHECK PREVIOUS CHARACTER
 8967  036176  000000          20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
 8968  036200  000000                  0
 8969  036202  000010                  8.
 8970  036204  103003                  BCC     .+8.            ;BR IF NO ERROR
 8971  036206                          ERROR                   ;REPORT STACKED ERROR
 8972  036206  104460                                                          TRAP    C$ERROR
 8973  036210                          ESCAPE  TST             ;SKIP TO END OF TEST
 8974  036210  104410                                                          TRAP    C$ESCAPE
 8975  036212  000026                                                          .WORD   L10101-.
 8976
 8977  036214  022702  036254          CMP     #TXEND2,R2
 8978  036220  001347                  BNE     5$
 8979                                  ;-----------------------------------------------------
 8980  036222  004537  011532          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
 8981  036226  000010                  8.
 8982  036230  103003                  BCC     .+8.            ;BR IF NO ERROR
 8983  036232                          ERROR                   ;REPORT STACKED ERROR
 8984  036232  104460                                                          TRAP    C$ERROR
 8985  036234                          ESCAPE  TST             ;SKIP TO END OF TEST
 8986  036234  104410                                                          TRAP    C$ESCAPE
 8987  036236  000002                                                          .WORD   L10101-.
```

```
 8988  036240
 8989  036240
 8990  036240  104401
 8991
 8992  036242     347
 8993  036243     030
 8994  036244     226
 8995  036245     226
 8996  036246     252
 8997  036247     101
 8998  036250     202
 8999  036251     226
 9000  036252     226
 9001  036253     000
 9002  036254     000
 9003          036256
 9004
```

```
ENDTST

;------------------------------
TXTBL2: .BYTE   347
        .BYTE   030
        .BYTE   226     ;SYNCH
        .BYTE   226     ;SYNCH
        .BYTE   252
        .BYTE   101
        .BYTE   202
        .BYTE   226     ;SYNCH
        .BYTE   226     ;SYNCH
        .BYTE   000
TXEND2: .BYTE   000
        .EVEN
;------------------------------
```

```
L10101:
        TRAP    C$ETST
```

```
                                .SBTTL  TEST 31 -- BCP RX LOST RXE TEST

                                ;++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
                                ;*
                                ;*      TEST 31 -- BCP RX LOST RXE TEST
                                ;*
                                ;*  THE USYRT IS INITIALIZED (CRC16,STRIPS,BCP MODE) AND A MESSAGE IS STARTED.
                                ;*  WHILE IN THE MIDDLE  OF  TEXT, RXE IS DROPPED AND THE REACTION OF THE
                                ;*  RECEIVER IS MONITORED.
                                ;*
                                ;*
                                ;--**********************************************************************************
                                ;
                                        BGNTST
                                                                                        T31::
036256                                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
036256  004737  005420
036262  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
036266  065626                          DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
036270  000000                          0                       ;USE 8 BIT CHARS
036272  103003                          BCC     .+8.            ;BR IF NO ERROR
036274                                  ERROR                   ;REPORT STACKED ERROR
036274  104460                                                                  TRAP    C$ ROR
036276                                  ESCAPE  TST             ;SKIP TO END OF TEST
036276  104410                                                                  TRAP    C$ESCAPE
036300  000334                                                                  .WORD   L10102-.

036302  004537  007676                  JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
036306  000226                          SYNCH
036310  000007                          7.
036312  103003                          BCC     .+8.            ;BR IF NO ERROR
036314                                  ERROR                   ;REPORT STACKED ERROR
036314  104460                                                                  TRAP    C$ERROR
036316                                  ESCAPE  TST             ;SKIP TO END OF TEST
036316  104410                                                                  TRAP    C$ESCAPE
036320  000314                                                                  .WORD   L10102-.

036322  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
036326  000000                          000
036330  000000                          0
036332  004537  007676                  JSR     R5,TXCHAR       ;LOAD 000, TX 2ND SYNCH
036336  000000                          000
036340  000010                          8.
036342  103003                          BCC     .+8.            ;BR IF NO ERROR
036344                                  ERROR                   ;REPORT STACKED ERROR
036344  104460                                                                  TRAP    C$ERROR
036346                                  ESCAPE  TST             ;SKIP TO END OF TEST
036346  104410                                                                  TRAP    C$ESCAPE
036350  000264                                                                  .WORD   L10102-.

036352  004537  007676                  JSR     R5,TXCHAR       ;LOAD 125, TX 000
036356  000125                          125
036360  000010                          8.
036362  103003                          BCC     .+8.            ;BR IF NO ERROR
036364                                  ERROR                   ;REPORT STACKED ERROR
036364  104460                                                                  TRAP    C$ERROR
```

```
9061  036366
9062  036366  104410                ESCAPE  TST             ;SKIP TO END OF TEST
9063  036370  000244                                                    TRAP    C$ESCAPE
9064                                                                      .WORD   L10102-.
9065  036372  004537  007676        JSR     R5,TXCHAR       ;LOAD 252, TX 125
9066  036376  000252                252
9067  036400  000010                8.
9068  036402  103003                BCC     .+8.            ;BR IF NO ERROR
9069  036404                        ERROR                   ;REPORT STACKED ERROR
9070  036404  104460                                                    TRAP    C$ERROR
9071  036406                        ESCAPE  TST             ;SKIP TO END OF TEST
9072  036406  104410                                                    TRAP    C$ESCAPE
9073  036410  000224                                                    .WORD   L10102-.
9074
9075  036412  004537  007676        JSR     R5,TXCHAR       ;LOAD 377, TX 252
9076  036416  000377                377
9077  036420  000010                8.
9078  036422  103003                BCC     .+8.            ;BR IF NO ERROR
9079  036424                        ERROR                   ;REPORT STACKED ERROR
9080  036424  104460                                                    TRAP    C$ERROR
9081  036426                        ESCAPE  TST             ;SKIP TO END OF TEST
9082  036426  104410                                                    TRAP    C$ESCAPE
9083  036430  000204                                                    .WORD   L10102-.
9084
9085  036432  004537  007676        JSR     R5,TXCHAR       ;LOAD 000
9086  036436  000000                000
9087  036440  000000                0
9088  036442  103003                BCC     .+8.            ;BR IF NO ERROR
9089  036444                        ERROR                   ;REPORT STACKED ERROR
9090  036444  104460                                                    TRAP    C$ERROR
9091  036446                        ESCAPE  TST             ;SKIP TO END OF TEST
9092  036446  104410                                                    TRAP    C$ESCAPE
9093  036450  000164                                                    .WORD   L10102-.
9094
9095  036452  004537  011364        JSR     R5,RCV1ST       ;CLOCK AND RCV 000
9096  036456  000000                0
9097  036460  103003                BCC     .+8.            ;BR IF NO ERROR
9098  036462                        ERROR                   ;REPORT STACKED ERROR
9099  036462  104460                                                    TRAP    C$ERROR
9100  036464                        ESCAPE  TST             ;SKIP TO END OF TEST
9101  036464  104410                                                    TRAP    C$ESCAPE
9102  036466  000146                                                    .WORD   L10102-.
9103
9104  036470  004537  010110        JSR     R5,RXCHAR       ;READ & CHK 000, RCV 125
9105  036474  000000                000
9106  036476  000000                0
9107  036500  000010                8.
9108  036502  103003                BCC     .+8.            ;BR IF NO ERROR
9109  036504                        ERROR                   ;REPORT STACKED ERROR
9110  036504  104460                                                    TRAP    C$ERROR
9111  036506                        ESCAPE  TST             ;SKIP TO END OF TEST
9112  036506  104410                                                    TRAP    C$ESCAPE
9113  036510  000124                                                    .WORD   L10102-.
9114
9115                        ;-----------------------------------------------------------
9116                        ; TX AND RX NOW SYNC'D. ONE CHARACTER HAS BEEN READ/VERIFIED....
                            ; NOW CLEAR RXEN (RECEIVER ENABLE).
```

```
9117
9118  036512  004537  003476          ;-----------------------------------------------------------------
9119  036516  120000                  JSR     R5,READ         ;READ VALUE OF VIAORB REGISTER
9120  036520  036546                  VIAORB                  ; AND PUT IT IN LOCATION PROVIDED
9121  036522  103003                  CGORB                   ;BELOW (CGORB).
9122  036524                          BCC     .+8.            ;BR IF NO ERROR
9123  036524  104460                  ERROR                   ;REPORT ERROR
9124  036526                                                                  TRAP    C$ERROR
9125  036526  104410                  ESCAPE  TST
9126  036530  000104                                                          TRAP    C$ESCAPE
9127                                                                          .WORD   L10102-.
9128  036532  042737  000100  036546  BIC     #RXEN,CGORB     ;CLEAR RXEN BIT OF ORB STATUS WORD
9129
9130  036540  004537  003734          JSR     R5,WRITEI       ;WRITE NEW STATUS WORD (W/RXEN=0)
9131  036544  120000                  VIAORB                  ;BACK INTO VIAORB
9132  036546  000000          CGORB:  .WORD   0               ;ACTUAL DATA WRITTEN TO VIAORB
9133  036550  103003                  BCC     .+8.            ;BR IF NO ERROR
9134  036552                          ERROR                   ;REPORT ERROR
9135  036552  104460                                                          TRAP    C$ERROR
9136  036554                          ESCAPE  TST
9137  036554  104410                                                          TRAP    C$ESCAPE
9138  036556  000056                                                          .WORD   L10102-.
9139                                  ;-----------------------------------------------------------------
9140                                  ; RXEN NOW CLEARED. CHECK USYRT STATUS, TX ANOTHER CHARACTER, AND
9141                                  ; CHECK USYRT STATUS AGAIN.
9142                                  ;-----------------------------------------------------------------
9143  036560  004537  005432          JSR     R5,CKUSTS       ;CHK USYRT STATUS FOR PROPER STATE
9144  036564  000104                  TXACT!TBMT              ;TXACT = 1, TBMT = 1
9145  036566  103003                  BCC     .+8.            ;BR IF NO ERROR
9146  036570                          ERROR
9147  036570  104460                                                          TRAP    C$ERROR
9148  036572                          ESCAPE  TST
9149  036572  104410                                                          TRAP    C$ESCAPE
9150  036574  000040                                                          .WORD   L10102-.
9151
9152  036576  004537  007676          JSR     R5,TXCHAR       ;LOAD/TRANSMIT 303
9153  036602  000303                  303
9154  036604  100010                  NCTBMT*256.!8.
9155  036606  103003                  BCC     .+8.            ;BR IF NO ERROR
9156  036610                          ERROR                   ;REPORT STACKED ERROR
9157  036610  104460                                                          TRAP    C$ERROR
9158  036612                          ESCAPE  TST             ;SKIP TO END OF TEST
9159  036612  104410                                                          TRAP    C$ESCAPE
9160  036614  000020                                                          .WORD   L10102-.
9161
9162  036616  004537  005432          JSR     R5,CKUSTS       ;CHK USYRT STATUS FOR PROPER STATE
9163  036622  000114                  TXACT!TSO!TBMT          ;TXACT = 1
9164  036624  103003                  BCC     .+8.            ;BR IF NO ERROR
9165  036626                          ERROR
9166  036626  104460                                                          TRAP    C$ERROR
9167  036630                          ESCAPE  TST
9168  036630  104410                                                          TRAP    C$ESCAPE
9169  036632  000002                                                          .WORD   L10102-.
9170  036634                  ENDTST
9171  036634                                                          L10102:
9172  036634  104401                                                          TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION


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

036636                                  BGNHRD
036636  000041                                                          .WORD L10103-L$HARD/2
036640                                                          L$HARD::

036640                                  GPRMA   ADDRES,0,0,160020,177776,YES
036640  000031                                                          .WORD   T$CODE
036642  036742                                                          .WORD   ADDRES
036644  160020                                                          .WORD   T$LOLIM
036646  177776                                                          .WORD   T$HILIM
036650                                  GPRMA   VECTOR,2,0,0,674,YES
036650  001031                                                          .WORD   T$CODE
036652  036770                                                          .WORD   VECTOR
036654  000000                                                          .WORD   T$LOLIM
036656  000674                                                          .WORD   T$HILIM
036660                                  GPRMD   PRIRTY,4,0,7000,4,7,YES
036660  002032                                                          .WORD   T$CODE
036662  037021                                                          .WORD   PRIRTY
036664  007000                                                          .WORD   7000
036666  000004                                                          .WORD   T$LOLIM
036670  000007                                                          .WORD   T$HILIM
036672                                  GPRMD   SW1.M,6,0,377,0,377,YES
036672  003032                                                          .WORD   T$CODE
036674  037052                                                          .WORD   SW1.M
036676  000377                                                          .WORD   377
036700  000000                                                          .WORD   T$LOLIM
036702  000377                                                          .WORD   T$HILIM
036704                                  GPRMD   SW2.M,10,0,377,0,377,YES
036704  004032                                                          .WORD   T$CODE
036706  037113                                                          .WORD   SW2.M
036710  000377                                                          .WORD   377
036712  000000                                                          .WORD   T$LOLIM
036714  000377                                                          .WORD   T$HILIM
036716                                  GPRMD   BDTY.M,12,0,7,0,2,YES
036716  005032                                                          .WORD   T$CODE
036720  037155                                                          .WORD   BDTY.M
036722  000007                                                          .WORD   7
036724  000000                                                          .WORD   T$LOLIM
036726  000002                                                          .WORD   T$HILIM
036730                                  GPRMD   BR.M,16,0,17,0,1,YES
036730  007032                                                          .WORD   T$CODE
036732  037240                                                          .WORD   BR.M
036734  000017                                                          .WORD   17
036736  000000                                                          .WORD   T$LOLIM
```

```
9229  036740  000001                                                                  .WORD   T$HILIM
9230
9231  036742                          ENDHRD
9232                                                                                  .EVEN
9233  036742                                                                L10103:
9234
9235                                  .NLIST  BEX
      036742  042504  044526  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
      036770  042504  044526  042503  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
      037021     104  053105  041511  PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
      037052  053523  052111  044103  SW1.M:  .ASCIZ  /SWITCH PACK # 1 (BOOT ADDRESS): /
      037113     123  044527  041524  SW2.M:  .ASCIZ  /SWITCH PACK # 2 (DDCMP ADDRESS): /
      037155     102  040517  042122  BDTY.M: .ASCIZ  /BOARD TYPE (0=M8064, 1=M8053-V.35, 2=M8053-EIA) : /
      037240  040502  042125  051040  BR.M:   .ASCIZ  /BAUD RATE (0=LOW (19.2K), 1=HIGH (56K)): /
                                      .LIST   BEX
9236                                  .EVEN
```

```
 9237                                   .SBTTL  SOFTWARE PARAMETER CODING SECTION
 9238
 9239
 9240                                   ;////////////////////////////////////////////////////////////////////
 9241                                   ;/ THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 9242                                   ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 9243                                   ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 9244                                   ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 9245                                   ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 9246                                   ;/ WITH THE OPERATOR.
 9247                                   ;////////////////////////////////////////////////////////////////////
 9248
 9249  037312                                   BGNSFT
 9250  037312  000000                                                          .WORD L10104-L$SOFT/2
 9251  037314                                                          L$SOFT::
 9252
 9253  037314                                   ENDSFT
 9254                                                                    .EVEN
 9255  037314                                                          L10104:
```

```
 9256                                   ****** PATCH AREA FOR DEBUG ******
 9257                                   .SBTTL ****** PATCH AREA FOR DEBUG ******
 9258  037314                           PATCH:
 9259            037414                         .=.+100
 9260  037414    000240                         NOP
 9261  037416    000240                         NOP
 9262  037420    000240                         NOP
 9263                                   ;*************************************
 9264
 9265                                   .SBTTL  'ENDMOD'' STATEMENT
 9266
 9267  037422                                   ENDMOD
 9268
 9269                                   .SBTTL  'LASTAD'' STATEMENT & END OF PROGRAM
 9270  037422                                   LASTAD
 9271                                                                        .EVEN
 9272  037422    000000                                                    .WORD   0
 9273  037424    000000                                                    .WORD   0
 9274  037426                           L$LAST::
 9275
 9276            000001                 .END
```

```
ABC    = 000160          1629#
ADDRES  036742           9192    9235#
ADR    = 000020 G        1561#
AD.HIT  021532           4674    4709#
AD.OK   021526           4698    4703#
APA    = 000200          1685#
APAD   = 100000          1694#
ASSEMB= 000010           1297
BADDAT  002442           2079#
BDATA   002402           2062#   2686*   2689    2692    2967*   3070*   3071*   3502*   3761*   3762*   3779*   3780*   4149
                         4178    4227    4272    4336    4361    4818*   5033*   5148*   5377    5379*   5380    5586*   5587*
                         5612*   5613*   5638*   5639*
BDRATE  002550           2126#   4645*   4903
BDTY.M  037155           9220    9235#
BIT0   = 000001 G        1534#   1579    1591    1633    1642    1661    1690    1715    1731    1749    1765    1907    1921
                         1943    1996    3101    3146    3191    3236    3281    3324    3371    3438    4611    6212    6514
                         7337    7373    7619    7655
BIT00  = 000001 G        1523#   1534
BIT01  = 000002 G        1522#   1533
BIT02  = 000004 G        1521#   1532
BIT03  = 000010 G        1520#   1531
BIT04  = 000020 G        1519#   1530
BIT05  = 000040 G        1518#   1529
BIT06  = 000100 G        1517#   1528
BIT07  = 000200 G        1516#   1527
BIT08  = 000400 G        1515#   1526
BIT09  = 001000 G        1514#   1525
BIT1   = 000002 G        1533#   1632    1660    1690    1715    1730    1748    1764    1906    1920    1942    1995    3395
                         6212    6574    7196    7232    7337    7373    7760    7796
BIT10  = 002000 G        1513#   1638    1666    1700    1701    1702
BIT11  = 004000 G        1512#   1637    1665    1698
BIT12  = 010000 G        1511#   1697
BIT13  = 020000 G        1510#   1696    2005
BIT14  = 040000 G        1509#   1695    1998    2004
BIT15  = 100000 G        1508#   1636    1664    1694    1992    2003
BIT2   = 000004 G        1532#   1631    1659    1690    1715    1729    1747    1763    1895    1920    1941    6212    6514
                         6574    7478    7514    7619    7655    7760    7796
BIT3   = 000010 G        1531#   1630    1658    1689    1714    1728    1746    1762    1895    1920    1940
BIT4   = 000020 G        1530#   1578    1629    1688    1713    1727    1745    1761    1895    1919    1939
BIT5   = 000040 G        1529#   1629    1687    1712    1726    1744    1760    1890    1918    1938    3903    6514    7373
                         7655
BIT6   = 000100 G        1528#   1590    1629    1686    1712    1725    1743    1759    1880    1918    1937    3923    6212
                         6574    7232    7373    7796
BIT7   = 000200 G        1527#   1589    1597    1628    1657    1685    1712    1724    1742    1758    1880    1918    1931
                         1953    2001    3740    3741    3912    5929    5968    6045    6157    6196    6514    6574    7514
                         7655    7796
BIT8   = 000400 G        1526#   1640    1668    1699    1700    1701
BIT9   = 001000 G        1525#   1639    1667    1699    1700    1999
BOE    = 000400 G        1565#
BRDTYP  002544           2124#   4643*   4887
BR.M    037240           9226    9235#
BSEL0   002472           2093#   2628    4681    4687
BSEL1   002474           2096#   2424*   2629
BSEL10  002512           2106#   2636
BSEL11  002514           2108#   2637
BSEL12  002516           2109#   2638
```

