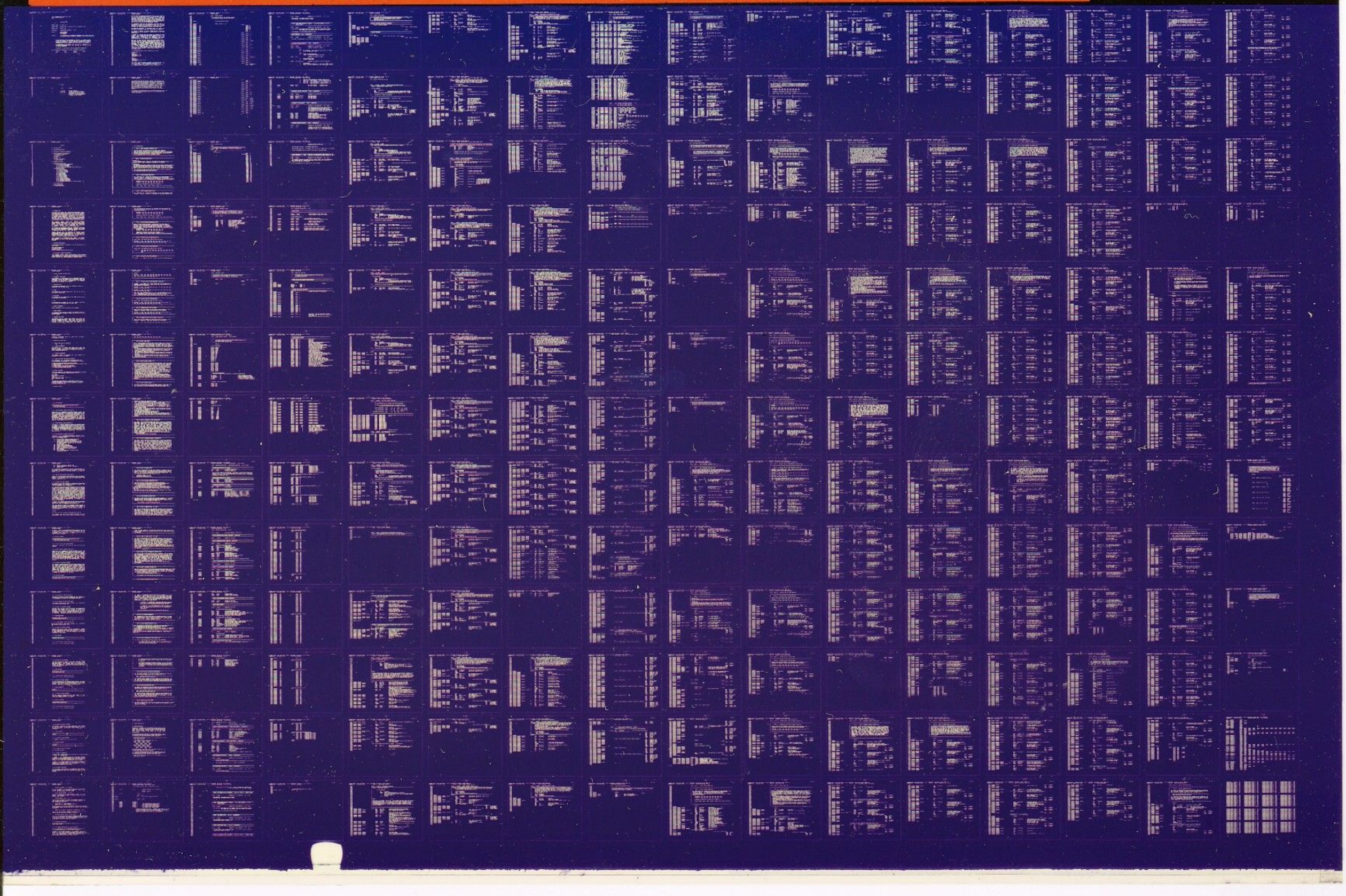
DMV11

DMV-11 LN UNT STC 1
CVDMCCO

AH-FZ68C-MC
FICHE 01 OF 02

FEB 1985
digital
Made In USA

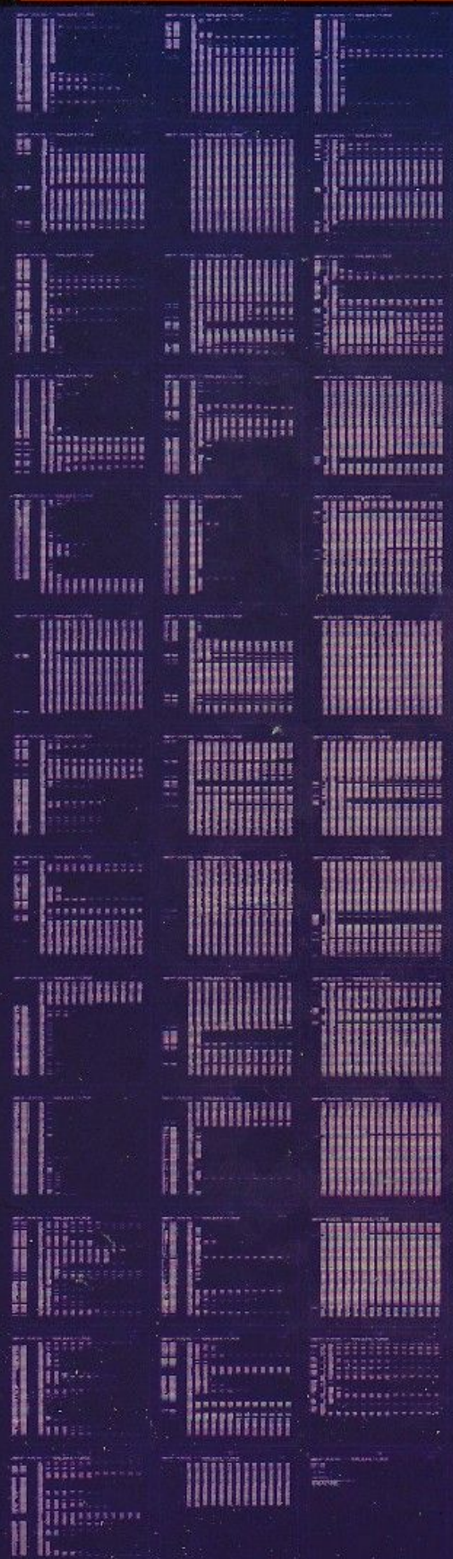


DMV11

DMV-11 LN UNT STC 1
CVDMCCO

AH-FZ68C-MC
FICHE 0Z OF 0Z

FEB 1985
digital
Made In USA



```
                        .TITLE CVDMCC0 DMV11 LINE UNIT DIAG1
                        .SBTTL PROGRAM DOCUMENT
                        .REM !



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

                        PRODUCT CODE:   AC-F267C-MC

                        PRODUCT NAME:   CVDMCC0 DMV-11 LINE UNIT STATIC DIAGNOSTIC PART 1

                        PRODUCT DATE:   AUGUST 1984

                        MAINTAINER:     DIAGNOSTICS MERRIMACK  CC:38P

                        AUTHORS:        CHRIS BRIENEN
                                        DAVE HOFFMAN
                                        RAY MARSHALL

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



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

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

                                COPYRIGHT (C) 1981,1984 BY DIGITAL EQUIPMENT CORPORATION

                        THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                                DIGITAL         PDP             UNIBUS          MASSBUS
                                DEC             DECUS           DECTAPE
```

```
                                    HISTORY
                                    -------

                      REV            DATE          REASON
                      ---            ----          ------

                       A           14-JAN-81       INITIAL RELEASE.
                       B           11-JUL-83       INSTALL OUTSTANDING PATCHES.
                       C           28-JUL-84       INCREASED MANY OF THE TIMING
                                                   PARAMETERS TO ALLOW PROGRAM TO
                                                   RUN ON A J-11 PROCESSOR(ORION).
                                                   (CHANGED BY NICK MCCAMY)
```

CONTENTS

1.0 INTRODUCTION

2.0 HARDWARE REQUIREMENTS

3.0 PRELIMINARY PROGRAM REQUIREMENTS

4.0 GENERAL PROGRAM CONSIDERATIONS
    4.1 DIAGNOSTIC SUPERVISOR
    4.2 EXECUTION TIME
    4.3 XXDP+
    4.4 ACT/SLIDE
    4.5 APT
    4.6 MEMORY MANAGEMENT
    4.7 ERROR LOGGING

5.0 PROGRAM LOAD MEDIA

6.0 OPERATING INSTRUCTIONS
    6.1 LOADING AND STARTING PROCEDURES
        6.1.1 LOADING PROCEDURES
        6.1.2 STARTING PROCEDURES
        6.1.3 ** STEPS FOR QUICK AND SIMPLE EXECUTION **
    6.2 INITIAL DIALOGUE
    6.3 PROGRAM OPTIONS
        6.3.1 START COMMAND
        6.3.2 RESTART COMMAND
        6.3.3 CONTINUE COMMAND
        6.3.4 PROCEED COMMAND
        6.3.5 ADD COMMAND
        6.3.6 DROP COMMAND
        6.3.7 PRINT COMMAND
        6.3.8 DISPLAY COMMAND
        6.3.9 FLAGS COMMAND
        6.3.10 ZFLAGS COMMAND
        6.3.11 CONTROL CHARACTERS
        6.3.12 HARDWARE PARAMETERS
        6.3.13 SOFTWARE PARAMETERS
        6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

7.0 TEST DESCRIPTIONS

8.0 ERROR INFORMATION
    8.1 ERROR REPORTING

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).

```
165
166                             4.0 GENERAL PROGRAM CONSIDERATIONS
167
168
169                             4.1 DIAGNOSTIC SUPERVISOR
170
171                             THIS  PROGRAM  IS  COMPATIBLE WITH THE STANDALONE DIAGNOSTIC
172                             SUPERVISOR, AND MUST BE LOADED TO BE  CO-RESIDENT  WITH  THE
173                             SUPERVISOR,  OR  BE  PREVIOUSLY COMBINED WITH THE SUPERVISOR
174                             AND LOADED AS A SINGLE FILE.  IN EITHER CASE,  THE  COMBINED
175                             PROGRAM WILL NOT EXCEED 16K OF MEMORY.
176
177
178                             4.2 EXECUTION TIME
179
180                             THE MAXIMUM TIME REQUIRED TO RUN THIS PROGRAM IS ABOUT 30
181                             SECONDS PER PASS FOR EACH UNIT.
182
183
184                             4.3 XXDP+
185
186                             THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND  MAY  BE  RUN  IN
187                             DUMP MODE OR CHAIN MODE.
188
189
190                             4.4 ACT/SLIDE
191
192                             THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN
193                             IN DUMP MODE OR CHAIN MODE.
194
195
196                             4.5 APT
197
198                             THIS PROGRAM MAY BE LOADED  BY  THE  APT  SYSTEM  (INCLUDING
199                             APT RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.
200
201
202                             4.6 MEMORY MANAGEMENT
203
204                             MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM.
205
206
207                             4.7 ERROR LOGGING
208
209                             AT THE END OF EACH PASS ON ALL UNITS, THE PROGRAM PRINTS OUT
210                             THE CUMULATIVE TOTAL NUMBER OF ERRORS SINCE THE LAST START OR
211                             RESTART COMMAND.
212
213
214                             5.0 PROGRAM LOAD MEDIA
215
216                             THIS PROGRAM  CAN  BE  LOADED  FROM  PAPER  TAPE  USING  THE
217                             ABSOLUTE  LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FRO..
218                             ANY MEDIA SUPPORTED BY XXDP+.   WHEN  USING  THE  PAPER  TAPE
219                             ABSOLUTE   LOADER,  THE  PROGRAM  SHOULD  BE  LOADED  FIRST,
220                             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-C-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>
****

```
445
446                             6.3.3.1 PASS SWITCH (/PASS:<PASS-CNT>)
447
448                             <PASS-CNT> IS SAME AS IN START COMMAND, BUT THE  DEFAULT  IS
449                             THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART.
450                             IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.
451
452
453                             6.3.3.2 FLAG SWITCH (/FLAGS:<FLAG-LIST>)
454
455                             <FLAG-LIST> IS SAME AS IN  START  COMMAND.  BUT  UNSPECIFIED
456                             FLAGS RETAIN THEIR CURRENT VALUE.
457
458
459                             6.3.3.3 EFFECT OF CONTINUE COMMAND
460
461                             CONTINUE  MUST  FOLLOW  A START OR RESTART, AND COMMAND MODE
462                             MUST HAVE  BEEN  ENTERED  DUE  TO  A  HALT  ON  ERROR  OR  A
463                             CONTROL/C.  THE  EFFECT  OF  THE  COMMAND  IS  TO  GO TO THE
464                             BEGINNING OF THE TEST THAT WAS BEING EXECUTED WHEN THE  HALT
465                             OR  CONTROL/C  TOOK PLACE.  SOFTWARE DIALOGUE MAY OPTIONALLY
466                             BE REEXECUTED.  HARDWARE PARAMETERS MAY NOT BE CHANGED.
467
468
469                             6.3.4 PROCEED COMMAND
470
471                             ************************************************************
472                             PRO(CEED)/FLAGS:<FLAG-LIST>
473                             ************************************************************
474
475
476                             6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
477
478                             <FLAG-LIST> IS AS IN  THE  START  COMMAND.  BUT  UNSPECIFIED
479                             FLAGS RETAIN THEIR CURRENT VALUE.
480
481
482                             6.3.4.2 EFFECT OF PROCEED COMMAND
483
484                             PROCEED  MUST FOLLOW A START, RESTART, OR CONTINUE.  COMMAND
485                             MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR.  THE EFFECT
486                             OF  THE  COMMAND  IS  TO  BEGIN  EXECUTION  AT  THE LOCATION
487                             FOLLOWING THE ERROR CALL.  NEITHER  HARDWARE  NOR  SOFTWARE
488                             PARAMETERS MAY BE ALTERED.
489
490
491                             6.3.5 ADD COMMAND
492
493                             ************************************************************
494                             ADD/UNITS:<UNIT-LIST>
495                             ************************************************************
496
497
498                             6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>
499
500                             <UNIT-LIST> IS AS IN THE RESTART COMMAND.
```

```
501
502
503                             6.3.5.2 EFFECT OF ADD COMMAND
504
505                             THE UNITS SPECIFIED ARE ADDED TO THE  TEST  SEQUENCE.   EACH
506                             UNIT  MUST  HAVE  A  P-TABLE  IN  MEMORY  DUE  TO AN EARLIER
507                             HARDWARE DIALOGUE.  THIS  COMMAND  MUST  BE  FOLLOWED  BY  A
508                             RESTART  OR  CONTINUE.   THE UNITS SWITCH MUST BE SPECIFIED.
509                             THE ADD COMMAND IS  MEANINGFUL  ONLY  FOR  UNITS  THAT  WERE
510                             PREVIOUSLY DROPPED.
511
512
513                             6.3.6 DROP COMMAND
514
515                             ************************************************************
516                             DRO(P)/UNITS:<UNIT-LIST>
517                             ************************************************************
518
519
520                             6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
521
522                             <UNIT-LIST> IS AS IN THE RESTART COMMAND.
523
524
525                             6.3.6.2 EFFECT OF DROP COMMAND
526
527                             THE UNITS SPECIFIED WILL BE DROPPED FROM TESTING.  THE UNITS
528                             WILL BE RESELECTED ONLY BY THE EXECUTION OF AN ADD OR  START
529                             COMMAND.   THE  UNITS  SWITCH MUST BE ENTERED.  THIS COMMAND
530                             MUST BE FOLLOWED BY A RESTART OR A CONTINUE COMMAND.
531
532
533                             6.3.7 PRINT COMMAND
534
535                             ************************************************************
536                             PRI(NT)
537                             ************************************************************
538
539
540                             6.3.7.1 EFFECT OF PRINT COMMAND
541
542                             THE TOTAL NUMBER OF ERRORS FOR  EACH  UNIT  SINCE  THE  LAST
543                             START  OR  RESTART  COMMAND  ARE  PRINTED.  THE ISR (INHIBIT
544                             STATISTICAL REPORTING) FLAG IS CLEARED.
545
546
547                             6.3.8 DISPLAY COMMAND
548
549                             ************************************************************
550                             DIS(PLAY)/UNITS:<UNIT-LIST>
551                             ************************************************************
552
553
554                             6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
555
556                             <UNIT-LIST> IS AS IN THE RESTART COMMAND.
```

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

```
613
614                                 1.   DEVICE CSR ADDRESS : (O) 160020?
615
616                                 THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE
617                                 ON  THE  LSI-BUS.   THE  ALLOWABLE  RANGE  IS  160020-177760
618                                 (OCTAL), AND THE DEFAULT VALUE IS 160020.
619
620                                 2.   DEVICE VECTOR ADDRESS : (O) 300 ?
621
622                                 THIS IS THE ADDRESS OF THE INPUT INTERRUPT VECTOR  FOR  THIS
623                                 DEVICE.   THE  ALLOWABLE  RANGE  IS 000-674 (OCTAL), AND THE
624                                 DEFAULT VALUE IS 300.
625
626                                 3.   DEVICE PRIORITY LEVEL : (O) 4 ?
627
628                                 THIS IS THE CPU PRIORITY AT WHICH THE INTERRUPT HANDLERS  OF
629                                 THIS  DEVICE  WILL BE EXECUTED.  THE ALLOWABLE RANGE IS 0-7,
630                                 AND THE DEFAULT VALUE IS 4.
631
632                                 4.   SWITCH PACK # 1 (BOOT ADDRESS): (O)  0 ?
633                                 5.   SWITCH PACK # 2 (DDCMP ADDRESS): (O)  0 ?
634
635                                 THESE REPRESENT THE TWO USER MODIFIABLE 8 POSITION DIP-SWITCHES.
636                                 THE ALLOWABLE RANGES FOR BOTH ARE 000-377 (OCTAL), AND THE DEFAULTS
637                                 ARE BOTH 0.
638
639                                 6.   BOARD TYPE (0=M8064, 1=M8053-V35, 2=M8053-EIA) : (O) 0 ?
640
641                                 THIS IS THE TYPE OF DMV-11 CURRENTLY INSTALLED. NOTE THAT THE
642                                 M8053 IS SWITCH SELECTABLE BETWEEN V.35 AND EIA.
643
644                                 7.   BAUD RATE (0=LOW (19.2K), 1=HIGH (56K)):
645
646                                 THIS  IS  THE SPEED AT WHICH THE DMV TRANSMITS AND RECEIVES DATA.
647                                 IN THE UNIT IS AN M8064, THE ANSWER IS IGNORED (DEFAULTS TO 56K).
648                                 IF THE UNIT IS AN M8053, THEN THE ANSWER SHOULD BE BASED ON THE
649                                 "SPEED SELECT SWITCH" LOCATED ON THE BOARD.
650
651
652                                 6.3.13 SOFTWARE PARAMETERS
653
654                                 NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 1  OF  THE
655                                 STATIC LOGIC TESTS.
656
657
658                                 6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE
659
660                                 THE  FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY
661                                 THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.
662
663                                 AS SOON AS THE QUESTION "# UNITS?"  IS  ANSWERED  (WITH  THE
664                                 NUMBER  N,  SAY)  SPACE IN CORE IS ALLOCATED FOR N P-TABLES.
665                                 ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE  IS  A
666                                 ONE-TO  ONE  CORRESPONDENCE  BETWEEN  THE HARDWARE PARAMETER
667                                 QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.
668
```

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.

```
725
726                     THE SECOND TIME THRU THE SERIES, TABLES 7 THRU THE END  ARE
727                     GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS
728                     PRINTED OUT FOR THE THE OPERATOR IN THE FORM  "UNIT  XX"  AT
729                     THE  BEGINNING  OF EACH SERIES).  QUESTION 1 IS RESPONDED TO
730                     BY A <CR>, SO SLOT ONE STAYS AT CONSTANT  75  IN  TABLES  7
731                     THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN.  SLOT TWO
732                     GETS THE VALUES 7,8,9,10,11 IN TABLES 7  THRU  11,  AND
733                     GETS AN 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN
734                     TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7
735                     THRU 15.
736
737                     THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT
738                     16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION
739                     (NAMELY QUESTION 2).
740
741
```

```
                        7.0 TEST DESCRIPTIONS

                        ;*****************************************************************************
                        ;*      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).
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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).
                        ;*
                        ;*  THIS TEST IS ONLY RUN ON THE FIRST PASS AFTER A "START" OR "RESTART".
                        ;*  ALL SUCCESSIVE PASSES WILL SKIP THIS 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
                        ;*
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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
```

```
798                             ;*  (BIT 0 @ A000) IN THE 6522 VIA. ALL REGISTERS ARE THEN CHECKED FOR
799                             ;*  THE PROPER CONTENTS. THE INITIALIZED STATE OF THE REGISTERS IS CHECKED
800                             ;*  AGAINST DATA PATTERN F.
801                             ;*
802                             ;*      PATTERN E: 377, 377, 377, 377, 377, 377, 377, 366.
803                             ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110.
804                             ;*
805                             ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS E & F:
806                             ;*
807                             ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
808                             ;*
809                             ;*********************************************************************************
810
811
812                             ;*********************************************************************************
813                             ;*      TEST 5  <USYRT REGISTER ADDRESSING TEST>
814                             ;*
815                             ;* FIRST, A MASTER CLEAR IS ISSUED, TO INITIALIZE THE USYRT REGS TO
816                             ;* PATTERN F. THEN, EACH REGISTER IS WRITTEN WITH A BYTE OF PATTERN J,
817                             ;* AND AFTER EACH IS WRITTEN, ALL ARE READ AND COMPARED TO THE CURRENT
818                             ;* EXPECTED VALUES. THIS IS PERFORMED FOR ALL REGISTERS -- INCLUDING THE
819                             ;* READ ONLY REGS -- IN ORDER TO MAKE SURE THAT EACH REGISTER ONLY RESPONDS
820                             ;* TO ITS OWN ADDRESS.
821                             ;*      PATTERN F: 000, 000, 000, 000, 000, 000, 000, 110
822                             ;*      PATTERN J: 000, 000, 001, 002, 004, 020, 040, 010
823                             ;*
824                             ;*      SEQUENCE OF REGISTERS AS USED WITH PATTERNS F & J:
825                             ;*      RDSRL, RDSRH, TDSRL, TDSRH, PCSARL, PCSARH, PCR, USYRT STATUS REG
826                             ;*********************************************************************************
827
828
829                             ;*********************************************************************************
830                             ;*      TEST 6  <R/W BIT TEST OF PCSAR HIGH BYTE>
831                             ;*
832                             ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN G.
833                             ;*
834                             ;*  PATTERN  G:  000,  001, 003, 004, 005, 007, 100, 101, 103, 104, 105,
835                             ;*  107, 000, 017, 027, 041, 200, 277, 103, 144, 115, 157, 000.
836                             ;*********************************************************************************
837
838
839                             ;*********************************************************************************
840                             ;*      TEST 7  <R/W BIT TEST OF S/AR REGISTER>
841                             ;*
842                             ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN H.
843                             ;*
844                             ;*  PATTERN  H:  125, 252, 000, 377, 000, 001, 002, 004, 010, 020, 040, 100,
845                             ;*               200, 000, 377, 376, 375, 373, 367, 357, 337, 277, 177, 377,
846                             ;*               000
847                             ;*********************************************************************************
848
849
850                             ;*********************************************************************************
851                             ;*      TEST 8  <R/W BIT TEST OF PCR REGISTER>
852                             ;*
853                             ;*  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, 367, 357, 030, 027, 377.
                        ;*
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      TEST 12 <PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE>
                        ;*
                        ;*  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
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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.
                        ;*****************************************************************************


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

```
1022
1023
1024                    ;*****************************************************************************
1025                    ;*      TEST 18 <BCP TX DISABLE TEST>
1026                    ;*
1027                    ;*  THE  USYRT IS INITIALIZED FOR BCP AND A MESSAGE IS STARTED. ONCE THE
1028                    ;*  SECOND  DATA  CHARACTER  IS LOADED INTO TXDB TXE IS DROPPED. TXSO IS
1029                    ;*  WATCHED TO ASSURE THAT THE CHARACTER BEING TRANSMITTED IS COMPLETED.
1030                    ;*  WHEN  IT  IS, THE USYRT SHOULD DROP TXA AND STOP TRANSMITTING -- THE
1031                    ;*  LAST CHARACTER LOADED INTO TXDB SHOULD BE LOST.
1032                    ;*
1033                    ;*      CHARACTERS LOADED: 125 252
1034                    ;*      CHARACTERS TRANSMITTED: 125
1035                    ;*****************************************************************************
1036
1037
1038                    ;*****************************************************************************
1039                    ;*      TEST 19 <FIFO STACKING CHARACTERS TEST>
1040                    ;*
1041                    ;*  THE USYRT IS SETUP FOR BCP MODE WITH NO ERROR DETECTION.
1042                    ;*  THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS
1043                    ;* TO FILL THE 8 CHARACTER RECEIVER FIFO WITH THE CHARACTERS:
1044                    ;*      1/2(SYNCH),000,377,125,252,347,030,303,1/2(074).
1045                    ;*  THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. NOTE
1046                    ;* THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE  CHARACTERS SINCE THEY
1047                    ;* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
1048                    ;* RDA = 0).
1049                    ;*  ALSO NOTE THAT DUE TO FIFO TIMING, TWO 'HALF CHARACTERS' ARE LOADED
1050                    ;* INTO THE FIFO (THE 1ST AND LAST CHARACTERS).
1051                    ;*
1052                    ;*****************************************************************************
1053
1054
1055                    ;*****************************************************************************
1056                    ;*      TEST 20 <BCP CHARACTER LENGTH TEST>
1057                    ;*
1058                    ;*  THE USYRT IS INITIALIZED FOR BCP WITH NO ERROR CHECKING. TXC IS MANUALLY
1059                    ;*  CONTROLLED UNTIL TWO SYNC CHARACTERS HAVE BEEN TRANSMITTED.  THEN  3
1060                    ;*  SUBTESTS FOLLOW, EACH ONE USING A DIFFERENT TRANSMIT CHARACTER LENGTH
1061                    ;*  STARTING AT FIVE (5) AND ENDING WITH SEVEN (7).
1062                    ;*
1063                    ;*       TEST PATTERN: 111 222 333 044 155 266 377
1064                    ;*****************************************************************************
1065
1066
1067                    ;*****************************************************************************
1068                    ;*      TEST 21 <BOP TX TABORT/(IDLE = 0) TEST>
1069                    ;*
1070                    ;*  THE USYRT IS INITIALIZED FOR "BIT-ORIENTED PROTOCOL" (BOP) WITH IDLE
1071                    ;*  SET TO ZERO. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
1072                    ;*  OBSERVING TXA -- IT SHOULD BE ASSERTED WITHIN TWO (2) CLOCK CYCLES.
1073                    ;*  NEXT, TXBE SHOULD GO HIGH; AT THIS TIME AN ALL ZEROS CHARACTER WILL BE
1074                    ;*  LOADED INTO TXDB DRIVING TXBE LOW. THE TRANSMITTER IS CLOCKED THROUGH
1075                    ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
1076                    ;*  ABORT TO BE TRANSMITTED. ALL CHARACTERS ARE CHECKED AT TXSO.
1077                    ;*  THIS TEST WILL GO NO FURTHER INTO THE TRANSMIT SEQUENCE SO THAT ONLY TWO
```

```
                        ;*  FLAGS, ONE ZERO CHARACTER, AND ONE ABORT CHARACTER IS SENT (INTO THE BIT
                        ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
                        ;*  THE SEQUENCE.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      TEST 22 <BOP TX TABORT/(IDLE = 1) TEST>
                        ;*
                        ;*  THE USYRT IS INITIALIZED FOR "BIT-ORIENTED PROTOCOL" (BOP) WITH IDLE
                        ;*  SET TO ONE. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
                        ;*  OBSERVING TXA -- 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. THE TRANSMITTER IS CLOCKED THROUGH
                        ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
                        ;*  ABORT TO BE TRANSMITTED. 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 FLAG CHARACTER IS SENT (INTO THE BIT
                        ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
                        ;*  THE SEQUENCE.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      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.
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      1EST 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
                        ;*****************************************************************************


                        ;*****************************************************************************
                        ;*      TEST 25 <BOP RX CHARACTER LENGTH TEST>
```

```
1134                    ;*
1135                    ;*  THE  USYRT  IS INITIALIZED FOR BOP WITH CRC-CCITT PRESET TO 1'S. TXC
1136                    ;*  IS   MANUALLY   CONTROLLED  UNTIL  TWO  FLAG  CHARACTERS  HAVE  BEEN
1137                    ;*  TRANSMITTED.  THEN  6  SUBTESTS  FOLLOW,  EACH ONE USING A DIFFERENT
1138                    ;*  TRANSMIT  CHARACTER LENGTH STARTING AT TWO (2) AND ENDING WITH SEVEN
1139                    ;*  (7).  IN  EACH  SUBTEST,  TWO  8  BIT CHARACTERS WILL BE TRANSMITTED
1140                    ;*  BEFORE  TXCL  IS  CHANGED TO THE CHARACTER LENGTH BEING TESTED. THIS
1141                    ;*  CORRESPONDS TO NORMAL USAGE WHERE EITHER:
1142                    ;*
1143                    ;*       1 -- A  MESSAGE  OF  CHARACTERS  WHICH ARE LESS THEN 8 BITS IS
1144                    ;*            SENT  AS  A  STREAM OF 8 BIT CHARACTERS AND THE REMAINING
1145                    ;*            BITS ARE SENT AS A CHARACTER OF LESS THEN 8 BITS OR
1146                    ;*
1147                    ;*       2 -- A  HEADER  OF  TWO 8 BIT CHARACTERS IS SENT FOLLOWED BY A
1148                    ;*            DATA  STREAM  OF DATA CHARACTERS WHICH MAY BE LESS THEN 8
1149                    ;*            BITS  IN  LENGTH  (I.E.  2,  3,  4,  5,  6,  OR 7 BIT
1150                    ;*            CHARACTERS).
1151                    ;*
1152                    ;*       THE TEST PATTERN IS: 123 321 111 222 333 044 155 266 377
1153                    ;*****************************************************************************
1154
1155
1156                    ;*****************************************************************************
1157                    ;*      TEST 26 <TX "SPACING SEQUENCE">
1158                    ;*
1159                    ;*  THE  TRANSMITTER IS INITIALIZED AND THE "SPACING SEQUENCE" IS FORCED
1160                    ;*  BY  ASSERTING  BOTH  TSOM  &  TEOM AT THE SAME TIME -- CHECK THE BIT
1161                    ;*  STREAM FOR ACCURACY (SPACES) AND COMPLETNESS (16 OF THEM). WHEN TXBE
1162                    ;*  GOES  HIGH  (=  1)  A  SMALL  MESSAGE  IS SENT.
1163                    ;*****************************************************************************
1164
1165
1166                    ;*****************************************************************************
1167                    ;*      TEST 27 <FIFO OVERRUN INTEGRITY TEST>
1168                    ;*
1169                    ;* THIS TEST BEGINS BY SYNCHRONIZING THE RECEIVER AND THEN PROCEEDS TO FILL
1170                    ;* THE 8 CHARACTER RECEIVER FIFO UNTIL RXOR WITH THE CHARACTERS:
1171                    ;*      (SYNCH),000,377,125,252,347,030,303,074,125.
1172                    ;* THESE CHARACTERS ARE THEN READ OFF OF THE FIFO AND CHECKED. OF IMPORTANCE
1173                    ;* IS THE INTEGRITY OF THE LAST OVERRUN-CAUSING FIFO CHARACTER (IT SHOULD
1174                    ;* REMAIN INTACT).
1175                    ;* NOTE THAT NO CLOCKS ARE PROVIDED WHEN RECEIVING THE  CHARACTERS SINCE THEY
1176                    ;* ARE SUPPLIED BY THE FIFO SUPPORT LOGIC IN GROUPS OF 4 TICKS (WHEN
1177                    ;* RDA = 0).
1178                    ;*
1179                    ;*****************************************************************************
1180
1181
1182                    ;*****************************************************************************
1183                    ;*      TEST 28 <BCP RX OVERRUN SET AND CLEAR TEST>
1184                    ;*
1185                    ;*  THE USYRT IS INITIALIZED AND THREE SUBTESTS ARE PERFORMED.
1186                    ;*
1187                    ;*     1 -- AN OVERRUN CONDITION IS FORCED, RECEIVER STATUS REGISTER IS
1188                    ;*          READ TWICE: ONCE TO VERIFY ROR BIT = 1, AND AGAIN TO VERIFY
1189                    ;*          THAT THE FIRST READ CLEARED ROR .
```

```
1190                                    ;*
1191                                    ;*      2 -- AN OVERRUN CONDITION IS FORCED (BY THE SAME TECHNIQUE USED IN
1192                                    ;*          (2), THE USYRT IS RESET AND THE PROPER STATE OF ALL REGISTERS
1193                                    ;*          IS VERIFIED.
1194                                    ;*
1195                                    ;*      3 -- AN OVERRUN CONDITION IS FORCED (AS ABOVE). RXE IS THEN DROPPED
1196                                    ;*           AND A DELAY IS PROVIDED TO ALLOW TIME FOR THE FIFO TO FLUSH
1197                                    ;*           (CAUSED BY RDA GOING LOW). RXE IS THEN RE-INITIALIZED AND ROR
1198                                    ;*           IS CHECKED = 0. THE RECEIVER IS THEN RE-SYNCHED AND THE TEST IS
1199                                    ;*           TERMINATED.
1200                                    ;*
1201                                    ;*****************************************************************************
1202
1203
1204                                    ;*****************************************************************************
1205                                    ;*      TEST 29 <BCP RX SYNC CHARACTER RECOGNITION>
1206                                    ;*
1207                                    ;*  THE  FOLLOWING  MESSAGE  IS INITIATED WITHOUT ASSERTING RXE AND ONCE
1208                                    ;*  THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
1209                                    ;*
1210                                    ;*     SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
1211                                    ;*     SYNC SYNC DATA DATA DATA SYNC SYNC
1212                                    ;*
1213                                    ;*  THE  RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
1214                                    ;*  THE  NEXT  TWO SYNC CHARACTERS FOR SYNCRONIZATION, THEN PASS THE REST
1215                                    ;*  OF THE MESSAGE (7 SYNC AND 6 DATA CHARACTERS) THROUGH RXDB REGISTER.
1216                                    ;*****************************************************************************
1217
1218
1219                                    ;*****************************************************************************
1220                                    ;*      TEST 30 <BCP RX STRIP-SYNC TEST>
1221                                    ;*
1222                                    ;*  THE  USYRT  IS INITIALIZED WITH THE STRIP-SYNC CONTROL BIT ASSERTED.
1223                                    ;*  THE  FOLLOWING  MESSAGE  IS THEN INITIATED WITHOUT ASSERTING RXE AND
1224                                    ;*  ONCE THE DATA IS BEING TRANSMITTED, RXE IS ASSERTED (IE: *):
1225                                    ;*
1226                                    ;*     SYNC * SYNC DATA DATA DATA SYNC SYNC SYNC SYNC SYNC DATA DATA DATA
1227                                    ;*     SYNC SYNC DATA DATA DATA SYNC SYNC
1228                                    ;*
1229                                    ;*  THE  RECEIVER SHOULD IGNORE THE FIRST STRING OF DATA CHARACTERS, USE
1230                                    ;*  THE  NEXT  TWO  SYNC  CHARACTERS FOR SYNCRONIZATION, IGNORE THE NEXT
1231                                    ;*  THREE  SYNC CHARACTERS, AND PASS THE REST OF THE MESSAGE (4 SYNC AND
1232                                    ;*  6 DATA CHARACTERS) THROUGH RXDB REGISTER.
1233                                    ;*****************************************************************************
1234
1235
1236                                    ;*****************************************************************************
1237                                    ;*      TEST 31 <BCP RX LOST RXE TEST>
1238                                    ;*
1239                                    ;*  THE USYRT IS INITIALIZED (CRC16,STRIPS,BCP MODE) AND A MESSAGE IS STARTED.
1240                                    ;*  WHILE IN THE MIDDLE  OF  TEXT, RXE IS DROPPED AND THE REACTION OF THE
1241                                    ;*  RECEIVER IS MONITORED.
1242                                    ;*
1243                                    ;*****************************************************************************
1244
1245
```