```
BSEL13  002520          2111#   2639
BSEL14  002522          2112#   2640
BSEL15  002524          2114#   2641
BSEL16  002526          2115#   2642
BSEL17  002530          2117#   2643    4633
BSEL2   002476          2097#   2431    2467*   2474    2514*   2521    2586*   2593    2630    4796
BSEL3   002500          2099#   2631
BSEL4   002502          2100#   2632
BSEL5   002504          2102#   2633
BSEL6   002506          2103#   2489    2634
BSEL7   002510          2105#   2635
BSR0    002256          2029#   2628*
BSR1    002260          2031#   2629*
BSR10   002276          2044#   2636*
BSR11   002300          2045#   2637*
BSR12   002302          2046#   2638*
BSR13   002304          2047#   2639*
BSR14   002306          2048#   2640*
BSR15   002310          2049#   2641*
BSR16   002312          2050#   2642*
BSR17   002314          2051#   2643*
BSR2    002262          20  3#  2630*
  R3    002264          203 #   2631*
  R4    002266          2037#   2632*
BSR5    002270          2039#   2633*
BSR6    002272          2041#   2634*
BSR7    002274          2043#   2635*
CARIER= 000100          1759#
CA1CTL= 000001          1921#
CA2CTL= 000016          1920#
CB1CTL= 000020          1919#
CB2CTL= 000340          19  #
CGORB   036546          91  0   9128*   9132#
CHKTSO  007116          3433#   3491    5878    5982    6106    6214    673o    6833    6930
CHPTYP  002454          2084#   4560*
CKRACT  005676          3141#   3905    3958    3985    4012    4023
CKRDA   006176          3231#   3914    3925    3961    3977    3982    8635    8850
CKROR   006475          3319#   8258    8426    8438    8447    8536
CKRSA   006336          3276#
CKSEQM  006636          3366#   3964
CKTACT  005536          3096#   3602    3638    3690    4009    4020
CKTBMT  006036          3186#   3585    3599    3647    3654    3695    3702    8094    8122    8145
CKUSTS  005432          3061#   3568    4780    5917    5956    6033    6145    6184    9143    9162
CRC16 = 001400          1699#   9023
CTS   = 000010          1762#
C$AU  = 000052          1297#   4753
C$AUTO= 000061          1297#   4707
C$BRK = 0   22          1297#
C$BSEG= 000004          1297#
C$BSUB= 000002          1297#   4856    4886    5200    5252    5306    5366    54  '   6511    6571    6631    7193    7334
                        7475    7616    7757    7898    8403    8461    8499
C$CEFG= 000045          1297#
C$CLCK= 000062          1297#
C$CLEA= 000012          1297#   4725
C$CLOS= 000035          1297#
C$  *   000006          1297#   4877
```

```
C$CVEC= 000036   1297#  4696
C$DCLN= 000044   1297#
C$DODU= 000051   1297#  4701
C$DRPT= 000024   1297#
C$DU  = 000053   1297#  4740
C$EDIT= 000003   1297#  1369
C$ERDI= 000055   1297#  4803  4822  4872  4908  5037  5152  5386  5591  5617  5643  5933  5972
                 6049  6161  6200
C$ERHR= 000056   1297#
C$ERRO= 000060   1297#  4775  4784  4894  5089  5189  5208  5241  5260  5295  5314  5355  5425
                 5440  5473  5515  5688  5709  5720  5728  5775  5785  5795  5805  5814  5825
                 5833  5873  5882  5892  5910  5921  5948  5960  5986  5996  6012  6022  6037
                 6064  6101  6110  6120  6138  6149  6176  6188  6218  6228  6241  6249  6283
                 6301  6311  6327  6338  6349  6385  6407  6430  6447  6458  6467  6517  6537
                 6555  6577  6597  6615  6637  6657  6675  6731  6740  6750  6768  6782  6792
                 6828  6837  6847  6865  6879  6889  6925  6934  6944  6962  6976  6986  7019
                 7037  7047  7057  7067  7076  7087  7097  7108  7118  7129  7140  7151  7199
                 7215  7225  7238  7251  7271  7282  7297  7311  7322  7340  7356  7366  7379
                 7392  7412  7423  7438  7452  7463  7481  7497  7507  7520  7533  7553  7564
                 7579  7593  7604  7622  7638  7648  7661  7674  7694  7705  7720  7734  7745
                 7763  7779  7789  7802  7815  7835  7846  7861  7875  7886  7904  7920  7930
                 7943  7956  7976  7987  8002  8016  8027  8098  8117  8126  8140  8149  8168
                 8182  8192  8228  8250  8262  8276  8287  8301  8320  8331  8342  8351  8411
                 8430  8442  8451  8469  8489  8507  8540  8578  8600  8610  8620  8630  8639
                 8652  8662  8672  8682  8691  8702  8716  8727  8739  8793  8815  8825  8835
                 8845  8854  8867  8877  8887  8897  8907  8917  8927  8936  8947  8961  8972
                 8984  9027  9037  9050  9060  9070  9080  9090  9099  9110  9123  9135  9147
                 9157  9166
C$ERSF= 000054   1297#
C$ERSO= 000057   1297#
C$ESCA= 000010   1297#  4777  4786  4808  4896  5042  5091  5157  5191  5210  5243  5262  5297
                 5316  5357  5391  5423  5442  5475  5517  5596  5622  5690  5711  5722  5730
                 5777  5787  5797  5807  5816  5827  5835  5875  5884  5894  5912  5923  5938
                 5950  5962  5977  5988  5998  6014  6024  6039  6054  6066  6103  6112  6122
                 6140  6151  6166  6178  6190  6205  6220  6230  6243  6251  6285  6303  6313
                 6329  6340  6351  6387  6409  6432  6449  6460  6469  6519  6539  6557  6579
                 6599  6617  6639  6659  6677  6733  6742  6752  6770  6784  6794  6830  6839
                 6849  6867  6881  6891  6927  6936  6946  6964  6978  6988  7021  7039  7049
                 7059  7069  7078  7089  7099  7110  7120  7131  7142  7153  7201  7217  7227
                 7240  7253  7273  7284  7299  7313  7324  7342  7358  7368  7381  7394  7414
                 7425  7440  7454  7465  7483  7499  7509  7522  7535  7555  7566  7581  7595
                 7606  7624  7640  7650  7663  7676  7696  7707  7722  7736  7747  7765  7781
                 7791  7804  7817  7837  7848  7863  7877  7888  7906  7922  7932  7945  7958
                 7978  7989  8004  8018  8029  8100  8119  8128  8142  8151  8170  8184  8194
                 8230  8252  8264  8278  8289  8303  8322  8333  8344  8353  8413  8432  8444
                 8453  8471  8491  8509  8542  8580  8602  8612  8622  8632  8641  8654  8664
                 8674  8684  8693  8704  8718  8729  8741  8795  8817  8827  8837  8847  8856
                 8869  8879  8889  8899  8909  8919  8929  8938  8949  8963  8974  8986  9029
                 9039  9052  9062  9072  9082  9092  9101  9112  9125  9137  9149  9159  9168
C$ESEG= 000005   1297#
C$ESUB= 000003   1297#  4882  4914  5215  5267  5321  5396  5447  6565  6625  6685  7328  7469
                 7610  7751  7892  8033  8457  8495  8546
C$ETST= 000001   1297#  4829  4918  5049  5097  5169  5221  5273  5327  5402  5453  5495  5537
                 5650  5735  5839  6070  6255  6355  6473  6688  6798  6895  6992  7157  8036
                 8198  8357  8549  8746  8990  9172
C$EXIT= 000032   1297#
```

```
C$GETB= 000026          1297#
C$GETW= 000027          1297#
C$GMAN= 000043          1297#
C$GPHR= 000042          1297#   4619
C$GPLO= 000030          1297#
C$GPRI= 000040          1297#
C$INIT= 000011          1297#   4652
C$INLP= 000020          1297#
C$MANI= 000050          1297#
C$MEM = 000031          1297#
C$MSG = 000023          1297#   4132    4159    4193    4237    4282    4317    4346    4966    4981
C$OPEN= 000034          1297#
C$PNTB= 000014          1297#   4144    4154    4173    4183    4189    4206    4212    4222    4232    4251    4257    4267
                        4277    4296    4302    4312    4331    4341    4410    4926    4962    4976
C$PNTF= 000017          1297#
C$PNTS= 000016          1297#
C$PNTX= 000015          1297#   4109    4125    4377    4387    4394    4404    4426    4436    4443    4453    4460    4470
                        4477    4487    4503    4513    4520    4530    4932    4938    4946    4956
C$QIO = 000377          1297#
C$RDBU= 000007          1297#
C$REFG= 000047          1297#   4577    4584    4591    4598
C$RESE= 000033          1297#   4648    4737
C$REVI= 000003          1297#   1368
C$RFLA= 000021          1297#
C$RPT = 000025          1297#
C$SEFG= 000046          1297#
C$SPRI= 000041          1297#
C$SVEC= 000037          1297#   4677
C$TPRI= 000013          1297#
DDCMP = 040000          1695#   5684    5771    5869    6097    6279    6381    6513    6573    6633    8224    8407    8465
                        8503    8574    8789    9023
DEVMAP  002460          2086#   4607*   4625*
DEVPTR  002462          2087#   4611*   4623*   4625    4626*
DFPTBL  002224  G       1467#
DIAGMC= 000000          1297
DOTBMT= 000007          1607#   4789    4800
DTR   = 000020          1745#   2949    2952    3025    3541    3544    8078    8081    8092
DTRL  = 000000          1750#
D.BUG = 000000          2007#
EF.CON= 000036  G       1541#   4597
EF.NEW= 000035  G       1542#   4590
EF.PWR= 000034  G       1543#
EF.RES= 000037  G       1540#   4583
EF.STA= 000040  G       1539#   4576
EIAV35= 000002          1995#
EM1     013103          4064#   5039
EM100   014425          3447    4064#
EM101   014445          3459    4064#
EM102   014471          3508    4064#
EM14    013332          3827    4064#
EM16    013346          3817    4064#
EM2     013142          2973    4064#
EM25    013366          2700    4064#   5388
EM26    013414          4064#
EM27    013446          4064#
EM28    013477          3391    3886    4064#
```

```
EM29    013520          3380    3896    4064#
EM3     013211          2439    4064#
EM30    013535          3415    3863    4064#
EM31    013556          3404    3873    4064#
EM34    013573          3768    4064#
EM35    013621          3794    4064#
EM36    013642          3804    4064#
EM39    013657          3840    4064#
EM4     013235          2483    2530    2601    4064#   4805
EM40    013701          3850    4064#
EM5     013252          4064#   5593    5645
EM54    013717          3755    4064#
EM6     013302          4064#   5619
EM66    013741          4064#   5154
EM68    013760          3077    4064#
EM69    014007          3110    4064#
EM70    014025          3121    4064#
EM71    014047          3155    4064#
EM72    014065          3166    4064#
EM73    014107          3200    4064#
EM74    014124          3211    4064#
EM75    014145          3245    4064#
EM76    014161          3256    4064#
EM77    014201          3290    4064#
EM78    014215          3301    4064#
EM90    014235          3333    4064#
EM91    014266          3344    4064#
EM92    014323          4064#   4824
EM98    014357          4064#   5935    6051    6163
EM99    014404          4064#   5974    6202
ENDEMB  011640          4064#   4186    4407
ENDIT   021412          4600    4649#
ENDPAT  003105          2356#
ENDTRN  011532          4007#   6463    8347    8735    8980
ERRBLK  002254  G       2023#   2440*   2484*   2531*   2602*   2701*   2974*   3078*   3111*   3122*   3156*   3167*   3201*
                        3212*   3246*   3257*   3291*   3302*   3334*   3345*   3381*   3392*   3405*   3416*   3448*   3460*
                        3509*   3756*   3769*   3795*   3805*   3818*   3828*   3841*   3851*   3864*   3874*   3887*   3897*
ERRFLG  002424          2072#
ERRMSG  002252  G       2022#   2439*   2483*   2530*   2601*   2700*   2973*   3077*   3110*   3121*   3155*   3166*   3200*
                        3211*   3245*   3256*   3290*   3301*   3333*   3344*   3380*   3391*   3404*   3415*   3447*   3459*
                        3508*   3755*   3768*   3794*   3804*   3817*   3827*   3840*   3850*   3863*   3873*   3886*   3896*
ERRNBR  002250  G       2021#   2438*   2482*   2529*   2600*   2699*   2972*   3076*   3109*   3120*   3154*   3165*   3199*
                        3210*   3244*   3255*   3289*   3300*   3332*   3343*   3379*   3390*   3403*   3414*   3446*   3458*
                        3507*   3754*   3767*   3793*   3803*   3816*   3826*   3839*   3849*   3862*   3872*   3885*   3895*
ERROR1  002452          2083#   4561*
ERRTYP  002246  G       2020#   2437*   2481*   2528*   2599*   2698*   2971*   3075*   3108*   3119*   3153*   3164*   3198*
                        3209*   3243*   3254*   3288*   3299*   3331*   3342*   3378*   3389*   3402*   3413*   3445*   3457*
                        3506*   3753*   3766*   3792*   3802*   3815*   3825*   3838*   3848*   3861*   3871*   3884*   3894*
ERR10   017372  G       2974    3078    3769    4199#   5040    5155
ERR11   017546  G       4244#   5594    5620    5646
ERR11$  020356          4279    4419#   4978
ERR12   017722  G       3111    3122    3156    3167    3201    3212    3246    3257    3291    3302    3334    3345    3381
                        3392    3405    3416    3448    3460    3756    3795    3805    3818    3828    3841    3851    3864
                        3874    3887    3897    4289#   5936    5975    6052    6164    6203
ERR12$  020710          4234    4314    4343    4496#
ERR13   020036  G       3509    4324#
```

CVDMCA.P11 10-DEC-80 09:14 CROSS REFERENCE TABLE -- USER SYMBOLS

```
ERR4     017030 G        2440    2484    2531    2602    4098#   4806
ERR5     017154 G        4137#   4825
ERR5$    020160          4129    4156    4370#
ERR7A    017252 G        2701    4164#   5389
EVL    = 000004 G        1559#
EVRC   = 002400          1701#
EXADD  = 000020          1713#
EXCON  = 000010          1714#
EXECUT= 000005           1606#
E$END  = 002100          1297#
E$LOAD= 000035           1297#   1392
FLGCA1= 000002           1942#
FLGCA2= 000001           1943#
FLGCB1= 000020           1939#
FLGCB2= 000010           1940#
FLGIRQ= 000200           1931#
FLGSR  = 000004          1941#
FLGT1  = 000100          1937#
FLGT2  = 000040          1938#
FMT10    012252          4064#   4151
FMT10A   012326          4064#
FMT11    012347          4064#   4384    4401
FMT15    012366          4064#   4170
FMT15A   012420          4064#
FMT19    012472          4064#
FMT2     011650          4064#   4141
FMT21    012523          4064#   4203    4248    4293    4328    4973
FMT22    012533          4064#   4209    4254    4299
FMT23    012555          4064#   4229    4274    4338
FMT24    012634          4064#   4423    4500
FMT25    012647          4064#   4433    4467    4510
FMT26    012677          4064#   4450    4484    4527
FMT27    012732          4064#   4219    4264    4309
FMT29    012741          4064#   4440    4457    4474    4517
FMT3     011705          4064#   4180
FMT30    012746          4064#
FMT31    013003          4064#
FMT32    013051          4064#
FMT4     011771          4064#   4374
FMT4A    012027          4064#
FMT4B    012062          4064#   4391
FMT4C    012067          4064#
FMT5     012122          4064#   4106
FMT5A    012165          4064#   4122
FRSTIM   002444          2080#   4563    4572*
F$AU   = 000015          1297#   4750    4752
F$AUTO= 000020           1297#   4670    4706
F$BGN  = 000040          1297#   1300    4098    4137    4164    4199    4244    4289    4324    4543    4556    4670    4720
                         4734    4750    4770    4777    4786    4808    4828    4851    4855    4881    4885    4896    4913
                         4917    4921    4969    5008    5042    5048    5073    5091    5096    5119    5157    5168    5185
                         5191    5199    5210    5214    5220    5237    5243    5251    5262    5266    5272    5291    5297
                         5305    5316    5320    5326    5351    5357    5365    5391    5395    5401    5417    5423    5431
                         5442    5446    5452    5469    5475    5494    5511    5517    5536    5564    5596    5622    5649
                         5679    5690    5711    5722    5730    5734    5766    5777    5787    5797    5807    5816    5827
                         5835    5838    5865    5875    5884    5894    5912    5923    5938    5950    5962    5977    5988
                         5998    6014    6024    6039    6054    6066    6069    6093    6103    6112    6122    6140    6151
```

```
                    6166  6178  6190  6205  6220  6230  6243  6251  6254  6274  6285  6303  6313
                    6329  6340  6351  6354  6377  6387  6409  6432  6449  6460  6469  6472  6504
                    6510  6519  6539  6557  6564  6570  6579  6599  6617  6624  6630  6639  6659
                    6677  6684  6687  6723  6733  6742  6752  6770  6784  6794  6797  6820  6830
                    6839  6849  6867  6881  6891  6894  6917  6927  6936  6946  6964  6978  6988
                    6991  7011  7021  7039  7049  7059  7069  7078  7089  7099  7110  7120  7131
                    7142  7153  7156  7186  7192  7201  7217  7227  7240  7253  7273  7284  7299
                    7313  7324  7327  7333  7342  7358  7368  7381  7394  7414  7425  7440  7454
                    7465  7468  7474  7483  7499  7509  7522  7535  7555  7566  7581  7595  7606
                    7609  7615  7624  7640  7650  7663  7676  7696  7707  7722  7736  7747  7750
                    7756  7765  7781  7791  7804  7817  7837  7848  7863  7877  7888  7891  7897
                    7906  7922  7932  7945  7958  7978  7989  8004  8018  8029  8032  8035  8061
                    8100  8119  8128  8142  8151  8170  8184  8194  8197  8220  8230  8252  8264
                    8278  8289  8303  8322  8333  8344  8353  8356  8399  8402  8413  8432  8444
                    8453  8456  8460  8471  8491  8494  8498  8509  8542  8545  8548  8569  8580
                    8602  8612  8627  8632  8641  8654  8664  8674  8684  8693  8704  8718  8729
                    8741  8745  8785  8795  8817  8827  8837  8847  8856  8869  8879  8889  8899
                    8909  8919  8929  8938  8949  8963  8974  8986  8989  9019  9029  9039  9052
                    9062  9072  9082  9092  9101  9112  9125  9137  9149  9159  9168  9171  9187
                    9250  9268
FSCLEA= 000007       1297#  4720  4724
FSDU  = 000016       1297#  4734  4739
FSEND = 000041       1297#  1300  4133  4160  4194  4238  4283  4318  4347  4653  4708  4726  4741
                     4754  4770  4777  4786  4808  4828  4830  4851  4855  4881  4883  4885  4896
                     4913  4915  4917  4919  4967  4982  5008  5042  5048  5050  5073  5091  5096
                     5098  5119  5157  5168  5170  5185  5191  5199  5210  5214  5216  5220  5222
                     5237  5243  5251  5262  5266  5268  5272  5274  5291  5297  5305  5316  5320
                     5322  5326  5328  5351  5357  5365  5391  5395  5397  5401  5403  5417  5423
                     5431  5442  5446  5448  5452  5454  5469  5475  5494  5496  5511  5517  5536
                     5538  5564  5596  5622  5649  5651  5679  5690  5711  5722  5730  5734  5736
                     5766  5777  5787  5797  5807  5816  5827  5835  5838  5840  5865  5875  5884
                     5894  5912  5923  5938  5950  5962  5977  5988  5998  6014  6024  6039  6054
                     6066  6069  6071  6093  6103  6112  6122  6140  6151  6166  6178  6190  6205
                     6220  6230  6243  6251  6254  6256  6274  6285  6303  6313  6329  6340  6351
                     6354  6356  6377  6387  6409  6432  6449  6460  6469  6472  6474  6504  6510
                     6519  6539  6557  6564  6566  6570  6579  6599  6617  6624  6626  6630  6639
                     6659  6677  6684  6686  6687  6689  6723  6733  6742  6752  6770  6784  6794
                     6797  6799  6820  6830  6839  6849  6867  6881  6891  6894  6896  6917  6927
                     6936  6946  6964  6978  6988  6991  6993  7011  7021  7039  7049  7059  7069
                     7078  7089  7099  7110  7120  7131  7142  7153  7156  7158  7186  7192  7201
                     7217  7227  7240  7253  7273  7284  7299  7313  7324  7327  7329  7333  7342
                     7358  7368  7381  7394  7414  7425  7440  7454  7465  7468  7470  7474  7483
                     7499  7509  7522  7535  7555  7566  7581  7595  7606  7609  7611  7615  7624
                     7640  7650  7663  7676  7696  7707  7722  7736  7747  7750  7752  7756  7765
                     7781  7791  7804  7817  7837  7848  7863  7877  7888  7891  7893  7897  7906
                     7922  7932  7945  7958  7978  7989  8004  8018  8029  8032  8034  8035  8037
                     8061  8100  8119  8128  8142  8151  8170  8184  8194  8197  8199  8220  8230
                     8252  8264  8278  8289  8303  8322  8333  8344  8353  8356  8358  8399  8402
                     8413  8432  8444  8453  8456  8458  8460  8471  8491  8494  8496  8498  8509
                     8542  8545  8547  8548  8550  8569  8580  8602  8612  8622  8632  8641  8654
                     8664  8674  8684  8693  8704  8718  8729  8741  8745  8747  8785  8795  8817
                     8827  8837  8847  8856  8869  8879  8889  8899  8909  8919  8929  8938  8949
                     8963  8974  8986  8989  8991  9019  9029  9039  9052  9062  9072  9082  9092
                     9101  9112  9125  9137  9149  9159  9168  9171  9173  9234  9256  9268
FSHARD= 000004       1297#  9187  9232
FSHW  = 000013       1297#  1465  1481
```

CVDMCA.P11 10-DEC-80 09:14 CROSS REFERENCE TABLE -- USER SYMBOLS

```
F$INIT= 000006          1297#   4556    4651
F$JMP  = 000050          1297#
F$MOD  = 000000          1297#   1300    9268
F$MSG  = 000011          1297#   4098    4131    4137    4158    4164    4192    4199    4236    4244    4281    4289    4316
                         4324    4345    4921    4965    4969    4980
F$PROT= 000021           1297#   4543    4548
F$PWR  = 000017          1297#
F$RPT  = 000012          1297#
F$SEG  = 000003          1297#
F$SOFT= 000005           1297#   9250    9254
F$SRV  = 000010          1297#
F$SUB  = 000002          1297#   4856    4881    4886    4913    5200    5214    5252    5266    5306    5320    5366    5395
                         5432    5446    6511    6564    6571    6624    6631    6684    7193    7327    7334    7468    7475
                         7609    7616    7750    7757    7891    7898    8032    8403    8456    8461    8494    8499    8545
F$SW   = 000014          1297#   1490    1495
F$TEST= 000001           1297#   4771    4828    4852    4917    5009    5048    5074    5096    5120    5168    5186    5220
                         5238    5272    5292    5326    5352    5401    5418    5452    5470    5494    5512    5536    5565
                         5649    5680    5734    5767    5838    5866    6069    6094    6254    6275    6354    6378    6472
                         6505    6687    6724    6797    6821    6894    6918    6991    7012    7156    7187    8035    8062
                         8197    8221    8356    8400    8548    8570    8745    8786    8989    9020    9171
GDATA   002400           2061#   2434*   2478*   2525*   2596*   2692    2965*   2966*   3069*   3501*   3759*   3760*   3777*
                         3778*   4099*   4101    4121    4150    4179    4228    4273    4337    4360    4800*   4817*   5031*
                         5032*   5146*   5147*   5203*   5255*   5309*   5369*   5380    5435*   5585*   5611*   5637*
GETBSR  004046           2628#
GETPRM  021264           4601    4615#   4624
GETURS  004402           2725#   4496
GETVRS  004502           2756#   4419
GETWSR  004210           2433    2477    2524    2595    2649#   4799    4811
GOODAT  002440           2078#
G$CNTO= 000200           1297#
G$DELM= 000372           1297#
G$DISP= 000003           1297#
G$EXCP= 000400           1297#
G$HILI= 000002           1297#
G$LOLI= 000001           1297#
G$ND   = 000000          1297#
G$OFFS= 000400           1297#   9191    9196    9201    9207    9213    9219    9225
G$OFSI= 000376           1297#   9191    9196    9201    9207    9213    9219    9225
G$PRMA= 000001           1297#   9191    9196
G$PRMD= 000002           1297#   9201    9207    9213    9219    9225
G$PRML= 000000           1297#
G$RADA= 000140           1297#
G$RADB= 000000           1297#
G$RADD= 000040           1297#
G$RADL= 000120           1297#
G$RADO= 000020           1297#   9191    9196    9201    9207    9213    9219    9225
G$XFER= 000004           1297#
G$YES  = 000010          1297#   9191    9196    9201    9207    9213    9219    9225
HDX    = 000004          1747#
HELP   = 000000          1288#   1322    1410    1456    2403
HOE    = 100000 G        1572#
IBE    = 010000 G        1569#
IDLE   = 000010          1689#
IDLES  = 004000          1698#   5771    5869    6097    6824    7195    9023
IDU    = 000040 G        1562#
IEI    = 000001          1579#
```

CVDMCA.P11 10-DEC-80 09:14 CROSS REFERENCE TABLE -- USER SYMBOLS

```
IEO    = 000020           1578#
IER    = 020000 G         1570#
INIDMV  005420            3047#  4852   5009   5020   5074   5120   5186   5238   5292   5352   5418   5470   5512
                          5565   5681   5768   5866   6094   6276   6378   6505   6724   6821   6918   7012   7187
                          8062   8221   8404   8462   8500   8571   8786   9020
INITRN  007400            3537#  5683   5770   5868   6096   6278   6380   6512   6572   6632   6726   6823   6920
                          7014   7194   7335   7476   7617   7758   7899   8223   8406   8464   8502   8573   8788
                          9022
INITT1  004556            2794#
INITT2  004756            2881#
INTFLG  002422            2070#
INTGRL= 000001            1996#
INTSC  = 000200           1953#
ISR    = 000100 G         1563#
IXE    = 004000 G         1568#
I$AU   = 000041           1297#  4750#  4754#
I$AUTO= 000041            1297#  4670#  4708#
I$CLN  = 000041           1297#  4720#  4726#
I$DU   = 000041           1297#  4734#  4741#
I$HRD  = 000041           9187#  9234#
I$INIT= 000041            1297#  4556#  4653#
I$MOD  = 000041           1297#  1300#  9268#
I$MSG  = 000041           1297#  4098#  4133#  4137#  4160#  4164#  4194#  4199#  4238#  4244#  4283#  4289#  4318#
                          4324#  4347#  4921#  4967#  4969#  4982#
I$PROT= 000040            1297#  4543#
I$PTAB= 000041            1297#
I$PWR  = 000041           1297#
I$RPT  = 000041           1297#
I$SEG  = 000041           1297#  4770   4851   4855   4885   5008   5073   5119   5185   5199   5237   5251   5291
                          5305   5351   5365   5417   5431   5469   5511   5564   5679   5766   5865   6093   6274
                          6377   6504   6510   6570   6630   6723   6820   6917   7011   7186   7192   7333   7474
                          7615   7756   7897   8061   8220   8399   8402   8460   8498   8569   8785   9019
I$SETU= 000041            1297#
I$SFT  = 000041           9250#  9256#
I$SRV  = 000041           1297#
I$SUB  = 000041           1297#  4770   4851   4855#  4881#  4883#  4885#  4896   4913#  4915#  5008   5073   5119
                          5185   5199#  5214#  5216#  5237   5251#  5266#  5268#  5291   5305#  5320#  5322#  5351
                          5365#  5395#  5397#  5417   5431#  5446#  5448#  5469   5511   5564   5679   5766   5865
                          6093   6274   6377   6504   6510#  6519   6539   6557   6564#  6566#  6570#  6579   6599
                          6617   6624#  6626#  6630#  6639   6659   6677   6684#  6686#  6723   6820   6917   7011
                          7186   7192#  7201   7217   7227   7240   7253   7273   7284   7299   7313   7324   7327#
                          7329#  7333#  7342   7358   7368   7381   7394   7414   7425   7440   7468#  7470#  7474#
                          7483   7499   7509   7522   7535   7555   7566   7581   7609#  7611#  7615#  7624   7640
                          7650   7663   7676   7696   7707   7722   7750#  7752#  7756#  7765   7781   7791   7804
                          7817   7837   7848   7863   7891#  7893#  7897#  7906   7922   7932   7945   7958   7978
                          7989   8004   8032#  8034#  8061   8220   8399   8402#  8413   8432   8453   8456#  8458#
                          8460#  8471   8494#  8496#  8498#  8509   8542   8545#  8547#  8569   8785   9019
I$TST  = 000041           1297#  4770#  4777   4786   4808   4828#  4830#  4851#  4855   4885   4917#  4919#  5008#
                          5042   5048#  5050#  5073#  5091   5096#  5098#  5119#  5157   5168#  5170#  5185#  5191
                          5199   5210   5220#  5222#  5237#  5243   5251   5262   5272#  5274#  5291#  5297   5305
                          5316   5326#  5328#  5351#  5357   5365   5391   5401#  5403#  5417#  5423   5431   5442
                          5452#  5454#  5469#  5475   5494#  5496#  5511#  5517   5536#  5538#  5564#  5596   5622
                          5649#  5651#  5679#  5690   5711   5722   5730   5734#  5736#  5766#  5777   5787   5797
                          5807   5816   5827   5835   5838#  5840#  5865#  5875   5884   5894   5912   5923   5938
                          5950   5962   5977   5988   5998   6014   6024   6039   6054   6066   6069#  6071#  6093#
                          6103   6112   6122   6140   6151   6166   6178   6190   6205   6220   6230   6243   6251
```

```
L$LAST  037426 G          1350    9274#
L$LOAD  002100 G          1391#
L$LUN   002074 G          1387#
L$MREV  002050 G          1367#
L$NAME  002000 G          1324#
L$PRIO  002042 G          1361#
L$PROT  021072 G          1402    4543#
L$PRT   002112 G          1401#
L$REPP  002062 G          1377#
L$REV   002010 G          1333#
L$SOFT  037314 G          9250    9251#
L$SPC   002056 G          1373#
L$SPCP  002020 G          1343#
L$SPTP  002024 G          1347#
L$STA   002030 G          1351#
L$SW    002246 G          1490    1491#
L$TEST  002114 G          1403#
L$TIML  002014 G          1339#
L$UNIT  002012 G          1337#
L10000  002244            1465    1481#
L10001  002246            1490    1495#
L10002  017152            4131#
L10003  017250            4158#
L10004  017370            4192#
L10005  017544            4236#
L10006  017720            4281#
L10007  020034            4316#
L10010  020132            4345#
L10012  021412            4651#
L10013  021530            4706#
L10014  021540            4724#
L10015  021544            4739#
L10016  021546            4752#
L10017  021726            4778    4787    4809    4828#
L10020  022114            4917#
L10021  022012            4881#
L10022  022112            4897    4913#
L10023  022306            4965#
L10024  022344            4980#
L10025  023112            5043    5048#
L10026  023176            5092    5096#
L10027  023422            5158    5168#
L10030  023514            5192    5211    5220#
L10031  023510            5214#
L10032  023606            5244    5263    5272#
L10033  023602            5266#
L10034  023700            5298    5317    5326#
L10035  023674            5320#
L10036  024030            5358    5392    5401#
L10037  024024            5395#
L10040  024122            5424    5443    5452#
L10041  024116            5446#
L10042  024176            5476    5494#
L10043  024252            5518    5536#
L10044  024574            5597    5623    5649#
L10045  024726            5691    5712    5723    5731    5734#
```

```
L10046  025110          5778    5788    5798    5808    5817    5828    5836    5838#
L10047  025616          5876    5885    5895    5913    5924    5939    5951    5963    5978    5989    5999    6015    6025
                        6040    6055    6067    6069#
L10050  026216          6104    6113    6123    6141    6152    6167    6179    6191    6206    6221    6231    6244    6252
                        6254#
L10051  026424          6286    6304    6314    6330    6341    6352    6354#
L10052  026666          6388    6410    6433    6450    6461    6470    6472#
L10053  027400          6687#
L10054  027056          6520    6540    6558    6564#
L10055  027226          6580    6600    6618    6624#
L10056  027376          6640    6660    6678    6684#
L10057  027606          6734    6743    6753    6771    6785    6795    6797#
L10060  030002          6831    6840    6850    6868    6882    6892    6894#
L10061  030176          6928    6937    6947    6965    6979    6989    6991#
L10062  030554          7022    7040    7050    7060    7070    7079    7090    7100    7111    7121    7132    7143    7154
                        7156#
L10063  033466          7455    7466    7596    7607    7737    7748    7878    7889    8019    8030    8035#
L10064  031146          7202    7218    7228    7241    7254    7274    7285    7300    7314    7325    7327#
L10065  031534          7343    7359    7369    7382    7395    7415    7426    7441    7468#
L10066  032122          7484    7500    7510    7523    7536    7556    7567    7582    7609#
L10067  032510          7625    7641    7651    7664    7677    7697    7708    7723    7750#
L10070  033076          7766    7782    7792    7805    7818    7838    7849    7864    7891#
L10071  033464          7907    7923    7933    7946    7959    7979    7990    8005    8032#
L10072  034054          8101    8120    8129    8143    8152    8171    8185    8195    8197#
L10073  034424          8231    8253    8265    8279    8290    8304    8323    8334    8345    8354    8356#
L10074  035042          8445    8492    8548#
L10075  034600          8414    8433    8454    8456#
L10076  034702          8472    8494#
L10077  035040          8510    8543    8545#
L10100  035502          8581    8603    8613    8623    8633    8642    8655    8665    8675    8685    8694    8705    8719
                        8730    8742    8745#
L10101  036240          8796    8818    8828    8838    8848    8857    8870    8880    8890    8900    8910    8920    8930
                        8939    8950    8964    8975    8987    8989#
L10102  036634          9030    9040    9053    9063    9073    9083    9093    9102    9113    9126    9138    9150    9160
                        9169    9171#
L10103  036742          9187    9233#
L10104  037314          9250    9255#
MCLR   = 000100         1590#   1592    2424    2434
MDMRDY= 000040          1760#
MLWRI   003744          2561    2586#
MPCSR   002472          2095#   4201    4246    4291    4326    4629    4971
MPIVEC  002532          2119#   4637*
MPOVEC  002534          2120#   4638*   4639*
MPRIOR  002536          2121#   4640*
MRDY   = 000200         1597#   2431    2474    2521    2593    4796
MREQ   = 000001         1591#   1592    2424    2434
MSTCLR  003374          2424#   3047    4772
NCRACT= 020000          2005#   7137    7148    7308    7319    7449    7460    7590    7601    7731    7742    7872    7883
                        8013    8024
NCTBMT= 000200          2001#   3644    6404    8247    8311    9154
NEWLIN  011645          4064#   4959
NEWST   021244          4593    4608#
NFCRDA= 040000          2004#   7148    7319    7460    7601    7742    7883    8024
NOCHK  = 003400         1700#   5684    5771    5869    6097    6279    6381    6513    6573    6633    6727    6824    6921
                        7015    8224    8407    8465    8503    8574    8789
NOCRDA= 100000          2003#   6427    6444    6455    6552    6612    6672    7268    7279    7294    7308    7319    7409
```