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

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

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

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

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

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

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

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

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

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

                        002000          .=2000

                                        .MCALL  SVC
002000                                  SVC     ; INITIALIZE SUPERVISOR MACROS

002000                                  BGNMOD  LU1MOD

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

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

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

002000                                  POINTER BGNAU,BGNDU,ERRTBL

                                        HEADER  CVDMC,C,0,10.,0

002000                                                          L$NAME::
002000  103                                                             .ASCII /C/
002001  126                                                             .ASCII /V/
002002  104                                                             .ASCII /D/
002003  115                                                             .ASCII /M/
002004  103                                                             .ASCII /C/
002005  000                                                             .BYTE   0
002006  000                                                             .BYTE   0
002007  000                                                             .BYTE   0
002010                                                          L$REV::
002010  103                                                             .ASCII  /C/
002011                                                          L$DEPO::
002011  060                                                             .ASCII  /0/
002012                                                          L$UNIT::
002012  000000                                                          .WORD   0
002014                                                          L$TIML::
002014  000012                                                          .WORD   10.
002016                                                          L$MPCP::
002016  036646                                                          .WORD   L$HARD
002020                                                          L$SPCP::
002020  000000                                                          .WORD   0
002022                                                          L$MPTP::
002022  002224                                                          .WORD   L$MM
002024                                                          L$SPTP::
002024  000000                                                          .WORD   0
002026                                                          L$LADP::
002026  037434                                                          .WORD   L$LAST
002030                                                          L$STA::
002030  000000                                                          .WORD   0
002032                                                          L$CO::
002032  000000                                                          .WORD   0
002034                                                          L$DTYP::
002034  000000                                                          .WORD   0
002036                                                          L$APT::
002036  000000                                                          .WORD   0
002040                                                          L$DTP::
002040  002124                                                          .WORD   L$DISPATCH
002042                                                          L$PRIO::
002042  000000                                                          .WORD   0
002044                                                          L$ENVI::
002044  000000                                                          .WORD   0
002046                                                          L$EXP1::
002046  000000                                                          .WORD   0
002050                                                          L$MREV::
002050     003                                                          .BYTE   C$REVISION
002051     003                                                          .BYTE   C$EDIT
```

```
1373  002052
1374  002052  000000
1375  002054  000000
1376  002056
1377  002056  000000
1378  002060
1379  002060  003306
1380  002062
1381  002062  000000
1382  002064
1383  002064  000000
1384  002066
1385  002066  000000
1386  002070
1387  002070  021554
1388  002072
1389  002072  021550
1390  002074
1391  002074  000000
1392  002076
1393  002076  003326
1394  002100
1395  002100  104035
1396  002102
1397  002102  002246
1398  002104
1399  002104  021100
1400  002106
1401  002106  021546
1402  002110
1403  002110  021422
1404  002112
1405  002112  021072
1406  002114
1407  002114  000000
1408  002116
1409  002116  000000
1410  002120
1411  002120  000000
1412
1413
1414                        .EVEN
```

```
L$EF::
        .WORD   0
        .WORD   0
L$SPC::
        .WORD   0
L$DEVP::
        .WORD   L$DVTYP
L$REPP::
        .WORD   0
L$EXP4::
        .WORD   0
L$EXP5::
        .WORD   0
L$AUT::
        .WORD   L$AU
L$DUT::
        .WORD   L$DU
L$LUN::
        .WORD   0
L$DESP::
        .WORD   L$DESC
L$LOAD::
        EMT     E$LOAD
L$ETP::
        .WORD   L$ERRTBL
L$ICP::
        .WORD   L$INIT
L$CCP::
        .WORD   L$CLEAN
L$ACP::
        .WORD   L$AUTO
L$PRT::
        .WORD   L$PROT
L$TEST::
        .WORD   0
L$DLY::
        .WORD   0
L$HTMF  :
        .WORD   0
```

```
1415                                    .SBTTL  DISPATCH TABLE
1416
1417  002122                            SLASH
1418                                    ;;//////////////////////////////////////////////////////////////////////
1419                                    ;/ THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1420                                    ;/ IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1421  002122                            SLASH
1422                                    ;;//////////////////////////////////////////////////////////////////////
1423
1424  002122                                    DISPATCH 31.
1425  002122  000037                                                       .WORD   31
1426  002124                                                          L$DISPATCH::
1427  002124  021556                                                       .WORD   T1
1428  002126  021736                                                       .WORD   T2
1429  002130  022736                                                       .WORD   T3
1430  002132  023122                                                       .WORD   T4
1431  002134  023206                                                       .WORD   T5
1432  002136  023432                                                       .WORD   T6
1433  002140  023524                                                       .WORD   T7
1434  002142  023616                                                       .WORD   T8
1435  002144  023710                                                       .WORD   T9
1436  002146  024040                                                       .WORD   T10
1437  002150  024132                                                       .WORD   T11
1438  002152  024206                                                       .WORD   T12
1439  002154  024262                                                       .WORD   T13
1440  002156  024604                                                       .WORD   T14
1441  002160  024736                                                       .WORD   T15
1442  002162  025120                                                       .WORD   T16
1443  002164  025626                                                       .WORD   T17
1444  002166  026226                                                       .WORD   T18
1445  002170  026434                                                       .WORD   T19
1446  002172  026712                                                       .WORD   T20
1447  002174  027422                                                       .WORD   T21
1448  002176  027616                                                       .WORD   T22
1449  002200  030012                                                       .WORD   T23
1450  002202  030206                                                       .WORD   T24
1451  002204  030564                                                       .WORD   T25
1452  002206  033506                                                       .WORD   T26
1453  002210  034064                                                       .WORD   T27
1454  002212  034450                                                       .WORD   T28
1455  002214  035052                                                       .WORD   T29
1456  002216  035530                                                       .WORD   T30
1457  002220  036264                                                       .WORD   T31
1458
```

```
                                .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          ;M3254&M3255 USED
002242  000001                          .WORD        1          ;BAUD RATE = 56 K
                                                                ;       0 = 19.2 K
                                                                ;       1 = 56 K

002244                                  ENDHW
002244                                                          L10000:
```

```
                                SOFTWARE P-TABLE

1485                            .SBTTL  SOFTWARE P-TABLE
1486
1487                            ;////////////////////////////////////////////////////////////////////////
1488                            ;/ THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
1489                            ;/ PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
1490                            ;////////////////////////////////////////////////////////////////////////
1491
1492  002244                            BGNSW   SW.TBL
1493  002244  000000                                                           .WORD   L10001-L$SW/2
1494  002246                                                                L$SW::
1495  002246                                                                SFPTBL::
1496
1497  002246                            ENDSW
1498  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
```

```
1555  000140  PRIO3== 140
1556  000100  PRIO2== 100
1557  000040  PRIO1== 40
1558  000000  PRIO0== 0
1559          ;
1560          ;OPERATOR FLAG BITS
1561          ;
1562  000004  EVL==       4
1563  000010  LOT==      10
1564  000020  ADR==      20
1565  000040  IDU==      40
1566  000100  ISR==     100
1567  000200  UAM==     200
1568  000400  BOE==     400
1569  001000  PNT==    1000
1570  002000  PRI==    2000
1571  004000  IXE==    4000
1572  010000  IBE==   10000
1573  020000  IER==   20000
1574  040000  LOE==   40000
1575  100000  HOE==  100000
```

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

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

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

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

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

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

```
                        .SBTTL  REGISTER DEFINITIONS -- USYRT


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

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

            120400      RDSRL   = 120400        ;ADDRESS OF THIS REG

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

            120401      RDSRH   = 120401        ;ADDRESS OF THIS REG

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

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

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

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

            120402      TDSRL   = 120402        ;ADDRESS OF THIS REG

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

            120403      TDSRH   = 120403        ;ADDRESS OF THIS REG

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

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

```
1669            002000                  TXAB    = BIT10         ;TRANSMIT ABORT
1670            001000                  TXEOM   = BIT9          ;TRANSMIT END-OF-MESSAGE
1671            000400                  TXSOM   = BIT8          ;TRANSMIT START-OF-MESSAGE
1672
1673                                    ;;**************************************************************************
1674                                    ;* USYRT "SYNC/SECONDARY ADDRESS" REGISTER
1675                                    ;;**************************************************************************
1676
1677            120404                  PCSARL  = 120404        ;ADDRESS OF THIS REG
1678            000226                  SYNCH   = 226           ;STANDARD SYNCH CHARACTER
1679
1680                                    ;;**************************************************************************
1681                                    ;* USYRT "MODE CONTROL"
1682                                    ;;**************************************************************************
1683
1684            120405                  PCSARH  = 120405        ;ADDRESS OF THIS REG
1685
1686                                            ;BIT DEFINITIONS ON BYTE BASIS:
1687
1688            000200                  APA     = BIT7          ;"ALL PARTIES ADDRESS" ENABLE
1689            000100                  PROTO   = BIT6          ;SPECIFIES BOP/CCP PROTOCOL -- 0 = BOP
1690            000040                  STRIP   = BIT5          ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1691            000020                  SECAD   = BIT4          ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1692            000010                  IDLE    = BIT3          ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1693            000007                  XYZ     = BIT2!BIT1!BIT0 ;CRC/PARITY SELECTION CONTROL
1694
1695                                            ;BIT DEFINITIONS ON WORD BASIS:
1696
1697            100000                  APAD    = BIT15         ;"ALL PARTIES ADDRESS" ENABLE
1698            040000                  DDCMP   = BIT14         ;CODE FOR DDCMP MODE
1699            020000                  STRIPS  = BIT13         ;STRIP EXTRA SYNC'S IN CCP MODE, SEE GA CHARS IN BOP
1700            010000                  SECADR  = BIT12         ;SECONDARY ADDRESS MODE -- BOP MODE ONLY
1701            004000                  IDLES   = BIT11         ;IDLE & SYNC CHAR. TRANSMISSION CONTROL
1702            001400                  CRC16   = BIT9!BIT8     ;CODE FOR CRC-16 SELECTION
1703            003400                  NOCHK   = BIT10!BIT9!BIT8 ;CODE FOR NO ERROR CHECKING
1704            002400                  EVRC    = BIT10!BIT8    ;CODE FOR VRC EVEN CHECK
1705            002000                  OVRC    = BIT10         ;CODE FOR VRC ODD CHECK
1706
1707                                    ;;**************************************************************************
1708                                    ;* USYRT "DATA LENGTH SELECT" REGISTER
1709                                    ;;**************************************************************************
1710
1711            120407                  PCR     = 120407        ;ADDRESS OF THIS REG
1712
1713                                            ;BIT DEFINITIONS:
1714
1715            000340                  TXDL    = BIT7!BIT6!BIT5 ;TRANSMIT DATA LENGTH SELECTION
1716            000020                  EXADD   = BIT4          ;EXTENDED ADDRESS FIELD -- NOT USED OR TESTED
1717            000010                  EXCON   = BIT3          ;EXTENDED CONTROL FIELD -- NOT USED OR TESTED
1718            000007                  RXDL    = BIT2!BIT1!BIT0 ;RECEIVER DATA LENGTH SELECTION
1719
1720                                    ;;**************************************************************************
1721                                    ;* USYRT STATUS REGISTER (ADDR. A400)
1722                                    ;;**************************************************************************
1723            122000                  USTATR = 122000         ;USYRT STATUS REGISTER ADDRESS = A400 (HEX)
1724
```

```
                                        ;BIT DEFINITIONS:

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

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

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

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

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

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

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

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

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


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

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

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

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


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

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

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

```
1791
1792                                    ;       INITIALIZED TO 001 (HEX = 01) -- PORT A IS READ ONLY (EXCEPT FOR
1793                                    ;                                       BIT0 WHICH ENABLES USYRT INTERNAL LOOPBACK).
1794
1795
1796
1797                                    ;;*****************************************************************************
1798                                    ;* TIMER 1 LOW ORDER (LATCH & COUNTER) -- "T1L-L" & "T1C-L" -- WRITE & READ
1799                                    ;;*****************************************************************************
1800
1801    120004                          VIAT1A  = 120004                ;ADDRESS OF THIS REGISTER -- HEX = A0X4
1802
1803                                    ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1804                                    ;  WHEN READING, LOW ORDER COUNTER IS READ.
1805
1806
1807
1808                                    ;;*****************************************************************************
1809                                    ;* TIMER 1 HIGH ORDER COUNTER & TRIGGER -- "T1L-H AND TRIGGER" & "T1C-H"
1810                                    ;*      -- WRITE & READ
1811                                    ;;*****************************************************************************
1812
1813    120005                          VIAT1B  = 120005                ;ADDRESS OF THIS REGISTER -- HEX = A0X5
1814
1815                                    ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1816                                    ;                 ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1817
1818                                    ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1819
1820
1821
1822                                    ;;*****************************************************************************
1823                                    ;* TIMER 1 LOW ORDER LATCH -- "T1L-L" -- READ/WRITE
1824                                    ;;*****************************************************************************
1825
1826    120006                          VIAT1C  = 120006                ;ADDRESS OF THIS REGISTER -- HEX = A0X6
1827
1828                                    ;  THE LOW ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1829                                    ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1830
1831
1832
1833                                    ;;*****************************************************************************
1834                                    ;* TIMER 1 HIGH ORDER LATCH -- "T1L-H" -- READ/WRITE
1835                                    ;;*****************************************************************************
1836
1837    120007                          VIAT1D  = 120007                ;ADDRESS OF THIS REGISTER -- HEX = A0X7
1838
1839                                    ;  THE HIGH ORDER LATCH IS READ OR LOADED.  THIS LATCH IS USED TO LOAD THE
1840                                    ;  COUNTER WHEN T1MODE (IN VIAACR) = 3
1841
1842
1843
1844                                    ;;*****************************************************************************
1845                                    ;* TIMER 2 LOW ORDER (LATCH & COUNTER) -- "T2L-L" & "T2C-L" -- WRITE & READ
1846                                    ;;*****************************************************************************
```

```
1847
1848          120010                  VIAT2A  = 120010                ;ADDRESS OF THIS REGISTER -- HEX = A0X8
1849
1850                                  ;  WHEN WRITING, LOW ORDER LATCH IS LOADED.
1851                                  ;  WHEN READING, LOW ORDER COUNTER IS READ.
1852
1853
1854
1855                                  ;;*************************************************************************
1856                                  ;* TIMER 2 HIGH ORDER COUNTER & TRIGGER -- "T2L-H AND TRIGGER" & "T2C-H"
1857                                  ;*      -- WRITE & READ
1858                                  ;;*************************************************************************
1859
1860          120011                  VIAT2B  = 120011                ;ADDRESS OF THIS REGISTER -- HEX = A0X9
1861
1862                                  ;  WHEN WRITING; HIGH ORDER LATCH IS LOADED, BOTH LOW & HIGH ORDER LATCHES
1863                                  ;                 ARE LOADED INTO THE COUNTER, AND THE COUNTER IS STARTED.
1864
1865                                  ;  WHEN READING, THE HIGH ORDER COUNTER IS READ.
1866
1867                                  ;;*************************************************************************
1868                                  ;* SHIFT REGISTER -- "SR" -- READ/WRITE
1869                                  ;;*************************************************************************
1870
1871          120012                  VIASR   = 120012                ;ADDRESS OF THIS REGISTER -- HEX = A0XA
1872
1873                                  ;  SHIFTING IS CONTROLLED BY THE SETTING OF VIASRC (ACR2 ---> ACR4) IN VIAACR
1874
1875
1876
1877                                  ;;*************************************************************************
1878                                  ;* AUXILIARY CONTROL REGISTER -- "ACR" -- READ/WRITE
1879                                  ;;*************************************************************************
1880
1881          120013                  VIAACR  = 120013                ;ADDRESS OF THIS REGISTER -- HEX = A0XB
1882
1883          000300                  T1MODE  = BIT7!BIT6             ;CONTROL THE MODE OF TIMER # 1
1884
1885                                                  ;BIT 7:
1886                                                  ; 0     PB7 DISABLED -- ONLY T1TO IN VIAIFR REFLECTS TIMEOUT
1887                                                  ; 1     PB7 & T1TO REFLECT TIMEOUT
1888
1889                                                  ;BIT 6:
1890                                                  ; 0     TIMER 1 IN ONE-SHOT MODE
1891                                                  ; 1     TIMER 1 IN CONTINUOUS SQUARE WAVE MODE
1892
1893          000040                  T2MODE  = BIT5                  ;CONTROLS THE MODE OF TIMER # 1
1894
1895                                                  ; 0     PULSE COUNTING MODE
1896                                                  ; 1     INTERVAL TIMER MODE
1897
1898          000034                  SRMODE  = BIT4!BIT3!BIT2        ;CONTROLS THE MODE OF THE SHIFT REGISTER
1899
1900                                                  ; 0     SR DISABLED
1901                                                  ; 1     SHIFT IN UNDER CONTROL OF T2, SHFT PULSES GEN'D ON CB1
1902                                                  ; 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
```

```
1959                                                    ;BITS IN THE INT. ENAB. REG..  IF = 1, THE
1960                                                    ;RESPECTIVE BITS WILL BE SET.
1961
1962                            ;  WHEN WRITING THIS REG., THE COMMENT ABOVE HOLDS.
1963
1964                            ;  WHEN READING THIS REG., THE CURRENT STATE OF THE INT. ENABLE REG. IS RETURNED.
1965
1966                            ;  THE BIT ASSIGNMENTS ARE THE SAME AS FOR VIAIFR AS DEFINED ABOVE.
1967
1968
1969
1970                            ;;*************************************************************************
1971                            ;* OUTPUT REGISTER A -- "ORA" -- READ ONLY (OR READ/WRITE UNDER CONTROL OF "DDPA")
1972                            ;;*************************************************************************
1973
1974            120017          VIAORA  = 120017                ;ADDRESS OF THIS REGISTER -- HEX = A0XF
1975
1976                            ;  THIS ADDRESS ACCESSES THE SAME DATA AS "VIAMS" EXCEPT THAT NO "HANDSHAKING"
1977                            ;  WILL TAKE PLACE (I.E. THERE IS NO CHANGE IN IRQ OR CA2 AS A RESULT OF
1978                            ;  READING ORA THROUGH THIS ADDRESS)
1979
1980                            ;THE BIT ASSIGNMENTS ARE THE SAME AS FOR "VIAMS" ABOVE.
1981
1982
1983
```

```
                        .SBTTL  REGISTER DEFINITIONS -- MISC

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

        121000          SWPBOT  = 121000                ;"BOOT ADDRESS" SWITCH PACK [A200]
        121400          SWPDOCMP = 121400               ;"DOCMP ADDRESS" SWITCH PACK [A300]

                        ;MISCELLANEOUS EQUATES

        100000          TCCHEK  = BIT15         ;FLAG TO REQUEST M3254.5 CHECK
        001000          RAMADR  = 001000        ;STARTING ADRS OF RAM PAGE 2 (ADRS C200 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 DURI'G "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)
```

```
                                .SBTTL  GLOBAL DATA SECTION

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

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

002246                                  ERRTBL
002246                                                                  L$ERRTBL::
002246  000000                  ERRTYP::        .WORD   0
002250  000000                  ERRNBR::        .WORD   0
002252  000000                  ERRMSG::        .WORD   0
002254  000000                  ERRBLK::        .WORD   0

                                ;;*********************************************************************
                                ;* STORAGE FOR DEVICE REGISTERS
                                ;;*********************************************************************
002256                          WSR0:                           ;STORAGE FOR DEVICE CSR REGISTERS
002256  000000                  BSR0:   .WORD   0
002260                          WSR2:
002260  000000                  BSR1:   .WORD   0
002262                          WSR4:
002262  000000                  BSR2:   .WORD   0
002264                          WSR6:
002264  000000                  BSR3:   .WORD   0
002266                          WSR10:
002266  000000                  BSR4:   .WORD   0
002270                          WSR12:
002270  000000                  BSR5:   .WORD   0
002272                          WSR14:
002272  000000                  BSR6:   .WORD   0
002274                          WSR16:
002274  000000                  BSR7:   .WORD   0
002276  000000                  BSR10:  .WORD   0
002300  000000                  BSR11:  .WORD   0
002302  000000                  BSR12:  .WORD   0
002304  000000                  BSR13:  .WORD   0
002306  000000                  BSR14:  .WORD   0
002310  000000                  BSR15:  .WORD   0
002312  000000                  BSR16:  .WORD   0
002314  000000                  BSR17:  .WORD   0

002316  000010                  UREGS:  .BLKW   8.              ;THE FIRST 7 ARE FOR THE USYRT'S ACTUAL
                                                                ;REGISTERS.  THE LAST ONE IS FOR THE STATUS
                                                                ;REG. (USTATR).
002336  000020                  VREGS:  .BLKW   16.             ;STORAGE FOR VIA REGISTERS FOR PRINTOUT
```

```
2060                                ;*************************************************************************
2061                                ;* MISCELLANEOUS STORAGE
2062                                ;*************************************************************************
2063  002376  000000  TDATA:  .WORD  0          ;TEST DATA
2064  002400  000000  GDATA:  .WORD  0          ;GOOD DATA
2065  002402  000000  BDATA:  .WORD  0          ;BAD DATA
2066  002404  000000  XDATA:  .WORD  0          ;EXCLUSIVE-OR BETWEEN GOOD AND BAD DATA
2067  002406  000000  SCRACH: .WORD  0          ;GEN'L PURPOSE SCRATCH WORD
2068  002410  000000  LOGDEV: .WORD  0          ;LOGICAL DEVICE NUMBER
2069  002412  000000  REGNUM: .WORD  0          ;CONTAINS A DEVICE REGISTER NUMBER
2070  002414  000000  PSTACK: .WORD  0          ;CONTAINS BASE LEVEL PROGRAM STACK POINTER
2071  002416  000000  PRIOR:  .WORD  0          ;CPU PRIORITY FOR PRINTOUT
2072  002420  000000  SUBRPC: .WORD  0          ;PC OF SUBR CALL FOR ERROR REPORTS
2073  002422  000000  INTFLG: .WORD  0          ;INTERRUPT RECEIVED FLAGS
2074                                ;  BIT 0 FOR TX, BIT 1 FOR RCV
2075  002424  000000  ERRFLG: .WORD  0          ;SUBROUTINE ERROR FLAG
2076  002426  000000  TIMFLG: .WORD  0          ;EVENT TIME-OUT FLAG
2077  002430  000000  RETADR: .WORD  0          ;SUBR ERROR RETURN ADDRESS
2078  002432  000000  REDBYT: .WORD  0          ;LO BYTE CONTAINS BYTE READ FROM LU REG
2079  002434  000000  WRIBYT: .WORD  0          ;LO BYTE CONTAINS BYTE TO LOAD INTO LU REG
2080  002436  000000  LOADAT: .WORD  0          ;CONTAINS TEST DATA LOADED INTO REG
2081  002440  000000  GOODAT: .WORD  0          ;STORAGE FOR EXPECTED DATA
2082  002442  000000  BADDAT: .WORD  0          ;STORAGE FOR ACTUAL DATA
2083  002444  000000  FRSTIM: .WORD  0          ;FLAG=0 IF PROGRAM JUST LOADED
2084  002446  000000  SAVE4:  .WORD  0          ;SAVE LOC 4 HERE (ERROR TRAP VECTOR)
2085  002450  000000  SAVE6:  .WORD  0          ;SAVE LOC 6 HERE (ERROR TRAP VECTOR)
2086  002452  000000  ERROR1: .WORD  0          ;SUBR ERR. BIT FLAGS (DEF'D IN GLOBAL EQUATES)
2087  002454  000000  CHPTYP: .WORD  0          ;USYRT CHIP TYPE, =0 FOR SMC, ELSE =1
2088  002456  000000  SAVLEN: .WORD  0          ;SAVED TX AND RCV CHAR LENGTHS
2089  002460  000000  DEVMAP: .WORD  0          ;BIT MAP OF ACTIVE DEVICES
2090  002462  000000  DEVPTR: .WORD  0          ;DEVICE MAP BIT POINTER
2091  002464  000000  UNIT:   .WORD  0          ;CONTAINS UNIT NO. (1 TO N)
2092  002466  000000  STARES: .WORD  0          ;FLAG TO SHOW NO. OF PASSES SINCE STA OR RES
2093  002470  000000  TSTNUM: .WORD  0          ;NO. OF CURRENT TEST (FOR SOME TESTS)
2094
```

```
2095                                    ;;********** CURRENT DEVICE PARAMETERS **************
2096  002472                            BSEL0:
2097  002472                            SEL0:
2098  002472  160020                    MPCSR:  .WORD   160020          ;POINTER TO DMV11 CSR'S
2099  002474  160021                    BSEL1:  .WORD   160021          ;POINTER TO BSEL1
2100  002476                            BSEL2:
2101  002476  160022                    SEL2:   .WORD   160022          ;POINTER TO SEL2
2102  002500  160023                    BSEL3:  .WORD   160023          ;POINTER TO BSEL3
2103  002502                            BSEL4:
2104  002502  160024                    SEL4:   .WORD   160024          ;POINTER TO SEL4
2105  002504  160025                    BSEL5:  .WORD   160025          ;POINTER TO BSEL5
2106  002506                            BSEL6:
2107  002506  160026                    SEL6:   .WORD   160026          ;POINTER TO SEL6
2108  002510  160027                    BSEL7:  .WORD   160027          ;POINTER TO BSEL7
2109  002512                            BSEL10:
2110  002512  160030                    SEL10:  .WORD   160030          ;POINTER TO SEL10
2111  002514  160031                    BSEL11: .WORD   160031          ;POINTER TO BSEL11
2112  002516                            BSEL12:
2113  002516  160032                    SEL12:  .WORD   160032          ;POINTER TO SEL12
2114  002520  160033                    BSEL13: .WORD   160033          ;POINTER TO BSEL13
2115  002522                            BSEL14:
2116  002522  160034                    SEL14:  .WORD   160034          ;POINTER TO SEL14
2117  002524  160035                    BSEL15: .WORD   160035          ;POINTER TO BSEL15
2118  002526                            BSEL16:
2119  002526  160036                    SEL16:  .WORD   160036          ;POINTER TO SEL16
2120  002530  160037                    BSEL17: .WORD   160037          ;POINTER TO BSEL17
2121
2122  002532  000300                    MPIVEC: .WORD      300          ;DMV11 INPUT INTERRUPT VECTOR
2123  002534  000304                    MPOVEC: .WORD      304          ;DMV11 OUTPUT INTERRUPT VECTOR
2124  002536  000240                    MPRIOR: .WORD      240          ;DMV11 DEVICE PRIORITY
2125  002540  000000                    LUSWI1: .WORD        0          ;LINE UNIT SWITCH PACK #1
2126  002542  000000                    LUSWI2: .WORD        0          ;LINE UNIT SWITCH PACK #2
2127  002544  000000                    BRDTYP: .WORD        0          ;0-M8064, 1-M8053/V.35,2-M8053/EIA
2128  002546  000000                    TSTCON: .WORD        0          ;TEST CONNECTOR INDICATOR
2129  002550  000001                    BDRATE: .WORD        1          ;BAUD RATE = 56 K
2130                                                                    ;        0 = 19.2 K
2131                                                                    ;        1 = 56 K
```