```
                    7420    7435    7550    7561    7576    7691    7702    7717    7832    7843    7858    7973    7984
                    7999    8273    8284    8298    8317    8328    8339
NOLOOP= 001000      1999#
NORXEN= 040000      1998#   5870    6098    6728    6825    6922    8575    8790
NULCLK= 000200      1742#   5579    5585    5605    5631    5637
OVRC  = 002000      1702#
O$APTS= 000000      1297#   1351
O$AU  = 000001      1297#   1321#   1383
O$BGNR= 000000      1297#   1377
O$BGNS= 000000      1297#   1343
O$DU  = 000001      1297#   1321#   1385
O$ERRT= 000001      1297#   1321#   1393
O$GNSW= 000000      1297#   1347
O$POIN= 000001      1297#   1321#   1409
O$SETU= 000000      1297#   1337    9272
PALENB= 000001      1907#
PATCH   037314      9258#
PATE    002652      2175#   5011    5076
PATF    002662      2186#   2955    5022    5121
PATG    002672      2197#   5124    5194    5195
PATH    002721      2223#   5195    5246    5247    5426    5427    5478    5479    5520    5521
PATI    002752      2251#   5247    5300    5301    5427    5479    5521
PATJ    003013      2287#   5129    5301
PATK    003023      2298#   5360    5361
PATL    003044      2318#   5361    5362
PATQ    003065      2338#
PATQB   003075      2348#
PBLENB= 000002      1906#
PCR   = 120407      1708#   3557    5310    7231    7372    7513    7654    7795    7936
PCSARH= 120405      1681#   3551    5204    6004    6211    8087
PCSARL= 120404      1674#   3547    5256    8084
PNT   = 001000 G    1566#
PRESET= 000001      1749#   2949    3541    8078
PRI   = 002000 G    1567#
PRIOR   002416      2068#
PRIRTY  037021      9202    9235#
PRI00 = 000000 G    1555#
PRI01 = 000040 G    1554#
PRI02 = 000100 G    1553#
PRI03 = 000140 G    1552#
PRI04 = 000200 G    1551#
PRI05 = 000240 G    1550#
PRI06 = 000300 G    1549#
PRI07 = 000340 G    1548#
PROTO = 000100      1686#   6005
PSTACK  002414      2067#   4558*
RABGA = 000004      1631#   3831    3833    3843
RAMADR= 001000      1993#
RCVBUF  003106      2362#
RCVDAT= 000002      1764#
RCVIST  011364      3952#   5705    5810    6323    7072    8687    8932    9095
RDA   = 000200      1724#   3238    3249    3973
RDSRH = 120401      1625#   3322    3369    3731    5526    5530
RDSRL = 120400      1619#   3734    5484    5488
READ    003476      2466#   2687    2729    2734    2759    5375    9118
READI   003610      2512#   2816    2903    2957    3062    3098    3143    3188    3233    3278    3321    3368    3435
```

```
                        3591    3730    3733    3744    3970    4857    4860    4889    5024    5140    5487    5529    5576
                        5602    5628    5926    5965    6042    6154    6193
REDBYT  002432          2075#
REDDAT  002572          2140#   5122    5130*   5137*   5143    5147
REDLOC= 000001          1602#   2467    2478    2514    2525
REDPAG= 000003          1604#
REGNUM  002412          2066#   2963*   2964*   3068*   3097*   3142*   3187*   3232*   3277*   3320*   3367*   3434*   3749*
                        3758*   3776*   3781*   4139    4165    4214    4259    4304    4819*   5029*   5030*   5145*   5196*
                        5248*   5302*   5383*   5428*   5584*   5610*   5636*
REG0    002602          2143#   5575*   5581*   5601*   5607*   5627*   5633*
REG1    002604          2144#
REG2    002606          2145#
REG3    002610          2146#
REG4    002612          2147#
REG5    002614          2148#
REG6    002616          2149#
REG7    002620          2150#
REOM  = 000002          1632#   3397    3408    3854    3856    3866
RERCHK= 000001          1642#   3782    7318    7459    7600    7730    7871    8012
RERR  = 000200          1628#   3785    3787    3797
RETADR  002430          2074#
RING  = 000200          1758#
ROR   = 000010          1630#   3326    3337    3808    3810    3820
RSA   = 000020          1727#   3283    3294
RSOM  = 000001          1633#   3373    3384    3877    3879    3889
RSTCHK  005126          2943#   5086    5725    5830    6246    8486
RTSND = 000010          1746#   2949    2952    3025    3541    3544    8078    8081    8092
RUN   = 000200          1589#   1592    2424    2434
RXABGA= 002000          1638#
RXACT = 000040          1726#   3148    3159
RXCHAR  010110          3727#   5714    5819    6332    6343    6424    6441    6452    6549    6609    6669    7081    7102
                        7123    7134    7145    7265    7276    7291    7305    7316    7406    7417    7432    7446    7457
                        7547    7558    7573    7587    7598    7688    7699    7714    7728    7739    7829    7840    7855
                        7869    7880    7970    7981    7996    8010    8021    8270    8281    8295    8314    8325    8336
                        8696    8721    8941    8966    9104
RXDL  = 000007          1715#   3738    6634    7901    7937
RXEN  = 000100          1743#   3577    3607    6318    8092    8534    8591    8806    9128
RXEOM = 001000          1639#   7146    7317    7458    7599    7740    7881    8022
RXERR = 100000          1636#
RXOR  = 004000          1637#
RXSOM = 000400          1640#   7082    7266    7407    7548    7689    7830    7971
SAVE4   002446          2081#   4565*   4569
SAVE6   002450          2082#   4566*   4570
SAVLEN  002456          2085#   3554*   3555*   3738    4562*
SCRACH  002406          2064#
SECAD = 000020          1688#
SECADR= 010000          1697#
SEL0    002472          2094#   2649
SEL10   002512          2107#   2653
SEL12   002516          2110#   2654
SEL14   002522          2113#   2655
SEL16   002526          2116#   2656
SEL2    002476          2098#   2650    4789*
SEL4    002502          2101#   2466*   2513*   2559*   2584*   2651
SEL6    002506          2104#   2536    2560*   2585*   2652
SERIAL  007256          3479#   5887    5905    5943    5991    6007    6017    6059    6115    6133    6171    6223    6236
```

CVDMCA.P11 10-DEC-80 09:14 CROSS REFERENCE TABLE -- USER SYMBOLS

```
                    6745   6763   6777   6787   6842   6860   6874   6884   6939   6957   6971   6981   8112
                    8135   8163   8177   8187
SFTVIA  005326      3013#  3048
SFPTBL  002246 G    1492#
SFR     = 000001    1731#
SPEED   = 000020    1761#
SRMODE  = 000034    1895#
STALL   004400      2712#
STARES  002466      2089#  4604*  4610*
STARST  021234      4579   4586   4603#
STEPLU  011614      3488   3589   3650   3698   3918   3967   4018   4041#  5953   6181   6320   6415   6438
                    7262   7403   7544   7685   7826   7967   8109   8435   8483   8521
STRIP   = 000040    1687#
STRIPS  = 020000    1696#  8789   9023
STRTML  = 000301    1592#
STUREG  004272      2678#  5205   5257   5311   5437
SUBRPC  002420      2069#  4559*
SVCGBL  = 000000    1297#  1300   1307#  1324   1333   1335   1337   1339   1341   1343   1345   1347   1349
                    1351   1353   1355   1357   1359   1361   1363   1365   1367   1370   1373   1375   1377
                    1379   1381   1383   1385   1387   1389   1391   1393   1395   1397   1399   1401   1403
                    1405   1407   1423   1466   1467   1491   1492   2019   2379   2391   4098   4137   4164
                    4199   4244   4289   4324   4543   4556   4670   4720   4734   4750   4921   4969   9188
                    9251   9274#  9275
SVCINS  = 000001    1297#  1304#  1325   1326   1327   1328   1329   1330   1331   1332   1334   1336   1338
                    1340   1342   1344   1346   1348   1350   1352   1354   1356   1358   1360   1362   1364
                    1366   1368   1369   1371   1372   1374   1376   1378   1380   1382   1384   1386   1388
                    1390   1392   1394   1396   1398   1400   1402   1404   1406   1408   1422   1424   1425
                    1426   1427   1428   1429   1430   1431   1432   1433   1434   1435   1436   1437   1438
                    1439   1440   1441   1442   1443   1444   1445   1446   1447   1448   1449   1450   1451
                    1452   1453   1454   1465   1490   2380   2383   2392   2399   2437   2438   2439   2440
                    2481   2482   2483   2484   2528   2529   2530   2531   2599   2600   2601   2602   2698
                    2699   2700   2701   2971   2972   2973   2974   3075   3076   3077   3078   3108   3109
                    3110   3111   3119   3120   3121   3122   3153   3154   3155   3156   3164   3165   3166
                    3167   3198   3199   3200   3201   3209   3210   3211   3212   3243   3244   3245   3246
                    3254   3255   3256   3257   3288   3289   3290   3291   3299   3300   3301   3302   3331
                    3332   3333   3334   3342   3343   3344   3345   3378   3379   3380   3381   3389   3390
                    3391   3392   3402   3403   3404   3405   3413   3414   3415   3416   3445   3446   3447
                    3448   3457   3458   3459   3460   3506   3507   3508   3509   3753   3754   3755   3756
                    3766   3767   3768   3769   3792   3793   3794   3795   3802   3803   3804   3805   3815
                    3816   3817   3818   3825   3826   3827   3828   3838   3839   3840   3841   3848   3849
                    3850   3851   3861   3862   3863   3864   3871   3872   3873   3874   3884   3885   3886
                    3887   3894   3895   3896   3897   4105   4106   4107   4108   4109   4110   4120   4121
                    4122   4123   4124   4125   4126   4132   4139   4140   4141   4142   4143   4144   4145
                    4148   4149   4150   4151   4152   4153   4154   4155   4159   4168   4169   4170   4171
                    4172   4173   4174   4177   4178   4179   4180   4181   4182   4183   4184   4186   4187
                    4188   4189   4190   4193   4201   4202   4203   4204   4205   4206   4207   4209   4210
                    4211   4212   4213   4217   4218   4219   4220   4221   4222   4223   4226   4227   4228
                    4229   4230   4231   4232   4233   4237   4246   4247   4248   4249   4250   4251   4252
                    4254   4255   4256   4257   4258   4262   4263   4264   4265   4266   4267   4268   4271
                    4272   4273   4274   4275   4276   4277   4278   4282   4291   4292   4293   4294   4295
                    4296   4297   4299   4300   4301   4302   4303   4307   4308   4309   4310   4311   4312
                    4313   4317   4326   4327   4328   4329   4330   4331   4332   4335   4336   4337   4338
                    4339   4340   4341   4342   4346   4372   4373   4374   4375   4376   4377   4378   4380
                    4381   4382   4383   4384   4385   4386   4387   4388   4390   4391   4392   4393   4394
                    4395   4397   4398   4399   4400   4401   4402   4403   4404   4405   4407   4408   4409
                    4410   4411   4421   4422   4423   4424   4425   4426   4427   4429   4430   4431   4432
```

4433 4434 4435 4436 4437 4439 4440 4441 4442 4443 4444 4446 4447
4448 4449 4450 4451 4452 4453 4454 4456 4457 4458 4459 4460 4461
4463 4464 4465 4466 4467 4468 4469 4470 4471 4473 4474 4475 4476
4477 4478 4480 4481 4482 4483 4484 4485 4486 4487 4488 4498 4499
4500 4501 4502 4503 4504 4506 4507 4508 4509 4510 4511 4512 4513
4514 4516 4517 4518 4519 4520 4521 4523 4524 4525 4526 4527 4528
4529 4530 4531 4576 4577 4579 4583 4584 4586 4590 4591 4593 4597
4598 4600 4618 4619 4620 4622 4648 4652 4673 4674 4675 4676 4677
4678 4695 4696 4700 4701 4707 4725 4737 4740 4753 4775 4777 4778
4784 4786 4787 4803 4804 4805 4806 4808 4809 4822 4823 4824 4825
4829 4856 4872 4873 4874 4875 4877 4882 4886 4894 4896 4897 4908
4909 4910 4911 4914 4918 4923 4924 4925 4926 4927 4929 4930 4931
4932 4933 4935 4936 4937 4938 4939 4941 4942 4943 4944 4945 4946
4947 4951 4952 4953 4954 4955 4956 4957 4959 4960 4961 4962 4963
4966 4971 4972 4973 4974 4975 4976 4977 4981 5037 5038 5039 5040
5042 5043 5049 5089 5091 5092 5097 5152 5153 5154 5155 5157 5158
5169 5189 5191 5192 5200 5208 5210 5211 5215 5221 5241 5243 5244
5252 5260 5262 5263 5267 5273 5295 5297 5298 5306 5314 5316 5317
5321 5327 5355 5357 5358 5366 5386 5387 5388 5389 5391 5392 5396
5402 5421 5423 5424 5432 5440 5442 5443 5447 5453 5473 5475 5476
5495 5515 5517 5518 5537 5591 5592 5593 5594 5596 5597 5617 5618
5619 5620 5622 5623 5643 5644 5645 5646 5650 5688 5690 5691 5709
5711 5712 5720 5722 5723 5728 5730 5731 5735 5775 5777 5778 5785
5787 5788 5795 5797 5798 5805 5807 5808 5814 5816 5317 5825 5827
5828 5833 5835 5836 5839 5873 5875 5876 5882 5884 5885 5892 5894
5895 5910 5912 5913 5921 5923 5924 5933 5934 5935 5936 5938 5939
5948 5950 5951 5960 5962 5963 5972 5973 5974 5975 5977 5978 5986
5986 5989 5996 5998 5999 6012 6014 6015 6022 6024 6025 6037 6039
6040 6049 6050 6051 6052 6054 6055 6064 6066 6067 6070 6101 6103
6104 6110 6112 6113 6120 6122 6123 6138 6140 6141 6149 6151 6152
6161 6162 6163 6164 6166 6167 6176 6178 6179 6188 6190 6191 6200
6201 6202 6203 6205 6206 6218 6220 6221 6228 6230 6231 6241 6243
6244 6249 6251 6252 6255 6283 6285 6286 6301 6303 6304 6311 6313
6314 6327 6329 6330 6338 6340 6341 6349 6351 6352 6355 6385 6387
6388 6407 6409 6410 6430 6432 6433 6447 6449 6450 6458 6460 6461
6467 6469 6470 6473 6511 6517 6519 6520 6537 6539 6540 6555 6557
6558 6565 6571 6577 6579 6580 6597 6599 6600 6615 6617 6618 6625
6631 6637 6639 6640 6657 6659 6660 6675 6677 6678 6685 6688 6731
6733 6734 6740 6742 6743 6750 6752 6753 6768 6770 6771 6782 6784
6785 6792 6794 6795 6798 6828 6830 6831 6837 6839 6840 6847 6849
6850 6865 6867 6868 6879 6881 6882 6889 6891 6892 6895 6925 6927
6928 6934 6936 6937 6944 6946 6947 6962 6964 6965 6976 6978 6979
6986 6988 6989 6992 7019 7021 7022 7037 7039 7040 7047 7049 7050
7057 7059 7060 7067 7069 7070 7076 7078 7079 7087 7089 7090 7097
7099 7100 7108 7110 7111 7118 7120 7121 7129 7131 7132 7140 7142
7143 7151 7153 7154 7157 7193 7199 7201 7202 7215 7217 7218 7225
7227 7228 7238 7240 7241 7251 7253 7254 7271 7273 7274 7282 7284
7285 7297 7299 7300 7311 7313 7314 7322 7324 7325 7328 7334 7340
7342 7343 7356 7358 7359 7366 7368 7369 7379 7381 7382 7392 7394
7395 7412 7414 7415 7423 7425 7426 7438 7440 7441 7452 7454 7455
7463 7465 7466 7469 7475 7481 7483 7484 7497 7499 7500 7507 7509
7510 7520 7522 7523 7533 7535 7536 7553 7555 7556 7564 7566 7567
7579 7581 7582 7593 7595 7596 7604 7606 7607 7610 7616 7622 7624
7625 7638 7640 7641 7648 7650 7651 7661 7663 7664 7674 7676 7677
7694 7696 7697 7705 7707 7708 7720 7722 7723 7734 7736 7737 7745
7747 7748 7751 7757 7763 7765 7766 7779 7781 7782 7789 7791 7792