```
2132
2133  002552  120400                  ;TABLE OF USYRT REGISTER ADDRESSES
2134  002554  120401                  USYREG: .WORD   120400          ;ADDRESS OF RDSRL
2135  002556  120402                          .WORD   120401          ;ADDRESS OF RDSRH
2136  002560  120403                          .WORD   120402          ;ADDRESS OF TDSRL
2137  002562  120404                          .WORD   120403          ;ADDRESS OF TDSRH
2138  002564  120405                          .WORD   120404          ;ADDRESS OF PCSARL
2139  002566  120407                          .WORD   120405          ;ADDRESS OF PCSARH
2140  002570  122000                          .WORD   120407          ;ADDRESS OF PCR
2141                                          .WORD   122000          ;ADDRESS OF USYRT STATUS REG
2142                                  ;***** STORAGE FOR DATA READ IN ADDRESS TESTS *****
2143  002572  000010                  REDDAT: .BLKB   8.
2144
2145                                  ;;****  GEN'L PURPOSE SCRATCH STORAGE  *****
2146  002602  000000                  REG0:   .WORD   0
2147  002604  000000                  REG1:   .WORD   0
2148  002606  000000                  REG2:   .WORD   0
2149  002610  000000                  REG3:   .WORD   0
2150  002612  000000                  REG4:   .WORD   0
2151  002614  000000                  REG5:   .WORD   0
2152  002616  000000                  REG6:   .WORD   0
2153  002620  000000                  REG7:   .WORD   0
2154
2155                                  ;;****  SCRATCH STORAGE FOR MESSAGE REPORTING   *****
2156  002622  000000                  TMP0:   .WORD   0
2157  002624  000000                  TMP1:   .WORD   0
2158  002626  000000                  TMP2:   .WORD   0
2159  002630  000000                  TMP3:   .WORD   0
2160  002632  000000                  TMP4:   .WORD   0
2161  002634  000000                  TMP5:   .WORD   0
2162  002636  000000                  TMP6:   .WORD   0
2163  002640  000000                  TMP7:   .WORD   0
2164
2165                                  ;***** INBUS LU REG BIT MASKS FOR UNPREDICTABLE BITS *****
2166  002642                          UPBITS:
2167  002642      377                         .BYTE   377             ;MASK FOR RDBR
2168  002643      000                         .BYTE   000             ;MASK FOR RDSR
2169  002644      000                         .BYTE   000             ;MASK FOR TDBR
2170  002645      360                         .BYTE   360             ;MASK FOR TDSR
2171  002646      000                         .BYTE   000             ;MASK FOR SSAR
2172  002647      000                         .BYTE   000             ;MASK FOR PCSAR
2173  002650      347                         .BYTE   347             ;MASK FOR PCR
2174
2175  002651      200                 TDSRNRW: .BYTE  200             ;TDSR NON-R/W BITS
```

```
2176                                    .SBTTL  DATA TEST PATTERNS
2177                                    ;***** DATA PATTERN E *****
2178  002652                            PATE:
2179  002652    377                             .BYTE   377
2180  002653    377                             .BYTE   377
2181  002654    377                             .BYTF   377
2182  002655    377                             .BYTE   377
2183  002656    377                             .BYTE   377
2184  002657    377                             .BYTE   377
2185  002660    377                             .BYTE   377
2186  002661    366                             .BYTE   366
2187
2188                                    ;***** DATA PATTERN F *****
2189  002662                            PATF:
2190  002662    000                             .BYTE   000
2191  002663    000                             .BYTE   000
2192  002664    000                             .BYTE   000
2193  002665    000                             .BYTE   000
2194  002666    000                             .BYTE   000
2195  002667    000                             .BYTE   000
2196  002670    000                             .BYTE   000
2197  002671    110                             .BYTE   110
2198
2199                                    ;***** DATA PATTERN G *****
2200  002672                            PATG:
2201  002672    000                             .BYTE   000
2202  002673    001                             .BYTE   001
2203  002674    003                             .BYTE   003
2204  002675    004                             .BYTE   004
2205  002676    005                             .BYTE   005
2206  002677    007                             .BYTE   007
2207  002700    100                             .BYTE   100
2208  002701    101                             .BYTE   101
2209  002702    103                             .BYTE   103
2210  002703    104                             .BYTE   104
2211  002704    105                             .BYTE   105
2212  002705    107                             .BYTE   107
2213  002706    000                             .BYTE   000
2214  002707    017                             .BYTE   017
2215  002710    027                             .BYTE   027
2216  002711    041                             .BYTE   041
2217  002712    200                             .BYTE   200
2218  002713    277                             .BYTE   277
2219  002714    103                             .BYTE   103
2220  002715    144                             .BYTE   144
2221  002716    115                             .BYTE   115
2222  002717    157                             .BYTE   157
2223  002720    000                             .BYTE   000
2224
2225                                    ;***** DATA PATTERN H *****
2226        002722                      .EVEN
2227  002722                            PATH:
2228  002722    125                             .BYTE   125
2229  002723    252                             .BYTE   252
2230  002724    000                             .BYTE   000
2231  002725    377                             .BYTE   377
```

```
 2232  002726  000                        .BYTE   000
 2233  002727  001                        .BYTE   001
 2234  002730  002                        .BYTE   002
 2235  002731  004                        .BYTE   004
 2236  002732  010                        .BYTE   010
 2237  002733  020                        .BYTE   020
 2238  002734  040                        .BYTE   040
 2239  002735  100                        .BYTE   100
 2240  002736  200                        .BYTE   200
 2241  002737  000                        .BYTE   000
 2242  002740  377                        .BYTE   377
 2243  002741  376                        .BYTE   376
 2244  002742  375                        .BYTE   375
 2245  002743  373                        .BYTE   373
 2246  002744  367                        .BYTE   367
 2247  002745  357                        .BYTE   357
 2248  002746  337                        .BYTE   337
 2249  002747  277                        .BYTE   277
 2250  002750  177                        .BYTE   177
 2251  002751  377                        .BYTE   377
 2252  002752  000                        .BYTE   000
 2253
 2254                                  ;***** DATA PATTERN I *****
 2255  002753                          PATI:
 2256  002753  000                        .BYTE   000
 2257  002754  041                        .BYTE   041
 2258  002755  102                        .BYTE   102
 2259  002756  143                        .BYTE   143
 2260  002757  204                        .BYTE   204
 2261  002760  245                        .BYTE   245
 2262  002761  306                        .BYTE   306
 2263  002762  347                        .BYTE   347
 2264  002763  000                        .BYTE   000
 2265  002764  001                        .BYTE   001
 2266  002765  002                        .BYTE   002
 2267  002766  004                        .BYTE   004
 2268  002767  040                        .BYTE   040
 2269  002770  100                        .BYTE   100
 2270  002771  200                        .BYTE   200
 2271  002772  000                        .BYTE   000
 2272  002773  346                        .BYTE   346
 2273  002774  345                        .BYTE   345
 2274  002775  343                        .BYTE   343
 2275  002776  307                        .BYTE   307
 2276  002777  247                        .BYTE   247
 2277  003000  147                        .BYTE   147
 2278  003001  347                        .BYTE   347
 2279  003002  242                        .BYTE   242
 2280  003003  105                        .BYTE   105
 2281  003004  347                        .BYTE   347
 2282  003005  010                        .BYTE   010
 2283  003006  020                        .BYTE   020
 2284  003007  367                        .BYTE   367
 2285  003010  357                        .BYTE   357
 2286  003011  030                        .BYTE   030
 2287  003012  027                        .BYTE   027
```

```
2288  003013  377                     .BYTE   377
2289
2290                                  ;***** DATA PATTERN J *****
2291  003014                          PATJ:
2292  003014  000                     .BYTE   000
2293  003015  000                     .BYTE   000
2294  003016  001                     .BYTE   001
2295  003017  002                     .BYTE   002
2296  003020  004                     .BYTE   004
2297  003021  020                     .BYTE   020
2298  003022  040                     .BYTE   040
2299  003023  010                     .BYTE   010
2300
2301                                  ;***** DATA PATTERN K *****
2302  003024                          PATK:
2303  003024  000                     .BYTE   000
2304  003025  377                     .BYTE   377
2305  003026  376                     .BYTE   376
2306  003027  375                     .BYTE   375
2307  003030  373                     .BYTE   373
2308  003031  376                     .BYTE   376
2309  003032  177                     .BYTE   177
2310  003033  377                     .BYTE   377
2311  003034  000                     .BYTE   000
2312  003035  001                     .BYTE   001
2313  003036  002                     .BYTE   002
2314  003037  004                     .BYTE   004
2315  003040  010                     .BYTE   010
2316  003041  200                     .BYTE   200
2317  003042  125                     .BYTE   125
2318  003043  252                     .BYTE   252
2319  003044  000                     .BYTE   000
2320
2321                                  ;***** DATA PATTERN L *****
2322  003045                          PATL:
2323  003045  000                     .BYTE   000
2324  003046  017                     .BYTE   017
2325  003047  016                     .BYTE   016
2326  003050  015                     .BYTE   015
2327  003051  013                     .BYTE   013
2328  003052  016                     .BYTE   016
2329  003053  017                     .BYTE   017
2330  003054  017                     .BYTE   017
2331  003055  000                     .BYTE   000
2332  003056  001                     .BYTE   001
2333  003057  002                     .BYTE   002
2334  003060  004                     .BYTE   004
2335  003061  010                     .BYTE   010
2336  003062  000                     .BYTE   000
2337  003063  005                     .BYTE   005
2338  003064  012                     .BYTE   012
2339  003065  000                     .BYTE   000
```

```
 2340
 2341                                   ;***** DATA PATTERN Q *****
 2342  003066    000                   PATQ:   .BYTE   000
 2343  003067    003                           .BYTE   003
 2344  003070    014                           .BYTE   014
 2345  003071    060                           .BYTE   060
 2346  003072    001                           .BYTE   001
 2347  003073    007                           .BYTE   007
 2348  003074    037                           .BYTE   037
 2349  003075    177                           .BYTE   177
 2350
 2351                                   ;***** DATA PATTERN INVERTED Q *****
 2352  003076    000                   PATQB:  .BYTE   000     ;INVERTED 000 (7 BIT)
 2353  003077    140                           .BYTE   140     ;INVERTED 003 (7 BIT)
 2354  003100    030                           .BYTE   030     ;INVERTED 014 (7 BIT)
 2355  003101    006                           .BYTE   006     ;INVERTED 060 (7 BIT)
 2356  003102    100                           .BYTE   100     ;INVERTED 001 (7 BIT)
 2357  003103    160                           .BYTE   160     ;INVERTED 007 (7 BIT)
 2358  003104    174                           .BYTE   174     ;INVERTED 037 (7 BIT)
 2359  003105    177                           .BYTE   177     ;INVERTED 177 (7 BIT)
 2360  003106                           ENDPAT:
 2361                                   .EVEN
```

```
2362
2363
2364
2365                                    ;*** RECEIVED DATA BUFFER (64. WORDS) ***
2366  003106  000100                    RCVBUF: .BLKW   64.
2367
2368
2369
2370
```

```
2371                                    .SBTTL  GLOBAL TEXT SECTION
2372
2373                                    ;*****************************************************************************
2374                                    ;*      THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
2375                                    ;*      MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
2376                                    ;*      MORE THAN ONE TEST.
2377                                    ;*****************************************************************************
2378
2379                                    ;;*************************************************************************
2380                                    ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
2381                                    ;;*************************************************************************
2382  003306                                    DEVTYP  <M8053 OR M8064>
2383  003306                                                                      L$DVTYP::
2384  003306  034115  032460  020063                                                      .ASCIZ  /M8053 OR M8064/
2385  003314  051117  046440  030070
2386  003322  032066     000
2387          003326                                                              .EVEN
2388
2389                                    ;;*************************************************************************
2390                                    ;* TITLE OF PROGRAM
2391                                    ;;*************************************************************************
2392
2393          000012                    .RADIX  10.
2394  003326                                    DESCRIPT        <DMV-11 LINE UNIT TESTS - PART 1 OF 3>
2395  003326                                                                      L$DESC::
2396  003326  046504  026526  030461                                                      .ASCIZ  /DMV-11 LINE UNI
2397  003334  046040  047111  020105
2398  003342  047125  052111  052040
2399  003350  051505  051524  026440
2400  003356  050040  051101  020124
2401  003364  020061  043117  031440
2402  003372     000
2403          003374                                                              .EVEN
2404          000010                    .RADIX  8.
2405
2406
```

```
                                .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    @#RDY,@#BSEL2   ;DID THE M-LOOP FINISH
003422  001023                          BNE     5$              ;YES,  GOOD.  RETURN
003424  004737  004210                  JSR     PC,GETUSR       ;GET BYTE SELECT REGISTERS
003430  012737  000301  002400          MOV     #RUN!MCLR!MREQ,@DATA    ;IDENTIFY REQUESTED FUNCTION
003436                                  GTDF    EM3,ERR4        ;"RDY" 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
```

```
2453                                    .SBTTL ....M-LOOP -- READ
2454                                    ;+**********************************************************************
2455                                    ; READ - READ THE SPECIFIED ADDRESS WITHIN THE DMV-11 (M8053)
2456                                    ;
2457                                    ;  CALLING SEQUENCE:
2458                                    ;
2459                                    ;       JSR     R5,READ
2460                                    ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
2461                                    ;       .WORD   <DESTINATION ADDRESS WITHIN LSI-11>
2462                                    ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2463                                    ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
2464                                    ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
2465                                    ;
2466                                    ; N$:   <RESUMPTION OF NORMAL PROCESSING>
2467                                    ;
2468                                    ;--**********************************************************************
2469
2470  003476  012577  177000            READ:   MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
2471  003502  112777  000001  176766            MOVB    #REDLOC,@SEL2   ;TELL M-LOOP TO GIVE US THE REQUESTED DATA
2472
2473  003510  010346                            MOV     R3,-(SP)
2474  003512  012703  001750                    MOV     #1000.,R3               ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2475  003516  077301                    1$:     SOB     R3,1$
2476  003520  012603                            MOV     (SP)+,R3
2477
2478  003522  132777  000200  176746            BITB    #RDY,@SEL2      ;DID THE M-LOOP FINISH
2479  003530  001023                            BNE     5$              ;YES,  GOOD.  RETURN
2480
2481  003532  004737  004210                    JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
2482  003536  012737  000001  002400            MOV     #REDLOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
2483  003544                                    GTDF    EM4,ERR4        ;"MRDY" TIMEOUT
2484                                            ;       QUEUE "DEVICE FATAL" ERROR # 2
2485  003544  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
2486  003552  012737  000002  002250                                                    MOV     #2,ERRNBR
2487  003560  012737  013235  002252                                                    MOV     #EM4,ERRMSG
2488  003566  012737  017030  002254                                                    MOV     #ERR4,ERRBLK
2489  003574  000261                            SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
2490  003576  000401                            BR      6$              ;RETURN WITH THAT INDICATION
2491
2492  003600  000241                    5$:     CLC                     ;INDICATE "NO ERROR"
2493  003602  117735  176700            6$:     MOVB    @SEL6,@(R5)+    ;PUT DATA WHERE CALLER WANTS IT
2494  003606  000205                            RTS     R5              ;RETURN
2495
2496
2497
2498
```

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

003610                  READI:
003610  012577  176666          MOV     (R5)+,@#SEL4        ;SETUP SOURCE POINTER
003614  112777  000001  176654  MOVB    @#REDLOC,@#SEL2     ;TELL M-LOOP TO GIVE US THE REQUESTED DATA

003622  010346          MOV     R3,-(SP)
003624  012703  001750  MOV     #1000.,R3           ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
003630  077301          1$:     SOB     R3,1$
003632  012603          MOV     (SP)+,R3

003634  132777  000200  176634  BITB    @#RDY,@#SEL2        ;DID THE M-LOOP FINISH
003642  001023          BNE     5$                  ;YES.  GOOD.  RETURN

003644  004737  004210          JSR     PC,GETMSR           ;GET BYTE SELECT REGISTERS
003650  012737  000001  002400  MOV     #REDLOC,@DATA       ;IDENTIFY REQUESTED FUNCTION
003656                  GTDF    EM4,ERR4            ;"RDY" TIMEOUT
                        ;       QUEUE "DEVICE FATAL" ERROR # 3
003656  012737  000001  002246                                  MOV     #T.EDF,ERRTYP
003664  012737  000003  002250                                  MOV     #3,ERRNBR
003672  012737  013235  002252                                  MOV     #EM4,ERRMSG
003700  012737  017030  002254                                  MOV     #ERR4,ERRBLK
003706  000261          SEC                         ;INDICATE AN ERROR HAS BEEN STACKED
003710  000401          BR      6$                  ;RETURN WITH THAT INDICATION

003712  000241          5$:     CLC                 ;INDICATE "NO ERROR"
003714  017725  176566  6$:     MOV     @#SEL6,(R5)+        ;PUT DATA WHERE CALLER WANTS IT
003720  000205          RTS     R5                  ;RETURN
```

....M-LOOP -- WRITE

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

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

```
2570                                    .SBTTL ....M-LOOP -- WRITE IMMEDIATE
2571                                    ;+**********************************************************************
2572                                    ; WRITEI - WRITE IMMEDIATE THE SPECIFIED DATA INTO THE SPECIFIED DMV-11 ADDRESS
2573                                    ;
2574                                    ; CALLING SEQUENCE:
2575                                    ;
2576                                    ;       JSR     R5,WRITEI
2577                                    ;       .WORD   <ADDRESS OF REGISTER WITHIN DMV-11>
2578                                    ;       .WORD   <DATA FIELD -- DATA TO BE WRITTEN IN DMV-11>
2579                                    ;       BCC     N$              ;IF NO ERROR OCCURED, PROCEED WITH ROUTINE
2580                                    ;       ERROR                   ;AN ERROR MESSAGE HAS BEEN STACKED:  PRINT IT
2581                                    ;       <ANY OTHER SPECIAL ERROR PROCESSING MAY BE DONE HERE  (I.E. CKLOOP)>
2582                                    ;
2583                                    ; N$:   <RESUMPTION OF NORMAL PROCESSING>
2584                                    ;
2585                                    ;--**********************************************************************
2586
2587  003734                            WRITEI:
2588  003734  012577  176542                   MOV     (R5)+,@SEL4     ;SETUP SOURCE POINTER
2589  003740  012577  176542                   MOV     (R5)+,@SEL6     ;MAKE DATA AVAILABLE TO M-LOOP
2590  003744  112777  000002  176524  MLWRI:  MOVB    #WRILOC,@BSEL2  ;TELL M-LOOP TO WRITE THE DATA
2591
2592  003752  010346                           MOV     R3,-(SP)
2593  003754  012703  001750                   MOV     #1000.,R3       ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
2594  003760  077301                    1$:    SOB     R3,1$
2595  003762  012603                           MOV     (SP)+,R3
2596
2597  003764  132777  000200  176504         BITB    #RDY,@BSEL2     ;DID THE M-LOOP FINISH
2598  003772  001023                           BNE     5$              ;YES.  GOOD.  RETURN
2599  003774  004737  004210                   JSR     PC,GETWSR       ;GET BYTE SELECT REGISTERS
2600  004000  012737  000002  002400         MOV     #WRILOC,GDATA   ;IDENTIFY REQUESTED FUNCTION
2601  004006                                   GTDF    EM4,ERR4        ;"RDY" TIMEOUT
2602                                          ;       QUEUE "DEVICE FATAL" ERROR # 4
2603  004006  012737  000001  002246                                                   MOV     #T.EDF,ERRTYP
2604  004014  012737  000004  002250                                                   MOV     #4,ERRNBR
2605  004022  012737  013235  002252                                                   MOV     #EM4,ERRMSG
2606  004030  012737  017030  002254                                                   MOV     #ERR4,ERRBLK
2607  004036  000261                           SEC                     ;INDICATE AN ERROR HAS BEEN STACKED
2608  004040  000401                           BR      6$              ;RETURN WITH THAT INDICATION
2609
2610  004042  000241                    5$:    CLC                     ;INDICATE "NO ERROR"
2611  004044  000205                    6$:    RTS     R5              ;RETURN
2612
2613
2614
2615
```

```
                                .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  117737  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     @#BSEL0,WSR0     ;MOVE THE 4 WORD REGISTERS TO THE OTHERWISE
004216  017737  176254  002260           MOV     @#BSEL2,WSR2     ;BYTE TABLE
004224  017737  176252  002262           MOV     @#BSEL4,WSR4
004232  017737  176250  002264           MOV     @#BSEL6,WSR6
004240  017737  176246  002266           MOV     @#BSEL10,WSR10
004246  017737  176244  002270           MOV     @#BSEL12,WSR12
004254  017737  176242  002272           MOV     @#BSEL14,WSR14
004262  017737  176240  002274           MOV     @#BSEL16,WSR16
004270  000207                           RTS     PC               ;RETURN TO CALLER
```

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

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,READ         ;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,ERRNUM
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
```

```
2708
2709
2710
2711                                    .SBTTL ....STALL -- DELAY FOR 10.5 MICRO-SEC'S (ON LSI-11)
2712                                    ;+*****************************************************************************
2713                                    ; STALL -- THIS SUBROUTINE STALLS FOR ABOUT 10.5 MICRO-SECONDS
2714                                    ;--*****************************************************************************
2715
2716  004400  000207                   STALL:  RTS     PC
2717
2718
2719
```

```
                                        .SBTTL

                                        ;*************************************************************************
                                        ;* GETURS - LOAD INTO THE 8 WORD STORAGE AREA (UREGS) THE CONTENTS OF THE
                                        ;*      VARIOUS USYRT REGISTERS
                                        ;*
                                        ;*        CALLING SEQUENCE:
                                        ;*
                                        ;*************************************************************************
004402  012737  002316  004444  GETURS: MOV     #UREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
004410  012737  120400  004442          MOV     #USYRT,4$       ;INIT POINTER TO REGISTER ADDRESSES

004416  005037  002334                  CLR     UREGS+14.       ;CLEAR STORAGE WORD
004422  004537  003476                  JSR     R5,READ         ;READ THE USYRT STATUS REGISTER
004426  122000                          .WORD   USTATR          ;STATUS REGISTER'S ADDRESS WITHIN DMV-11
004430  002334                          .WORD   UREGS+14.       ;ADDRESS ALLOCATED TO THAT REG. W/IN "UREGS"

004432  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
004436  004537  003476                  JSR     R5,READ         ;READ A LINE UNIT REG
004442  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
004444  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE

004446  005237  004442          6$:     INC     4$              ;INCREMENT REG NO.
004452  023727  004442  120406          CMP     4$,#USYRT+6     ;THIS IS NOT A VALID REGISTER ADDRESS
004460  001772                          BEQ     6$              ;SO IT MUST BE BYPASSED

004462  062737  000002  004444          ADD     #2.5$           ;ADVANCE ADDRESS OF STORAGE AREA POINTER
004470  023727  004442  120410          CMP     4$,#USYRT+10    ;SEE IF ALL REGS READ YET
004476  001355                          BNE     3$              ;BR IF NOT

004500  000207                          RTS     PC              ;RETURN


                                        ;*************************************************************************
                                        ;* GETVRS: - LOAD INTO THE 16 WORD STORAGE AREA (VREGS) THE CONTENTS OF THE
                                        ;*      VARIOUS VIA REGISTERS.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*************************************************************************
004502  012737  002336  004530  GETVRS: MOV     #VREGS,5$       ;INIT POINTER TO REG STORAGE TABLE
004510  012737  120000  004526          MOV     #VIA,4$         ;INIT POINTER TO REGISTER ADDRESSES
004516  005077  000006          3$:     CLR     @5$             ;CLEAR STORAGE WORD
004522  004537  003476                  JSR     R5,READ         ;READ A VIA REG
004526  000000                  4$:     .WORD   0               ;REGISTER ADDRESS GOES HERE
004530  000000                  5$:     .WORD   0               ;STORAGE ADRS IN TABLE GOES HERE
004532  005237  004526          6$:     INC     4$              ;INCREMENT REG NO.
004536  062737  000002  004530          ADD     #2.5$           ;INCREMENT STORAGE ADRS
004544  023727  004526  120020          CMP     4$,#VIA+16.     ;SEE IF ALL VIA REGS READ YET
004552  001361                          BNE     3$              ;BR IF NOT
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
```

```
2827  004660  010137  004672                  MOV     R1,4$           ;PASS THE NEW REG. SETTING TO APPROPRIATE CALL
2828
2829  004664  004537  003734                  JSR     R5,WRITEI       ;WRITE TO
2830  004670  120013                          VIAACR                  ;THE VIA'S ACR
2831  004672  000000                  4$:     .WORD   0               ;THE NEW REGISTER SETTING
2832
2833  004674  004537  003734                  JSR     R5,WRITEI         RITE TO
2834  004700  120006                          VIAT1C                  ..OW ORDER LATCH REGISTER (T1L-L)
2835  004702  000000                  7$:     .WORD   0               ;THE VALUE PASSED
2836
2837  004704  113737  004703  004720          MOVB    7$+1,8$         ;SETUP FOR AND
2838  004712  004537  003734                  JSR     R5,WRITEI       ;WRITE TO
2839  004716  120007                          VIAT1D                  ;HIGH ORDER LATCH REGESTER (T1L-H)
2840  004720  000000                  8$:     .WORD   0               ;THE VALUE PASSED
2841
2842  004722  004537  003734                  JSR     R5,WRITEI       ;WRITE TO
2843  004726  120004                          VIAT1A                  ;LOW ORDER LATCH & COUNTER (T1L-L & T1C-L)
2844  004730  000000                  10$:    .WORD   0               ;THE VALUE PASSED
2845
2846  004732  113737  004731  004746          MOVB    10$+1,11$       ;SETUP FOR AND
2847  004740  004537  003734                  JSR     R5,WRITEI       ;WRITE TO
2848  004744  120005                          VIAT1B                  ;HIGH ORDER COUNTER (T1C-H) <ALSO STARTS CTR>
2849  004746  000000                  11$:    .WORD   0               ;THE VALUE PASSED
2850
2851                                  ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST RETURN!
2852
2853  004750  012601                          MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
2854  004752  005205                          INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
2855                                                                  ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
2856
2857  004754  000205                          RTS     R5              ;NOW, RETURN
2858
2859
```

```
                        .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
```

```
 2916  005062  004537  003734                  JSR     R5,WRITEI       ;WRITE TO
 2917  005066  120013                          VIAACR                  ;THE VIA'S ACR
 2918  005070  000000                  4$:     .WORD   0               ;THE NEW REGISTER SETTING
 2919
 2920  005072  004537  003734                  JSR     R5,WRITEI       ;WRITE TO
 2921  005076  120010                          VIAT2A                  ;LOW ORDER LATCH & COUNTER (T2L-L & T2C-L)
 2922  005100  000000                  10$:    .WORD   0               ;THE VALUE PASSED
 2923
 2924  005102  113737  005101  005116          MOVB    10$+1,11$       ;SETUP FOR AND
 2925  005110  004537  003734                  JSR     R5,WRITEI       ;WRITE TO
 2926  005114  120011                          VIAT2B                  ;HIGH ORDER COUNTER (T2C-H) <ALSO STARTS CTR>
 2927  005116  000000                  11$:    .WORD   0               ;THE VALUE PASSED
 2928
 2929                                  ; DON'T WAIT AROUND FOR ANYTHING TO HAPPEN -- JUST (JEST) RETURN!
 2930
 2931  005120  012601                          MOV     (SP)+,R1        ;BUT FIRST RESTORE R1
 2932  005122  005205                          INC     R5              ;AND PUT R5 BACK ON A WORD BOUNDRY (THE LAST
 2933                                                                  ;PASSED PARAM. WAS A BYTE, NOT A WORD!)
 2934
 2935  005124  000205                          RTS     R5              ;THEN RETURN
 2936
```

```
                        .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$,BDATA        ;GET ACTUAL DATA
                        ;STACK "USYRT NOT CLEARED BY PROGRAM RESET" MSG
005234                  GTDF    EM2,ERR10
                        ;       QUEUE "DEVICE FATAL" ERROR @ 6
005234  012737  000001  002246                  MOV     @T.EDF,ERRTYP
005242  012737  000006  002250                  MOV     @6,ERRHDR
005250  012737  013142  002252                  MOV     @EM2,ERRMSG
005256  012737  017372  002254                  MOV     @ERR10,ERRBLK
005264  000261                  SEC             ;SET C BIT TO FLAG ERROR
005266  000406                  BR      10$     ;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
```

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

                        ;*************************************************************************
                        ;* 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$,BDATA
                                ;STACK "USYRT STATUS INCORRECT" ERROR
005500                                  GTDF    EM68,ERR10
                                ;       QUEUE "DEVICE FATAL" ERROR # 7
005500  012737  000001  002246                                          MOV     #T.EDF,ERRTYP
005506  012737  000007  002250                                          MOV     #7,ERRNAM
005514  012737  013760  002252                                          MOV     #EM68,EMSG
005522  012737  017372  002254                                          MOV     #ERR10,ERRBLK
005530  000261                          SEC                     ;SET C BIT FOR ERROR
005532  005205                  2$:     INC     R5              ;INCREMENT R5 PAST ARGUMENT
005534  000205                          RTS     R5              ;RETURN
```

```
                                        ....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)

                                        .SBTTL ....CKTACT -- CHECK TRANSMITTER ACTIVE (TXACT)
                                        ;**************************************************************************
                                        ;* CKTACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TXACT IN THE USYRT
                                        ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                        ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                        ;*
                                        ;*      CALLING SEQUENCE :
                                        ;*      JSR     R5,CKTACT
                                        ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TXACT>
                                        ;**************************************************************************
005536                                  CKTACT:
005536  012737  000007  002412                  MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
005544  004537  003610                          JSR     R5,READI        ;READ USYRT STATUS
005550  122000                                  USTATR
005552  000000                          1$:     .WORD   0
005554  032725  000001                          BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TXACT
005560  001422                                  BEQ     2$              ;BR IF EXPECTED TXACT = 0
005562  132737  000004  005552                  BITB    #TXACT,1$       ;SEE IF TXACT = 1
005570  001040                                  BNE     3$              ;BR IF TXACT = 1
                                        ;STACK "TXACT NOT SET" MSG
005572                                          GTDF    EM69,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 8
005572  012737  000001  002246                                                          MOV     #T.EDF,ERRTYP
005600  012737  000010  002250                                                          MOV     #8.,ERRNBR
005606  012737  014007  002252                                                          MOV     #EM69,ERRMSG
005614  012737  017722  002254                                                          MOV     #ERR12,ERRBLK
005622  000261                                  SEC                     ;SET C BIT TO FLAG ERROR
005624  000423                                  BR      4$              ;TAKE ERROR EXIT
005626  132737  000004  005552          2$:     BITB    #TXACT,1$       ;SEE IF TXACT = 0
005634  001416                                  BEQ     3$              ;BR IF TXACT = 0
                                        ;STACK "TXACT NOT CLEARED" MSG
005636                                          GTDF    EM70,ERR12
                                                                        ;       QUEUE "DEVICE FATAL" ERROR # 9
005636  012737  000001  002246                                                          MOV     #T.EDF,ERRTYP
005644  012737  000011  002250                                                          MOV     #9.,ERRNBR
005652  012737  014025  002252                                                          MOV     #EM70,ERRMSG
005660  012737  017722  002254                                                          MOV     #ERR12,ERRBLK
005666  000261                                  SEC                     ;SET C BIT TO FLAG ERROR
005670  000401                                  BR      4$              ;TAKE ERROR EXIT
005672  000241                          3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
005674  000205                          4$:     RTS     R5              ;RETURN
```

```
                                .SBTTL ....CKRACT -- CHECK RECEIVER ACTIVE (RXACT)
                                ;**************************************************************************
                                ;* CKRACT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RXACT IN THE USYRT
                                ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,CKRACT
                                ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF RXACT>
                                ;**************************************************************************
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
```

```
                                .SBTTL ....CKTBMT -- CHECK TRANSMIT BUFFER EMPTY
                                ;**************************************************************************
                                ;* CKTBMT - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF TBMT IN THE USYRT
                                ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,CKTBMT
                                ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF TBMT>
                                ;**************************************************************************
006036                          CKTBMT:
006036  012737  000007  002412          MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
006044  004537  003610                  JSR     R5,READI        ;READ USYRT STATUS
006050  122000                          USTATR
006052  000000                  1$:     .WORD   0
006054  032725  000001                  BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF TBMT
006060  001422                          BEQ     2$              ;BR IF EXPECTED TBMT = 0
006062  132737  000100  006052          BITB    #TBMT,1$        ;SEE IF TBMT = 1
006070  001040                          BNE     3$              ;BR IF TBMT = 1
                                ;STACK "TBMT NOT SET" MSG
006072                                  GTDF    EM73,ERR12
                                                        ;       QUEUE "DEVICE FATAL" ERROR # 12
006072  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006100  012737  000014  002250                                                  MOV     #12,ERRNBR
006106  012737  014107  002252                                                  MOV     #EM73,ERRMSG
006114  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006122  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006124  000423                          BR      4$              ;TAKE ERROR EXIT
006126  132737  000100  006052  2$:     BITB    #TBMT,1$        ;SEE IF TBMT = 0
006134  001416                          BEQ     3$              ;BR IF TBMT = 0
                                ;STACK "TBMT NOT CLEARED" MSG
006136                                  GTDF    EM74,ERR12
                                                        ;       QUEUE "DEVICE FATAL" ERROR # 13
006136  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
006144  012737  000015  002250                                                  MOV     #13,ERRNBR
006152  012737  014124  002252                                                  MOV     #EM74,ERRMSG
006160  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
006166  000261                          SEC                     ;SET C BIT TO FLAG ERROR
006170  000401                          BR      4$              ;TAKE ERROR EXIT
006172  000241                  3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
006174  000205                  4$:     RTS     R5              ;RETURN
```

```
                                .SBTTL ....CKRDA -- CHECK RECEIVE DATA AVAILABLE
                                ;**********************************************************************
                                ;* CKRDA - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RDA IN THE USYRT
                                ;*       STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
                                ;*       STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
                                ;*
                                ;*       CALLING SEQUENCE :
                                ;*       JSR     R5,CKRDA
                                ;*       .WORD   <BIT 0 IS EXPECTED VALUE OF RDA>
                                ;**********************************************************************
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
```

```
3270                                    .SBTTL ....CKRSA -- CHECK RECEIVER STATUS AVAILABLE
3271                                    ;**************************************************************************
3272                                    ;* CKRSA - THIS SUBROUTINE CHECKS FOR THE PROPER STATE OF RSA IN THE USYRT
3273                                    ;*      STATUS REGISTER, AND REPORTS AN ERROR IF IT IS NOT PROPERLY SET TO THE
3274                                    ;*      STATE OF BIT 0 IN THE WORD FOLLOWING THE CALL.
3275                                    ;*
3276                                    ;*      CALLING SEQUENCE :
3277                                    ;*      JSR     R5,CKRSA
3278                                    ;*      .WORD   <BIT 0 IS EXPECTED VALUE OF RSA>
3279                                    ;**************************************************************************
3280  006336                            CKRSA:
3281  006336  012737  000007  002412            MOV     #7,REGNUM       ;SET REG NO. FOR POSSIBLE ERROR REPORT
3282  006344  004537  003610                    JSR     R5,READI        ;READ USYRT STATUS
3283  006350  122000                            USTATR
3284  006352  000000                    1$:     .WORD   0
3285  006354  032725  000001                   BIT     #BIT0,(R5)+     ;GET EXPECTED STATE OF RSA
3286  006360  001422                            BEQ     2$              ;BR IF EXPECTED RSA = 0
3287  006362  132737  000020  006352           BITB    #RSA,1$         ;SEE IF RSA = 1
3288  006370  001040                            BNE     3$              ;BR IF RSA = 1
3289                                    ;STACK "RSA NOT SET" MSG
3290  006372                                    GTDF    EM77,ERR12
3291                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 16
3292  006372  012737  000001  002246                                                                    MOV     #T.EDF,ERRTYP
3293  006400  012737  000020  002250                                                                    MOV     #16,ERRNBR
3294  006406  012737  014201  002252                                                                    MOV     #EM77,ERRMSG
3295  006414  012737  017722  002254                                                                    MOV     #ERR12,ERRBLK
3296  006422  000261                            SEC                     ;SET C BIT TO FLAG ERROR
3297  006424  000423                            BR      4$              ;TAKE ERROR EXIT
3298  006426  132737  000020  006352  2$:      BITB    #RSA,1$         ;SEE IF RSA = 0
3299  006434  001416                            BEQ     3$              ;BR IF RSA = 0
3300                                    ;STACK "RSA NOT CLEARED" MSG
3301  006436                                    GTDF    EM78,ERR12
3302                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 17
3303  006436  012737  000001  002246                                                                    MOV     #T.EDF,ERRTYP
3304  006444  012737  000021  002250                                                                    MOV     #17,ERRNBR
3305  006452  012737  014215  002252                                                                    MOV     #EM78,ERRMSG
3306  006460  012737  017722  002254                                                                    MOV     #ERR12,ERRBLK
3307  006466  000261                            SEC                     ;SET C BIT TO FLAG ERROR
3308  006470  000401                            BR      4$              ;TAKE ERROR EXIT
3309  006472  000241                    3$:     CLC                     ;CLEAR C BIT FOR NO ERRORS
3310  006474  000205                    4$:     RTS     R5              ;RETURN
3311
3312
```

```
                                .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
```

```
3413  007054  001416                          BEQ     5$                ;BR IF REOM = 0
3414                                  ;STACK "REOM NOT CLEARED" MSG
3415  007056                                  GTDF    EM30,ERR12
3416                                                    ;       QUEUE "DEVICE FATAL" ERROR # 23
3417  007056  012737  000001  002246                                                  MOV     #T.EDF,ERRTYP
3418  007064  012737  000027  002250                                                  MOV     #23,ERRNBR
3419  007072  012737  013535  002252                                                  MOV     #EM30,ERRMSG
3420  007100  012737  017722  002254                                                  MOV     #ERR12,ERRBLK
3421  007106  000261                          SEC                       ;SET C BIT TO FLAG ERROR
3422  007110  000401                          BR      6$                ;TAKE ERROR EXIT
3423  007112  000241                  5$:     CLC                       ;CLEAR C BIT FOR NO ERRORS
3424  007114  000205                  6$:     RTS     R5                ;RETURN
3425
3426
```

```
                                .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)>
                                ;**************************************************************************
007400                          INITRN:
007400  010146                          MOV     R1,-(SP)        ;SAVE R1
007402  004537  003734                  JSR     R5,WRITEI       ;RESET THE USYRT
007406  120000                          VIAORB
007410  000031                          RTSND!DTR!PRESET
007412  004537  003734                  JSR     R5,WRITEI       ;CLEAR USYRT RESET BIT
007416  120000                          VIAORB
007420  000030                          RTSND!DTR
007422  112537  007434                  MOVB    (R5)+,1$        ;GET VALUE TO LOAD INTO USYRT PCSARL
007426  004537  003734                  JSR     R5,WRITEI       ;LOAD USYRT PCSARL
007432  120404                          PCSARL
007434  000000                  1$:     .WORD   0
007436  112537  007450                  MOVB    (R5)+,2$        ;GET VALUE TO LOAD INTO PCSARH
007442  004537  003734                  JSR     R5,WRITEI       ;LOAD USYRT PCSARH
007446  120405                          PCSARH
007450  000000                  2$:     .WORD   0
007452  112537  007476                  MOVB    (R5)+,3$        ;GET VALUE TO LOAD INTO PCR
007456  005037  002456                  CLR     SAVLEN
007462  113737  007476  002456          MOVB    3$,SAVLEN       ;SAVE CHAR LENGTH BITS
007470  004537  003734                  JSR     R5,WRITEI       ;LOAD USYRT PCR
007474  120407                          PCR
007476  000000                  3$:     .WORD   0
007500  004537  003734                  JSR     R5,WRITEI       ;SET ACR FOR T1 ONE-SHOT MODE
007504  120013                          VIAACR
007506  000200                          200
007510  004537  003734                  JSR     R5,WRITEI       ;LOAD VIA T1L-L
007514  120006                          VIAT1C
007516  000300                          300
007520  004537  003734                  JSR     R5,WRITEI       ;LOAD VIA T1L-H
007524  120007                          VIAT1D
007526  000000                          000
007530  004537  005432                  JSR     R5,CKUSTS       ;CHK USYRT STATUS FOR INIT'D STATE
007534  000110                          110                     ; TBMT = 1, TSO = 1
007536  103454                          BCS     7$              ;IF ERROR, EXIT SUBROUTINE

007540  013737  007674  007560          MOV     20$,13$         ;* SET UP DEFAULT VIAORB PARAMETERS
007546  142537  007560                  BICB    (R5)+,13$       ;* CLEAR ANY SPECIFIED VIAORB BITS.

007552  004537  003734                  JSR     R5,WRITEI       ;SET UP USYRT
```

```
3580  007556  120000                        VIAORB
3581  007560  000142              13$:      TXEN!RXEN!TTLOOP        ;* THIS VALUE MIGHT BE MODIFIED ABOVE
3582
3583  007562  004537  003734                JSR     R5,WRITEI       ;SET TSOM IN USYRT
3584  007566  120403                        TDSRH
3585  007570  000001                        TSOM
3586  007572  004537  003734                JSR     R5,WRITEI       ;LOAD SYNCH CHAR INTO TX BUF
3587  007576  120402                        TDSRL
3588  007600  000226                        SYNCH
3589  007602  004537  006036                JSR     R5,CKTBMT       ;CHK FOR TBMT = 0
3590  007606  000000                        0
3591  007610  103427                        BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3592  007612  005001                        CLR     R1              ;INIT CYCLE COUNTER
3593  007614  004537  011614          4$:   JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3594  007620  000001                        1
3595  007622  004537  003610                JSR     R5,READI        ;READ USYRT STATUS REG
3596  007626  122000                        USTATR
3597  007630  000000              5$:       .WORD   0
3598  007632  132737  000100  007630        BITB    #TBMT,5$        ;SEE IF TBMT IS SET YET
3599  007640  001010                        BNE     6$              ;BR IF YES
3600  007642  005201                        INC     R1              ;INCR CYCLE COUNTER
3601  007644  020127  000003                CMP     R1,#3           ;SEE IF 3 CYCLES DONE YET
3602  007650  002761                        BLT     4$              ;BR IF LESS THAN 3 CYCLES
3603  007652  004537  006036                JSR     R5,CKTBMT       ;GO STACK "TBMT NOT SET" MSG
3604  007656  000001                        1
3605  007660  103403                        BCS     7$              ;IF ERROR, EXIT SUBROUTINE
3606  007662  004537  005536          6$:   JSR     R5,CKTACT       ;CHK FOR TXACT = 1
3607  007666  000001                        1
3608  007670  012601                7$:     MOV     (SP)+,R1        ;RESTORE R1
3609  007672  000205                        RTS     R5              ;RETURN (IF C = 1, WE HAD AN ERROR)
3610
3611  007674  000142                20$:    TXEN!RXEN!TTLOOP        ;DEFAULT VALUE FOR VIAORB: ENABLE
3612                                        ;TX AND RX ON USYRT, ASSERT RTS, DTR
3613
```

```
                                .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>
                                ;**************************************************************************
010010                          TXCTRL:
010010  010146                         MOV     R1,-(SP)        ;SAVE R1
010012  010246                         MOV     R2,-(SP)        ;SAVE R2
010014  012537  010026                 MOV     (R5)+,2$        ;GET DATA FOR TDSRH
010020  004537  003734                 JSR     R5,WRITEI       ;LOAD DATA INTO TDSRH
010024  120403                         TDSRH
010026  000000                 2$:     .WORD   0
010030  005001                         CLR     R1              ;INIT CYCLE COUNT AND CLEAR C BIT
010032  012502                         MOV     (R5)+,R2        ;GET DESIRED NO. OF CYCLES
010034  001422                         BEQ     6$              ;BR IF NO CLOCKING DONE
010036  004537  005536         3$:     JSR     R5,CKTACT       ;CHECK TXACT = 1
010042  000001                         1
010044  103416                         BCS     6$              ;BR TO EXIT IF ERROR
010046  020102                         CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
010050  001411                         BEQ     5$              ;BR IF YES
010052  004537  006036                 JSR     R5,CKTBMT       ;CHECK FOR TBMT = 0
010056  000000                         0
010060  103410                         BCS     6$              ;BR TO EXIT IF ERROR
010062  004537  011614                 JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
010066  000001                         1
010070  005201                         INC     R1              ;INCR CYCLE COUNT
010072  000761                         BR      3$              ;KEEP CLOCKING
010074  004537  006036         5$:     JSR     R5,CKTBMT       ;CHK TBMT = 1
010100  000001                         1
010102  012602                 6$:     MOV     (SP)+,R2        ;RESTORE R2
010104  012601                         MOV     (SP)+,R1        ;RESTORE R1
010106  000205                         RTS     R5              ;RETURN (WITH C BIT = 1 IF ERROR)
```

```
                                .SBTTL ....RXCHAR -- RECEIVE A CHARACTER
                                ;**********************************************************************
                                ;* RXCHAR - THIS SUBROUTINE READS THE USYRT RDSR AND CHECKS THE CONTENTS
                                ;*       AGAINST THE DATA PASSED IN THE WORD FOLLOWING THE CALL.
                                ;*       IF BIT0 = 0 IN THE SECOND WORD FOLLOWING THE CALL, THE RERR BIT IS
                                ;*       NOT CHECKED AGAINST THE EXPECTED VALUE. THEN, IT CLOCKS
                                ;*       THE LINE UNIT FOR THE NO. OF CYCLES PASSED IN THE THIRD WORD
                                ;*       FOLLOWING THE CALL. THE PROGRAM CONTINUALLY MONITORS RDA AND RXACT.
                                ;*       IF THE SUBROUTINE DETECTS AN ERROR, THE ERROR INFORMATION
                                ;*       IS STACKED, AND THE C-BIT SET, WHICH LEAVES THE ERROR REPORTING AT THE
                                ;*       DISCRETION OF THE CALLING ROUTINE OR SUBROUTINE.
                                ;*
                                ;*       CALLING SEQUENCE :
                                ;*       JSR     R5,RXCHAR
                                ;*       .WORD   <EXPECTED RDSRL IN LO BYTE, RDSRH IN HI BYTE>
                                ;*       .WORD   <=0 FOR NO RERR CHK, =1 FOR RERR CHK>
                                ;*       .WORD   <NUMBER OF CYCLES TO CLOCK (IN LO BYTE)>
                                ;*               <SPECIAL DISABLE SWITCHES: NOCRDA,NFCRDA,NCRACT(IN HI BYTE)>
                                ;**********************************************************************
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
```

```
3768  010302                                        GTDF    EM34,ERR10
3769                                                ;       QUEUE "DEVICE FATAL" ERROR # 28
3770  010302  012737  000001  002246                                    MOV     #T.EDF,ERRTYP
3771  010310  012737  000034  002250                                    MOV     #28,ERRNBR
3772  010316  012737  013573  002252                                    MOV     #EM34,ERRMSG
3773  010324  012737  017372  002254                                    MOV     #ERR10,ERRBLK
3774  010332  000137  011350                JMP     20$             ;TAKE ERROR EXIT
3775  010336  116501  000001        6$:     MOVB    1(R5),R1        ;GET RDSRH
3776  010342  042701  177400                BIC     #177400,R1      ;MASK OFF UNUSED BITS
3777  010346  123701  010122                CMPB    2$,R1           ;COMPARE RCV'D STATUS TO EXPECTED
3778  010352  001016                        BNE     7$              ;BR IF MISMATCH
3779  010354  000137  011234                JMP     17$             ;CONTINUE
3780  010360  012737  000001  002412        MOV     #1,REGNUM       ;SET USYRT REG NO. FOR RDSRH
3781  010366  005037  002400                CLR     GDATA           ;SET EXPECTED DATA
3782  010372  110137  002400                MOVB    R1,GDATA
3783  010376  005037  002402                CLR     BDATA           ;SET ACTUAL DATA
3784  010402  113737  010122  002402        MOVB    2$,BDATA
3785  010410  012737  000001  002412  7$:   MOV     #1,REGNUM       ;SET REG NO. FOR PRINTOUT
3786  010416  032765  000001  000002        BIT     #RERCHK,2(R5)   ;SEE IF RCV ERROR BIT SHOULD BE IGNORED
3787  010424  001447                        BEQ     9$              ;BR IF YES
3788                                ;CHECK RERR BIT
3789  010426  132701  000200                BITB    #RERR,R1        ;SEE IF EXPECTED BIT = 1
3790  010432  001022                        BNE     8$              ;BR IF YES
3791  010434  132737  000200  010122        BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 0
3792  010442  001440                        BEQ     9$              ;BR IF YES
3793                                ;STACK "RERR NOT CLEARED" MSG
3794  010444                                GTDF    EM35,ERR12
3795                                                ;       QUEUE "DEVICE FATAL" ERROR # 29
3796  010444  012737  000001  002246                                    MOV     #T.EDF,ERRTYP
3797  010452  012737  000035  002250                                    MOV     #29,ERRNBR
3798  010460  012737  013621  002252                                    MOV     #EM35,ERRMSG
3799  010466  012737  017722  002254                                    MOV     #ERR12,ERRBLK
3800  010474  000137  011350                JMP     20$             ;TAKE ERROR EXIT
3801  010500  132737  000200  010122  8$:   BITB    #RERR,2$        ;SEE IF ACTUAL BIT = 1
3802  010506  001016                        BNE     9$              ;BR IF YES
3803                                ;STACK "RERR NOT SET" MSG
3804  010510                                GTDF    EM36,ERR12
3805                                                ;       QUEUE "DEVICE FATAL" ERROR # 30
3806  010510  012737  000001  002246                                    MOV     #T.EDF,ERRTYP
3807  010516  012737  000036  002250                                    MOV     #30,ERRNBR
3808  010524  012737  013642  002252                                    MOV     #EM36,ERRMSG
3809  010532  012737  017722  002254                                    MOV     #ERR12,ERRBLK
3810  010540  000137  011350                JMP     20$             ;TAKE ERROR EXIT
3811                                ;CHECK ROR BIT
3812  010544  132701  000010        9$:     BITB    #ROR,R1         ;SEE IF EXPECTED BIT = 1
3813  010550  001022                        BNE     10$             ;BR IF YES
3814  010552  132737  000010  010122        BITB    #ROR,2$         ;SEE IF ACTUAL BIT = 0
3815  010560  001440                        BEQ     11$             ;BR IF YES
3816                                ;STACK "ROR NOT CLEARED" MSG
3817  010562                                GTDF    EM16,ERR12
3818                                                ;       QUEUE "DEVICE FATAL" ERROR # 31
3819  010562  012737  000001  002246                                    MOV     #T.EDF,ERRTYP
3820  010570  012737  000037  002250                                    MOV     #31,ERRNBR
3821  010576  012737  013346  002252                                    MOV     #EM16,ERRMSG
3822  010604  012737  017722  002254                                    MOV     #ERR12,ERRBLK
3823  010612  000137  011350                JMP     20$             ;TAKE ERROR EXIT
```

```
 3824  010616  132737  000010  010122  10$:    BITB    #ROR,2$          ;SEE IF ACTUAL BIT = 1
 3825  010624  001016                           BNE     11$              ;BR IF YES
 3826                                 ;STACK "ROR NOT SET" MSG
 3827  010626                                   GTDF    EM14,ERR12
 3828                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 32
 3829  010626  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
 3830  010634  012737  000040  002250                                                    MOV     #32,ERRNBR
 3831  010642  012737  013332  002252                                                    MOV     #EM14,ERRMSG
 3832  010650  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
 3833  010656  000137  011350                   JMP     20$              ;TAKE ERROR EXIT
 3834                                 ;CHECK RABGA BIT
 3835  010662  132701  000004          11$:    BITB    #RABGA,R1        ;SEE IF EXPECTED BIT = 1
 3836  010666  001022                           BNE     12$              ;BR IF YES
 3837  010670  132737  000004  010122          BITB    #RABGA,2$        ;SEE IF ACTUAL BIT = 0
 3838  010676  001440                           BEQ     13$              ;BR IF YES
 3839                                 ;STACK "RABGA NOT CLEARED" MSG
 3840  010700                                   GTDF    EM39,ERR12
 3841                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 33
 3842  010700  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
 3843  010706  012737  000041  002250                                                    MOV     #33,ERR  R
 3844  010714  012737  013657  002252                                                    MOV     #EM39,ER MSG
 3845  010722  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
 3846  010730  000137  011350                   JMP     20$              ;TAKE ERROR EXIT
 3847  010734  132737  000004  010122  12$:    BITB    #RABGA,2$        ;SEE IF ACTUAL BIT = 1
 3848  010742  001016                           BNE     13$              ;BR IF YES
 3849                                 ;STACK "RABGA NOT SET" MSG
 3850  010744                                   GTDF    EM40,ERR12
 3851                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 34
 3852  010744  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
 3853  010752  012737  000042  002250                                                    MCV     #34,ERR  R
 3854  010760  012737  013701  002252                                                    MOV     #EM40,ER MSG
 3855  010766  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
 3856  010774  000137  011350                   JMP     20$              ;TAKE ERROR EXIT
 3857                                 ;CHECK REOM BIT
 3858  011000  132701  000002          13$:    BITB    #REOM,R1         ;SEE IF EXPECTED BIT = 1
 3859  011004  001022                           BNE     14$              ;BR IF YES
 3860  011006  132737  000002  010122          BITB    #REOM,2$         ;SEE IF ACTUAL BIT = 0
 3861  011014  001440                           BEQ     15$              ;BR IF YES
 3862                                 ;STACK "REOM NOT CLEARED" MSG
 3863  011016                                   GTDF    EM30,ERR12
 3864                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 35
 3865  011016  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
 3866  011024  012737  000043  002250                                                    MOV     #35,ERR  R
 3867  011032  012737  013535  002252                                                    MOV     #EM30,ER MSG
 3868  011040  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
 3869  011046  000137  011350                   JMP     20$              ;TAKE ERROR EXIT
 3870  011052  132737  000002  010122  14$:    BITB    #REOM,2$         ;SEE IF ACTUAL BIT = 1
 3871  011060  001016                           BNE     15$              ;BR IF YES
 3872                                 ;STACK "REOM NOT SET" MSG
 3873  011062                                   GTDF    EM31,ERR12
 3874                                                                    ;       QUEUE "DEVICE FATAL" ERROR # 36
 3875  011062  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
 3876  011070  012737  000044  002250                                                    MOV     #36,ERR  R
 3877  011076  012737  013556  002252                                                    MOV     #EM31,ER MSG
 3878  011104  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
 3879  011112  000137  011350                   JMP     20$              ;TAKE ERROR EXIT
```

```
3880                                    ;CHECK RSOM BIT
3881  011116  132701  000001            15$:    BITB    #RSOM,R1        ;SEE IF EXPECTED BIT = 1
3882  011122  001022                            BNE     16$             ;BR IF YES
3883  011124  132737  000001  010122            BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 0
3884  011132  001440                            BEQ     17$             ;BR IF YES
3885                                    ;STACK "RSOM NOT CLEARED" MSG
3886  011134                                    GTDF    EM28,ERR12
3887                                                    ;       QUEUE "DEVICE FATAL" ERROR # 37
3888  011134  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
3889  011142  012737  000045  002250                                                    MOV     #37,ERRNBR
3890  011150  012737  013477  002252                                                    MOV     #EM28,ERRMSG
3891  011156  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
3892  011164  000137  011350                    JMP     20$             ;TAKE ERROR EXIT
3893  011170  132737  000001  010122  16$:     BITB    #RSOM,2$        ;SEE IF ACTUAL BIT = 1
3894  011176  001016                            BNE     17$             ;BR IF YES
3895                                    ;STACK "RSOM NOT SET" MSG
3896  011200                                    GTDF    EM29,ERR12
3897                                                    ;       QUEUE "DEVICE FATAL" ERROR # 38
3898  011200  012737  000001  002246                                                    MOV     #T.EDF,ERRTYP
3899  011206  012737  000046  002250                                                    MOV     #38,ERRNBR
3900  011214  012737  013520  002252                                                    MOV     #EM29,ERRMSG
3901  011222  012737  017722  002254                                                    MOV     #ERR12,ERRBLK
3902  011230  000137  011350                    JMP     20$             ;TAKE ERROR EXIT
3903
3904  011234  116502  000004            17$:    MOVB    4(R5),R2        ;GET DESIRED NO. OF CYCLES
3905  011240  005001                            CLR     R1              ;INIT CYCLE COUNT
3906
3907  011242  136527  000005  000040  18$:     BITB    5(R5),#BIT5     ;* IS RXACT CHECK TO BE DISABLED ?
3908  011250  001004                            BNE     31$             ;* BR IF YES
3909  011252  004537  005676                    JSR     R5,CKRACT       ;CHK FOR RACT = 1
3910  011256  000001                            1
3911  011260  103433                            BCS     20$             ;BR TO EXIT IF ERROR
3912
3913  011262  020102                    31$:    CMP     R1,R2           ;SEE IF REQUIRED CYCLES DONE YET
3914  011264  001415                            BEQ     19$             ;BR IF YES
3915
3916  011266  136527  000005  000200            BITB    5(R5),#BIT7     ;* SEE IF INITIAL RDA CHECK DESIRED
3917  011274  001004                            BNE     22$             ;* BR IF NO
3918  011276  004537  006176                    JSR     R5,CKRDA        ;CHK FOR RDA = 0
3919  011302  000000                            0
3920  011304  103421                            BCS     20$             ;BR TO EXIT IF ERROR
3921
3922  011306  004537  011614            22$:    JSR     R5,STEPLU       ;CLOCK LU FOR 1 CYCLE
3923  011312  000001                            1
3924  011314  005201                            INC     R1              ;INCR CYCLE COUNT
3925  011316  000751                            BR      18$             ;CONTINUE CLOCKING
3926
3927  011320  136527  000005  000100  19$:     BITB    5(R5),#BIT6     ;* IS FINAL RDA CHECK TO BE SKIPPED ?
3928  011326  001004                            BNE     30$             ;* BR IF YES
3929  011330  004537  006176                    JSR     R5,CKRDA        ;CHK RDA = 1
3930  011334  000001                            1
3931  011336  103404                            BCS     20$             ;BR IF ERROR
3932
3933  011340  062705  000006            30$:    ADD     #6,R5           ;FIX UP RETURN ADRS
3934  011344  000241                            CLC                     ;SET C = 0 FOR NO ERROR
3935  011346  000403                            BR      21$             ;TAKE ERROR-FREE EXIT
```

```
 3936  011350  062705  000006          20$:    ADD     #6,R5           ;FIX UP RETURN ADDRESS
 3937  011354  000261                          SEC                     ;SET C BIT FOR ERROR
 3938  011356  012602                  21$:    MOV     (SP)+,R2        ;RESTORE R2
 3939  011360  012601                          MOV     (SP)+,R1        ;RESTORE R1
 3940  011362  000205                          RTS     R5              ;RETURN
```

```
                                .SBTTL ....RCV1ST -- RECEIVE FIRST CHARACTER OF MESSAGE
                                ;**********************************************************************
                                ;* RCV1ST - THIS SUBROUTINE RECEIVES THE FIRST CHAR OF A MESSAGE AND MONITURS
                                ;*      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)
```

```
                                .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>
                                ;**********************************************************************
011532                          ENDTRN:
011532  012537  011572                  MOV     (R5)+,2$        ;GET DESIRED NO. OF CYCLES TO CLOCK
011536  004537  005536                  JSR     R5,CKTACT       ;CHK FOR TXACT = 1
011542  000001                          1
011544  103422                          BCS     6$              ;BR IF ERROR
011546  004537  005676                  JSR     R5,CKRACT       ;CHK FOR RXACT = 1
011552  000001                          1
011554  103416                          BCS     6$
011556  004537  003734                  JSR     R5,WRITEI       ;CLEAR TXEN AND RXEN IN USYRT
011562  120000                          VIAORB                  ;** AND KEEP TTLOOP ENABLED **
011564  000002                          TTLOOP                  ;
011566  004537  011614                  JSR     R5,STEPLU       ;CLOCK LU FOR DESIRED NO. OF CYCLES
011572  000000                  2$:     .WORD   0
011574  004537  005536                  JSR     R5,CKTACT       ;CHK FOR TXACT = 0
011600  000000                          0
011602  103403                          BCS     6$              ;BR IF ERROR
011604  004537  005676                  JSR     R5,CKRACT       ;CHK FOR RXACT = 0
011610  000000                          0
011612  000205                  6$:     RTS     R5
```

```
                                .SBTTL ....STEPLU -- CLOCK THE USYRT N TIMES
                                ;**********************************************************************
                                ;* STEPLU - THIS SUBROUTINE CLOCKS THE LINE UNIT FOR THE NUMBER OF CYCLES
                                ;*      PASSED IN THE WORD FOLLOWING THE CALL. THE VIA ACR MUST BE PREVIOUSLY
                                ;*      SET UP FOR T1 ONE-SHOT MODE, AND THE T1 LATCHES MUST BE PREVIOUSLY SET
                                ;*      TO CONTROL THE WIDTH OF THE CLOCK PULSE. ALL THAT THIS SUBROUTINE
                                ;*      DOES IS TO LOAD 000 INTO THE HI BYTE OF THE T1 COUNTER, FOR THE
                                ;*      DESIRED NUMBER OF TIMES.
                                ;*
                                ;*      CALLING SEQUENCE :
                                ;*      JSR     R5,STEPLU
                                ;*      .WORD   <NUMBER OF CYCLES TO CLOCK>
                                ;**********************************************************************
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
```

```
4061                                    .SBTTL  GLOBAL ERROR REPORT SECTION
4062
4063                                    ;////////////////////////////////////////////////////////////////
4064                                    ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
4065                                    ;/      THAT ARE USED IN MORE THAN ONE TEST.
4066                                    ;////////////////////////////////////////////////////////////////
4067
4068                                    .NLIST  BEX
011640  047045  047045  000    ENDEMB: .ASCIZ  /<BELL><BELL>/
011645  045     000116         NEWLIN: .ASCIZ  /<BELL>/                ;USED TO TERMINATE ERROR MESSAGES

011650  047045  040445  040506 FMT2:   .ASCIZ  /<BELL><BELL>FAILING REG = <BELL>T<BELL>SEL<BELL>01/
011705  045     022516  020101 FMT3:   .ASCIZ  /<BELL><BELL> EXPECTED: <BELL>03<BELL>  ACTUAL: <BELL>03<BELL>  XOR: <BELL>03/
011771  045     022516  052101 FMT4:   .ASCIZ  /<BELL><BELL>THE CONTENTS OF ALL<BELL>T<BELL><BELL>T/
012027  045     022516  030523 FMT4A:  .ASCIZ  /<BELL><BELL>S1<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03/
012062  047045  052045  000    FMT4B:  .ASCIZ  /<BELL><BELL>T/
012067  045     022516  032523 FMT4C:  .ASCIZ  /<BELL><BELL>S5<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03/
012122  047045  040445  020040 FMT5:   .ASCIZ  /<BELL><BELL>A  WHEN <BELL>03<BELL>A LOADED INTO BSEL1/
012165  045     022516  020101 FMT5A:  .ASCIZ  /<BELL><BELL>A  ATTEMPTING "M-LOOP" FUNCTION CODE <BELL>02<BELL>A (<BELL>T<BELL>A)/

012252  047045  040445  020040 FMT10:  .ASCIZ  /<BELL><BELL>A  EXPECTED:<BELL>08<BELL>A  ACTUAL:<BELL>08<BELL>A  XOR:<BELL>08/
012326  040445  020040  051514 FMT10A: .ASCIZ  /<BELL>A  LSI ADDR:<BELL>08/
012347  045     022516  034117 FMT11:  .ASCIZ  /<BELL><BELL>08<BELL>08<BELL>08<BELL>08/
012366  040445  020040  042504 FMT15:  .ASCIZ  /<BELL>A  DETECTED IN <BELL>T<BELL>T<BELL>A --/
012420  040445  020040  042504 FMT15A: .ASCIZ  /<BELL>A  DETECTED @ TEST PATTERN ELEMENT # <BELL>D2/

012472  047045  040445  042524 FMT19:  .ASCIZ  /<BELL><BELL>ATEST <BELL>D2<BELL>A NOT RUN<BELL>/
012523  045     022524  033117 FMT21:  .ASCIZ  /<BELL>T<BELL>06<BELL>/
012533  045     022516  043101 FMT22:  .ASCIZ  /<BELL><BELL>FAILING REG: /
012555  045     042501  050130 FMT23:  .ASCIZ  /<BELL>EXPECTED: <BELL>03<BELL>S5<BELL>ACTUAL: <BELL>03<BELL>S5<BELL>XOR: <BELL>03<BELL>/
012634  047045  052045  047045 FMT24:  .ASCIZ  /<BELL><BELL>T<BELL><BELL>T<BELL>/
012647  045     031517  051445 FMT25:  .ASCIZ  /<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03<BELL>/
012677  045     032123  047445 FMT26:  .ASCIZ  /<BELL>S4<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03<BELL>S5<BELL>03<BELL>/
012732  052045  052045  047045 FMT27:  .ASCIZ  /<BELL>T<BELL>T<BELL>/
012741  045     022524  000116 FMT29:  .ASCIZ  /<BELL>T<BELL>/
012746  047045  040445  047506 FMT30:  .ASCIZ  /<BELL><BELL>FOR BAUD RATE SPECIFIED./
013003  045     022516  044501 FMT31:  .ASCIZ  /<BELL><BELL>AIMPROPER CONNECTOR TYPE SPECIFIED/
013051  045     022516  043101 FMT32:  .ASCIZ  /<BELL><BELL>FOR 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  /REOI 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  EM136:  .ASCIZ  /RERR NOT SET/
013657     122  041101  040507  EM139:  .ASCIZ  /RABGA NOT CLEARED/
013701     122  041101  040507  EM140:  .ASCIZ  /RABGA NOT SET/
013717     124  020130  047105  EM154:  .ASCIZ  /TX UNDERRUN ERROR/
013741     122  043505  046040  EM166:  .ASCIZ  /REG MISCOMPARE/
013760  051525  051131  020124  EM168:  .ASCIZ  /USYRT STATUS INCORRECT/
014007     124  040530  052103  EM169:  .ASCIZ  /TXACT NOT SET/
014025     124  040530  052103  EM170:  .ASCIZ  /TXACT NOT CLEARED/
014047     122  040530  052103  EM171:  .ASCIZ  /RXACT NOT SET/
014065     122  040530  052103  EM172:  .ASCIZ  /RXACT NOT CLEARED/
014107     124  046502  020124  EM173:  .ASCIZ  /TBMT NOT SET/
014124  041124  052115  047040  EM174:  .ASCIZ  /TBMT NOT CLEARED/
014145     122  040504  047040  EM175:  .ASCIZ  /RDA NOT SET/
014161     122  040504  047040  EM176:  .ASCIZ  /RDA NOT CLEARED/
014201     122  040523  047040  EM177:  .ASCIZ  /RSA NOT SET/
014215     122  040523  047040  EM178:  .ASCIZ  /RSA NOT CLEARED/
014235     122  041505  044505  EM190:  .ASCIZ  /RECEIVER OVERRUN NOT SET/
014266  042522  042503  053111  EM191:  .ASCIZ  /RECEIVER OVERRUN NOT CLEARED/
014373     124  046502  020124  EM192:  .ASCIZ  /TBMT INTERRUPT TEST FAILURE/
014457     124  051105  020122  EM198:  .ASCIZ  /TERR BIT NOT CLEARED/
014404  042524  051122  041040  EM199:  .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
4069                            .EVEN
4070
4071
```

```
                                        .SBTTL ....TEXT ADDRESS TABLES FOR ERROR HANDLERS -- "TXT__T"
                                        ;-----------------------------------------------------------------------
                                        ;----------- TEXT ADDRESS TABLES USED BY ERROR HANDLERS -----------------
                                        ;-----------------------------------------------------------------------

016702  016113  016117  016133  TXTMLT: .WORD   TXTML0,TXTML1,TXTML2,TXTML3,TXTML4,TXTML5,TXTML6,TXTML7
016710  016150  016172  016213
016716  016243  016255

016722  016312                          .WORD   TXTVR
016724  016330  016334  016340  TXTVRT: .WORD   TXTVR0,TXTVR1,TXTVR2,TXTVR3,TXTVR4,TXTVR5,TXTVR6,TXTVR7
016732  016345  016352  016357
016740  016364  016371
016744  016376  016403  016410          .WORD   TXTVR8,TXTVR9,TXTVRA,TXTVRB,TXTVRC,TXTVRD,TXTVRE,TXTVRF
016752  016413  016417  016423
016760  016427  016433

016764  016437                          .WORD   TXTNP
016766  016444  016454  016464  TXTNPT: .WORD   TXTNP0,TXTNP1,TXTNP2,TXTNP3,TXTNP4,TXTNP5,TXTNP6,TXTNP7,TXTNP8
016774  016474  016511  016526
017002  016543  016557  016573
017010  016622  016630  016636  TXTURT: .WORD   TXTUR0,TXTUR1,TXTUR2,TXTUR3,TXTUR4,TXTUR5,TXTUR6,TXTUR7
017016  016644  016652  016661
017024  016670  016674
```

```
                        ;-------------------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR4 -- M-LOOP TIMEOUT ERROR HANDLING
                        ;-------------------------------------------------------------------------------
                                BGNMSG  ERR4
                                                                                        ERR4::
017030  105037  002401          CLRB    GDATA+1                 ;MAKE SURE BIT 8 DOESN'T PRINT!
017034  010146                  MOV     R1,-(SP)                ;SAVE THE WORKING REGISTER
017036  013701  002400          MOV     GDATA,R1                ;SAVE THIS FOR LATER
017042  022701  000017          CMP     #17,R1                  ;WAS THIS AN M-LOOP REQUEST?
017046  002012                  BGE     5$                      ;YES, THEN REPORT THE FUNCTION CODE
017050                          PRINTX  #FMT5,R1                ;NO,  THEN IT MUST BE A BSEL1 SETTING
017050  010146                                                                  MOV     R1,-(SP)
017052  012746  012122                                                          MOV     #FMT5,-(SP)
017056  012746  000002                                                          MOV     #2,-(SP)
017062  010600                                                                  MOV     SP,R0
017064  104415                                                                  TRAP    C$PNTX
017066  062706  000006                                                          ADD     #6,SP
017072  000424                  BR      20$
                        
017074  001001          5$:     BNE     6$                      ;IF IT WAS A 17, THIS IS A "NOP" AND
017076  005001                  CLR     R1                      ;  THE TEXT POINTER MUST SO REFLECT.
017100  022701  000007  6$:     CMP     #7,R1                   ;IS FUNCTION CODE > 7?
017104  002002                  BGE     7$                      ;NO,  THAN WE CAN HANDLE IT
017106  012701  000006          MOV     #6,R1                   ;YES, THAN IT'S UNDEFINED -- SAY SO
017112  006301          7$:     ASL     R1                      ;CONVERT TO A WORD OFFSET
017114                          PRINTX  #FMT5A,GDATA,TXTMLT(R1) ;REPORT THE FAILING FUNCTION
017114  016146  016702                                                          MOV     TXTMLT(R1),-(SP)
017120  013746  002400                                                          MOV     GDATA,-(SP)
017124  012746  012165                                                          MOV     #FMT5A,-(SP)
017130  012746  000003                                                          MOV     #3,-(SP)
017134  010600                                                                  MOV     SP,R0
017136  104415                                                                  TRAP    C$PNTX
017140  062706  000010                                                          ADD     #10,SP
                        
017144  012601  20$:     MOV     (SP)+,R1                ;RESTORE THE WORKING REGISTER
017146  004737  020160          JSR     PC,ERR5$                ;DUMP THE SELECT REGISTERS
017152                          ENDMSG
017152                                                                  L10002:
017152  104423                                                                  TRAP    C$MSG
                        ;-------------------------------------------------------------------------------
                        .SBTTL ....ERROR HANDLER -- ERR5 -- WORD SELECT REG. ERRORS
                        ;-------------------------------------------------------------------------------
017154                          BGNMSG  ERR5
017154                                                                  ERR5::
017154                          PRINTB  #FMT2,#TXT5,REGNUM
017154  013746  002412                                                          MOV     REGNUM,-(SP)
017160  012746  015074                                                          MOV     #TXT5,-(SP)
017164  012746  011650                                                          MOV     #FMT2,-(SP)
017170  012746  000003                                                          MOV     #3,-(SP)
017174  010600                                                                  MOV     SP,R0
017176  104414                                                                  TRAP    C$PNTB
017200  062706  000010                                                          ADD     #10,SP
017204  004737  020134          JSR     PC,XORGB
017210                          PRINTB  #FMT10,GDATA,BDATA,XDATA
017210  013746  002404                                                          MOV     XDATA,-(SP)
017214  013746  002402                                                          MOV     BDATA,-(SP)
```

```
4154  017220  013746  002400                                                  MOV     GDATA,-(SP)
4155  017224  012746  012252                                                  MOV     #FMT10,-(SP)
4156  017230  012746  000004                                                  MOV     #4,-(SP)
4157  017234  010600                                                          MOV     SP,R0
4158  017236  104414                                                          TRAP    C$PNTB
4159  017240  062706  000012                                                  ADD     #12,SP
4160  017244  004737  020160          JSR     PC,ERR5$                ;DUMP THE SELECT REGISTERS
4161  017250                          ENDMSG
4162  017250                                                        L10003:
4163  017250  104423                                                          TRAP    C$MSG
4164                                  ;-------------------------------------------------------------------------
4165                                  .SBTTL ....ERROR HANDLER -- ERR7A -- USYRT REGISTER ERRORS
4166                                  ;-------------------------------------------------------------------------
4167  017252                          BGNMSG  ERR7A
4168  017252                                                        ERR7A::
4169  017252  113701  002412          MOVB    REGNUM,R1
4170  017256  006301                  ASL     R1              ;AS PASSED, THIS WAS A BYTE OFFSET
4171  017260                          PRINTB  #FMT15,#TXTUR,TXTURT(R1)
4172  017260  016146  017010                                                  MOV     TXTURT(R1),-(SP)
4173  017264  012746  016607                                                  MOV     #TXTUR,-(SP)
4174  017270  012746  012366                                                  MOV     #FMT15,-(SP)
4175  017274  012746  000003                                                  MOV     #3,-(SP)
4176  017300  010600                                                          MOV     SP,R0
4177  017302  104414                                                          TRAP    C$PNTB
4178  017304  062706  000010                                                  ADD     #10,SP
4179  017310  004737  020134          JSR     PC,XORGB
4180  017314                          PRINTB  #FMT3,GDATA,BDATA,XDATA
4181  017314  013746  002404                                                  MOV     XDATA,-(SP)
4182  017320  013746  002402                                                  MOV     BDATA,-(SP)
4183  017324  013746  002400                                                  MOV     GDATA,-(SP)
4184  017330  012746  011705                                                  MOV     #FMT3,-(SP)
4185  017334  012746  000004                                                  MOV     #4,-(SP)
4186  017340  010600                                                          MOV     SP,R0
4187  017342  104414                                                          TRAP    C$PNTB
4188  017344  062706  000012                                                  ADD     #12,SP
4189  017350                          PRINTB  #ENDEMB
4190  017350  012746  011640                                                  MOV     #ENDEMB,-(SP)
4191  017354  012746  000001                                                  MOV     #1,-(SP)
4192  017360  010600                                                          MOV     SP,R0
4193  017362  104414                                                          TRAP    C$PNTB
4194  017364  062706  000004                                                  ADD     #4,SP
4195  017370                          ENDMSG
4196  017370                                                        L10004:
4197  017370  104423                                                          TRAP    C$MSG
4198                                  ;-------------------------------------------------------------------------
4199                                  .SBTTL ....ERROR HANDLER -- ERR10 -- USYRT REG ERROR (XOR, REG PRINTOUT)
4200                                  ;-------------------------------------------------------------------------
4201
4202  017372                          BGNMSG  ERR10
4203  017372                                                        ERR10::
4204  017372                          PRINTB  #FMT21,#TXT12,MPCSR
4205  017372  013746  002472                                                  MOV     MPCSR,-(SP)
4206  017376  012746  015541                                                  MOV     #TXT12,-(SP)
4207  017402  012746  012523                                                  MOV     #FMT21,-(SP)
4208  017406  012746  000003                                                  MOV     #3,-(SP)
4209  017412  010600                                                          MOV     SP,R0
```

```
4210  017414  104414                                                          TRAP    C$PNTB
4211  017416  062706  000010                                                  ADD     #10,SP
4212  017422                        PRINTB  #FMT22
4213  017422  012746  012533                                                  MOV     #FMT22,-(SP)
4214  017426  012746  000001                                                  MOV     #1,-(SP)
4215  017432  010600                                                          MOV     SP,R0
4216  017434  104414                                                          TRAP    C$PNTB
4217  017436  062706  000004                                                  ADD     #4,SP
4218  017442  013701  002412              MOV     REGNUM,R1
4219  017446  006301                      ASL     R1                      ;GET PTR TO USYRT REG ASCII
4220  017450                        PRINTB  #FMT27,#TXTUR,TXTURT(R1)
4221  017450  016146  017010                                                  MOV     TXTURT(R1),-(SP)
4222  017454  012746  016607                                                  MOV     #TXTUR,-(SP)
4223  017460  012746  012732                                                  MOV     #FMT27,-(SP)
4224  017464  012746  000003                                                  MOV     #3,-(SP)
4225  017470  010600                                                          MOV     SP,R0
4226  017472  104414                                                          TRAP    C$PNTB
4227  017474  062706  000010                                                  ADD     #10,SP
4228  017500  004737  020134              JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
4229  017504                        PRINTB  #FMT23,GDATA,BDATA,XDATA
4230  017504  013746  002404                                                  MOV     XDATA,-(SP)
4231  017510  013746  002402                                                  MOV     BDATA,-(SP)
4232  017514  013746  002400                                                  MOV     GDATA,-(SP)
4233  017520  012746  012555                                                  MOV     #FMT23,-(SP)
4234  017524  012746  000004                                                  MOV     #4,-(SP)
4235  017530  010600                                                          MOV     SP,R0
4236  017532  104414                                                          TRAP    C$PNTB
4237  017534  062706  000012                                                  ADD     #12,SP
4238  017540  004737  020710              JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
4239  017544                        ENDMSG
4240  017544                                                          L10005:
4241  017544  104423                                                          TRAP    C$MSG
4242
4243                        ;-------------------------------------------------------------------------
4244                        .SBTTL ....ERROR HANDLER -- ERR11 -- VIA REG ERROR (XOR, REG PRINTOUT)
4245                        ;-------------------------------------------------------------------------
4246
4247  017546                        BGNMSG  ERR11
4248  017546                                                          ERR11::
4249  017546                        PRINTB  #FMT21,#TXT12,MPCSR
4250  017546  013746  002472                                                  MOV     MPCSR,-(SP)
4251  017552  012746  015541                                                  MOV     #TXT12,-(SP)
4252  017556  012746  012523                                                  MOV     #FMT21,-(SP)
4253  017562  012746  000003                                                  MOV     #3,-(SP)
4254  017566  010600                                                          MOV     SP,R0
4255  017570  104414                                                          TRAP    C$PNTB
4256  017572  062706  000010                                                  ADD     #10,SP
4257  017576                        PRINTB  #FMT22
4258  017576  012746  012533                                                  MOV     #FMT22,-(SP)
4259  017602  012746  000001                                                  MOV     #1,-(SP)
4260  017606  010600                                                          MOV     SP,R0
4261  017610  104414                                                          TRAP    C$PNTB
4262  017612  062706  000004                                                  ADD     #4,SP
4263  017616  013701  002412              MOV     REGNUM,R1
4264  017622  006301                      ASL     R1                      ;GET PTR TO VIA REG ASCII
4265  017624                        PRINTB  #FMT27,#TXTVR,TXTVRT(R1)
```

```
4266  017624  016146  016724                                                  MOV     TXTVRT(R1),-(SP)
4267  017630  012746  016312                                                  MOV     #TXTVR,-(SP)
4268  017634  012746  012732                                                  MOV     #FMT27,-(SP)
4269  017640  012746  000003                                                  MOV     #3,-(SP)
4270  017644  010600                                                          MOV     SP,R0
4271  017646  104414                                                          TRAP    C$PNTB
4272  017650  062706  000010                                                  ADD     #10,SP
4273  017654  004737  020134          JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
4274  017660                          PRINTB  #FMT23,GDATA,BDATA,XDATA
4275  017660  013746  002404                                                  MOV     XDATA,-(SP)
4276  017664  013746  002402                                                  MOV     BDATA,-(SP)
4277  017670  013746  002400                                                  MOV     GDATA,-(SP)
4278  017674  012746  012555                                                  MOV     #FMT23,-(SP)
4279  017700  012746  000004                                                  MOV     #4,-(SP)
4280  017704  010600                                                          MOV     SP,R0
4281  017706  104414                                                          TRAP    C$PNTB
4282  017710  062706  000012                                                  ADD     #12,SP
4283  017714  004737  020356          JSR     PC,ERR11$               ;GET & PRINT VIA REGISTERS
4284  017720                          ENDMSG
4285  017720                                                          L10006:
4286  017720  104423                                                          TRAP    C$MSG
4287
4288
4289                          ;-------------------------------------------------------------------
4290                          .SBTTL ....ERROR HANDLER -- ERR12 -- USYRT REG ERROR (USYRT PRINTOUT)
4291                          ;-------------------------------------------------------------------
4292  017722                  BGNMSG  ERR12
4293  017722                                                          ERR12::
4294  017722                  PRINTB  #FMT21,#TXT12,MPCSR
4295  017722  013746  002472                                                  MOV     MPCSR,-(SP)
4296  017726  012746  015541                                                  MOV     #TXT12,-(SP)
4297  017732  012746  012523                                                  MOV     #FMT21,-(SP)
4298  017736  012746  000003                                                  MOV     #3,-(SP)
4299  017742  010600                                                          MOV     SP,R0
4300  017744  104414                                                          TRAP    C$PNTB
4301  017746  062706  000010                                                  ADD     #10,SP
4302  017752                  PRINTB  #FMT22
4303  017752  012746  012533                                                  MOV     #FMT22,-(SP)
4304  017756  012746  000001                                                  MOV     #1,-(SP)
4305  017762  010600                                                          MOV     SP,R0
4306  017764  104414                                                          TRAP    C$PNTB
4307  017766  062706  000004                                                  ADD     #4,SP
4308  017772  013701  002412          MOV     REGNUM,R1
4309  017776  006301                  ASL     R1                      ;GET PTR TO USYRT REG ASCII
4310  020000                          PRINTB  #FMT27,#TXTUR,TXTURT(R1)
4311  020000  016146  017010                                                  MOV     TXTURT(R1),-(SP)
4312  020004  012746  016607                                                  MOV     #TXTUR,-(SP)
4313  020010  012746  012732                                                  MOV     #FMT27,-(SP)
4314  020014  012746  000003                                                  MOV     #3,-(SP)
4315  020020  010600                                                          MOV     SP,R0
4316  020022  104414                                                          TRAP    C$PNTB
4317  020024  062706  000010                                                  ADD     #10,SP
4318  020030  004737  020710          JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
4319  020034                          ENDMSG
4320  020034                                                          L10007:
4321  020034  104423                                                          TRAP    C$MSG
```