```
TDSRNR  002651          2172#
TEOM  = 000002          1660#   6968    7114    7260    7401    7542    7683    7824    7965    8106
TERR  = 000200          1657#
TGA   = 000010          1658#   6968
TIMFLG  002426          2073#
TM    = 000004          1763#
TMP0    002622          2153#   4679*   4697    4709*
TMP1    002624          2154#
TMP2    002626          2155#
TMP3    002630          2156#
TMP4    002632          2157#
TMP5    002634          2158#
TMP6    002636          2159#
TMP7    002640          2160#
TSO   = 000010          1728#   3440    3452    5918    5957    6034    6146    6185    9163
TSOM  = 000001          1661#   3581    5694    5698    5702    6028    6289    6391    6523    6583    6643    7025    7205
                        7346    7487    7628    7769    7910    8106    8132    8233    8417    8420    8475    8478    8513
                        8516    8584    8799
TSTCON  002546          2125#   4644*
TSTNUM  002470          2090#
TTLOOP= 000002          1748#   3577    3607    4017    6318    8092    8526    8534    8591    8806
TXAB  = 002000          1666#
TXACT = 000004          1729#   3103    3114    5918    5957    6034    6146    6185    9144    9163
TXCHAR  007676          3627#   5780    5790    5800    6296    6306    6402    6532    6592    6652    7032    7042    7052
                        7062    7092    7210    7220    7233    7246    7351    7361    7374    7387    7492    7502    7515
                        7528    7633    7643    7656    7669    7774    7784    7797    7810    7915    7925    7938    7951
                        8245    8309    8595    8605    8615    8625    8647    8657    8667    8677    8711    8810    8820
                        8830    8840    8862    8872    8882    8892    8902    8912    8922    8956    9032    9045    9055
                        9065    9075    9085    9152
TXCTRL  010010          3680#   5693    5697    5701    5901    6027    6129    6288    6292    6390    6393    6522    6525
                        6582    6585    6642    6645    6755    6773    6852    6870    6949    6967    7024    7028    7113
                        7204    7207    7259    7345    7348    7400    7486    7489    7541    7627    7630    7682    7768
                        7771    7823    7909    7912    7964    8105    8131    8159    8232    8236    8416    8419    8422
                        8474    8477    8480    8512    8515    8518    8583    8586    8798    8801    9042
TXDL  = 000340          1712#   3738    6634    7937
TXEN  = 000040          1744#   3577    3607    8092    8526    8534    8591    8806
TXEND1  035521          8732    8761#
TXEND2  036254          8977    9002#
TXEND3  026702          6412    6435    6485#
TXEND4  034441          8255    8305    8370#
TXEOM = 001000          1667#
TXERR = 100000          1664#
TXGA  = 004000          1665#
TXSOM = 000400          1668#
TXTBL1  035504          8707    8748#
TXTBL2  036242          8952    8992#
TXTBL3  026670          6399    6421    6475#
TXTBL4  034426          8242    8292    8359#
TXTMLT  016702          4073#   4120
TXTML0  016113          4064#   4073
TXTML1  016117          4064#   4073
TXTML2  016133          4064#   4073
TXTML3  016150          4064#   4073
TXTML4  016172          4064#   4073
TXTML5  016213          4064#   4073
TXTML6  016243          4064#   4073
```

TXTML7	016255	4064#	4073				
TXTNP	016437	4064#	4085				
TXTNPT	016766	4086#					
TXTNP0	016444	4064#	4086				
TXTNP1	016454	4064#	4086				
TXTNP2	016464	4064#	4086				
TXTNP3	016474	4064#	4086				
TXTNP4	016511	4064#	4086				
TXTNP5	016526	4064#	4086				
TXTNP6	016543	4064#	4086				
TXTNP7	016557	4064#	4086				
TXTNP8	016573	4064#	4084				
TXTNUL	016111	4064#					
TXTUR	016607	4064#	4160	4218	4308		
TXTURT	017010	4089#	4168	4217	4307		
TXTUR0	016622	4064#	4089				
TXTUR1	016630	4064#	4089				
TXTUR2	016636	4064#	4089				
TXTUR3	016644	4064#	4089				
TXTUR4	016652	4064#	4089				
TXTUR5	016661	4064#	4089				
TXTUR6	016670	4064#	4089				
TXTUR7	016674	4064#	4089				
TXTVR	016312	4064#	4077	4263			
TXTVRA	016410	4064#	4081				
TXTVRB	016413	4064#	4081				
TXTVRC	016417	4064#	4081				
TXTVRD	016423	4064#	4081				
TXTVRE	016427	4064#	4081				
TXTVRF	016433	4064#	4081				
TXTVRT	016724	4078#	4262				
TXTVR0	016330	4064#	4078				
TXTVR1	016334	4064#	4078				
TXTVR2	016340	4064#	4078				
TXTVR3	016345	4064#	4078				
TXTVR4	016352	4064#	4078				
TXTVR5	016357	4064#	4078				
TXTVR6	016364	4064#	4078				
TXTVR7	016371	4064#	4078				
TXTVR8	016376	4064#	4081				
TXTVR9	016403	4064#	4081				
TXT1	014541	4064#					
TXT10	015341	4064#					
TXT11	015361	4064#					
TXT11A	015433	4064#					
TXT11B	015471	4064#					
TXT12	015541	4064#	4202	4247	4292	4327	4972
TXT13	015567	4064#	4499				
TXT14	015604	4064#	4498				
TXT15	015642	4064#	4516				
TXT16	015704	4064#	4422				
TXT17	015717	4064#	4421				
TXT18	015754	4064#	4439				
TXT19	016015	4064#	4456				
TXT2	014577	4064#					
TXT2A	014641	4064#					

CVDMCA.P11 10-DEC-80 09:14 CROSS PEFERENCE TABLE -- USER SYMBOLS

```
TXT2B   014700          4064#
TXT20   016051          4064#   4473
TXT3    014743          4064#
TXT4    014773          4064#   4372
TXT4A   015033          4064#   4390
TXT5    015074          4064#   4140
TXT6    015076          4064#   4373
TXT7    015121          4064#
TXT7A   015211          4064#
TXT8    015301          4064#
TXT9    015321          4064#
TXU     =  000002       1730#   3747    5957    6185
T$ARGC= 000003          1325#   1326#   1327#   1328#   1329#   1330#   4105#   4110    4120#   4126    4139#   4145    4148#
                        4155    4168#   4174    4177#   4184    4186#   4190    4201#   4207    4209#   4213    4217#   4223
                        4226#   4233    4246#   4252    4254#   4258    4262#   4268    4271#   4278    4291#   4297    4299#
                        4303    4307#   4313    4326#   4332    4335#   4342    4372#   4378    4380#   4388    4390#   4395
                        4397#   4405    4407#   4411    4421#   4427    4429#   4437    4439#   4444    4446#   4454    4456#
                        4461    4463#   4471    4473#   4478    4480#   4488    4498#   4504    4506#   4514    4516#   4521
                        4523#   4531    4923#   4927    4929#   4933    4935#   4939    4941#   4947    4951#   4957    4959#
                        4963    4971#   4977
T$CODE= 007032          9191#   9196#   9201#   9207#   9213#   9219#   9225#
T$ERRN= 000065          1297#   4804#   4823#   4873#   4909#   5038#   5153#   5387#   5592#   5618#   5644#   5934#   5973#
                        6050#   6162#   6201#
T$EXCP= 000000          9191#   9195    9196#   9200    9201#   9206    9207#   9212    9213#   9218    9219#   9224    9225#
                        9230
T$FLAG= 000040          4777#   4786#   4808#   4896#   5042#   5091#   5157#   5191#   5210#   5243#   5262#   5297#   5316#
                        5357#   5391#   5423#   5442#   5475#   5517#   5596#   5622#   5690#   5711#   5722#   5730#   5777#
                        5787#   5797#   5807#   5816#   5827#   5835#   5875#   5884#   5894#   5912#   5923#   5938#   5950#
                        5962#   5977#   5988#   5998#   6014#   6024#   6039#   6054#   6066#   6103#   6112#   6122#   6140#
                        6151#   6166#   6178#   6190#   6205#   6220#   6230#   6243#   6251#   6285#   6303#   6313#   6329#
                        6340#   6351#   6387#   6409#   6432#   6449#   6460#   6469#   6519#   6535#   6557#   6579#   6599#
                        6617#   6639#   6659#   6677#   6733#   6742#   6752#   6770#   6784#   6794#   6830#   6839#   6849#
                        6867#   6881#   6891#   6927#   6936#   6946#   6964#   6978#   6988#   7021#   7039#   7049#   7059#
                        7069#   7078#   7089#   7099#   7110#   7120#   7131#   7142#   7153#   7201#   7217#   7227#   7240#
                        7253#   7273#   7284#   7299#   7313#   7324#   7342#   7358#   7368#   7381#   7394#   7414#   7425#
                        7440#   7454#   7465#   7483#   7499#   7509#   7522#   7535#   7555#   7566#   7581#   7595#   7606#
                        7624#   7640#   7650#   7663#   7676#   7696#   7707#   7722#   7736#   7747#   7765#   7781#   7791#
                        7804#   7817#   7837#   7848#   7863#   7877#   7888#   7906#   7922#   7932#   7945#   7958#   7978#
                        7989#   8004#   8018#   8029#   8100#   8119#   8128#   8142#   8151#   8170#   8184#   8194#   8230#
                        8252#   8264#   8278#   8289#   8303#   8322#   8333#   8344#   8353#   8413#   8432#   8444#   8453#
                        8471#   8491#   8509#   8542#   8580#   8602#   8612#   8622#   8632#   8641#   8654#   8664#   8674#
                        8684#   8693#   8704#   8718#   8729#   8741#   8795#   8817#   8827#   8837#   8847#   8856#   8869#
                        8879#   8889#   8899#   8909#   8919#   8929#   8938#   8949#   8963#   8974#   8986#   9029#   9039#
                        9052#   9062#   9072#   9082#   9092#   9101#   9112#   9125#   9137#   9149#   9159#   9168#
T$GMAN= 000000          1297#
T$HILI= 000001          9191#   9194    9196#   9199    9201#   9205    9207#   9211    9213#   9217    9219#   9223    9225#
                        9229
T$LAST= 000001          1297#   9272#
T$LOLI= 000000          9191#   9193    9196#   9198    9201#   9204    9207#   9210    9213#   9216    9219#   9222    9225#
                        9228
T$LSYM= 010000          1297#   1482    1496    4132    4159    4193    4237    4282    4317    4346    4652    4707    4725
                        4740    4753    4829    4882    4914    4918    4966    4981    5049    5097    5169    5215    5221
                        5267    5273    5321    5327    5396    5402    5447    5453    5495    5537    5650    5735    5839
                        6070    6255    6355    6473    6565    6625    6685    6688    6798    6895    6992    7157    7328
                        7469    7610    7751    7892    8033    8036    8198    8357    8457    8495    8546    8549    8746
                        8990    9172    9234    9256
```

```
TSLTNO= 000037          9275#
TSNEST= 177777          1297#  1300#  1465#  1481#  1490#  1495#  4098#  4131#  4137#  4158#  4164#  4192#  4199#
                        4236#  4244#  4281#  4289#  4316#  4324#  4345#  4543#  4548#  4556#  4651#  4670#  4706#
                        4720#  4724#  4734#  4739#  4750#  4752#  4771#  4828#  4852#  4856#  4881#  4686#  4913#
                        4917#  4921#  4965#  4969#  4980#  5009#  5048#  5074#  5096#  5120#  5168#  5186#  5200#
                        5214#  5220#  5238#  5252#  5266#  5272#  5292#  5306#  5320#  5326#  5352#  5366#  5395#
                        5401#  5418#  5432#  5446#  5452#  5470#  5494#  5512#  5536#  5565#  5649#  5680#  5734#
                        5767#  5838#  5866#  6069#  6094#  6254#  6275#  6354#  6378#  6472#  6505#  6511#  6564#
                        6571#  6624#  6631#  6684#  6687#  6724#  6797#  6821#  6894#  6918#  6991#  7012#  7156#
                        7187#  7193#  7327#  7334#  7468#  7475#  7609#  7616#  7750#  7757#  7891#  7898#  8032#
                        8035#  8062#  8197#  8221#  8356#  8400#  8403#  8456#  8461#  8494#  8499#  8545#  8548#
                        8570#  8745#  8786#  8989#  9020#  9171#  9187#  9232#  9250#  9254#  9268#
TSNS0 = 000000           1300#  9268
TSNS1 = 000005           1465#  1481   1490#  1495   4098#  4131   4137#  4158   4164#  4192   4199#  4236   4244#
                         4281   4289#  4316   4324#  4345   4543#  4548   4556#  4651   4670#  4706   4720#  4724
                         4734#  4739   4750#  4752   4771#  4828   4852#  4917   4921#  4965   4969#  4980   5009#
                         5048   5074#  5096   5120#  5168   5186#  5220   5238#  5272   5292#  5326   5352#  5401
                         5418#  5452   5470#  5494   5512#  5536   5565#  5649   5680#  5734   5767#  5838   5866#
                         6069   6094#  6254   6275#  6354   6378#  6472   6505#  6687   6724#  6797   6821#  6894
                         6918#  6991   7012#  7156   7187#  8035   8062#  8197   8221#  8356   8400#  8548   8570#
                         8745   8786#  8989   9020#  9171   9187#  9232   9250#  9274
TSNS2 = 000002           4856#  4881   4886#  4913   5200#  5214   5252#  5266   5306#  5320   5366#  5395   5432#
                         5446   6511#  6564   6571#  6624   6631#  6684   7193#  7327   7334#  7468   7475#  7609
                         7616#  7750   7757#  7891   7898#  8032   8403#  8456   8461#  8494   8499#  8545
TSPTNU= 000000           1297#
TSSAVL= 177777           1297#
TSSEGL= 177777           1297#
TSSUBN= 000000           1297#  4770#  4851#  4855#  4885#  5008#  5073#  5119#  5185#  5199#  5237#  5251#  5291#
                         5305#  5351#  5365#  5417#  5431#  5469#  5511#  5564#  5679#  5766#  5865#  6093#  6274#
                         6377#  6504#  6510#  6570#  6630#  6723#  6820#  6917#  7011#  7186#  7192#  7333#  7474#
                         7615#  7756#  7897#  8061#  8220#  8399#  8402#  8460#  8498#  8569#  8785#  9019#
TSTAGL= 177777           1297#
TSTAGN= 010105           1297#  1465#  1490#  4098#  4137#  4164#  4199#  4244#  4289#  4324#  4543#  4556#  4670#
                         4720#  4734#  4750#  4771#  4852#  4856#  4886#  4921#  4969#  5009#  5074#  5120#  5186#
                         5200#  5238#  5252#  5292#  5306#  5352#  5366#  5418#  5432#  5470#  5512#  5565#  5680#
                         5767#  5866#  6094#  6275#  6378#  6505#  6511#  6571#  6631#  6724#  6821#  6918#  7012#
                         7187#  7193#  7334#  7475#  7616#  7757#  7898#  8062#  8221#  8400#  8403#  8461#  8499#
                         8570#  8786#  9020#  9187#  9250#
TSTEMP= 000000           1415#  1419#  1424#  1425#  1426#  1427#  1428#  1429#  1430#  1431#  1432#  1433#  1434#
                         1435#  1436#  1437#  1438#  1439#  1440#  1441#  1442#  1443#  1444#  1445#  1446#  1447#
                         1448#  1449#  1450#  1451#  1452#  1453#  1454#  1455#  1481#  1495#  4131#  4158#  4192#
                         4236#  4281#  4316#  4345#  4548#  4651#  4706#  4724#  4739#  4752#  4777#  4778   4786#
                         4787   4808#  4809   4828#  4881#  4896#  4897   4913#  4917#  4965#  4980#  5042#  5043
                         5048#  5091#  5092   5096#  5157#  5158   5168#  5191#  5192   5210#  5211   5214#  5220#
                         5243#  5244   5262#  5263   5266#  5272#  5297#  5298   5316#  5317   5320#  5326#  5357#
                         5358   5391#  5392   5395#  5401#  5423#  5424   5442#  5443   5446#  5452#  5475#  5476
                         5494#  5517#  5518   5536#  5596#  5597   5622#  5623   5649#  5690#  5691   5711#  5712
                         5722#  5723   5730#  5731   5734#  5777#  5778   5787#  5788   5797#  5798   5807#  5808
                         5816#  5817   5827#  5828   5835#  5836   5838#  5875#  5876   5884#  5885   5894#  5895
                         5912#  5913   5923#  5924   5938#  5939   5950#  5951   5962#  5963   5977#  5978   5988#
                         5989   5998#  5999   6014#  6015   6024#  6025   6039#  6040   6054#  6055   6066#  6067
                         6069#  6103#  6104   6112#  6113   6122#  6123   6140#  6141   6151#  6152   6166#  6167
                         6178#  6179   6190#  6191   6205#  6206   6220#  6221   6230#  6231   6243#  6244   6251#
                         6252   6254#  6285#  6286   6303#  6304   6313#  6314   6329#  6330   6340#  6341   6351#
                         6352   6354#  6387#  6388   6409#  6410   6432#  6433   6449#  6450   6460#  6461   6469#
                         6470   6472#  6519#  6520   6539#  6540   6557#  6558   6564#  6579#  6580   6599#  6600
```