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

                                BGNMSG  ERR13
                                                                                ERR13::
                                PRINTB  #FMT21,#TXT12,MPCSR
020036  013746  002472                                                          MOV     MPCSR,-(SP)
020042  012746  015541                                                          MOV     #TXT12,-(SP)
020046  012746  012523                                                          MOV     #FMT21,-(SP)
020052  012746  000003                                                          MOV     #3.-(SP)
020056  010600                                                                  MOV     SP,R0
020060  104414                                                                  TRAP    C$PNTB
020062  062706  000010                                                          ADD     #10,SP
020066  004737  020134                JSR     PC,XORGB                ;COMPUTE XOR OF GOOD AND BAD DATA
020072                                  PRINTB  #FMT23,GDATA,BDATA,XDATA
020072  013746  002404                                                          MOV     XDATA,-(SP)
020076  013746  002402                                                          MOV     BDATA,-(SP)
020102  013746  002400                                                          MOV     GDATA,-(SP)
020106  012746  012555                                                          MOV     #FMT23,-(SP)
020112  012746  000004                                                          MOV     #4.-(SP)
020116  010600                                                                  MOV     SP,R0
020120  104414                                                                  TRAP    C$PNTB
020122  062706  000012                                                          ADD     #12,SP
020126  004737  020710                JSR     PC,ERR12$               ;GET & PRINT USYRT REGISTERS
020132                                  ENDMSG
020132                                                                  L10010:
020132  104423                                                                  TRAP    C$MSG

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

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

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

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$:
020160                          PRINTX  #FMT4,#TXT6,#TXT4
020160  012746  014773                                                          MOV     #TXT4,(SP)
020164  012746  015076                                                          MOV     #TXT6,-(SP)
```

```
4378  020170  012746  011771                                                    MOV     #FMT4,-(SP)
4379  020174  012746  000003                                                    MOV     #3,-(SP)
4380  020200  010600                                                           MOV     SP,R0
4381  020202  104415                                                           TRAP    C$PNTX
4382  020204  062706  000010                                                   ADD     #10,SP
4383  020210                        PRINTX  #FMT11,WSR0,WSR2,WSR4,WSR6 ;DUMP THE SELECT REGISTERS
4384  020210  013746  002264                                                    MOV     WSR6,-(SP)
4385  020214  013746  002262                                                    MOV     WSR4,-(SP)
4386  020220  013746  002260                                                    MOV     WSR2,-(SP)
4387  020224  013746  002256                                                    MOV     WSR0,-(SP)
4388  020230  012746  012347                                                    MOV     #FMT11,-(SP)
4389  020234  012746  000005                                                    MOV     #5,-(SP)
4390  020240  010600                                                           MOV     SP,R0
4391  020242  104415                                                           TRAP    C$PNTX
4392  020244  062706  000014                                                   ADD     #14,SP
4393  020250                        PRINTX  #FMT4B,#TXT4A
4394  020250  012746  015033                                                    MOV     #TXT4A,-(SP)
4395  020254  012746  012062                                                    MOV     #FMT4B,-(SP)
4396  020260  012746  000002                                                    MOV     #2,-(SP)
4397  020264  010600                                                           MOV     SP,R0
4398  020266  104415                                                           TRAP    C$PNTX
4399  020270  062706  000006                                                   ADD     #6,SP
4400  020274                        PRINTX  #FMT11,WSR10,WSR12,WSR14,WSR16 ;DUMP THE SELECT REGISTERS
4401  020274  013746  002274                                                    MOV     WSR16,-(SP)
4402  020300  013746  002272                                                    MOV     WSR14,-(SP)
4403  020304  013746  002270                                                    MOV     WSR12,-(SP)
4404  020310  013746  002266                                                    MOV     WSR10,-(SP)
4405  020314  012746  012347                                                    MOV     #FMT11,-(SP)
4406  020320  012746  000005                                                    MOV     #5,-(SP)
4407  020324  010600                                                           MOV     SP,R0
4408  020326  104415                                                           TRAP    C$PNTX
4409  020330  062706  000014                                                   ADD     #14,SP
4410  020334                        PRINTB  #ENDEMB
4411  020334  012746  011640                                                    MOV     #ENDEMB,-(SP)
4412  020340  012746  000001                                                    MOV     #1,-(SP)
4413  020344  010600                                                           MOV     SP,R0
4414  020346  104414                                                           TRAP    C$PNTB
4415  020350  062706  000004                                                   ADD     #4,SP
4416  020354  000207                RTS     PC
4417
4418                          ;-------------------------------------------------------------------
4419                          .SBTTL .....ERROR HANDLER SUBROUTINE -- ERR11$
4420                          ;-------------------------------------------------------------------
4421                          ;       COMMON ERROR SUBROUTINE TO GET/PRINT VIA REGISTERS
4422
4423  020356  004737  004502  ERR11$: JSR     PC,GETVRS               ;GET VIA REGS FOR PRINTOUT
4424  020362                        PRINTX  #FMT24,#TXT16,#TXT17
4425  020362  012746  015717                                                    MOV     #TXT17,-(SP)
4426  020366  012746  015704                                                    MOV     #TXT16,-(SP)
4427  020372  012746  012634                                                    MOV     #FMT24,-(SP)
4428  020376  012746  000003                                                    MOV     #3,-(SP)
4429  020402  010600                                                           MOV     SP,R0
4430  020404  104415                                                           TRAP    C$PNTX
4431  020406  062706  000010                                                   ADD     #10,SP
4432  020412                        PRINTX  #FMT25,VREGS+0,VREGS+2,VREGS+4,VREGS+6
4433  020412  013746  002344                                                    MOV     VREGS+6,-(SP)
```

```
 4434  020416  013746  002342                                                  MOV     VREGS+4,-(SP)
 4435  020422  013746  002340                                                  MOV     VREGS+2,-(SP)
 4436  020426  013746  002336                                                  MOV     VREGS+0,-(SP)
 4437  020432  012746  012647                                                  MOV     #FMT25,-(SP)
 4438  020436  012746  000005                                                  MOV     #5,-(SP)
 4439  020442  010600                                                          MOV     SP,R0
 4440  020444  104415                                                          TRAP    C$PNTX
 4441  020446  062706  000014                                                  ADD     #14,SP
 4442  020452                      PRINTX  #FMT29  #TXT18
 4443  020452  012746  015754                                                  MOV     #TXT18,-(SP)
 4444  020456  012746  012741                                                  MOV     #FMT29,-(SP)
 4445  020462  012746  000002                                                  MOV     #2,-(SP)
 4446  020466  010600                                                          MOV     SP,R0
 4447  020470  104415                                                          TRAP    C$PNTX
 4448  020472  062706  000006                                                  ADD     #6,SP
 4449  020476                      PRINTX  #FMT26,VREGS+8.,VREGS+10.,VREGS+12.,VREGS+14.
 4450  020476  013746  002354                                                  MOV     VREGS+14.,-(SP)
 4451  020502  013746  002352                                                  MOV     VREGS+12.,-(SP)
 4452  020506  013746  002350                                                  MOV     VREGS+10.,-(SP)
 4453  020512  013746  002346                                                  MOV     VREGS+8.,-(SP)
 4454  020516  012746  012677                                                  MOV     #FMT26,-(SP)
 4455  020522  012746  000005                                                  MOV     #5,-(SP)
 4456  020526  010600                                                          MOV     SP,R0
 4457  020530  104415                                                          TRAP    C$PNTX
 4458  020532  062706  000014                                                  ADD     #14,SP
 4459  020536                      PRINTX  #FMT29,#TXT19
 4460  020536  012746  016015                                                  MOV     #TXT19,-(SP)
 4461  020542  012746  012741                                                  MOV     #FMT29,-(SP)
 4462  020546  012746  000002                                                  MOV     #2,-(SP)
 4463  020552  010600                                                          MOV     SP,R0
 4464  020554  104415                                                          TRAP    C$PNTX
 4465  020556  062706  000006                                                  ADD     #6,SP
 4466  020562                      PRINTX  #FMT25,VREGS+16.,VREGS+18.,VREGS+20.,VREGS+22.
 4467  020562  013746  002364                                                  MOV     VREGS+22.,-(SP)
 4468  020566  013746  002362                                                  MOV     VREGS+20.,-(SP)
 4469  020572  013746  002360                                                  MOV     VREGS+18.,-(SP)
 4470  020576  013746  002356                                                  MOV     VREGS+16.,-(SP)
 4471  020602  012746  012647                                                  MOV     #FMT25,-(SP)
 4472  020606  012746  000005                                                  MOV     #5,-(SP)
 4473  020612  010600                                                          MOV     SP,R0
 4474  020614  104415                                                          TRAP    C$PNTX
 4475  020616  062706  000014                                                  ADD     #14,SP
 4476  020622                      PRINTX  #FMT29,#TXT20
 4477  020622  012746  016051                                                  MOV     #TXT20,-(SP)
 4478  020626  012746  012741                                                  MOV     #FMT29,-(SP)
 4479  020632  012746  000002                                                  MOV     #2,-(SP)
 4480  020636  010600                                                          MOV     SP,R0
 4481  020640  104415                                                          TRAP    C$PNTX
 4482  020642  062706  000006                                                  ADD     #6,SP
 4483  020646                      PRINTX  #FMT26,VREGS+24.,VREGS+26.,VREGS+28.,VREGS+30.
 4484  020646  013746  002374                                                  MOV     VREGS+30.,-(SP)
 4485  020652  013746  002372                                                  MOV     VREGS+28.,-(SP)
 4486  020656  013746  002370                                                  MOV     VREGS+26.,-(SP)
 4487  020662  013746  002366                                                  MOV     VREGS+24.,-(SP)
 4488  020666  012746  012677                                                  MOV     #FMT26,-(SP)
 4489  020672  012746  000005                                                  MOV     #5,-(SP)
```

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

020676  010600                                                          MOV     SP,R0
020700  104415                                                          TRAP    C$PNTX
020702  062706  000014                                                  ADD     #14,SP
020706  000207                  RTS     PC

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

020710  004737  004402  ERR12$: JSR     PC,GETURS               ;GET USYRT REGS FOR PRINTOUT
020714                          PRINTX  #FMT24,#TXT13,#TXT14
020714  012746  015604                                                  MOV     #TXT14,-(SP)
020720  012746  015567                                                  MOV     #TXT13,-(SP)
020724  012746  012634                                                  MOV     #FMT24,-(SP)
020730  012746  000003                                                  MOV     #3,-(SP)
020734  010600                                                          MOV     SP,R0
020736  104415                                                          TRAP    C$PNTX
020740  062706  000010                                                  ADD     #10,SP
020744                          PRINTX  #FMT25,UREGS+0,UREGS+2,UREGS+4,UREGS+6
020744  013746  002324                                                  MOV     UREGS+6,-(SP)
020750  013746  002322                                                  MOV     UREGS+4,-(SP)
020754  013746  002320                                                  MOV     UREGS+2,-(SP)
020760  013746  002316                                                  MOV     UREGS+0,-(SP)
020764  012746  012647                                                  MOV     #FMT25,-(SP)
020770  012746  000005                                                  MOV     #5,-(SP)
020774  010600                                                          MOV     SP,R0
020776  104415                                                          TRAP    C$PNTX
021000  062706  000014                                                  ADD     #14,SP
021004                          PRINTX  #FMT29,#TXT15
021004  012746  015642                                                  MOV     #TXT15,-(SP)
021010  012746  012741                                                  MOV     #FMT29,-(SP)
021014  012746  000002                                                  MOV     #2,-(SP)
021020  010600                                                          MOV     SP,R0
021022  104415                                                          TRAP    C$PNTX
021024  062706  000006                                                  ADD     #6,SP
021030                          PRINTX  #FMT26,UREGS+10,UREGS+12,UREGS+14,UREGS+16
021030  013746  002334                                                  MOV     UREGS+16,-(SP)
021034  013746  002332                                                  MOV     UREGS+14,-(SP)
021040  013746  002330                                                  MOV     UREGS+12,-(SP)
021044  013746  002326                                                  MOV     UREGS+10,-(SP)
021050  012746  012677                                                  MOV     #FMT26,-(SP)
021054  012746  000005                                                  MOV     #5,-(SP)
021060  010600                                                          MOV     SP,R0
021062  104415                                                          TRAP    C$PNTX
021064  062706  000014                                                  ADD     #14,SP
021070  000207                  RTS     PC
```

```
                                .SBTTL  LOAD DEVICE PROTECTION TABLE

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

021072                                  BGNPROT
021072                                                                  L$PROT::
021072  177777                        .WORD   -1      ;DON'T CHK CSR ADRS
021074  177777                        .WORD   -1      ;DON'T CHK MASSBUS UNIT NO.
021076  177777                        .WORD   -1      ;DON'T CHK DRIVE NO.
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  103473                                                                  BCS     ENDIT
021232  000414                          BR      GETPRM

021234                          STARST:
```

```
 4608  021234  005037  002466                  CLR     STARES          ;CLEAR FLAG TO SHOW JUST HAD STA OR RES
 4609
 4610                                          ;CLEAR DEVICE MAP
 4611  021240  005037  002460                  CLR     DEVMAP
 4612  021244                                  NEWST:
 4613  021244  012737  177777  002410          MOV     #-1,LOGDEV      ;RESET LOGICAL DEVICE TO -1
 4614  021252  005237  002466                  INC     STARES          ;INCREMENT NO. OF PASSES SINCE STA OR RES
 4615  021256  012737  000001  002462          MOV     #BIT0,DEVPTR    ;INIT DEVICE MAP BIT POINTER
 4616
 4617                                          ; GET UNIBUS ADDRESS, VECTOR, PRIORITY LEVEL, SWITCH PACKS, TEST
 4618                                          ;   CONNECTOR INFORMATION FOR THIS LOGICAL DEVICE
 4619  021264                                  GETPRM:
 4620  021264  005237  002410                  INC     LOGDEV          ;INCREMENT LOGICAL DEVICE NUMBER
 4621  021270                                  GPHARD  LOGDEV,R1       ;GET P-TABLE POINTER INTO R1
 4622  021270  013700  002410                                                          MOV     LOGDEV,R0
 4623  021274  104442                                                                  TRAP    C$GPHRD
 4624  021276  010001                                                                  MOV     R0,R1
 4625  021300                                  BCOMPLETE       10$     ;BR IF DEVICE AVAILABLE
 4626  021300  103403                                                                  BCS     10$
 4627  021302  006337  002462                  ASL     DEVPTR          ;SHIFT DEVICE POINTER
 4628  021306  000766                          BR      GETPRM          ;SKIP THIS DEVICE
 4629  021310  053737  002462  002460  10$:    BIS     DEVPTR,DEVMAP   ;SET BIT FOR THIS DEVICE
 4630  021316  006337  002462                  ASL     DEVPTR          ;SHIFT BIT POINTER
 4631
 4632  021322  012102                          MOV     (R1)+,R2        ;R2=CSR ADDR VALUE
 4633  021324  012703  002472                  MOV     #MPCSR,R3       ;R3=POINTER TO CSR ADDR STORAGE AREA
 4634
 4635  021330  010223                  11$:    MOV     R2,(R3)+        ;PUT CSR ADDRESSES IN 'BSEL' AREA
 4636  021332  005202                          INC     R2              ;BUMP BSEL ADDR
 4637  021334  022703  002532                  CMP     #BSEL17+2,R3    ;ALL 16 ADDRESSES MOVED ?
 4638  021340  001373                          BNE     11$             ;NO: DO ANOTHER ADDRESS
 4639                                                                  ;YES: CONTINUE
 4640
 4641  021342  011137  002532                  MOV     (R1),MPIVEC     ;GET DMV11 INPUT INTRPT VECTOR
 4642  021346  012137  002534                  MOV     (R1)+,MPOVEC
 4643  021352  062737  000004  002534          ADD     #4,MPOVEC       ;GET DMV11 OUTPUT INTRPT VECTOR
 4644  021360  012137  002536                  MOV     (R1)+,MPRIOR    ;GET DMV11 DEVICE PRIORITY
 4645  021364  012137  002540                  MOV     (R1)+,LUSWI1    ;GET LU SWITCH PACK #1
 4646  021370  012137  002542                  MOV     (R1)+,LUSWI2    ;GET LU SWITCH PACK #2
 4647  021374  012137  002544                  MOV     (R1)+,BRDTYP    ;GET DMV-11 BOARD TYPE
 4648  021400  012137  002546                  MOV     (R1)+,TSTCON    ;GET TEST CONNECTOR INDICATOR
 4649  021404  011137  002550                  MOV     (R1),BDRATE     ;GET BAUD RATE FOR THIS DEVICE
 4650                                  ;ISSUE LSI BUS RESET, TO INIT DMV11
 4651  021410                                  BRESET
 4652  021410  104433                                                                  TRAP    C$RESET
 4653  021412  005000                          CLR     R0              ;# TIME DELAY TO ALLOW COMPLETION
 4654  021414  000240                  15$:    NOP                     ;# OF DMV11 MICRODIAGNOSTICS.
 4655  021416  077002                          SOB     R0,15$          ;#
 4656  021420                                  ENDIT:
 4657  021420                                  ENDINIT
 4658  021420                                                                  L10012:
 4659  021420  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.
                                ;
                                ;---------------------------------------------------------------------

021422                                  BGNAUTO
021422                                                                  L$AUTO::

021422                                  SETVEC  #4,#AD.HIT,#0   ;SETUP INVALID-ADDRESS TRAP VECTOR
021422  012746  000000                                                  MOV     #0,-(SP)
021426  012746  021540                                                  MOV     #AD.HIT,-(SP)
021432  012746  000004                                                  MOV     #4,-(SP)
021436  012746  000003                                                  MOV     #3,-(SP)
021442  104437                                                          TRAP    C$SVEC
021444  062706  000010                                                  ADD     #10,SP
021450  005037  002622                  CLR     TMP0            ;INITIALIZE TRAP FLAG REGISTER
021454  012702  000001                  MOV     #1,R2           ;FLAG BIT
021460  013703  002472                  MOV     BSEL0,R3        ;INIT ADDRESS POINTER

021464  105723                  1$:     TSTB    (R3)+           ;ACCESS THE CSR'S BY BYTES.
021466  006302                          ASL     R2
021470  103375                          BCC     1$

021472  013703  002472                  MOV     BSEL0,R3        ;RE-INIT ADDRESS POINTER
021476  012702  000001                  MOV     #1,R2           ;RE-INIT FLAG BIT
021502  005723                  2$:     TST     (R3)+           ;ACCESS THE CSR'S BY WORDS.
021504  006302                          ASL     R2
021506  006302                          ASL     R2
021510  103374                          BCC     2$

021512                                  CLRVEC  #4              ;RESTORE THE VECTOR TO DS
021512  012700  000004                                                  MOV     #4,R0
021516  104436                                                          TRAP    C$CVEC
021520  005737  002622                  TST     TMP0            ;DID WE GET HIT WITH AN INVALID ADDRESS TRAP?
021524  001403                          BEQ     AD.OK           ;NO.  EXIT TEST
021526                                  DODU    LOGDEV          ;YES, DROP THIS LOGICAL DEV.
021526  013700  002410                                                  MOV     LOGDEV,R0
021532  104451                                                          TRAP    C$DODU
```

```
 4710  021534  000240                AD.OK:  NOP                     ;(FOR PATCHING IN A HALT IF NECESSARY)
 4711
 4712  021536                              ENDAUTO
 4713  021536                                                                      L10013:
 4714  021536  104461                                                                      TRAP    C$AUTO
 4715
 4716  021540  050237  002622        AD.HIT: BIS     R2,TMP0         ;FLAG THE HIT IF WE GET IT!
 4717  021544  000002                      RTI                     ;RETURN
 4718
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

021546                                  BGNCLN
021546                                                                  L$CLEAN::

                                        ENDCLN
021546                                                                  L10014:
021546  104412                                                                  TRAP    C$CLEAN
```

```
                        .BTTL  DROP UNIT SECTION

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

021550                          BGNDU
021550                                                                  L$DU::
                        ;ISSUE UNIBUS RESET TO CLEAN UP
021550                          BRESET
021550  104433                                                                TRAP    C$RESET
021552                          ENDDU
021552                                                                  L10015:
021552  104453                                                                  TRAP    C$DU
```

```
                        .SBTTL  ADD UNIT SECTION

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

                                BGNAU
                                                                        L$AU::
                                ENDAU
                                                                        L10016:
                                                                                ;RAP    C$AU
```

```
                                .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).
                                ;*
                                ;--*********************************************************************
                                ;
021556                          ;       BGNTST                                          T1::

021556  004737  003374                  JSR     PC,@#STCLR      ;PUT THE MICROPROCESSOR IN THE MAINTENANCE LOOP
021562  103003                          BCC     .+8.            ;IF NO ERROR, PROCEED
021564                                  ERROR                   ;ELSE, REPORT IT AND
021564  104460                                                                  TRAP    C$ERROR
021566                                  ESCAPE  TST             ;       EXIT THIS TEST
021566  104410                                                                  TRAP    C$ESCAPE
021570  000144                                                                  .WORD   L10017-.

021572  004537  005432                  JSR     R5,CKUSTS       ;CHK USYRT STATUS FOR INIT STATE
021576  000110                          110                     ;TBMT=1, TSO=1
021600  103003                          BCC     .+8.            ;IF ERROR, PRINT REPORT
021602                                  ERROR
021602  104460                                                                  TRAP    C$ERROR
021604                                  ESCAPE  TST             ; AND SKIP REMAINDER OF TEST
021604  104410                                                                  TRAP    C$ESCAPE
021606  000126                                                                  .WORD   L10017-.

021610  012777  000007  160660          MOV     #DOTBMT,@BSEL2  ;ISSUE "TBMT TEST" COMMAND

021616  010346                          MOV     R3,-(SP)
021620  012703  001750                  MOV     #1000.,R3       ;WAIT FOR THE M-LOOP TO FINISH THE OPERATION
021624  077301                  1$:     SOB     R3,1$
021626  012603                          MOV     (SP)+,R3

021630  132777  000200  160640          BITB    #MRDY,@BSEL2    ;"MRDY" SHOULD BE HIGH BY NOW.
021636  001013                          BNE     5$              ;BR IF NO ERROR

021640  004737  004210                  JSR     PC,GETWSR       ;GET SELECT REGISTERS
021644  012737  000007  002400          MOV     #DOTBMT,GDATA   ;IDENTIFY REQUESTED FUNCTION
021652                                  GEDF    EM4,ERR4        ;REPORT "MRDY" TIMEOUT ERROR...
                                        ;               "DEVICE FATAL" ERROR # 39
021652  104455                                                                  TRAP    C$ERDF
021654  000047                                                                  .WORD   39
021656  013235                                                                  .WORD   EM4
021660  017030                                                                  .WORD   ERR4
021662                                  ESCAPE  TST             ;AND EXIT TEST
021662  104410                                                                  TRAP    C$ESCAPE
021664  000050                                                                  .WORD   L10017-.
```

```
 4818  021666  004737  004210          5$:     JSR     PC,GETWSR       ;GET CURRENT REGISTER CONTENTS
 4819  021672  023727  002260  100200          CMP     WSR2,#100200    ;COMPARE BYTE SELECT REGISTERS 2 AND 3
 4820                                                                  ;REG 2 = 200 -- "MRDY" SET/COMMAND CLEARED
 4821                                                                  ;REG 3 = 200 -- "TBMT" INTERRUPT BIT SET
 4822  021700  001415                          BEQ     7$              ;BR IF NO ERROR.
 4823
 4824  021702  012737  100200  002400          MOV     #100200,GDATA   ;GET THE GOOD DATA
 4825  021710  013737  002260  002402          MOV     WSR2,BDATA      ;GET THE BAD DATA
 4826  021716  012737  000002  002412          MOV     #2,REGNUM       ;GET THE REGISTER NUMBER
 4827  021724                                  GEDF    EM92,ERR5       ;REPORT "TBMT INTERRUPT FAILURE" ERROR.
 4828                                                                  ;       "DEVICE FATAL" ERROR # 40
 4829  021724  104455                                                                  TRAP    C$ERDF
 4830  021726  000050                                                                  .WORD   40
 4831  021730  014323                                                                  .WORD   EM92
 4832  021732  017154                                                                  .WORD   ERR5
 4833
 4834  021734                          7$:     ENDTST
 4835  021734                                                                  L10017:
 4836  021734  104401                                                                  TRAP    C$ETST
 4837
```

```
                                .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).
                                ;*
                                ;*
                                ;--**********************************************************************
                                ;
                                ;       BGNTST
                                                                                        T2::
021736  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11 (MAINT LOOP)
                                ;-------------------------------------------------------------------
                                        BGNSUB
                                                                                T2.1:
021742  104402                                                                          TRAP    C$BSUB
021744  004537  003610                  JSR     R5,READI        ;GET "DDCMP ADDRESS"
021750  121400                          .WORD   SWPDDCMP
021752  000000          T3.SW1: .WORD   0               ;  (IT WILL BE PUT HERE)
021754  004537  003610                  JSR     R5,READI        ;GET "BOOT ADDRESS"
021760  121000                          .WORD   SWPBOT
021762  000000          T3.SW2: .WORD   0               ;  (IT WILL BE PUT HERE)

021764  123737  021752  002542          CMPB    T3.SW1,LUSWI2   ;# DOES "DDCMP ADDRESS" MATCH P-TABLE VALUE
021772  001004                          BNE     T3.ERR          ;NO,  REPORT ERROR

021774  123737  021762  002540          CMPB    T3.SW2,LUSWI1   ;# DOES BOOT ADDRESS" MATCH P-TABLE VALUE
022002  001405                          BEQ     T3.OK           ;NO,  REPORT ERROR

022004          T3.ERR: GEDF    T3.EMD,T3.EM1   ;REPORT SWITCH SETTINGS DON'T MATCH P-TABLE
                                                ;       "DEVICE FATAL" ERROR # 41
022004  104455                                                                  TRAP    C$ERDF
022006  000051                                                                  .WORD   41
022010  022354                                                                  .WORD   T3.EMD
022012  022124                                                                  .WORD   T3.EM1
022014                          CKLOOP
022014  104406                                                                  TRAP    C$CLP1

022016  000240          T3.OK:  NOP
022020                          ENDSUB
022020                                                                  L10021:
022020  104403                                                                  TRAP    C$ESUB
                                ;-------------------------------------------------------------------
022022                          BGNSUB
022022                                                                  T2.2:
022022  104402                                                                  TRAP    C$BSUB
```

```
4894  022024  005737  002544          TST     BRDTYP          ;IS THIS AN M8064 ?
4895  022030  001433                  BEQ     10$             ; IF YES: THEN SKIP THIS SUBROUTINE
4896  022032  004537  003610          JSR     R5,READI        ;GET VIAORA REGISTER
4897  022036  120017                  VIAORA
4898  022040  000000          1$:     000                     ;STATUS WORD GOES HERE
4899  022042  103003                  BCC     .+8.
4900  022044                          ERROR
4901  022044  104460                                                          TRAP    C$ERROR
4902  022046                          ESCAPE  SUB
4903  022046  104410                                                          TRAP    C$ESCAPE
4904  022050  000050                                                          .WORD   L10022-.
4905  022052  142737  000357  022040          BICB    #357,1$         ;CLEAR ALL BUT SPEED SELECT BIT(PA4)
4906  022060  106237  022040              ASRB    1$              ;RIGHT JUSTIFY SPOSEL SWITCH BIT FOR
4907  022064  106237  022040              ASRB    1$              ; COMPARISON WITH OPERATOR'S REPLY
4908  022070  106237  022040              ASRB    1$
4909  022074  106237  022040              ASRB    1$
4910  022100  123737  022040  002550      CMPB    1$,BDRATE       ;IS THE SWITCH IN THE DESIRED POSITION?
4911  022106  001404                      BEQ     10$             ; IF YES: END THE SUBROUTINE/TEST
4912
4913  022110                          GEDF    T3.6,T3.EM2     ;REPORT ERROR
4914                                                          ;       "DEVICE FATAL" ERROR # 42
4915  022110  104455                                                          TRAP    C$ERDF
4916  022112  000052                                                          .WORD   42
4917  022114  022652                                                          .WORD   T3.6
4918  022116  022316                                                          .WORD   T3.EM2
4919  022120                  10$:    ENDSUB
4920  022120                                                          L10022:
4921  022120  104403                                                          TRAP    C$ESUB
4922                          ;-----------------------------------------------------------------
4923  022122                  T3.END: ENDTST
4924  022122                                                          L10020:
4925  022122  104401                                                          TRAP    C$ETST
4926
4927  022124                          BGNMSG  T3.EM1
4928  022124                                                          T3.EM1::
4929  022124                          PRINTB  #T3.1           ;PRINT ERROR MESSAGE
4930  022124  012746  022400                                                  MOV     #T3.1,-(SP)
4931  022130  012746  000001                                                  MOV     #1,-(SP)
4932  022134  010600                                                          MOV     SP,R0
4933  022136  104414                                                          TRAP    C$PNTB
4934  022140  062706  000004                                                  ADD     #4,SP
4935  022144                          PRINTX  #T3.2
4936  022144  012746  022465                                                  MOV     #T3.2,-(SP)
4937  022150  012746  000001                                                  MOV     #1,-(SP)
4938  022154  010600                                                          MOV     SP,R0
4939  022156  104415                                                          TRAP    C$PNTX
4940  022160  062706  000004                                                  ADD     #4,SP
4941  022164                          PRINTX  #T3.3
4942  022164  012746  022514                                                  MOV     #T3.3,-(SP)
4943  022170  012746  000001                                                  MOV     #1,-(SP)
4944  022174  010600                                                          MOV     SP,R0
4945  022176  1044.5                                                          TRAP    C$PNTX
4946  022200  062706  000004                                                  ADD     #4,SP
4947  022204                          PRINTX  #T3.4,LUSWI2,LUSWI1
4948  022204  013746  002540                                                  MOV     LUSWI1,-(SP)
4949  022210  013746  002542                                                  MOV     LUSWI2,-(SP)
```

```
 4950  022214  012746  022545                                                          MOV     #T3.4,-(SP)
 4951  022220  012746  000003                                                          MOV     #3,-(SP)
 4952  022224  010600                                                                 MOV     SP,R0
 4953  022226  104415                                                                 TRAP    C$PNTX
 4954  022230  062706  000010                                                         ADD     #10,SP
 4955  022234  105037  021753          CLRB    T3.SW1+1        ;MAKE SURE BITS 8 THROUGH 15 AREN'T REPORTED
 4956  022240  105037  021763          CLRB    T3.SW2+1        ; -- ESPECIALLY BIT 8!!
 4957  022244                          PRINTX  #T3.5,T3.SW1,T3.SW2
 4958  022244  013746  021762                                                          MOV     T3.SW2,-(SP)
 4959  022250  013746  021752                                                          MOV     T3.SW1,-(SP)
 4960  022254  012746  022610                                                          MOV     #T3.5,-(SP)
 4961  022260  012746  000003                                                          MOV     #3,-(SP)
 4962  022264  010600                                                                 MOV     SP,R0
 4963  022266  104415                                                                 TRAP    C$PNTX
 4964  022270  062706  000010                                                         ADD     #10,SP
 4965  022274                          PRINTB  #NEWLIN
 4966  022274  012746  011645                                                          MOV     #NEWLIN,-(SP)
 4967  022300  012746  000001                                                          MOV     #1,-(SP)
 4968  022304  010600                                                                 MOV     SP,R0
 4969  022306  104414                                                                 TRAP    C$PNTB
 4970  022310  062706  000004                                                         ADD     #4,SP
 4971  022314                          ENDMSG
 4972  022314                                                                  L10023:
 4973  022314  104423                                                                 TRAP    C$MSG
 4974
 4975  022316                          BGNMSG  T3.EM2
 4976  022316                                                                  T3.EM2::
 4977  022316                          PRINTB  #FMT21,#TXT12,MPCSR
 4978  022316  013746  002472                                                          MOV     MPCSR,-(SP)
 4979  022322  012746  015541                                                          MOV     #TXT12,-(SP)
 4980  022326  012746  012523                                                          MOV     #FMT21,-(SP)
 4981  022332  012746  000003                                                          MOV     #3,-(SP)
 4982  022336  010600                                                                 MOV     SP,R0
 4983  022340  104414                                                                 TRAP    C$PNTB
 4984  022342  062706  000010                                                         ADD     #10,SP
 4985  022346  004737  020356          JSR     PC,ERR11$       ;GET AND PRINT VIA REGISTERS
 4986  022352                          ENDMSG
 4987  022352                                                                  L10024:
 4988  022352  104423                                                                 TRAP    C$MSG
 4989
 4990                                  .NLIST  BEX
       022354  053523  052111  044103 T3.EMD: .ASCIZ  'SWITCH SETTING TEST'
       022400  047045  040445  053523 T3.1:   .ASCIZ  /##ASWITCH PACK SETTING DOES NOT MATCH P-TABLE ENTRY/
       022465     045  022516  031123 T3.2:   .ASCIZ  '##S22#A DOCMP    BOOT'
       022514  047045  051445  031062 T3.3:   .ASCIZ  '##S22#AADDRESS  ADDRESS'
       022545     045  022516  033123 T3.4:   .ASCIZ  '##S6#AP-TABLE VALUES:#S1#03#S6#03'
       022610  047045  051445  022466 T3.5:   .ASCIZ  '##S6#AREAD FROM DMV:#S2#03#S6#03'
       022652  050123  051504  046105 T3.6:   .ASCIZ /SPDSEL SWITCH DOESN'T MATCH P-TABLE BAUD RATE ENTRY/
                                      .LIST   BEX
 4991                                 .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
022736                                                                                  T3::
022736  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11
022742  005001                          CLR     R1              ;INIT USYRT REG ADRS POINTER
022744  012702  002652                  MOV     #PATE,R2        ;INIT PATTERN E POINTER
022750  016137  002552  022766  1$:     MOV     USYREG(R1),2$   ;GET USYRT REG ADRS
022756  112237  022770                  MOVB    (R2)+,3$        ;GET A PATTERN BYTE
022762  004537  003734                  JSR     R5,WRITEI       ;WRITE A USYRT REG
022766  000000                  2$:     .WORD   0               ;USYRT REG ADRS GOES HERE
022770  000000                  3$:     .WORD   0               ;DATA BYTE GOES HERE
022772  062701  000002                  ADD     #2,R1           ;INCR REG ADRS PTR
022776  020127  000020                  CMP     R1,#16.         ;SEE IF ALL REGS WRITTEN YET
023002  002762                          BLT     1$              ;BR IF NOT
023004  004737  005420                  JSR     PC,INIDMV       ;ISSUE MASTER CLEAR
023010  005001                          CLR     R1              ;INIT USYRT REG ADRS PTR
023012  012702  002662                  MOV     #PATF,R2        ;INIT DATA PATTERN POINTER
023016  016137  002552  023030  6$:     MOV     USYREG(R1),7$   ;SET USYRT READ ADDRESS
023024  004537  003610                  JSR     R5,READI        ;READ A USYRT REG
023030  000000                  7$:     .WORD   0               ;USYRT REG ADRS GOES HERE
023032  000000                  8$:     .WORD   0               ;DATA READ IS RETURNED HERE
023034  123722  023032                  CMPB    8$,(R2)+        ;SEE IF REG CONTAINS EXPECTED DATA
023040  001422                          BEQ     9$              ;BR IF MATCH
023042  010137  002412                  MOV     R1,REGNUM       ;SET USYRT REG NO. FOR PRINTOUT
023046  006237  002412                  ASR     REGNUM          ;GET WORD OFFSET
023052  005037  002400                  CLR     GDATA           ;GET EXPECTED DATA
023056  116237  177777  002400          MOVB    -1(R2),GDATA
023064  013737  023032  002402          MOV     8$,BDATA        ;GET ACTUAL DATA
                        ;REPORT REG NOT CLEARED BY MASTER CLEAR
023072                                  GEDF    EM1,ERR10
                                                        ;       "DEVICE FATAL" ERROR # 43
023072  104455                                                  TRAP    C$ERDF
023074  000053                                                  .WORD   43
023076  013103                                                  .WORD   EM1
023100  017372                                                  .WORD   ERR10
```

```
5048 023102                         ESCAPE  TST
5049 023102 104410                                                          TRAP    C$ESCAPE
5050 023104 000014                                                          .WORD   L10025-.
5051 023106 062701 000002   9$:     ADD     #2,R1           ;INCR USYRT REG ADRS PTR
5052 023112 020127 000020           CMP     R1,#16.         ;SEE IF ALL REGS READ YET
5053 023116 002737                  BLT     6$              ;BR IF NOT
5054 023120                 ENDTST
5055 023120                                                        L10025:
5056 023120 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, PCSRL, PCSRH, PCR, USYRT STATUS REG
                        ;*
                        ;*
                        ;-*****************************************************************************
                        ;
                        ;       BGNTST
023122                                                                          T4::
023122  004737  005420          JSR     PC,INITDMV      ;INIT DMV-11
023126  005001                  CLR     R1              ;INIT USYRT REG ADRS POINTER
023130  012702  002652          MOV     @PATE,R2        ;INIT PATTERN E POINTER
023134  016137  002552  023152  1$:     MOV     USYREG(R1),2$   ;GET USYRT REG ADRS
023142  112237  023154          MOVB    (R2)+,3$        ;GET A PATTERN BYTE
023146  004537  003734          JSR     R5,WRITEI       ;WRITE A USYRT REG
023152  000000                  2$:     .WORD   0       ;USYRT REG ADRS GOES HERE
023154  000000                  3$:     .WORD   0       ;DATA BYTE GOES HERE
023156  062701  000002          ADD     @2,R1           ;INCR REG ADRS PTR
023162  020127  000020          CMP     R1,@16.         ;SEE IF ALL REGS WRITTEN YET
023166  002762                  BLT     1$              ;BR IF NOT

023170  004537  005126          JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
023174  103003                  BCC     .+8.            ;BR IF NO ERROR
023176                          ERROR                   ;REPORT STACKED ERROR
023176  104460                                                          TRAP    C$ERROR
023200                          ESCAPE TST              ;SKIP TO END OF TEST
023200  104410                                                          TRAP    C$ESCAPE
023202  000002                                                          .WORD   L10026 .

                        ENDTST
023204                                                          L10026:
023204  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
  023206                                                                                  T5::
  023206  004737  005420                        JSR     PC,INIDMV           ;ISSUE MASTER CLEAR TO INIT DMV
  023212  012702  002662                        MOV     #PATF,R2            ;INIT PATTERN I POINTER
  023216  012703  002572                        MOV     #REDDAT,R3          ;INIT PTR TO EXPECTED DATA AREA
  023222  112223                        1$:     MOVB    (R2)+,(R3)+         ;MOV PATTERN F INTO REDDAT TABLE
  023224  020227  002672                        CMP     R2,#PATG
  023230  103774                                BLO     1$
  023232  005001                                CLR     R1                  ;INIT USYRT REG ADRS PTR FOR WRITING
  023234  005002                                CLR     R2                  ;INIT INDEX FOR WRITING
  023236  016137  002552  023302        2$:     MOV     USYREG(R1),3$       ;SET USYRT REG ADRS
  023244  116237  002014  023304                MOVB    PATJ(R2),4$         ;SET DATA FOR WRITE
  023252  113762  023304  002572                MOVB    4$,REDDAT(R2)       ;SET EXPECTED DATA BYTE
  023260  023727  023302  120402                CMP     3$,#TDSRL           ;SEE IF WRITING TDSRL
  023266  001003                                BNE     10$                 ;BR IF NOT
  023270  142737  000100  002601                BICB    #100,REDDAT+7       ;FIX EXPECTED USTAT VALUE
  023276  004537  003734                10$:    JSR     R5,WRITEI           ;WRITE BYTE INTO A USYRT REG
  023302  000000                        3$:     .WORD   0                   ;REG ADRS GOES HERE
  023304  000000                        4$:     .WORD   0                   ;DATA BYTE GOES HERE
  023306  005003                                CLR     R3                  ;INIT USYRT REG ADRS PTR FOR READING
  023310  005004                                CLR     R4                  ;INIT INDEX FOR READING
  023312  016337  002552  023324        5$:     MOV     USYREG(R3),6$       ;SET USYRT REG ADRS
  023320  004537  003610                        JSR     R5,READI            ;READ A USYRT REG
  023324  000000                        6$:     .WORD   0                   ;REG ADRS GOES HERE
  023326  000000                        7$:     .WORD   0                   ;DATA BYTE READ IS RETURNED HERE
  023330  123764  023326  002572                CMPB    7$,REDDAT(R4)       ;SEE IF BYTE READ MATCHES EXPECTED BYTE
  023336  001420                                BEQ     8$                  ;BR IF MATCH
  023340  010437  002412                        MOV     R4,REGNUM           ;SET FAILING REG NO. FOR ERROR REPORT
  023344  005037  002400                        CLR     GDATA               ;GET EXPECTED DATA
  023350  116437  002572  002400                MOVB    REDDAT(R4),GDATA
  023356  013737  023326  002402                MOV     7$,BDATA            ;GET ACTUAL DATA
                                        ;REPORT REGISTER MISCOMPARE ERROR
  023364                                        GEDF    EM66,ERR10
                                                                            ;       "DEVICE FATAL" ERROR # 44
  023364  104455                                                                                  TRAP    C$ERDF
  023366  000054                                                                                  .WORD   44
```

```
 5161  023370  013741                                                          .WORD   EM66
 5162  023372  017372                                                          .WORD   ERR10
 5163  023374                          ESCAPE  TST
 5164  023374  104410                                                          TRAP    C$ESCAPE
 5165  023376  000032                                                          .WORD   L10027-.
 5166  023400  062703  000002    8$:     ADD     #2,R3       ;INCR USYRT REG ADRS PTR FOR READING
 5167  023404  005204                    INC     R4          ;INCR READ INDEX
 5168  023406  020427  000010          CMP     R4,#10      ;SEE IF ALL REGS READ YET
 5169  023412  002737                    BLT     5$          ;BR IF NOT ALL READ YET
 5170  023414  062701  000002          ADD     #2,R1       ;INCR USYRT REG ADRS PTR FOR WRITING
 5171  023420  005202                    INC     R2          ;INCR WRITE INDEX
 5172  023422  020227  000010          CMP     R2,#10      ;SEE IF ALL REGS WRITTEN YET
 5173  023426  002703                    BLT     2$          ;BR IF NOT ALL WRITTEN YET
 5174  023430                    ENDTST
 5175  023430                                                      L10027:
 5176  023430  104401                                                          TRAP    C$ETST
```

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

5177                    .SBTTL  TEST 6 -- R/W BIT TEST OF PCSAR HIGH BYTE
5178
5179                    ;+**********************************************************************
5180                    ;*
5181                    ;*      TEST 6 -- R/W BIT TEST OF PCSAR HIGH BYTE
5182                    ;*
5183                    ;*
5184                    ;*  WRITE, READ, AND COMPARE EACH WORD OF DATA PATTERN G.
5185                    ;*
5186                    ;*  PATTERN  G:  000,  001, 003, 004, 005, 007, 100, 101, 103, 104, 105,
5187                    ;*  107, 000, 017, 027, 041, 200, 277, 103, 144, 115, 157, 000.
5188                    ;*
5189                    ;--**********************************************************************
5190                    ;
5191                    ;       BGNTST
5192    023432                                                                  T6::
5193    023432  004737  005420          JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
5194    023436  103003                  BCC     30$             ;IF AN ERROR OCCURED,
5195    023440                          ERROR                   ;REPORT IT &
5196    023440  104460                                                          TRAP    C$ERROR
5197    023442                          ESCAPE  TST             ;  EXIT
5198    023442  104410                                                          TRAP    C$ESCAPE
5199    023444  000056                                                          .WORD   L10030-.
5200
5201    023446  012701  002672  30$:    MOV     @PATG,R1            ;POINT TO PATTERN TABLE
5202    023452  012703  000026          MOV     #<PATH-PATG-2>,R3   ;GET # OF ENTRIES IN TABLE
5203    023456  012737  000005  002412  MOV     #5,REGNUM           ;ERROR INDEX FOR PCSARH
5204
5205    023464                  T7.LP:  BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
5206    023464                                                                  T6.1:
5207    023464  104402                                                          TRAP    C$BSUB
5208
5209    023466  111137  002376          MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR "STUREG"
5210    023472  112137  002400          MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR "STUREG"
5211    023476  012700  120405          MOV     #PCSARH,R0      ;SPECIFY THE REGISTER BEING TESTED
5212    023502  004737  004272          JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
5213    023506  103003                  BCC     10$             ;WAS AN ERROR FOUND?
5214    023510                          ERROR                   ;YES, REPORT IT AND
5215    023510  104460                                                          TRAP    C$ERROR
5216    023512                          ESCAPE  TST             ;     EXIT FROM THE TEST.  "CXLOOP" IS IMPLIED
5217    023512  104410                                                          TRAP    C$ESCAPE
5218    023514  000006                                                          .WORD   L10030-.
5219
5220    023516                  10$:    ENDSUB
5221    023516                                                          L10031:
5222    023516  104403                                                          TRAP    C$ESUB
5223
5224    023520  077317                  SOB     R3,T7.LP        ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
5225                                                            ;TEST 11.  ELSE, FALL OUT OF LOOP AND TEST
5226    023522                  ENDTST
5227    023522                                                          L10030:
5228    023522  104401                                                          TRAP    C$ETST
```

```
                                .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
                                ;*
                                ;--**********************************************************************
                                ;
023524                          ;       BGNTST
                                                                                T7::
023524  004737  005420          JSR     PC,INIDMV       ;INIT DMV & START UP THE MAINT. LOOP
023530  103003                  BCC     30$             ;IF AN ERROR OCCURED,
023532                          ERROR                   ;REPORT IT &
023532  104460                                                                  TRAP    C$ERROR
023534                          ESCAPE  TST             ;  EXIT
023534  104410                                                                  TRAP    C$ESCAPE
023536  000056                                                                  .WORD   L10032-.