```
                    6879  6881  6889  6891  6895  6925  6927  6934  6936  6944  6946  6962  6964
                    6976  6978  6986  6988  6992  7019  7021  7037  7039  7047  7049  7057  7059
                    7067  7069  7076  7078  7087  7089  7097  7099  7108  7110  7118  7120  7129
                    7131  7140  7142  7151  7153  7157  7193  7199  7201  7215  7217  7225  7227
                    7238  7240  7251  7253  7271  7273  7282  7284  7297  7299  7311  7313  7322
                    7324  7328  7334  7340  7342  7356  7358  7366  7368  7379  7381  7392  7394
                    7412  7414  7423  7425  7438  7440  7452  7454  7463  7465  7469  7475  7481
                    7483  7497  7499  7507  7509  7520  7522  7533  7535  7553  7555  7564  7566
                    7579  7581  7593  7595  7604  7606  7610  7616  7622  7624  7638  7640  7648
                    7650  7661  7663  7674  7676  7694  7696  7705  7707  7720  7722  7734  7736
                    7745  7747  7751  7757  7763  7765  7779  7781  7789  7791  7802  7804  7815
                    7817  7835  7837  7846  7848  7861  7863  7875  7877  7886  7888  7892  7898
                    7904  7906  7920  7922  7930  7932  7943  7945  7956  7958  7976  7978  7987
                    7989  8002  8004  8016  8018  8027  8029  8033  8036  8098  8100  8117  8119
                    8126  8128  8140  8142  8149  8151  8168  8170  8182  8184  8192  8194  8198
                    8228  8230  8250  8252  8262  8264  8276  8278  8287  8289  8301  8303  8320
                    8322  8331  8333  8342  8344  8351  8353  8357  8403  8411  8413  8430  8432
                    8442  8444  8451  8453  8457  8461  8469  8471  8489  8491  8495  8499  8507
                    8509  8540  8542  8546  8549  8578  8580  8600  8602  8610  8612  8620  8622
                    8630  8632  8639  8641  8652  8654  8662  8664  8672  8674  8682  8684  8691
                    8693  8702  8704  8716  8718  8727  8729  8739  8741  8746  8793  8795  8815
                    8817  8825  8827  8835  8837  8845  8847  8854  8856  8867  8869  8877  8879
                    8887  8889  8897  8899  8907  8909  8917  8919  8927  8929  8936  8938  8947
                    8949  8961  8963  8972  8974  8984  8986  8990  9027  9029  9037  9039  9050
                    9052  9060  9062  9070  9072  9080  9082  9090  9092  9099  9101  9110  9112
                    9123  9125  9135  9137  9147  9149  9157  9159  9166  9168  9172
T$TSTS= 000001       1297#  4771#  4852#  5009#  5074#  5120#  5186#  5238#  5292#  5352#  5418#  5470#  5512#
                    5565#  5680#  5767#  5866#  6094#  6275#  6378#  6505#  6724#  6821#  6918#  7012#  7187#
                    8062#  8221#  8400#  8570#  8786#  9020#
T$$AU = 010016       4750#  4752
T$$AUT= 010013       4670#  4706
T$$CLE= 010014       4720#  4724
T$$DU = 010015       4734#  4739
T$$HAR= 010103       9187#  9233
T$$HW = 010000       1465#  1481
T$$INI= 010012       4556#  4651
T$$MSG= 010024       4098#  4131  4137#  4158  4164#  4192  4199#  4236  4244#  4281  4289#  4316  4324#
                    4345  4921#  4965  4969#  4980
T$$PRO= 010011       4543#
T$$SOF= 010104       9250#  9255
T$$SUB= 010077       4856#  4881  4886#  4896  4913  5200#  5214  5252#  5266  5306#  5320  5366#  5395
                    5432#  5446  6511#  6519  6539  6557  6564  6571#  6579  6599  6617  6624  6631#
                    6639  6659  6677  6684  7193#  7201  7217  7227  7240  7253  7273  7284  7299
                    7313  7324  7327  7334#  7342  7358  7368  7381  7394  7414  7425  7440  7468
                    7475#  7483  7499  7509  7522  7535  7555  7566  7581  7609  7616#  7624  7640
                    7650  7663  7676  7696  7707  7722  7750  7757#  7765  7781  7791  7804  7817
                    7837  7848  7863  7891  7898#  7906  7922  7932  7945  7958  7978  7989  8004
                    8032  8403#  8413  8432  8453  8456  8461#  8471  8494  8499#  8509  8542  8545
T$$SW = 010001       1490#  1495
T$$TES= 010102       4771#  4777  4786  4808  4828  4852#  4917  5009#  5042  5048  5074#  5091  5096
                    5120#  5157  5168  5186#  5191  5210  5220  5238#  5243  5262  5272  5292#  5297
                    5316  5326  5352#  5357  5391  5401  5418#  5423  5442  5452  5470#  5475  5494
                    5512#  5517  5536  5565#  5596  5622  5649  5680#  5690  5711  5722  5730  5734
                    5767#  5777  5787  5797  5807  5816  5827  5835  5838  5866#  5875  5884  5894
                    5912  5923  5938  5950  5962  5977  5988  5998  6014  6024  6039  6054  6066
                    6069  6094#  6103  6112  6122  6140  6151  6166  6178  6190  6205  6220  6230
```

```
                        6243    6251    6254    6275#   6285    6303    6313    6329    6340    6351    6354    6378#   6387
                        6409    6432    6449    6460    6469    6472    6505#   6687    6724#   6733    6742    6752    6770
                        6784    6794    6797    6821#   6830    6839    6849    6867    6881    6891    6894    6918#   6927
                        6936    6946    6964    6978    6988    6991    7012#   7021    7039    7049    7059    7069    7078
                        7089    7099    7110    7120    7131    7142    7153    7156    7187#   7454    7465    7595    7606
                        7736    7747    7877    7888    8018    8029    8035    8062#   8100    8119    8128    8142    8151
                        8170    8184    8194    8197    8221#   8230    8252    8264    8278    8289    8303    8322    8333
                        8344    8353    8356    8400#   8444    8491    8548    8570#   8580    8602    8612    8622    8632
                        8641    8654    8664    8674    8684    8693    8704    8718    8729    8741    8745    8786#   8795
                        8817    8827    8837    8847    8856    8869    8879    8889    8899    8909    8919    8929    8938
                        8949    8963    8974    8986    8989    9020#   9029    9039    9052    9062    9072    9082    9092
                        9101    9112    9125    9137    9149    9159    9168    9171
T.EDF  = 000001         2007#   2437    2481    2528    2599    2698    2971    3075    3108    3119    3153    3164    3198
                        3209    3243    3254    3288    3299    3331    3342    3378    3389    3402    3413    3445    3457
                        3506    3753    3766    3792    3802    3815    3825    3838    3848    3861    3871    3884    3894
T.EHRD= 000002          2007#
T.ESF  = 000000         2007#
T.ESFT= 000003          2007#
T1      021550 G        1424    4770#
T1MODE= 000300          1880#
T10     024032 G        1433    5417#
T10.LP  023732          5364#   5398
T10.1   024064          5431#
T11     024124 G        1434    5469#
T11.LP  024064          5430#   5449
T12     024200 G        1435    5511#
T13     024254 G        1436    5564#
T14     024576 G        1437    5679#
T15     024730 G        1438    5766#
T16     025112 G        1439    5865#
T17     025620 G        1440    6093#
T18     026220 G        1441    6274#
T19     026426 G        1442    6377#
T2      021730 G        1425    4851#
T2MODE= 000040          1890#
T2.1    021734          4855#
T2.2    022014          4885#
T20     026704 G        1443    6504#
T20.1   026710          6510#
T20.2   027060          6570#
T20.3   027230          6630#
T21     027414 G        1444    6723#
T22     027610 G        1445    6820#
T23     030004 G        1446    6917#
T24     030200 G        1447    7011#
T24TBL  027402          6529    6542    6545    6560    6589    6602    6605    6620    6649    6662    6665    6680    6690#
T25     030556 G        1448    7186#
T25.1   030562          7192#
T25.2   031150          7333#
T25.3   031536          7474#
T25.4   032124          7615#
T25.5   032512          7756#
T25.6   033100          7897#
T26     033500 G        1449    8061#
T27     034056 G        1450    8220#
T28     034442 G        1451    8399#
```

```
T28.1    034442          8402#
T28.2    034602          8460#
T28.3    034704          8498#
T29      035044 G        1452    8569#
T3       022730 G        1426    5008#
T3.EHD   022346          4874    4983#
T3.EM1   022116 G        4875    4921#
T3.EM2   022310 G        4911    4969#
T3.END   022114          4916#
T3.ERR   021776          4865    4870#
T3.OK    022010          4868    4879#
T3.SW1   021744          4859#   4864    4948*   4952
T3.SW2   021754          4862#   4867    4949*   4951
T3.1     022372          4923    4983#
T3.2     022457          4929    4983#
T3.3     022506          4935    4983#
T3.4     022537          4943    4983#
T3.5     022602          4953    4983#
T3.6     022644          4910    4983#
T30      035522 G        1453    8785#
T30TBL   033470          7243    7256    7287    7302    7384    7397    7428    7443    7525    7538    7569    7584    7666
                         7679    7710    7725    7807    7820    7851    7866    7948    7961    7992    8007    8038#
T31      036256 G        1454    9019#
T4       023114 G        1427    5073#
T5       023200 G        1428    5119#
T6       023424 G        1429    5185#
T6.1     023456          5199#
T7       023516 G        1430    5237#
T7.LP    023456          5198#   5217
T7.1     023550          5251#
T8       023610 G        1431    5291#
T8.LP    023550          5250#   5269
T8.1     023642          5305#
T9       023702 G        1432    5351#
T9.LP    023642          5304#   5323
T9.1     023732          5365#
UAM    = 000200 G        1564#
UMAINT= 000001           1765#
UNIT     002464          2088#
UPBITS   002642          2163#
UREGS    002316          2053#   2725    2728*   2731    4506    4507    4508    4509    4523    4524    4525    4526
USTATR= 120000           1720#   2730    3063    3099    3144    3189    3234    3279    3436    3592    3745    3971
USYREG   002552          2130#   2956    5012    5023    5077    5128    5139
USYRT  = 120400          1613#   2726    2739    2743
VECTOR   036770          9197    9235#
VIA    = 120000          1734#   2757    2764
VIAACR= 120013           1878#   2817    2826    2904    2913    3027    3560    8067
VIADPA= 120003           1784#   3018
VIADPB= 120002           1772#   3015
VIAIER= 120016           1951#   2813    2900    3033
VIAIFR= 120015           1929#
VIAMS  = 120001          1756#
VIAORA= 120017           1971#   3021    4890
VIAORB= 120000           1740#   2948    2951    3024    3540    3543    3576    4016    5577    5603    5629    6317    8077
                         8080    8091    8525    8533    8590    8805    9     9131
VIAPCR= 120014           1916#   3030
```

```
VIASR = 120012          1868#
VIAT1A= 120004          1798#   2839    5567
VIAT1B= 120005          1810#   2844    4045    5570
VIAT1C= 120006          1823#   2830    3563    8070
VIAT1D= 120007          1834#   2835    3566    8073
VIAT2A= 120010          1845#   2917
VIAT2B= 120011          1857#   2922
VREGS   002336          2056#   2756    4429    4430    4431    4432    4446    4447    4448    4449    4463    4464    4465
                        4466    4480    4481    4482    4483
WAIT50  005312          2990#   8529
WRIBYT  002434          2076#
WRILOC= 000002          1603#   2586    2596
WRIPAG= 000004          1605#
WRITE   003722          2559#   2681    5371
WRITEI  003734          2583#   2812    2825    2829    2834    2838    2843    2899    2912    2916    2921    2947    2950
                        3014    3017    3020    3023    3026    3029    3032    3539    3542    3546    3550    3556    3559
                        3562    3565    3575    3579    3582    3631    3684    4015    4044    5014    5079    5134    5483
                        5525    5566    5569    5897    6003    6125    6210    6316    6759    6856    6953    7230    7371
                        7512    7653    7794    7935    8066    8069    8072    8076    8079    8083    8086    8090    8156
                        8173    8524    8532    8589    8804    9130
WSR0    002256          2028#   2649*   4383
WSR10   002266          2036#   2653*   4400
WSR12   002270          2038#   2654*   4399
WSR14   002272          2040#   2655*   4398
WSR16   002274          2042#   2656*   4397
WSR2    002260          2030#   2650*   4382    4812    4818
WSR4    002262          2032#   2651*   4381
WSR6    002264          2034#   2652*   4380
XDATA   002404          2063#   4148    4177    4226    4271    4335    4361*   4362*
XORGB   020134          4146    4175    4224    4269    4333    4359#
XYZ   = 000007          1690#   6005    8088
X$ALWA= 000000          1297#
X$FALS= 000040          1297#
X$OFFS= 000400          1297#
X$TRUE= 000020          1297#
$E    = 000065          2007#   2436#   2480#   2527#   2598#   2697#   2970#   3074#   3107#   3118#   3152#   3163#   3197#
                        3208#   3242#   3253#   3287#   3298#   3330#   3341#   3377#   3388#   3401#   3412#   3444#   3476#
                        3505#   3752#   3765#   3791#   3801#   3814#   3824#   3837#   3847#   3860#   3870#   3883#   3893*
                        4802#   4821#   4871#   4907#   5036#   5151#   5385#   5590#   5616#   5642#   5932#   5971#   6048#
                        6160#   6199#
$LSTIN= 000001          1302#
$LSTTA= 000001          1303#
$T    = 000037          2007#   4755#   4831#   4985#   5050#   5098#   5170#   5222#   5274#   5328#   5403#   5454#   5496#
                        5538#   5651#   5736#   5841#   6071#   6256#   6356#   6488#   6702#   6799#   6896#   6993#   7158#
                        8047#   8199#   8373#   8550#   8764#   9005#
.     - 037426          1293#   2053#   2056#   2140#   2357#   2362#   2383#   2399#   3511    4773    4778    4782    4787
                        4809    4892    4897    5043    5087    5092    5158    5192    5211    5244    5263    5298    5317
                        5358    5392    5424    5443    5476    5518    5597    5623    5686    5691    5707    5712    5718
                        5723    5726    5731    5773    5778    5783    5788    5793    5798    5803    5808    5812    5817
                        5823    5828    5831    5836    5871    5876    5880    5885    5890    5895    5908    5913    5919
                        5924    5939    5946    5951    5958    5963    5978    5984    5989    5994    5999    6010    6015
                        6020    6025    6035    6040    6055    6062    6067    6099    6104    6108    6113    6118    6123
                        6136    6141    6147    6152    6167    6174    6179    6186    6191    6206    6216    6221    6226
                        6231    6239    6244    6247    6252    6281    6286    6299    6304    6309    6314    6325    6330
                        6336    6341    6347    6352    6383    6388    6405    6410    6428    6433    6445    6450    6456
                        6461    6465    6470    6486#   6515    6520    6535    6540    6553    6558    6575    6580    6595
```

6600	6613	6618	6635	6640	6655	6660	6673	6678	6729	6734	6738	6743
6748	6753	6766	6771	6780	6785	6790	6795	6826	6831	6835	6840	6845
6850	6863	6868	6877	6882	6887	6892	6923	6928	6932	6937	6942	6947
6960	6965	6974	6979	6984	6989	7017	7022	7035	7040	7045	7050	7055
7060	7065	7070	7074	7079	7085	7090	7095	7100	7106	7111	7116	7121
7127	7132	7138	7143	7149	7154	7197	7202	7213	7218	7223	7228	7236
7241	7249	7254	7269	7274	7280	7285	7295	7300	7309	7314	7320	7325
7338	7343	7354	7359	7364	7369	7377	7382	7390	7395	7410	7415	7421
7426	7436	7441	7450	7455	7461	7466	7479	7484	7495	7500	7505	7510
7518	7523	7531	7536	7551	7556	7562	7567	7577	7582	7591	7596	7602
7607	7620	7625	7636	7641	7646	7651	7659	7664	7672	7677	7692	7697
7703	7708	7718	7723	7732	7737	7743	7748	7761	7766	7777	7782	7787
7792	7800	7805	7813	7818	7833	7838	7844	7849	7859	7864	7873	7878
7884	7889	7902	7907	7918	7923	7928	7933	7941	7946	7954	7959	7974
7979	7985	7990	8000	8005	8014	8019	8025	8030	8045#	8096	8101	8115
8120	8124	8129	8138	8143	8147	8152	8166	8171	8180	8185	8190	8195
8226	8231	8248	8253	8260	8265	8274	8279	8285	8290	8299	8304	8318
8323	8329	8334	8340	8345	8349	8354	8409	8414	8428	8433	8440	8445
8449	8454	8467	8472	8487	8492	8505	8510	8538	8543	8576	8581	8598
8603	8608	8613	8618	8623	8628	8633	8637	8642	8650	8655	8660	8665
8670	8675	8680	8685	8689	8694	8700	8705	8714	8719	8725	8730	8737
8742	8791	8796	8813	8818	8823	8828	8833	8838	8843	8848	8852	8857
8865	8870	8875	8880	8885	8890	8895	8900	8905	8910	8915	8920	8925
8930	8934	8939	8945	8950	8959	8964	8970	8975	8982	8987	9003#	9025
9030	9035	9040	9048	9053	9058	9063	9068	9073	9078	9083	9086	9093
9097	9102	9108	9113	9121	9126	9133	9138	9145	9150	9155	9160	9164
9169	9259#											

CVDMCA.P11 10-DEC-80 09:14 CROSS REFERENCE TABLE -- MACRO NAMES

```
BCOMPL      1#  1297#  4578  4585  4592  4599  4621
BERROR      1#  1297#
BGNAU       1#  1297#  4749
BGNAUT      1#  1297#  4669
BGNCLN      1#  1297#  4719
BGNDU       1#  1297#  4733
BGNHRD      1#  1297#  9186
BGNHW       1#  1297#  1464
BGNINI      1#  1297#  4555
BGNMOD      1#  1297#  1299
BGNMSG      1#  1297#  4097  4136  4163  4198  4243  4288  4323  4920  4968
BGNPRO      1#  1297#  4542
BGNPTA      1#  1297#
BGNRPT      1#  1297#
BGNSEG      1#  1297#
BGNSET      1#  1297#
BGNSFT      1#  1297#  9249
BGNSRV      1#  1297#
BGNSUB      1#  1297#  4854  4884  5198  5250  5304  5364  5430  6509  6569  6629  7191  7332  7473
         7614   7755   7896  8401  8459  8497
BGNSW       1#  1297#  1489
BGNTST      1#  1297#  4770  4851  5008  5073  5119  5185  5237  5291  5351  5417  5469  5511  5564
         5679   5766   5865  6093  6274  6377  6504  6723  6820  6917  7011  7186  8061  8220  8399
         8569   8785   9019
BNCOMP      1#  1297#
BNERRO      1#  1297#
BREAK       1#  1297#
BRESET      1#  1297#  4647  4736
CKLOOP      1#  1297#  4876
CLOCK       1#  1297#
CLOSE       1#  1297#
CLRVEC      1#  1297#  4694
COMMEN      1#  1297#
DELAY       1#  1297#
DESCRI      1#  1297#  2390
DEVTYP      1#  1297#  2378
DISPAT      1#  1297#  1421
DISPLA      1#  1297#
DOCLN       1#  1297#
DODU        1#  1297#  4699
DORPT       1#  1297#
ENDAU       1#  1297#  4751
ENDAUT      1#  1297#  4705
ENDCLN      1#  1297#  4723
ENDCOM      1#  1297#
ENDDU       1#  1297#  4738
ENDHRD      1#  1297#  9231
ENDHW       1#  1297#  1480
ENDINI      1#  1297#  4650
ENDMOD      1#  1297#  9267
ENDMSG      1#  1297#  4130  4157  4191  4235  4280  4315  4344  4964  4979
ENDPRO      1#  1297#  4547
ENDPTA      1#  1297#
ENDRPT      1#  1297#
ENDSEG      1#  1297#
ENDSET      1#  1297#
```

```
ENDSFT      1#  1297#  9253
ENDSRV      1#  1297#
ENDSUB      1#  1297#  4880  4912  5213  5265  5319  5394  5445  6563  6623  6683  7326  7467  7608
         7749   7890   8031  8455  8493  8544
ENDSW       1#  1297#  1494
ENDTST      1#  1297#  4827  4916  5047  5095  5167  5219  5271  5325  5400  5451  5493  5535  5648
         5733   5837   6068  6253  6353  6471  6686  6796  6893  6990  7155  8034  8196  8355  8547
         8744   8988   9170
EQUALS      1#  1297#  1504
ERRDF       1#  1297#  4803  4822  4872  4908  5037  5152  5386  5591  5617  5643  5933  5972  6049
         6161   6200
ERRHRD      1#  1297#
ERROR       1#  1297#  4774  4783  4893  5088  5188  5207  5240  5259  5294  5313  5354  5420  5439
         5472   5514   5687  5708  5719  5727  5774  5784  5794  5804  5813  5824  5832  5872  5881
         5891   5909   5920  5947  5959  5985  5995  6011  6021  6036  6063  6100  6109  6119  6137
         6148   6175   6187  6217  6227  6240  6248  6282  6300  6310  6326  6337  6348  6384  6406
         6429   6446   6457  6466  6516  6536  6554  6576  6596  6614  6636  6656  6674  6730  6739
         6749   6767   6781  6791  6827  6836  6846  6864  6878  6888  6924  6933  6943  6961  6975
         6985   7018   7036  7046  7056  7066  7075  7086  7096  7107  7117  7128  7139  7150  7198
         7214   7224   7237  7250  7270  7281  7296  7310  7321  7339  7355  7365  7378  7391  7411
         7422   7437   7451  7462  7480  7496  7506  7519  7532  7552  7563  7578  7592  7603  7621
         7637   7647   7660  7673  7693  7704  7719  7733  7744  7762  7778  7788  7801  7814  7834
         7845   7860   7874  7885  7903  7919  7929  7942  7955  7975  7986  8001  8015  8026  8097
         8116   8125   8139  8148  8167  8181  8191  8227  8249  8261  8275  8286  8300  8319  8330
         8341   8350   8410  8429  8441  8450  8468  8488  8506  8539  8577  8599  8609  8619  8629
         8638   8651   8661  8671  8681  8690  8701  8715  8726  8738  8792  8814  8824  8834  8844
         8853   8866   8876  8886  8896  8906  8916  8926  8935  8946  8960  8971  8983  9026  9036
         9049   9059   9069  9079  9089  9098  9109  9122  9134  9146  9156  9165
ERRSF       1#  1297#
ERRSOF      1#  1297#
ERRTBL      1#  1297#  2018
ESCAPE      1#  1297#  4776  4785  4807  4895  5041  5090  5156  5190  5209  5242  5261  5296  5315
         5356   5390   5422  5441  5474  5516  5595  5621  5689  5710  5721  5729  5776  5786  5796
         5806   5815   5826  5834  5874  5883  5893  5911  5922  5937  5949  5961  5976  5987  5997
         6013   6023   6038  6053  6065  6102  6111  6121  6139  6150  6165  6177  6189  6204  6219
         6229   6242   6250  6284  6302  6312  6328  6339  6350  6386  6408  6431  6448  6459  6468
         6518   6538   6556  6578  6598  6616  6638  6658  6676  6732  6741  6751  6769  6783  6793
         6829   6838   6848  6866  6880  6890  6926  6935  6945  6963  6977  6987  7020  7038  7048
         7058   7068   7077  7089  7098  7109  7119  7130  7141  7152  7200  7216  7226  7239  7252
         7272   7283   7298  7312  7323  7341  7357  7367  7380  7393  7413  7424  7439  7453  7464
         7482   7498   7508  7521  7534  7554  7565  7580  7594  7605  7623  7639  7649  7662  7675
         7695   7706   7721  7735  7746  7764  7780  7790  7803  7816  7836  7847  7862  7876  7887
         7905   7921   7931  7944  7957  7977  7988  8003  8017  8028  8099  8118  8127  8141  8150
         8169   8183   8193  8229  8251  8263  8277  8288  8302  8321  8332  8343  8352  8412  8431
         8443   8452   8470  8490  8508  8541  8579  8601  8611  8621  8631  8640  8653  8663  8673
         8683   8692   8703  8717  8728  8740  8794  8816  8826  8836  8846  8855  8868  8878  8888
         8898   8908   8918  8928  8937  8948  8962  8973  8985  9028  9038  9051  9061  9071  9081
         9091   9100   9111  9124  9136  9148  9158  9167
EXIT        1#  1297#
FEQUAL      1#  1297#
GEDF     2007#  4801   4820  4870  4906  5035  5150  5384  5589  5615  5641  5931  5970  6047  6159
         6198
GEHRD    2007#
GESF     2007#
GESFT    2007#
GETBYT      1#  1297#
```

```
GETPRI      1#  1297#
GETWOR      1#  1297#
GMANIA      1#  1297#
GMANID      1#  1297#
GMANIL      1#  1297#
GPHARD      1#  1297#  4617
GPRMA       1#  1297#  9190  9195
GPRMD       1#  1297#  9200  9206  9212  9218  9224
GPRML       1#  1297#
GTDF     2007#  2435   2479  2526  2597  2696  2969  3073  3106  3117  3151  3162  3196  3207  3241
         3252   3286   3297  3329  3340  3376  3387  3400  3411  3443  3455  3504  3751  3764  3790
         3800   3813   3823  3836  3846  3859  3869  3882  3892
GTHRD    2007#
GTSF     2007#
GTSFT    2007#
HEADER      1#  1297#  1323
INLOOP      1#  1297#
IOSETU      1#  1297#
IOSTAR      1#  1297#
KT11        1#  1297#
LASTAD      1#  1297#  9270
MANUAL      1#  1297#
MEMORY      1#  1297#
MSG      4755#  4761   4831#  4837  4985#  4991  5050#  5056  5098#  5104  5170#  5176  5222#  5228  5274#
         5280   5328#  5334   5403#  5409   5454#  5460   5496#  5502   5538#  5544   5651#  5657   5736#  5742
         5841#  5847   6071#  6077   6256#  6262   6356#  6362   6488#  6494   6702#  6708   6799#  6805   6896#
         6902   6993#  6999   7158#  7164   8047#  8053   8199#  8205   8373#  8379   8550#  8556   8764#  8770
         9005#  9011
MSBYTE      1#  1297#  1324#  1330  1331  1332
MSCHEC      1#  1297#
MSCNTO      1#  1297#  9191#  9196#  9201#  9207#  9213#  9219#  9225#
MSCOUN      1#  1297#  4105#  4120#  4139#  4148#  4168#  4177#  4186#  4201#  4209#  4217#  4226#  4246#  4254#
         4262#  4271#  4291#  4299#  4307#  4326#  4335#  4372#  4380#  4390#  4397#  4407#  4421#  4429#  4439#
         4446#  4456#  4463#  4473#  4480#  4498#  4506#  4516#  4523#  4923#  4929#  4935#  4941#  4951#  4959#
         4971#
MSDATA      1#  1297#  1324#  1333  1335  1337  1339  1341  1343  1345  1347  1349  1351  1353  1355
         1357   1359   1361   1363#  1365  1367  1370  1373  1375  1377  1379  1381  1383  1385  1387
         1389   1391   1393   1395  1397  1399  1401  1403  1405  1407  2379#  2391#
MSDECR      1#  1297#  1481#  1495#  4131#  4158#  4192#  4236#  4281#  4316#  4345#  4548#  4651#  4706#  4724#
         4739#  4752#  4828#  4881#  4913#  4917#  4965#  4980#  5048#  5096#  5168#  5214#  5220#  5266#  5272#
         5320#  5326#  5395#  5401#  5446#  5452#  5494#  5536#  5649#  5734#  5838#  6069#  6254#  6354#  6472#
         6564#  6624#  6684#  6687#  6797#  6894#  6991#  7156#  7327#  7468#  7609#  7750#  7891#  8032#  8035#
         8197#  8356#  8456#  8494#  8545#  8548#  8745#  8989#  9171#  9232#  9254#  9268#
MSDEFA      1#  1297#  9191#  9196#  9201#  9207#  9213#  9219#  9225#
MSENDE      1#  1297#  1481#  1495#  4131#  4158#  4192#  4236#  4281#  4316#  4345#  4651#  4706#  4724#  4739#
         4752#  4828#  4881#  4913#  4917#  4965#  4980#  5048#  5096#  5168#  5214#  5220#  5266#  5272#  5320#
         5326#  5395#  5401#  5446#  5452#  5494#  5536#  5649#  5734#  5838#  6069#  6254#  6354#  6472#  6564#
         6624#  6684#  6687#  6797#  6894#  6991#  7156#  7327#  7468#  7609#  7750#  7891#  8032#  8035#  8197#
         8356#  8456#  8494#  8545#  8548#  8745#  8989#  9171#  9232#  9254#  9268#
MSERRI      1#  1297#  4803#  4822#  4872#  4908#  5037#  5152#  5386#  5591#  5617#  5643#  5933#  5972#  6049#
         6161#  6200#
MSESCA      1#  1297#  4777#  4778  4786#  4787  4808#  4809  4896#  4897  5042#  5043  5091#  5092  5157#
         5158   5191#  5192   5210#  5211   5243#  5244   5262#  5263   5297#  5298   5316#  5317   5357#  5358
         5391#  5392   5423#  5424   5442#  5443   5475#  5476   5517#  5518   5596#  5597   5622#  5623   5690#
         5691   5711#  5712   5722#  5723   5730#  5731   5777#  5778   5787#  5788   5797#  5798   5807#  5808
         5816#  5817   5827#  5828   5835#  5836   5875#  5876   5884#  5885   5894#  5895   5912#  5913   5923#
```

```
                5924    5938#   5939    5950#   5951    5962#   5963    5977#   5978    5988#   5989    5998#   5999    6014#   6015
                6024#   6025    6039#   6040    6054#   6055    6066#   6067    6103#   6104    6112#   6113    6122#   6123    6140#
                6141    6151#   6152    6166#   6167    6178#   6179    6190#   6191    6205#   6206    6220#   6221    6230#   6231
                6243#   6244    6251#   6252    6285#   6286    6303#   6304    6313#   6314    6329#   6330    6340#   6341    6351#
                6352    6387#   6388    6409#   6410    6432#   6433    6449#   6450    6460#   6461    6469#   6470    6519#   6520
                6539#   6540    6557#   6558    6579#   6580    6599#   6600    6617#   6618    6639#   6640    6659#   6660    6677#
                6678    6733#   6734    6742#   6743    6752#   6753    6770#   6771    6784#   6785    6794#   6795    6830#   6831
                6839#   6840    6849#   6850    6867#   6868    6881#   6882    6891#   6892    6927#   6928    6936#   6937    6946#
                6947    6964#   6965    6978#   6979    6988#   6989    7021#   7022    7039#   7040    7049#   7050    7059#   7060
                7069#   7070    7078#   7079    7089#   7090    7099#   7100    7110#   7111    7120#   7121    7131#   7132    7142#
                7143    7153#   7154    7201#   7202    7217#   7218    7227#   7228    7240#   7241    7253#   7254    7273#   7274
                7284#   7285    7299#   7300    7313#   7314    7324#   7325    7342#   7343    7358#   7359    7368#   7369    7381#
                7382    7394#   7395    7414#   7415    7425#   7426    7440#   7441    7454#   7455    7465#   7466    7483#   7484
                7499#   7500    7509#   7510    7522#   7523    7535#   7536    7555#   7556    7566#   7567    7581#   7582    7595#
                7596    7606#   7607    7624#   7625    7640#   7641    7650#   7651    7663#   7664    7676#   7677    7696#   7697
                7707#   7708    7722#   7723    7736#   7737    7747#   7748    7765#   7766    7781#   7782    7791#   7792    7804#
                7805    7817#   7818    7837#   7838    7848#   7849    7863#   7864    7877#   7878    7888#   7889    7906#   7907
                7922#   7923    7932#   7933    7945#   7946    7958#   7959    7978#   7979    7989#   7990    8004#   8005    8018#
                8019    8029#   8030    8100#   8101    8119#   8120    8128#   8129    8142#   8143    8151#   8152    8170#   8171
                8184#   8185    8194#   8195    8230#   8231    8252#   8253    8264#   8265    8278#   8279    8289#   8290    8303#
                8304    8322#   8323    8333#   8334    8344#   8345    8353#   8354    8413#   8414    8432#   8433    8444#   8445
                8453#   3454    8471#   8472    8491#   8492    8509#   8510    8542#   8543    8580#   8581    8602#   8603    8612#
                8613    8622#   8623    8632#   8633    8641#   8642    8654#   8655    8664#   8665    8674#   8675    8684#   8685
                8693#   8694    8704#   8705    8718#   8719    8729#   8730    8741#   8742    8795#   8796    8817#   8818    8827#
                8828    8837#   8838    8847#   8848    8856#   8857    8869#   8870    8879#   8880    8889#   8890    8899#   8900
                8909#   8910    8919#   8920    8929#   8930    8938#   8939    8949#   8950    8963#   8964    8974#   8975    8986#
                8987    9029#   9030    9039#   9040    9052#   9053    9062#   9063    9072#   9073    9082#   9083    9092#   9093
                9101#   9102    9112#   9113    9125#   9126    9137#   9138    9149#   9150    9159#   9160    9168#   9169
MSESCS            1#    1297#   4777#   4786#   4808#   4896#   5042#   5091#   5157#   5191#   5210#   5243#   5262#   5297#   5316#
                5357#   5391#   5423#   5442#   5475#   5517#   5596#   5622#   5690#   5711#   5722#   5730#   5777#   5787#   5797#
                5804#   5816#   5827#   5835#   5875#   5934#   5894#   5912#   5923#   5938#   5950#   5962#   5977#   5988#   5998#
                6014#   6024#   6039#   6054#   6066#   6103#   6112#   6122#   6140#   6151#   6166#   6178#   6190#   6205#   6220#
                6230#   6243#   6251#   6285#   6303#   6313#   6329#   6340#   6351#   6387#   6409#   6432#   6449#   6460#   6469#
                6519#   6539#   6557#   6579#   6599#   6617#   6639#   6659#   6677#   6733#   6742#   6752#   6770#   6784#   6794#
                6830#   6839#   6849#   6867#   6881#   6891#   6927#   6936#   6946#   6964#   6978#   6988#   7021#   7039#   7049#
                7059#   7069#   7078#   7089#   7099#   7110#   7120#   7131#   7142#   7153#   7201#   7217#   7227#   7240#   7253#
                7273#   7284#   7299#   7313#   7324#   7342#   7358#   7368#   7381#   7394#   7414#   7425#   7440#   7454#   7465#
                7483#   7499#   7509#   7522#   7535#   7555#   7566#   7581#   7595#   7606#   7624#   7640#   7650#   7663#   7676#
                7696#   7707#   7722#   7736#   7747#   7765#   7781#   7791#   7804#   7817#   7837#   7848#   7863#   7877#   7888#
                7906#   7922#   7932#   7945#   7958#   7978#   7989#   8004#   8018#   8029#   8100#   8119#   8128#   8142#   8151#
                8170#   8184#   8194#   8230#   8252#   8264#   8278#   8289#   8303#   8322#   8333#   8344#   8353#   8413#   8432#
                8444#   8453#   8471#   8491#   8509#   8542#   8580#   8602#   8612#   8622#   8632#   8641#   8654#   8664#   8674#
                8684#   8693#   8704#   8718#   8729#   8741#   8795#   8817#   8827#   8837#   8847#   8856#   8869#   8879#   8889#
                8899#   8909#   8919#   8929#   8938#   8949#   8963#   8974#   8986#   9029#   9039#   9052#   9062#   9072#   9082#
                9092#   9101#   9112#   9125#   9137#   9149#   9159#   9168#
MSEXCP            1#    1297#   9191#   9196#   9201#   9207#   9213#   9219#   9225#
MSEXIT            1#    1297#
MSEXSE            1#    1297#
MSEXTJ            1#    1297#
MSGEN             1#    1297#   1300#   1324#   1333#   1335#   1337#   1339#   1341#   1343#   1345#   1347#   1349#   1351#   1353#
                1355#   1357#   1359#   1361#   1363#   1365#   1367#   1370#   1373#   1375#   1377#   1379#   1381#   1383#   1385#
                1387#   1389#   1391#   1393#   1395#   1397#   1399#   1401#   1403#   1405#   1407#   1423#   1466#   1467#   1481#
                1491#   1492#   1495#   2019#   2379#   2391#   4098#   4131#   4137#   4158#   4164#   4192#   4199#   4236#   4244#
                4281#   4289#   4316#   4324#   4345#   4543#   4556#   4651#   4670#   4706#   4720#   4724#   4734#   4739#   4750#
                4752#   4770#   4828#   4851#   4855#   4881#   4885#   4913#   4917#   4921#   4965#   4969#   4980#   5008#   5048#
                5073#   5096#   5119#   5168#   5185#   5199#   5214#   5220#   5237#   5251#   5266#   5272#   5291#   5305#   5320#
```