023540  012701  002722  30$:    MOV     #PATH,R1            ;POINT TO PATTERN TABLE
023544  012703  000027          MOV     #<PATI-PATH-2>,R3   ;GET # OF ENTRIES IN TABLE
023550  012737  000004  002412  MOV     #4,REGNUM           ;ERROR INDEX FOR S/AR BYTE

023556                  T8.LP:  BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
023556                                                                          T7.1:
023556  104402                                                                  TRAP    C$BSUB

023560  111137  002376          MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR "STUREG"
023564  112137  002400          MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR "STUREG"
023570  012700  120404          MOV     #PCSARL,R0      ;SPECIFY THE REGISTER BEING TESTED
023574  004737  004272          JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
023600  103003                  BCC     10$             ;WAS AN ERROR FOUND?
023602                          ERROR                   ;YES, REPORT IT AND
023602  104460                                                                  TRAP    C$ERROR
023604                          ESCAPE  TST             ;       EXIT FROM THE TEST.  "CXLOOP" IS IMPLIED
023604  104410                                                                  TRAP    C$ESCAPE
023606  000006                                                                  .WORD   L10032-.

023610                  10$:    ENDSUB
023610                                                                  L10033:
023610  104403                                                                  TRAP    C$ESUB

023612  077317                  SOB     R3,T8.LP        ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
                                                        ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
023614                  ENDTST
023614                                                                  L10032:
023614  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, 367, 357, 030, 027, 377.
                                ;*
                                ;*
                                ;-*********************************************************************
                                ;
                                ;       BGNTST
023616                                                                          T8::
023616  004737  005420                  JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
023622  103003                  BCC     30$             ;IF AN ERROR OCCURED,
023624                          ERROR                   ;REPORT IT &
023624  104460                                                                  TRAP    C$ERROR
023626                          ESCAPE  TST             ;  EXIT
023626  104410                                                                  TRAP    C$ESCAPE
023630  000056                                                                  .WORD   L10034-.

023632  012701  002753          30$:    MOV     #PATI,R1                ;POINT TO PATTERN TABLE
023636  012703  000037                  MOV     #<PATJ-PATI-2>,R3       ;GET # OF ENTRIES IN TABLE
023642  012737  000006  002412          MOV     #6,REGNUM               ;ERROR INDEX FOR PCR REGISTER

023650                          T9.LP:  BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
023650                                                                          T8.1:
023650  104402                                                                  TRAP    C$BSUB

023652  111137  002376                  MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR "STUREG"
023656  112137  002400                  MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR "STUREG"
023662  012700  120407                  MOV     #PCR,R0         ;SPECIFY THE REGISTER BEING TESTED
023666  004737  004272                  JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
023672  103003                  BCC     10$             ;WAS AN ERROR FOUND?
023674                          ERROR                   ;YES, REPORT IT AND
023674  104460                                                                  TRAP    C$ERROR
023676                          ESCAPE  TST             ;       EXIT FROM THE TEST.  "CXLOOP" IS IMPLIED
023676  104410                                                                  TRAP    C$ESCAPE
023700  000006                                                                  .WORD   L10034-.

023702                          10$:    ENDSUB
023702                                                                          L10035:
023702  104403                                                                  TRAP    C$ESUB

023704  077317                  SOB     R3,T9.LP        ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
                                                        ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
023706                          ENDTST
023706                                                                          L10034:
023706  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.
                                ;*
                                ;--*********************************************************************
                                ;
023710                          ;       BGNTST                                                  T9::
023710  004737  005420                  JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
023714  103003                  BCC     30$             ;IF AN ERROR OCCURED,
023716                          ERROR                   ;REPORT IT &
023716  104460                                                                  TRAP    C$ERROR
023720                          ESCAPE  TST             ;  EXIT
023720  104410                                                                  TRAP    C$ESCAPE
023722  000114                                                                  .WORD   L10036-.

023724  012701  003024          30$:    MOV     #PATK,R1                ;POINT TO PATTERN TABLE
023730  012703  000017                  MOV     #<PATL-PATK-2>,R3       ;GET # OF ENTRIES IN TABLE
023734  012702  003045                  MOV     #PATL,R2                ;POINT TO "EXPECTED" DATA PATTERN TABLE

023740                          T10.LP: BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
023740                                                                          T9.1:
023740  104402                                                                  TRAP    C$BSUB

023742  112137  002376                  MOVB    (R1)+,TDATA     ;SETUP TEST DATA BYTE
023746  112237  002400                  MOVB    (R2)+,GDATA     ;SETUP EXPECTED DATA BYTE

023752  004537  003722                  JSR     R5,WRITE        ;WRITE TO DMV-11
023756  120403                  .WORD   TDSRH           ; DMV-11 ADDRESS WRITTEN TO
023760  002376                  .WORD   TDATA           ; LOCATION OF DATA WRITTEN

023762  004537  003476                  JSR     R5,READ         ;READ FROM DMV-11
023766  120403                  .WORD   TDSRH           ; DMV-11 ADDRESS READ FROM
023770  002402                  .WORD   BDATA           ; LOCATION WHERE READ DATA IS PUT

023772  042737  177760  002402          BIC     #177760,BDATA   ;MASK OUT "DON'T CARE" BITS
024000  023737  002400  002402          CMP     GDATA,BDATA     ;READ DATA = EXPECTED DATA ?
024006  001411                  BEQ     10$             ;WAS AN ERROR FOUND?

024010  012737  000003  002412          MOV     #3,REGNUM       ;YES: SET UP REGISTER NUMBER
```

```
5391  024016                          GEDF    EM25,ERR7A      ;REPORT IT AND
5392                                                          ;       "DEVICE FATAL" ERROR # 45
5393  024016  104455                                                  TRAP    C$ERDF
5394  024020  000055                                                  .WORD   45
5395  024022  013366                                                  .WORD   EM25
5396  024024  017252                                                  .WORD   ERR7A
5397  024026                          ESCAPE  TST             ;  EXIT FROM THE TEST.  "CKLOOP" IS IMPLIED
5398  024026  104410                                                  TRAP    C$ESCAPE
5399  024030  000006                                                  .WORD   L10036-.
5400
5401  024032                      10$:    ENDSUB
5402  024032                                                      L10037:
5403  024032  104403                                                  TRAP    C$ESUB
5404
5405  024034  077337                          SOB     R3,T10.LP       ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
5406                                                          ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
5407  024036                      ENDTST
5408  024036                                                      L10036:
5409  024036  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::
024040                  JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
024040  004737  005420  BCC     30$             ;IF AN ERROR OCCURED,
024044  103003          ERROR                   ;REPORT IT &
024046                                                                          TRAP    C$ERROR
024046  104460          ESCAPE  TST             ;  EXIT
024050                                                                          TRAP    C$ESCAPE
024050  104410                                                                  .WORD   L10040-.
024052  000056
                        30$:    MOV     #PATH,R1        ;POINT TO PATTERN TABLE
024054  012701  002722  MOV     #<PATI-PATH-2>,R3 ;GET # OF ENTRIES IN TABLE
024060  012703  000027  MOV     #2,REGNUM       ;ERROR INDEX FOR TXDB REGISTER
024064  012737  000002  002412
                        T11.LP: BGNSUB                  ;THE SUBTEST ONLY TESTS THE ONE PATTERN
024072                                                                  T10.1:
024072                                                                  TRAP    C$BSUB
024072  104402
                                MOVB    (R1),TDATA      ;SETUP TEST DATA BYTE FOR "STUREG"
024074  111137  002376  MOVB    (R1)+,GDATA     ;SETUP EXPECTED DATA BYTE FOR "STUREG"
024100  112137  002400  MOV     #TDSRL,R0       ;SPECIFY THE REGISTER BEING TESTED
024104  012700  120402  JSR     PC,STUREG       ;PERFORM STATIC TEST OF THE SPECIFIED REGISTER
024110  004737  004272  BCC     10$             ;WAS AN ERROR FOUND?
024114  103003          ERROR                   ;YES, REPORT IT AND
024116                                                                  TRAP    C$ERROR
024116  104460          ESCAPE  TST             ;       EXIT FROM THE TEST.  "CKLOOP" IS IMPLIED
024120                                                                  TRAP    C$ESCAPE
024120  104410                                                          .WORD   L10040-.
024122  000006
                        10$:    ENDSUB
024124                                                                  L10041:
024124                                                                  TRAP    C$ESUB
024124  104403
                        ;       SOB     R3,T11.LP       ;IF THERE IS IN FACT MORE DATA, LOOP BACK TO
024126  077317                                          ;TEST IT.  ELSE, FALL OUT OF LOOP AND TEST
024130                  ENDTST
024130                                                                  L10040:
024130  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::
024132                  
024132  004737  005420          JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
024136  103003                  BCC     30$             ;IF AN ERROR OCCURED,
024140                          ERROR                   ;REPORT IT &
024140  104460                                                          TRAP    C$ERROR
024142                          ESCAPE  TST             ;  EXIT
024142  104410                                                          TRAP    C$ESCAPE
024144  000040                                                          .WORD   L10042-.

024146  012701  002722  30$:    MOV     #PATH,R1            ;POINT TO PATTERN TABLE
024152  012703  000027          MOV     #<PATI-PATH-2>,R3   ;GET # OF ENTRIES IN TABLE

024156  012137  024170  20$:    MOV     (R1)+,2$

024162  004537  003734          JSR     R5,WRITEI       ;WRITE TO DMV-11
024166  120400                  .WORD   RDSRL           ; DMV-11 ADDRESS WRITTEN TO
024170  000000          2$:     .WORD   0               ; ACTUAL DATA WRITTEN

024172  004537  003610          JSR     R5,READI        ;READ FROM DMV-11
024176  120400                  .WORD   RDSRL           ; DMV-11 ADDRESS READ FROM
024200  000000                  .WORD   0               ; READ DATA IS PUT HERE

024202  077313                  SOB     R3,20$          ;IF MORE DATA, LOOP BACK TO WRITE/READ IT,
                                                        ;  ELSE, FALL OUT OF LOOP AND TEST
024204                  ENDTST
024204                                                                  L10042:
024204  104401                                                                  TRAP    C$ETST
```

```
                                        .SBTTL  TEST 12 -- PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE

                                        ;**********************************************************************
                                        ;*
                                        ;*      TEST 12 -- PSEUDO R/W BIT TEST OF RDSR'S HIGH BYTE
                                        ;*
                                        ;*  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
        024206                                                                         T12::
        024206  004737  005420                  JSR     PC,INIDMV       ;INIT DMV/VIA & START UP MAINT. LOOP
        024212  103003                          BCC     30$             ;IF AN ERROR OCCURED,
        024214                                  ERROR                   ;REPORT IT &
        024214  104460                                                                          TRAP    C$ERROR
        024216                                  ESCAPE  TST             ;  EXIT
        024216  104410                                                                          TRAP    C$ESCAPE
        024220  000040                                                                          .WORD   L10043-.

        024222  012701  002722          30$:    MOV     #PATH,R1                ;POINT TO PATTERN TABLE
        024226  012703  000027                  MOV     #<PATI-PATH-2>,R3       ;GET # OF ENTRIES IN TABLE

        024232  012137  024244          20$:    MOV     (R1)+,2$

        024236  004537  003734                  JSR     R5,WRITEI       ;WRITE TO DMV-11
        024242  120401                          .WORD   RDSRH           ; DMV-11 ADDRESS WRITTEN TO
        024244  000000                  2$:     .WORD   0               ; ACTUAL DATA WRITTEN

        024246  004537  003610                  JSR     R5,READI        ;READ FROM DMV-11
        024252  120401                          .WORD   RDSRH           ; DMV-11 ADDRESS READ FROM
        024254  000000                          .WORD   0               ; READ DATA IS PUT HERE

        024256  077313                          SOB     R3,20$          ;IF MORE DATA, LOOP BACK TO WRITE/READ IT.
                                                                        ;  ELSE, FALL OUT OF LOOP AND TEST
        024260                          ENDTST
        024260                                                                         L10043:
        024260  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
024262                                                                          T13::
024262  004737  005420                  JSR     PC,INIDMV       ;INIT DMV11
024266  004537  003734                  JSR     R5,WRITEI       ;LOAD T1C-L, T1L-L FOR 56K BAUD
024272  120004                          VIAT1A
024274  000017                          15.
024276  004537  003734                  JSR     R5,WRITEI       ;LOAD T1C-H, T1L-H, START CLOCK
024302  120005                          VIAT1B
024304  000000                          000
                                ;-------------------------------------------------------------------------
                                ;WAIT FOR NULCLK BIT TO BE SET TO 1
                                ;-------------------------------------------------------------------------
024306  012737  170000  002602          MOV     #170000,REG0    ;INIT PROGRAM TIMER
024314  004537  003610          1$:     JSR     R5,READI        ;READ VIA ORB
024320  120000                          VIAORB
024322  000000                  2$:     .WORD   0
024324  132737  000200  024322          BITB    #NULCLK,2$      ;SEE IF CLOCK BIT SET
024332  001025                          BNE     3$              ;BR IF SET
024334  005237  002602                  INC     REG0            ;INCR TIMER
024340  001365                          BNE     1$              ;BR IF TIMER DID NOT TIME OUT
                                                                ; (TIME OUT = SEVERAL HUNDRED MICRO-SEC)
024342  012737  000000  002412          MOV     #0,REGNUM       ;SET VIA REG NO. FOR PRINTOUT
024350  012737  000200  002400          MOV     #NULCLK,GDATA   ;GET EXPECTED DATA
024356  013737  024322  002402          MOV     2$,BDATA        ;GET ACTUAL DATA
024364  042737  000177  002402          BIC     #177,BDATA      ;CLEAR UNUSED BITS
                                ;REPORT NULL CLK STUCK AT 0
                                        GEDF    EM5,ERR11
024372                                                          ;       "DEVICE FATAL" ERROR # 46
024372  104455                                                                  TRAP    C$ERDF
024374  000056                                                                  .WORD   46
024376  013252                                                                  .WORD   EM5
```

```
5601  024400  017546                                                          .WORD   ERR11
5602  024402                          ESCAPE  TST
5603  024402  104410                                                          TRAP    C$ESCAPE
5604  024404  000176                                                          .WORD   L10044-.
5605                          ;-------------------------------------------------------------
5606                          ;WAIT FOR NULCLK BIT TO BE CLEARED TO 0
5607                          ;-------------------------------------------------------------
5608  024406  012737  170000  002602  3$:     MOV     #170000,REG0    ;INIT PROGRAM TIMER
5609  024414  004537  003610          4$:     JSR     R5,READI        ;READ VIA ORB
5610  024420  120000                          VIAORB
5611  024422  000000                  5$:     .WORD   0
5612  024424  132737  000200  024422          BITB    #NULCLK,5$      ;SEE IF CLOCK BIT CLEARED
5613  024432  001425                          BEQ     6$              ;BR IF CLEARED
5614  024434  005237  002602                  INC     REG0            ;INCR TIMER
5615  024440  001365                          BNE     4$              ;BR IF TIMER DID NOT TIME OUT
5616                                                                  ; (TIME OUT = SEVERAL HUNDRED MICRO-SEC)
5617  024442  012737  000000  002412          MOV     #0,REGNUM       ;SET VIA REG NO. FOR PRINTOUT
5618  024450  012737  000000  002400          MOV     #000,GDATA      ;GET EXPECTED DATA
5619  024456  013737  024422  002402          MOV     5$,BDATA        ;GET ACTUAL DATA
5620  024464  042737  000177  002402          BIC     #177,BDATA      ;CLEAR UNUSED BITS
5621                          ;REPORT NULL CLK STUCK AT 1
5622  024472                  GEDF    EM6,ERR11
5623                                                                  ;       "DEVICE FATAL" ERROR # 47
5624  024472  104455                                                          TRAP    C$ERDF
5625  024474  000057                                                          .WORD   47
5626  024476  013302                                                          .WORD   EM6
5627  024500  017546                                                          .WORD   ERR11
5628  024502                          ESCAPE  TST
5629  024502  104410                                                          TRAP    C$ESCAPE
5630  024504  000076                                                          .WORD   L10044-.
5631                          ;-------------------------------------------------------------
5632                          ;WAIT FOR NULCLK BIT TO BE SET TO 1 AGAIN
5633                          ;-------------------------------------------------------------
5634  024506  012737  170000  002602  6$:     MOV     #170000,REG0    ;INIT PROGRAM TIMER
5635  024514  004537  003610          7$:     JSR     R5,READI        ;READ VIA ORB
5636  024520  120000                          VIAORB
5637  024522  000000                  8$:     .WORD   0
5638  024524  132737  000200  024522          BITB    #NULCLK,8$      ;SEE IF CLOCK BIT SET
5639  024532  001023                          BNE     9$              ;BR IF SET
5640  024534  005237  002602                  INC     REG0            ;INCR TIMER
5641  024540  001365                          BNE     7$              ;BR IF TIMER DID NOT TIME OUT
5642                                                                  ; (TIME OUT = SEVERAL HUNDRED MICRO-SEC)
5643  024542  012737  000000  002412          MOV     #0,REGNUM       ;SET VIA REG NO. FOR PRINTOUT
5644  024550  012737  000200  002400          MOV     #NULCLK,GDATA   ;GET EXPECTED DATA
5645  024556  013737  024522  002402          MOV     8$,BDATA        ;GET ACTUAL DATA
5646  024564  042737  000177  002402          BIC     #177,BDATA      ;CLEAR UNUSED BITS
5647                          ;REPORT NULL CLK STUCK AT 0
5648  024572                  GEDF    EM5,ERR11
5649                                                                  ;       "DEVICE FATAL" ERROR # 48
```

```
5650  024572  104455                                                    TRAP    C$ERDF
5651  024574  000060                                                    .WORD   48
5652  024576  013252                                                    .WORD   EMT5
5653  024600  017546                                                    .WORD   ERR11
5654  024602                        9$:
5655  024602                        ENDTST
5656  024602                                                L10044:
5657  024602  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
024604                                                                          T14::

024604  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

024610  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
024614  043525                          DDCMP!NOCHK!125         ;SET DDCMP,NO CHECK,S/AR(SYNC)=125
024616  000000                          0                       ;USE 8 BIT CHARS
024620  103003                          BCC     .+8.            ;BR IF NO ERROR
024622                                  ERROR                   ;REPORT STACKED ERROR
024622  104460                                                                          TRAP    C$ERROR
024624                                  ESCAPE  TST             ;SKIP TO END OF TEST
024624  104410                                                                          TRAP    C$ESCAPE
024626  000106                                                                          .WORD   L10045-.

024630  004537  010010                  JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
024634  000001                          TSOM                    ;AND KNOCK DOWN TBMT
024636  000007                          7.

024640  004537  010010                  JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
024644  000001                          TSOM                    ;AND KNOCK DOWN TBMT
024646  000010                          8.

024650  004537  010010                  JSR     R5,TXCTRL       ;OUTPUT 3RD SYNC CHARACTER (125)
024654  000001                          TSOM                    ;AND KNOCK DOWN TBMT
024656  000010                          8.

024660  004537  011364                  JSR     R5,RCV1ST       ;CLOCK AND RCV TSOM
024664  000000                          0
```

```
5714  024666  103003                  BCC     .+8.            ;BR IF NO ERROR
5715  024670                          ERROR                   ;REPORT STACKED ERROR
5716  024670  104460                                                  TRAP    C$ERROR
5717  024672                          ESCAPE  TST             ;SKIP TO END OF TEST
5718  024672  104410                                                  TRAP    C$ESCAPE
5719  024674  000040                                                  .WORD   L10045-.
5720
5721  024676  004537  010110          JSR     R5,RXCHAR       ;READ AND CHECK FOR S/AR 125 SYNC
5722  024702  000125                  125                     ; AND CLOCK IN NEXT ONE.
5723  024704  000000                  0
5724  024706  000010                  8.
5725  024710  103003                  BCC     .+8.            ;BR IF NO ERROR
5726  024712                          ERROR                   ;REPORT STACKED ERROR
5727  024712  104460                                                  TRAP    C$ERROR
5728  024714                          ESCAPE  TST             ;SKIP TO END OF TEST
5729  024714  104410                                                  TRAP    C$ESCAPE
5730  024716  000016                                                  .WORD   L10045-.
5731
5732  024720  004537  005126          JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
5733  024724  103003                  BCC     .+8.            ;BR IF NO ERROR
5734  024726                          ERROR                   ;REPORT STACKED ERROR
5735  024726  104460                                                  TRAP    C$ERROR
5736  024730                          ESCAPE  TST
5737  024730  104410                                                  TRAP    C$ESCAPE
5738  024732  000002                                                  .WORD   L10045-.
5739
5740  024734                  ENDTST
5741  024734                                                  L10045:
5742  024734  104401                                                  TRAP    C$ETST
```

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

                        .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
024736                                                                  T15::
                        
024736  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
                        
024742  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
024746  047626                  DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
024750  000000                  0                       ;USE 8 BIT CHARS
024752  103003                  BCC     .+8.            ;BR IF NO ERROR
024754                          ERROR                   ;REPORT STACKED ERROR
024754  104460                                                          TRAP    C$ERROR
024756                          ESCAPE  TST             ;SKIP TO END OF TEST
024756  104410                                                          TRAP    C$ESCAPE
024760  000136                                                          .WORD   L10046-.
                        
024762  004537  007676          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
024766  000226                  SYNCH
024770  000007                  7.
024772  103003                  BCC     .+8.            ;BR IF NO ERROR
024774                          ERROR                   ;REPORT STACKED ERROR
024774  104460                                                          TRAP    C$ERROR
024776                          ESCAPE  TST             ;SKIP TO END OF TEST
024776  104410                                                          TRAP    C$ESCAPE
025000  000116                                                          .WORD   L10046-.
                        
025002  004537  007676          JSR     R5,TXCHAR       ;LOAD 125, TX 2ND SYNCH
025006  000125                  125
```

```
5799  025010  000010                    8.
5800  025012  103003                    BCC     .+8.            ;BR IF NO ERROR
5801  025014                            ERROR                   ;REPORT STACKED ERROR
5802  025014  104460                                                    TRAP    C$ERROR
5803  025016                            ESCAPE  TST             ;SKIP TO END OF TEST
5804  025016  104410                                                    TRAP    C$ESCAPE
5805  025020  000076                                                    .WORD   L10046-.
5806
5807  025022  004537  007676            JSR     R5,TXCHAR       ;LOAD 377, TX 125
5808  025026  000377                    377
5809  025030  000010                    8.
5810  025032  103003                    BCC     .+8.            ;BR IF NO ERROR
5811  025034                            ERROR                   ;REPORT STACKED ERROR
5812  025034  104460                                                    TRAP    C$ERROR
5813  025036                            ESCAPE  TST             ;SKIP TO END OF TEST
5814  025036  104410                                                    TRAP    C$ESCAPE
5815  025040  000056                                                    .WORD   L10046-.
5816
5817  025042  004537  011364            JSR     R5,RCV1ST       ;CLOCK AND RCV 125
5818  025046  000000                    0
5819  025050  103003                    BCC     .+8.            ;BR IF NO ERROR
5820  025052                            ERROR                   ;REPORT STACKED ERROR
5821  025052  104460                                                    TRAP    C$ERROR
5822  025054                            ESCAPE  TST             ;SKIP TO END OF TEST
5823  025054  104410                                                    TRAP    C$ESCAPE
5824  025056  000040                                                    .WORD   L10046-.
5825
5826  025060  004537  010110            JSR     R5,RXCHAR       ;READ AND CHECK FOR TXDB 125 SYNC
5827  025064  000125                    125
5828  025066  000000                    0
5829  025070  000010                    8.
5830  025072  103003                    BCC     .+8.            ;BR IF NO ERROR
5831  025074                            ERROR                   ;REPORT STACKED ERROR
5832  025074  104460                                                    TRAP    C$ERROR
5833  025076                            ESCAPE  TST             ;SKIP TO END OF TEST
5834  025076  104410                                                    TRAP    C$ESCAPE
5835  025100  000016                                                    .WORD   L10046-.
5836
5837  025102  004537  005126            JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
5838  025106  103003                    BCC     .+8.            ;BR IF NO ERROR
5839  025110                            ERROR                   ;REPORT STACKED ERROR
5840  025110  104460                                                    TRAP    C$ERROR
5841  025112                            ESCAPE TST              ;SKIP TO END OF TEST
5842  025112  104410                                                    TRAP    C$ESCAPE
5843  025114  000002                                                    .WORD   L10046-.
5844  025116                    ENDTST
5845  025116                                                    L10046:
5846  025116  104401                                                    TRAP    C$ETST
5847
```

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

                                        .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
025120                                                                  T16::
025120  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

025124  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
025130  047626                          DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
025132  040000                          NORXEN                  ;USE 8 BIT CHARS, RECEIVER DISABLED
025134  103003                          BCC     .+8.            ;BR IF NO ERROR
025136                                  ERROR                   ;REPORT STACKED ERROR
025136  104460                                                          TRAP    C$ERROR
025140                                  ESCAPE  TST             ;SKIP TO END OF TEST
025140  104410                                                          TRAP    C$ESCAPE
025142  000462                                                          .WORD   L10047-.

025144  004537  007116                  JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "SYNCH"
025150  000000                          0                       ; CHARACTER (SHOULD BE 0)
025152  103003                          BCC     .+8.            ;BR IF NO ERROR
025154                                  ERROR                   ;REPORT STACKED ERROR
025154  104460                                                          TRAP    C$ERROR
025156                                  ESCAPE  TST             ;AND EXIT TEST
025156  104410                                                          TRAP    C$ESCAPE
025160  000444                                                          .WORD   L10047-.