```
                4262#  4271#  4291#  4299#  4307#  4326#  4335#  4372#  4380#  4390#  4397#  4407#  4421#  4429#  4439#
                4446#  4456#  4463#  4473#  4480#  4498#  4506#  4516#  4523#  4673#  4923#  4929#  4935#  4941#  4951#
                4959#  4971#
MSPUT1             1#  1297#  4105#  4106   4107   4120#  4121   4122   4123   4139#  4140   4141   4142   4148#  4149
                4150   4151   4152   4168#  4169   4170   4171   4177#  4178   4179   4180   4181   4186#  4187   4201#
                4202   4203   4204   4209#  4210   4217#  4218   4219   4220   4226#  4227   4228   4229   4230   4246#
                4247   4248   4249   4254#  4255   4262#  4263   4264   4265   4271#  4272   4273   4274   4275   4291#
                4292   4293   4294   4299#  4300   4307#  4308   4309   4310   4326#  4327   4328   4329   4335#  4336
                4337   4338   4339   4372#  4373   4374   4375   4380#  4381   4382   4383   4384   4385   4390#  4391
                4392   4397#  4398   4399   4400   4401   4402   4407#  4408   4421#  4422   4423   4424   4429#  4430
                4431   4432   4433   4434   4439#  4440   4441   4446#  4447   4448   4449   4450   4451   4456#  4457
                4458   4463#  4464   4465   4466   4467   4468   4473#  4474   4475   4480#  4481   4482   4483   4484
                4485   4498#  4499   4500   4501   4506#  4507   4508   4509   4510   4511   4516#  4517   4518   4523#
                4524   4525   4526   4527   4528   4673#  4674   4675   4676   4923#  4924   4929#  4930   4935#  4936
                4941#  4942   4943   4944   4951#  4952   4953   4954   4959#  4960   4971#  4972   4973   4974
MSRADI             1#  1297#  9191#  9196#  9201#  9207#  9213#  9219#  9225#
MSRBRO             1#  1297#
MSRNRO             1#  1297#  4618#  4620
MSSETS             1#  1297#  1300#  1465#  1490#  4098#  4137#  4164#  4199#  4244#  4289#  4324#  4543#  4556#  4670#
                4720#  4734#  4750#  4771#  4852#  4856#  4886#  4921#  4969#  5009#  5074#  5120#  5186#  5200#  5238#
                5252#  5292#  5306#  5352#  5366#  5418#  5432#  5470#  5512#  5565#  5680#  5767#  5866#  6094#  6275#
                6378#  6505#  6511#  6571#  6631#  6724#  6821#  6918#  7012#  7187#  7193#  7334#  7475#  7616#  7757#
                7898#  8062#  8221#  8400#  8403#  8461#  8499#  8570#  8786#  9020#  9187#  9250#
MSSTAR             1#  1297#
MSSVC              1#  1297#  4105#  4109   4120#  4125   4131#  4132   4139#  4144   4148#  4154   4158#  4159   4168#
                4173   4177#  4183   4186#  4189   4192#  4193   4201#  4206   4209#  4212   4217#  4222   4226#  4232
                4236#  4237   4246#  4251   4254#  4257   4262#  4267   4271#  4277   4281#  4282   4291#  4296   4299#
                4302   4307#  4312   4316#  4317   4326#  4331   4335#  4341   4345#  4346   4372#  4377   4380#  4387
                4390#  4394   4397#  4404   4407#  4410   4421#  4426   4429#  4436   4439#  4443   4446#  4453   4456#
                4460   4463#  4470   4473#  4477   4480#  4487   4498#  4503   4506#  4513   4516#  4520   4523#  4530
                4576#  4577   4583#  4584   4590#  4591   4597#  4598   4618#  4619   4648#  4651#  4652   4673#  4677
                4695#  4696   4700#  4701   4706#  4707   4724#  4725   4737#  4739#  4740   4752#  4753   4775#  4777#
                4784#  4786#  4803   4808#  4822   4828#  4829   4855#  4856   4872   4877#  4381#  4882   4885#  4886
                4894#  4896#  4908   4913#  4914   4917#  4918   4923#  4926   4929#  4932   4935#  4938   4941#  4946
                4951#  4956   4959#  4962   4965#  4966   4971#  4976   4980#  4981   5037   5042#  5048#  5049   5089#
                5091#  5096#  5097   5152   5157#  5168#  5169   5189#  5191#  5199#  5200   5208#  5210#  5214#  5215
                5220#  5221   5241#  5243#  5251#  5252   5260#  5262#  5266#  5267   5272#  5273   5295#  5297#  5305#
                5306   5314#  5316#  5320#  5321   5326#  5327   5355#  5357#  5365#  5366   5386   5391#  5395#  5396
                5401#  5402   5421#  5423#  5431#  5432   5440#  5442#  5446#  5447   5452#  5453   5473#  5475#  5494#
                5495   5515#  5517#  5536#  5537   5591   5596#  5617   5622#  5643   5649#  5650   5688#  5690#  5709#
                5711#  5720#  5722#  5728#  5730#  5734#  5735   5775#  5777#  5785#  5787#  5795#  5797#  5805#  5807#
                5814#  5816#  5825#  5827#  5833#  5835#  5838#  5839   5873#  5875#  5882#  5884#  5892#  5894#  5910#
                5912#  5921#  5923#  5933   5938#  5948#  5950#  5960#  5962#  5972   5977#  5986#  5988#  5996#  5998#
                6012#  6014#  6022#  6024#  6037#  6039#  6049   6054#  6064#  6066#  6069#  6070   6101#  6103#  6110#
                6112#  6120#  6122#  6138#  6140#  6149#  6151#  6161   6166#  6176#  6178#  6188#  6190#  6200   6205#
                6218#  6220#  6228#  6230#  6241#  6243#  6249#  6251#  6254#  6255   6283#  6285#  6301#  6303#  6311#
                6313#  6327#  6329#  6338#  6340#  6349#  6351#  6354#  6355   6385#  6387#  6407#  6409#  6430#  6432#
                6447#  6449#  6458#  6460#  6467#  6469#  6472#  6473   6510#  6511   6517#  6519#  6537#  6539#  6555#
                6557#  6564#  6565   6570#  6571   6577#  6579#  6597#  6599#  6615#  6617#  6624#  6625   6630#  6631
                6637#  6639#  6657#  6659#  6675#  6677#  6684#  6685   6687#  6688   6731#  6733#  6740#  6742#  6750#
                6752#  6768#  6770#  6782#  6784#  6792#  6794#  6797#  6798   6828#  6830#  6837#  6839#  6847#  6849#
                6865#  6867#  6879#  6881#  6889#  6891#  6894#  6895   6925#  6927#  6934#  6936#  6944#  6946#  6962#
                6964#  6976#  6978#  6986#  6988#  6991#  6992   7019#  7021#  7037#  7039#  7047#  7049#  7057#  7059#
                7067#  7069#  7076#  7078#  7087#  7089#  7097#  7099#  7108#  7110#  7118#  7120#  7129#  7131#  7140#
                7142#  7151#  7153#  7156#  7157   7192#  7193   7199#  7201#  7215#  7217#  7225#  7227#  7238#  7240#
                7251#  7253#  7271#  7273#  7282#  7284#  7297#  7299#  7311#  7313#  7322#  7324#  7327#  7328   7333#
```

7334 7340# 7342# 7356# 7358# 7366# 7368# 7379# 7381# 7392# 7394# 7412# 7414# 7423# 7425#
7438# 7440# 7452# 7454# 7463# 7465# 7468# 7469 7474# 7475 7481# 7483# 7497# 7499# 7507#
7509# 7520# 7522# 7533# 7535# 7553# 7555# 7564# 7566# 7579# 7581# 7593# 7595# 7604# 7606#
7609# 7610 7615# 7616 7622# 7624# 7638# 7640# 7648# 7650# 7661# 7663# 7674# 7676# 7694#
7696# 7705# 7707# 7720# 7722# 7734# 7736# 7745# 7747# 7750# 7751 7756# 7757 7763# 7765#
7779# 7781# 7789# 7791# 7802# 7804# 7815# 7817# 7835# 7837# 7846# 7848# 7861# 7863# 7875#
7877# 7886# 7888# 7891# 7892 7897# 7898 7904# 7906# 7920# 7922# 7930# 7932# 7943# 7945#
7956# 7958# 7976# 7978# 7987# 7989# 8002# 8004# 8016# 8018# 8027# 8029# 8032# 8033 8035#
8036 8098# 8100# 8117# 8119# 8126# 8128# 8140# 8142# 8149# 8151# 8168# 8170# 8182# 8184#
8192# 8194# 8197# 8198 8228# 8230# 8250# 8252# 8262# 8264# 8276# 8278# 8287# 8289# 8301#
8303# 8320# 8322# 8331# 8333# 8342# 8344# 8351# 8353# 8356# 8357 8402# 8403 8411# 8413#
8430# 8432# 8442# 8444# 8451# 8453# 8456# 8457 8460# 8461 8469# 8471# 8489# 8491# 8494#
8495 8498# 8499 8507# 8509# 8540# 8542# 8545# 8546 8548# 8549 8578# 8580# 8600# 8602#
8610# 8612# 8620# 8622# 8630# 8632# 8639# 8641# 8652# 8654# 8662# 8664# 8672# 8674# 8682#
8684# 8691# 8693# 8702# 8704# 8716# 8718# 8727# 8729# 8739# 8741# 8745# 8746 8793# 8795#
8815# 8817# 8825# 8827# 8835# 8837# 8845# 8847# 8854# 8856# 8867# 8869# 8877# 8879# 8887#
8889# 8897# 8899# 8907# 8909# 8917# 8919# 8927# 8929# 8936# 8938# 8947# 8949# 8961# 8963#
8972# 8974# 8984# 8986# 8989# 8990 9027# 9029# 9037# 9039# 9050# 9052# 9060# 9062# 9070#
9072# 9080# 9082# 9090# 9092# 9099# 9101# 9110# 9112# 9123# 9125# 9135# 9137# 9147# 9149#
9157# 9159# 9166# 9168# 9171# 9172

MSTLAB 1# 1297# 4109# 4125# 4132# 4144# 4154# 4159# 4173# 4183# 4189# 4193# 4206# 4212# 4222#
4232# 4237# 4251# 4257# 4267# 4277# 4282# 4296# 4302# 4312# 4317# 4331# 4341# 4346# 4377#
4387# 4394# 4404# 4410# 4426# 4436# 4443# 4453# 4460# 4470# 4477# 4487# 4503# 4513# 4520#
4530# 4577# 4584# 4591# 4598# 4619# 4648# 4652# 4677# 4696# 4701# 4707# 4725# 4737# 4740#
4753# 4775# 4777# 4784# 4786# 4803# 4808# 4822# 4829# 4856# 4872# 4877# 4882# 4886# 4894#
4896# 4908# 4914# 4918# 4926# 4932# 4938# 4946# 4956# 4962# 4966# 4976# 4981# 5037# 5042#
5049# 5089# 5091# 5097# 5152# 5157# 5169# 5189# 5191# 5200# 5208# 5210# 5215# 5221# 5241#
5243# 5252# 5260# 5262# 5267# 5273# 5295# 5297# 5306# 5314# 5316# 5321# 5327# 5355# 5357#
5366# 5386# 5391# 5396# 5402# 5421# 5423# 5432# 5440# 5442# 5447# 5453# 5473# 5475# 5495#
5515# 5517# 5537# 5591# 5596# 5617# 5622# 5643# 5650# 5688# 5690# 5709# 5711# 5720# 5722#
5728# 5730# 5735# 5775# 5777# 5785# 5787# 5795# 5797# 5805# 5807# 5814# 5816# 5825# 5827#
5833# 5835# 5839# 5873# 5875# 5882# 5884# 5892# 5894# 5910# 5912# 5921# 5923# 5933# 5938#
5948# 5950# 5960# 5962# 5972# 5977# 5986# 5988# 5996# 5998# 6012# 6014# 6022# 6024# 6037#
6039# 6049# 6054# 6064# 6066# 6070# 6101# 6103# 6110# 6112# 6120# 6122# 6138# 6140# 6149#
6151# 6161# 6166# 6176# 6178# 6188# 6190# 6200# 6205# 6218# 6220# 6228# 6230# 6241# 6243#
6249# 6251# 6255# 6283# 6285# 6301# 6303# 6311# 6313# 6327# 6329# 6338# 6340# 6349# 6351#
6355# 6385# 6387# 6407# 6409# 6430# 6432# 6447# 6449# 6458# 6460# 6467# 6469# 6473# 6511#
6517# 6519# 6537# 6539# 6555# 6557# 6565# 6571# 6577# 6579# 6597# 6599# 6615# 6617# 6625#
6631# 6637# 6639# 6657# 6659# 6675# 6677# 6685# 6688# 6731# 6733# 6740# 6742# 6750# 6752#
6768# 6770# 6782# 6784# 6792# 6794# 6798# 6828# 6830# 6837# 6839# 6847# 6849# 6865# 6867#
6879# 6881# 6889# 6891# 6895# 6925# 6927# 6934# 6936# 6944# 6946# 6962# 6964# 6976# 6978#
6986# 6988# 6992# 7019# 7021# 7037# 7039# 7047# 7049# 7057# 7059# 7067# 7069# 7076# 7078#
7087# 7089# 7097# 7099# 7108# 7110# 7118# 7120# 7129# 7131# 7140# 7142# 7151# 7153# 7157#
7193# 7199# 7201# 7215# 7217# 7225# 7227# 7238# 7240# 7251# 7253# 7271# 7273# 7282# 7284#
7297# 7299# 7311# 7313# 7322# 7324# 7328# 7334# 7340# 7342# 7356# 7358# 7366# 7368# 7379#
7381# 7392# 7394# 7412# 7414# 7423# 7425# 7438# 7440# 7452# 7454# 7463# 7465# 7469# 7475#
7481# 7483# 7497# 7499# 7507# 7509# 7520# 7522# 7533# 7535# 7553# 7555# 7564# 7566# 7579#
7581# 7593# 7595# 7604# 7606# 7610# 7616# 7622# 7624# 7638# 7640# 7648# 7650# 7661# 7663#
7674# 7676# 7694# 7696# 7705# 7707# 7720# 7722# 7734# 7736# 7745# 7747# 7751# 7757# 7763#
7765# 7779# 7781# 7789# 7791# 7802# 7804# 7815# 7817# 7835# 7837# 7846# 7848# 7861# 7863#
7875# 7877# 7886# 7888# 7892# 7898# 7904# 7906# 7920# 7922# 7930# 7932# 7943# 7945# 7956#
7958# 7976# 7978# 7987# 7989# 8002# 8004# 8016# 8018# 8027# 8029# 8033# 8036# 8098# 8100#
8117# 8119# 8126# 8128# 8140# 8142# 8149# 8151# 8168# 8170# 8182# 8184# 8192# 8194# 8198#
8228# 8230# 8250# 8252# 8262# 8264# 8276# 8278# 8287# 8289# 8301# 8303# 8320# 8322# 8331#
8333# 8342# 8344# 8351# 8353# 8357# 8403# 8411# 8413# 8430# 8432# 8442# 8444# 8451# 8453#
8457# 8461# 8469# 8471# 8489# 8491# 8495# 8499# 8507# 8509# 8540# 8542# 8546# 8549# 8578#

CVDMCA.P11 10-DEC-80 09:14 CROSS REFERENCE TABLE -- MACRO NAMES

```
        6163    6164    6200#   6201    6202    6203    9191#   9196#   9201#   9207#   9213#   9219#   9225#   9272    9273
MSXFER     1#   1297#
NEWTST  2007#   4755    4831    4985    5050    5098    5170    5222    5274    5328    5403    5454    5496    5538    5651
        5736    5841    6071    6256    6356    6488    6702    6799    6896    6993    7158    8047    8199    8373    8550
        8764    9005
NTST    2007#   4755    4831    4985    5050    5098    5170    5222    5274    5328    5403    5454    5496    5538    5651
        5736    5841    6071    6256    6356    6488    6702    6799    6896    6993    7158    8047    8199    8373    8550
        8764    9005
OPEN       1#   1297#
POINTE     1#   1297#   1320
PRINTB     1#   1297#   4138    4147    4167    4176    4185    4200    4208    4216    4225    4245    4253    4261    4270
        4290    4298    4306    4325    4334    4406    4922    4958    4970
PRINTF     1#   1297#
PRINTS     1#   1297#
PRINTX     1#   1297#   4104    4119    4371    4379    4389    4396    4420    4428    4438    4445    4455    4462    4472
        4479    4497    4505    4515    4522    4928    4934    4940    4950
READBU     1#   1297#
READEF     1#   1297#   4575    4582    4589    4596
RFLAGS     1#   1297#
SETDF   2007#   2437    2481    2528    2599    2698    2971    3075    3108    3119    3153    3164    3198    3209    3243
        3254    3288    3299    3331    3342    3378    3389    3402    3413    3445    3457    3506    3753    3766    3792
        3802    3815    3825    3838    3848    3861    3871    3884    3894
SETHRD  2007#
SETPRI     1#   1297#
SETSF   2007#
SETSFT  2007#
SETVEC     1#   1297#   4672
SLASH      1#   1297#   1414    1418
STARS      1#   1297#
SVC        1#   1295#   1296
T$GEN   2007#   2437    2481    2528    2599    2698    2971    3075    3108    3119    3153    3164    3198    3209    3243
        3254    3288    3299    3331    3342    3378    3389    3402    3413    3445    3457    3506    3753    3766    3792
        3802    3815    3825    3838    3848    3861    3871    3884    3894
XFER       1#   1297#
XFERF      1#   1297#
XFERT      1#   1297#
$GEDF   2007#   4802    4821    4871    4907    5036    5151    5385    5590    5616    5642    5932    5971    6048    6160
        6199
$GEHRD  2007#
$GESF   2007#
$GESFT  2007#
$GTDF   2007#   2436    2480    2527    2598    2697    2970    3074    3107    3118    3152    3163    3197    3208    3242
        3253    3287    3298    3330    3341    3377    3388    3401    3412    3444    3456    3505    3752    3765    3791
        3801    3814    3824    3837    3847    3860    3870    3883    3893
$GTHRD  2007#
$GTSF   2007#
$GTSFT  2007#
```

. ABS. 037426 000

ERRORS DETECTED: 0

CVDMCA.BIN,CVDMCA.SEQ/CRF/SOL=SVC34R.MAC,CVDMCA.P11
RUN-TIME: 40 52 6 SECONDS

CVDMCA.P11 10-DEC-80 09:14 CROSS REFERENCE TABLE -- MACRO NAMES

RUN-TIME RATIO: 204/99=2.0
CORE USED: 24K (47 PAGES)