025162  004537  007256                  JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF "SYNCH" CHARACTER
025166  000007                          7.                      ; (OFF OF TSO BIT)
025170  000151                          151                     ; EXPECTED BIT SEQUENCE (0010110)
025172  103003                          BCC     .+8.            ;BR IF NO ERROR
025174                                  ERROR                   ;REPORT STACKED ERROR
025174  104460                                                          TRAP    C$ERROR
025176                                  ESCAPE  TST             ;AND EXIT TEST
025176  104410                                                          TRAP    C$ESCAPE
025200  000424                                                          .WORD   L10047-.
```

```
5904  025202  004537  003734                JSR     R5,WRITEI       ;LOAD 1ST DATA CHARACTER (000)
5905  025206  120402                        TDSRL
5906  025210  000000                        000
5907
5908  025212  004537  010010                JSR     R5,TXCTRL       ;CLEAR TSOM
5909  025216  000000                        000
5910  025220  000000                        0
5911
5912  025222  004537  007256                JSR     R5,SERIAL       ;READ 2ND SYNCH CHARACTER VIA TSO
5913  025226  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
5914  025230  000151                        151                     ; EXPECTED BIT SEQUENCE (010010110)
5915  025232  103003                        BCC     .+8.            ;BR IF NO ERROR
5916  025234                                ERROR                   ;REPORT STACKED ERROR
5917  025234  104460                                                                TRAP    C$ERROR
5918  025236                                ESCAPE  TST             ;AND EXIT TEST
5919  025236  104410                                                                TRAP    C$ESCAPE
5920  025240  000364                                                                .WORD   L10047-.
5921                                ;-----------------------------------------------------------------
5922                                ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
5923                                ;-----------------------------------------------------------------
5924  025242  004537  005432                JSR     R5,CKUSTS       ;CHECK USYRT STATUS
5925  025246  000114                        TXACT!TBMT!TSO          ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
5926  025250  103003                        BCC     .+8.            ;BR IF NO ERROR
5927  025252                                ERROR                   ;REPORT STACKED ERROR
5928  025252  104460                                                                TRAP    C$ERROR
5929  025254                                ESCAPE  TST             ;SKIP TO END OF TEST
5930  025254  104410                                                                TRAP    C$ESCAPE
5931  025256  000346                                                                .WORD   L10047-.
5932
5933  025260  004537  003610                JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
5934  025264  120403                        TDSRH                   ; TERR BIT IS IN TDSRH
5935  025266  000000                10$:    000                     ; TDSRH STORED HERE
5936  025270  133727  025266  000200        BITB    10$,#BIT7       ;CHECK TERR BIT
5937  025276  001406                        BEQ     15$             ; BR IF TERR NOT SET
5938  025300                                GEDF    EM98,ERR12      ;REPORT ERROR: "TERR NOT CLEAR"
5939                                                                ;       "DEVICE FATAL" ERROR # 49
5940  025300  104455                                                                TRAP    C$ERDF
5941  025302  000061                                                                .WORD   49
5942  025304  014357                                                                .WORD   EM98
5943  025306  017722                                                                .WORD   ERR12
5944  025310                                ESCAPE  TST             ;AND EXIT TEST
5945  025310  104410                                                                TRAP    C$ESCAPE
5946  025312  000312                                                                .WORD   L10047-.
5947                                ;-----------------------------   -----------------------------------
5948                                ; NOW READ 1ST DATA CHARACTER (  ) CHECK TERR=1,TXU=1)
5949                                ;-----------------------------------------------------------------
5950  025314  004537  007256        15$:    JSR     R5,SERIAL       ;READ DATA CHARACTER (000) VIA TSO
5951  025320  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
5952  025322  000000                        000                     ; EXPECTED BIT SEQUENCE (00000000)
5953  025324  103003                        BCC     .+8.            ;BR IF NO ERROR
5954  025326                                ERROR                   ;REPORT STACKED ERROR
5955  025326  104460                                                                TRAP    C$ERROR
5956  025330                                ESCAPE  TST             ;AND EXIT TEST
5957  025330  104410                                                                TRAP    C$ESCAPE
5958  025332  000272                                                                .WORD   L10047-.
5959
```

```
 5960  025334  004537  011614                JSR     R5,STEPLU       ;GENERATE 1 TICK TO UNDERRUN TX
 5961  025340  000001                        1
 5962
 5963  025342  004537  005432                JSR     R5,CKUSTS       ;CHECK USYRT STATUS
 5964  025346  000116                        TBMT!TSO!TXACT!TXU      ; FOR TBMT AND TX UNDERRUN
 5965  025350  103003                        BCC     .+8.            ;BR IF NO ERROR
 5966  025352                                ERROR                   ;REPORT STACKED ERROR
 5967  025352  104460                                                                TRAP    C$ERROR
 5968  025354                                ESCAPE  TST             ;SKIP TO END OF TEST
 5969  025354  104410                                                                TRAP    C$ESCAPE
 5970  025356  000246                                                                .WORD   L10047-.
 5971
 5972  025360  004537  003610                JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
 5973  025364  120403                        TDSRH                   ; TERR BIT IS IN TDSRH
 5974  025366  000000                20$:    000                     ; TDSRH STORED HERE
 5975  025370  133727  025366  000200        BITB    20$,#BIT7       ;CHECK TERR BIT
 5976  025376  001006                        BNE     25$             ; BR IF TERR IS SET
 5977  025400                                GEDF    EM99,ERR12      ;REPORT ERROR: "TERR NOT SET"
 5978                                                                ;       "DEVICE FATAL" ERROR # 50
 5979  025400  104455                                                                TRAP    C$ERDF
 5980  025402  000062                                                                .WORD   50
 5981  025404  014404                                                                .WORD   EM99
 5982  025406  017722                                                                .WORD   ERR12
 5983  025410                                ESCAPE  TST             ;AND EXIT TEST
 5984  025410  104410                                                                TRAP    C$ESCAPE
 5985  025412  000212                                                                .WORD   L10047-.
 5986                                ;---------------------------------------------------------------
 5987                                ; NOW CHECK MARK CHARACTER (RESULT OF UNDERRUNNING TRANSMITTER)
 5988                                ;---------------------------------------------------------------
 5989  025414  004537  007116        25$:    JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "MARK"
 5990  025420  000001                        1                       ; CHARACTER (BIT SHOULD=1)
 5991  025422  103003                        BCC     .+8.            ;BR IF NO ERROR
 5992  025424                                ERROR                   ;REPORT STACKED ERROR
 5993  025424  104460                                                                TRAP    C$ERROR
 5994  025426                                ESCAPE  TST             ;AND EXIT TEST
 5995  025426  104410                                                                TRAP    C$ESCAPE
 5996  025430  000174                                                                .WORD   L10047-.
 5997
 5998  025432  004537  007256                JSR     R5,SERIAL       ;READ MARK CHARACTER VIA SO
 5999  025436  000007                        7                       ; 7 BIT CHAR/CLOCK TICKS
 6000  025440  000177                        177                     ; EXPECTED BIT SEQUENCE (1111111)
 6001  025442  103003                        BCC     .+8.            ;BR IF NO ERROR
 6002  025444                                ERROR                   ;REPORT STACKED ERROR
 6003  025444  104460                                                                TRAP    C$ERROR
 6004  025446                                ESCAPE  TST             ;AND EXIT TEST
 6005  025446  104410                                                                TRAP    C$ESCAPE
 6006  025450  000154                                                                .WORD   L10047-.
 6007                                ;---------------------------------------------------------------
 6008                                ; SET IDLE=0, VERIFY SYNCH CHARACTER IS OUTPUT
 6009                                ;---------------------------------------------------------------
 6010  025452  004537  003734                JSR     R5,WRITEI       ;SET IDLE BIT=0
 6011  025456  120405                        PCSARH
 6012  025460  000107                        PROTO!XYZ
 6013
 6014  025462  004537  007256                JSR     R5,SERIAL       ;READ MARK CHARACTER VIA TSO
 6015  025466  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
```

```
6016  025470  000377                    377                     ; EXPECTED BIT SEQUENCE (11111111)
6017  025472  103003                    BCC     .+8.            ;BR IF NO ERROR
6018  025474                            ERROR                   ;REPORT STACKED ERROR
6019  025474  104460                                                            TRAP    C$ERROR
6020  025476                            ESCAPE  TST             ;AND EXIT TEST
6021  025476  104410                                                            TRAP    C$ESCAPE
6022  025500  000124                                                            .WORD   L10047-.
6023
6024  025502  004537  007256            JSR     R5,SERIAL       ;READ SYNCH CHARACTER VIA TSO
6025  025506  000010                    8.                      ; 8 BIT CHAR/CLOCK TICKS
6026  025510  000151                    151                     ; EXPECTED BIT SEQUENCE (010010110)
6027  025512  103003                    BCC     .+8.            ;BR IF NO ERROR
6028  025514                            ERROR                   ;REPORT STACKED ERROR
6029  025514  104460                                                            TRAP    C$ERROR
6030  025516                            ESCAPE  TST             ;AND EXIT TEST
6031  025516  104410                                                            TRAP    C$ESCAPE
6032  025520  000104                                                            .WORD   L10047-.
6033
6034  025522  004537  010010            JSR     R5,TXCTRL       ;* ASSERT TSOM (SHOULD CLEAR TXU,TERR)
6035  025526  000001                    TSOM                    ;*
6036  025530  000000                    0                       ;*
6037                                ;-----------------------------------------------------------------
6038                                ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
6039                                ;-----------------------------------------------------------------
6040  025532  004537  005432            JSR     R5,CKUSTS       ;CHECK USYRT STATUS
6041  025536  000014                    TXACT!TSO               ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
6042  025540  103003                    BCC     .+8.            ;BR IF NO ERROR
6043  025542                            ERROR                   ;REPORT STACKED ERROR
6044  025542  104460                                                            TRAP    C$ERROR
6045  025544                            ESCAPE  TST             ;SKIP TO END OF TEST
6046  025544  104410                                                            TRAP    C$ESCAPE
6047  025546  000056                                                            .WORD   L10047-.
6048
6049  025550  004537  003610            JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
6050  025554  120403                    TDSRH                   ; TERR BIT IS IN TDSRH
6051  025556  000000                17$: 000                     ; TDSPH STORED HERE
6052  025560  133727  025556  000200    BITB    17$,#BIT7       ;CHECK TERR BIT
6053  025566  001406                    BEQ     18$             ; BR IF TERR NOT SET
6054  025570                            GEDF    EM98,ERR12      ;REPORT ERROR: "TERR NOT CLEAR"
6055                                                            ;       "DEVICE FATAL" ERROR # 51
6056  025570  104455                                                            TRAP    C$ERDF
6057  025572  000063                                                            .WORD   51
6058  025574  014357                                                            .WORD   EM98
6059  025576  017722                                                            .WORD   ERR12
6060  025600                            ESCAPE  TST             ;AND EXIT TEST
6061  025600  104410                                                            TRAP    C$ESCAPE
6062  025602  000022                                                            .WORD   L10047-.
6063                                ;-----------------------------------------------------------------
6064                                ;READ/CHECK FOR SYNCH CHARACTER
6065                                ;-----------------------------------------------------------------
6066  025604  004537  007256        18$: JSR     R5,SERIAL       ;READ !SYNCH! CHARACTER VIA TSO
6067  025610  000010                    8.                      ; 8 BIT CHAR/CLOCK TICKS
6068  025612  000151                    151                     ; EXPECTED BIT SEQUENCE (10010110)
6069  025614  103003                    BCC     .+8.            ;BR IF NO ERROR
6070  025616                            ERROR                   ;REPORT STACKED ERROR
6071  025616  104460                                                            TRAP    C$ERROR
```

```
6072  025620                          ESCAPE  TST             ;AND EXIT TEST
6073  025620  104410                                                  TRAP    C$ESCAPE
6074  025622  000002                                                  .WORD   L10047-.
6075  025624                  ENDTST
6076  025624                                                  L10047:
6077  025624  104401                                                  TRAP    C$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
025626                                                                          T17::
025626  004737  005420                          JSR     PC,INIDMV           ;INIT DMV-11. ENTER M-LOOP

025632  004537  007400                          JSR     R5,INITRN           ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
025636  047626                                  DDCMP!IDLES!NOCHK!SYNCH ;SET DDCMP,STRIP,IDLE,NO CHECK,SYNCH=226
025640  040000                                  NORXEN                  ;USE 8 BIT CHARS. RECEIVER DISABLED
025642  103003                                  BCC     .+8.            ;BR IF NO ERROR
025644                                          ERROR                   ;REPORT STACKED ERROR
025644  104460                                                                          TRAP    C$ERROR
025646                                          ESCAPE  TST             ;SKIP TO END OF TEST
025646  104410                                                                          TRAP    C$ESCAPE
025650  000354                                                                          .WORD   L10050-.

025652  004537  007116                          JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "SYNCH"
025656  000000                                  0                       ; CHARACTER (SHOULD BE 0)
025660  103003                                  BCC     .+8.            ;BR IF NO ERROR
025662                                          ERROR                   ;REPORT STACKED ERROR
025662  104460                                                                          TRAP    C$ERROR
025664                                          ESCAPE  TST             ;AND EXIT TEST
025664  104410                                                                          TRAP    C$ESCAPE
025666  000336                                                                          .WORD   L10050-.

025670  004537  007256                          JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF "SYNCH" CHARACTER
025674  000007                                  7.                      ; (OFF OF TSO BIT)
025676  000151                                  151                     ; EXPECTED BIT SEQUENCE (0010110)
025700  103003                                  BCC     .+8.            ;BR IF NO ERROR
025702                                          ERROR                   ;REPORT STACKED ERROR
025702  104460                                                                          TRAP    C$ERROR
025704                                          ESCAPE  TST             ;AND EXIT TEST
025704  104410                                                                          TRAP    C$ESCAPE
025706  000316                                                                          .WORD   L10050-.

025710  004537  003734                          JSR     R5,WRITEI       ;LOAD 1ST DATA CHARACTER (000)
025714  120402                                  TDSRL
```

```
6134  025716  000000                          000
6135
6136  025720  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
6137  025724  000000                          000
6138  025726  000000                          0
6139
6140  025730  004537  007256                  JSR     R5,SERIAL       ;READ 2ND SYNCH CHARACTER VIA TSO
6141  025734  000010                          8.                      ; 8 BIT CHAR/CLOCK TICKS
6142  025736  000151                          151                     ; EXPECTED BIT SEQUENCE (010010110)
6143  025740  103003                          BCC     .+8.            ;BR IF NO ERROR
6144  025742                                  ERROR                   ;REPORT STACKED ERROR
6145  025742  104460                                                                  TRAP    C$ERROR
6146  025744                                  ESCAPE  TST             ;AND EXIT TEST
6147  025744  104410                                                                  TRAP    C$ESCAPE
6148  025746  000256                                                                  .WORD   L10050-.
6149                                  ;---------------- ------------------------------------------------------
6150                                  ; VERIFY THAT TXU AND TERR BITS ARE ZERO AT THIS POINT
6151                                  ;-------------------------------------------------------------------
6152  025750  004537  005432                  JSR     R5,CKUSTS       ;CHECK USYRT STATUS
6153  025754  000114                          TXACT!TBMT!TSO          ; FOR TRANSMITTER ACTIVE (NO TXU YET!)
6154  025756  103003                          BCC     .+8.            ;BR IF NO ERROR
6155  025760                                  ERROR                   ;REPORT STACKED ERROR
6156  025760  104460                                                                  TRAP    C$ERROR
6157  025762                                  ESCAPE  TST             ;SKIP TO END OF TEST
6158  025762  104410                                                                  TRAP    C$ESCAPE
6159  025764  000240                                                                  .WORD   L10050-.
6160
6161  025766  004537  003610                  JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
6162  025772  120403                          TDSRH                   ; TERR BIT IS IN TDSRH
6163  025774  000000                  10$:    000                     ; TDSRH STORED HERE
6164  025776  133727  025774  000200          BITB    10$,#BIT7       ;CHECK TERR BIT
6165  026004  001406                          BEQ     15$             ; BR IF TERR NOT SET
6166  026006                                  GEDF    EM98,ERR12      ;REPORT ERROR: "TERR NOT CLEAR"
6167                                                                  ;       "DEVICE FATAL" ERROR # 52
6168  026006  104455                                                                  TRAP    C$ERDF
6169  026010  000064                                                                  .WORD   52
6170  026012  014357                                                                  .WORD   EM98
6171  026014  017722                                                                  .WORD   ERR12
6172  026016                                  ESCAPE  TST             ;AND EXIT TEST
6173  026016  104410                                                                  TRAP    C$ESCAPE
6174  026020  000204                                                                  .WORD   L10050-.
6175                                  ;-------------------------------------------------------------------
6176                                  ; NOW READ 1ST DATA CHARACTER (AND CHECK TERR=1,TXU=1)
6177                                  ;-------------------------------------------------------------  ----
6178  026022  004537  007256          15$:    JSR     R5,SERIAL       ;READ DATA CHARACTER (000) VIA TSO
6179  026026  000010                          8.                      ; 8 BIT CHAR/CLOCK TICKS
6180  026030  000000                          000                     ; EXPECTED BIT SEQUENCE (00000000)
6181  026032  103003                          BCC     .+8.            ;BR IF NO ERROR
6182  026034                                  ERROR                   ;REPORT STACKED ERROR
6183  026034  104460                                                                  TRAP    C$ERROR
6184  026036                                  ESCAPE  TST             ;AND EXIT TEST
6185  026036  104410                                                                  TRAP    C$ESCAPE
6186  026040  000164                                                                  .WORD   L10050-.
6187
6188  026042  004537  011614                  JSR     R5,STEPLU       ;GENERATE 1 TICK TO UNDERRUN TX
6189  026046  000001                          1
```

```
6190
6191  026050  004537  005432          JSR     R5,CKUSTS       ;CHECK USYRT STATUS
6192  026054  000116                  TBMT!TSO!TXACT!TXU      ; FOR TBMT AND TX UNDERRUN
6193  026056  103003                  BCC     .+8.            ;BR IF NO ERROR
6194  026060                          ERROR                   ;REPORT STACKED ERROR
6195  026060  104460                                                  TRAP    C$ERROR
6196  026062                          ESCAPE  TST             ;SKIP TO END OF TEST
6197  026062  104410                                                  TRAP    C$ESCAPE
6198  026064  000140                                                  .WORD   L10050-.
6199
6200  026066  004537  003610          JSR     R5,READI        ;GET TX ERROR BIT (TO SEE IF SET)
6201  026072  120403                  TDSRH                   ; TERR BIT IS IN TDSRH
6202  026074  000000          20$:    000                     ; TDSRH STORED HERE
6203  026076  133727  026074  000200  BITB    20$,@BIT7       ;CHECK TERR BIT
6204  026104  001006                  BNE     25$             ; BR IF TERR IS SET
6205  026106                          GEDF    EM99,ERR12      ;REPORT ERROR: "TERR NOT SET"
6206                                                          ;       "DEVICE FATAL" ERROR # 53
6207  026106  104455                                                  TRAP    C$ERDF
6208  026110  000065                                                  .WORD   53
6209  026112  014404                                                  .WORD   EM99
6210  026114  017722                                                  .WORD   ERR12
6211  026116                          ESCAPE  TST             ;AND EXIT TEST
6212  026116  104410                                                  TRAP    C$ESCAPE
6213  026120  000104                                                  .WORD   L10050-.
6214                          ;-----------------------------------------------------------
6215                          ; NOW CHECK MARK CHARACTER (RESULT OF UNDERRUNNING TRANSMITTER)
6216                          ;-----------------------------------------------------------
6217  026122  004537  003734  25$:     JSR     R5,WRITEI       ;CLEAR IDLE BIT
6218  026126  120405                  PCSARH
6219  026130  000107                  BIT6!BIT2!BIT1!BIT0     ;DDCMP FORMAT, NO ERROR CHECKING
6220
6221  026132  004537  007116          JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "MARK"
6222  026136  000001                  1                       ; CHARACTER (BIT SHOULD=1)
6223  026140  103003                  BCC     .+8.            ;BR IF NO ERROR
6224  026142                          ERROR                   ;REPORT STACKED ERROR
6225  026142  104460                                                  TRAP    C$ERROR
6226  026144                          ESCAPE  TST             ;AND EXIT TEST
6227  026144  104410                                                  TRAP    C$ESCAPE
6228  026146  000056                                                  .WORD   L10050-.
6229
6230  026150  004537  007256          JSR     R5,SERIAL       ;READ MARK CHARACTER VIA TSO
6231  026154  000007                  7                       ; 7 BIT CHAR/CLOCK TICKS
6232  026156  000177                  177                     ; EXPECTED BIT SEQUENCE (1111111)
6233  026160  103003                  BCC     .+8.            ;BR IF NO ERROR
6234  026162                          ERROR                   ;REPORT STACKED ERROR
6235  026162  104460                                                  TRAP    C$ERROR
6236  026164                          ESCAPE  TST             ;AND EXIT TEST
6237  026164  104410                                                  TRAP    C$ESCAPE
6238  026166  000036                                                  .WORD   L10050-.
6239
6240                          ;-----------------------------------------------------------
6241                          ; READ/CHECK FOR SYNCH, RESET/CHECK USYRT REGISTERS
6242                          ;-----------------------------------------------------------
6243  026170  004537  007256          JSR     R5,SERIAL       ;READ SYNCH CHARACTER VIA TSO
6244  026174  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6245  026176  000151                  151                     ; EXPECTED BIT SEQUENCE (00000000)
```

```
 6246  026200  103003                  BCC     .+8.            ;BR IF NO ERROR
 6247  026202                          ERROR                   ;REPORT STACKED ERROR
 6248  026202  104460                                                  TRAP    C$ERROR
 6249  026204                          ESCAPE  TST             ;AND EXIT TEST
 6250  026204  104410                                                  TRAP    C$ESCAPE
 6251  026206  000016                                                  .WORD   L10050-.
 6252
 6253  026210  004537  005126          JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
 6254  026214  103003                  BCC     .+8.            ;BR IF NO ERROR
 6255  026216                          ERROR                   ;REPORT STACKED ERROR
 6256  026216  104460                                                  TRAP    C$ERROR
 6257  026220                          ESCAPE TST              ;SKIP TO END OF TEST
 6258  026220  104410                                                  TRAP    C$ESCAPE
 6259  026222  000002                                                  .WORD   L10050-.
 6260  026224                  ENDTST
 6261  026224                                                  L10050:
 6262  026224  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
026226                                                                          T18::

026226  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

026232  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026236  043626                          DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
026240  000000                          0                       ;USE 8 BIT CHARS
026242  103003                          BCC     .+8.            ;BR IF NO ERROR
026244                                  ERROR                   ;REPORT STACKED ERROR
026244  104460                                                                  TRAP    C$ERROR
026246                                  ESCAPE  TST             ;SKIP TO END OF TEST
026246  104410                                                                  TRAP    C$ESCAPE
026250  000162                                                                  .WORD   L10051-.

026252  004537  010010                  JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
026256  000001                          TSOM                    ;AND KNOCK DOWN TBMT
026260  000007                          7.

026262  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM (GET READY TO SEND DATA)
026266  000000                          000
026270  000000                          0

026272  004537  007676                  JSR     R5,TXCHAR       ;LOAD 125, TX 2ND SYNCH
026276  000125                          125
026300  000010                          8.
026302  103003                          BCC     .+8.            ;BR IF NO ERROR
026304                                  ERROR                   ;REPORT STACKED ERROR
026304  104460                                                                  TRAP    C$ERROR
026306                                  ESCAPE  TST             ;SKIP TO END OF TEST
026306  104410                                                                  TRAP    C$ESCAPE
026310  000122                                                                  .WORD   L10051-.

026312  004537  007676                  JSR     R5,TXCHAR       ;LOAD 252
026316  000252                          252
026320  000000                          0
026322  103003                          BCC     .+8.            ;BR IF NO ERROR
026324                                  ERROR                   ;REPORT STACKED ERROR
026324  104460                                                                  TRAP    C$ERROR
```

```
6319  026326                                ESCAPE  TST             ;SKIP TO END OF TEST
6320  026326  104410                                                                TRAP    C$ESCAPE
6321  026330  000102                                                                .WORD   L10051-.
6322
6323  026332  004537  003734                JSR     R5,WRITEI       ;DROP TRANSMIT ENABLE (TXE)
6324  026336  170000                        VIAORB
6325  026340  000102                        RXEN!TTLOOP
6326
6327  026342  004537  011614                JSR     R5,STEPLU       ;CLOCK IN 125
6328  026346  000007                        7.                      ;***
6329
6330  026350  004537  011364                JSR     R5,RCV1ST       ;CLOCK AND RCV 125
6331  026354  000000                        0
6332  026356  103003                        BCC     .+8.            ;BR IF NO ERROR
6333  026360                                ERROR                   ;REPORT STACKED ERROR
6334  026360  104460                                                                TRAP    C$ERROR
6335  026362                                ESCAPE  TST             ;SKIP TO END OF TEST
6336  026362  104410                                                                TRAP    C$ESCAPE
6337  026364  000046                                                                .WORD   L10051-.
6338
6339  026366  004537  010110                JSR     R5,RXCHAR       ;READ AND CHECK FOR 125
6340  026372  000125                        125
6341  026374  000000                        0
6342  026376  000010                        8.                      ;AND CLOCK IN NEXT CHAR.
6343  026400  103003                        BCC     .+8.            ;BR IF NO ERROR
6344  026402                                ERROR                   ;REPORT STACKED ERROR
6345  026402  104460                                                                TRAP    C$ERROR
6346  026404                                ESCAPE  TST             ;SKIP TO END OF TEST
6347  026404  104410                                                                TRAP    C$ESCAPE
6348  026406  000024                                                                .WORD   L10051-.
6349
6350  026410  004537  010110                JSR     R5,RXCHAR       ;READ AND CHECK FOR 377
6351  026414  000377                        377                     ;** IF 252 IS READ, THEN DROPPING
6352  026416  000000                        0                       ;** TXEN DIDN'T WORK !!!
6353  026420  000010                        8.
6354  026422  103003                        BCC     .+8.            ;BR IF NO ERROR ****
6355  026424                                ERROR                   ;REPORT STACKED ERROR ****
6356  026424  104460                                                                TRAP    C$ERROR
6357  026426                                ESCAPE  TST             ;SKIP TO END OF TEST ****
6358  026426  104410                                                                TRAP    C$ESCAPE
6359  026430  000002                                                                .WORD   L10051-.
6360  026432                        ENDTST
6361  026432                                                                L10051:
6362  026432  104401                                                                TRAP    C$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
                                ;* ROA = 0).
                                ;*  ALSO NOTE THAT DUE TO FIFO TIMING, TWO 'HALF CHARACTERS' ARE LOADED
                                ;* INTO THE FIFO (THE 1ST AND LAST CHARACTERS).
                                ;*
                                ;*
                                ;---*********************************************************************
                                ;
                                ;       BGNTST
026434                                                                  T19::
026434  004737  005420                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
026440  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026444  043626                          DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
026446  000000                          0                       ;USE 8 BIT CHARS
026450  103003                          BCC     .+8.            ;BR IF NO ERROR
026452                                  ERROR                   ;REPORT STACKED ERROR
026452  104460                                                          TRAP    C$ERROR
026454                                  ESCAPE  TST             ;SKIP TO END OF TEST
026454  104410                                                          TRAP    C$ESCAPE
026456  000216                                                          .WORD   L10052-.

026460  004537  010010                  JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
026464  000001                          TSOM                    ;
026466  000007                          7.
026470  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
026474  000000                          000
026476  000000                          0
                                ;---------------------------------------------------------
                                ; FILL THE FIFO WITH CHARACTERS (DATA1 - DATA8)
                                ;---------------------------------------------------------
026500  012702  026676                  MOV     #TXTBL3,R2      ;SET UP TABLE POINTER
026504  112237  026514          5$:     MOVB    (R2)+,10$       ;SETUP TRANSMIT CHARACTER

026510  004537  007676                  JSR     R5,TXCHAR       ;TRANSMIT A CHARACTER
026514  000000                  10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
026516  100010                          NCTBMT*256.!8.          ;NO CHECK OF INITIAL TBMT=0
026520  103003                          BCC     .+8.            ;BR IF NO ERROR
026522                                  ERROR                   ;REPORT STACKED ERROR
026522  104460                                                          TRAP    C$ERROR
026524                                  ESCAPE  TST             ;SKIP TO END OF TEST
026524  104410                                                          TRAP    C$ESCAPE
026526  000146                                                          .WORD   L10052-.
```

```
6419  026530  022702  026710                  CMP     #TXEND3,R2
6420  026534  001363                          BNE     5$
6421
6422  026536  004537  011614                  JSR     R5,STEPLU       ;ADD A FEW TX TICKS TO COMPLETELY
6423  026542  000005                          5                       ; FILL UP FIFO.
6424                                  ;-----------------------------------------------------------
6425                                  ; THE FIFO IS NOW FULL OF CHARACTERS.
6426                                  ; NOW READ/VERIFY THEM.
6427                                  ;-----------------------------------------------------------
6428  026544  012702  026676                  MOV     #TXTBL3,R2      ;SET UP TABLE POINTER
6429  026550  112237  026560          15$:    MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
6430
6431  026554  004537  010110                  JSR     R5,RXCHAR       ;READ & CHK CHARACTER
6432  026560  000000          20$:    000                             ;** HOLE FOR EXPECTED RECEIVE CHAR.
6433  026562  000000                          0                       ;
6434  026564  100000                          NOCRDA          ;NO INITIAL CHECK OF RDA=0
6435  026566  103003                          BCC     .+8.            ;BR IF NO ERROR
6436  026570                                  ERROR                   ;REPORT STACKED ERROR
6437  026570  104460                                                          TRAP    C$ERROR
6438  026572                                  ESCAPE  TST             ;SKIP TO END OF TEST
6439  026572  104410                                                          TRAP    C$ESCAPE
6440  026574  000100                                                          .WORD   L10052-.
6441
6442  026576  022702  026706                  CMP     #TXEND3-2,R2
6443  026602  001362                          BNE     15$
6444
6445  026604  004537  011614                  JSR     R5,STEPLU       ;CLOCK IN LAST FEW CHARACTERS OFF
6446  026610  000015                          15                      ; OF FIFO.
6447
6448  026612  004537  010110                  JSR     R5,RXCHAR       ;READ & CHK CHARACTER
6449  026616  000303                          303
6450  026620  000000                          0
6451  026622  100000                          NOCRDA          ;NO INITIAL CHECK OF RDA=0
6452  026624  103003                          BCC     .+8.            ;BR IF NO ERROR
6453  026626                                  ERROR                   ;REPORT STACKED ERROR
6454  026626  104460                                                          TRAP    C$ERROR
6455  026630                                  ESCAPE  TST             ;SKIP TO END OF TEST
6456  026630  104410                                                          TRAP    C$ESCAPE
6457  026632  000042                                                          .WORD   L10052-.
6458
6459  026634  004537  010110                  JSR     R5,RXCHAR       ;READ & CHK CHARACTER
6460  026640  000074                          074
6461  026642  000000                          0
6462  026644  100000                          NOCRDA          ;NO INITIAL CHECK OF RDA=0
6463  026646  103003                          BCC     .+8.            ;BR IF NO ERROR
6464  026650                                  ERROR                   ;REPORT STACKED ERROR
6465  026650  104460                                                          TRAP    C$ERROR
6466  026652                                  ESCAPE  TST             ;SKIP TO END OF TEST
6467  026652  104410                                                          TRAP    C$ESCAPE
6468  026654  000020                                                          .WORD   L10052-.
6469
6470  026656  004537  011532                  JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
6471  026662  000010                          8.
6472  026664  103003                          BCC     .+8.            ;BR IF NO ERROR
6473  026666                                  ERROR                   ;REPORT STACKED ERROR
6474  026666  104460                                                          TRAP    C$EPROR
```

```
 6475  026670                          ESCAPE  TST             ;SKIP TO END OF TEST
 6476  026670  104410                                                  TRAP    C$ESCAPE
 6477  026672  000002                                                  .WORD   L10052-.
 6478  026674                        ENDTST
 6479  026674                                                  L10052:
 6480  026674  104401                                                  TRAP    C$ETST
 6481                                  ;------------------------------
 6482  026676     226                TXTBL3: .BYTE   226     ;SYNCH
 6483  026677     226                        .BYTE   226     ;SYNCH
 6484  026700     000                        .BYTE   000
 6485  026701     377                        .BYTE   377
 6486  026702     125                        .BYTE   125
 6487  026703     252                        .BYTE   252
 6488  026704     347                        .BYTE   347
 6489  026705     030                        .BYTE   030
 6490  026706     303                        .BYTE   303
 6491  026707     074                        .BYTE   074
 6492  026710     000                TXEND3: .BYTE   000
 6493          026712                        .EVEN
 6494                                  ;------------------------------
```

```
                                        TEST 20 -- BCP CHARACTER LENGTH TEST

                                .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::
026712  004737  005420          ;       JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
                                ;-----------------------------------------------------------------
                                ; SUBROUTINE # 1: 5 BIT CHARACTERS
                                ;-----------------------------------------------------------------
                                        BGNSUB
                                                                                T20.1:
                                                                                        TRAP    C$BSUB
026716  104402
026720  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
026724  043626                          DDCMP!NOCHK!SYNCH       ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
026726  000245                          BIT7!BIT5!BIT2!BIT0     ;TXCL=RXCL=5 BITS
026730  103003                          BCC     .+8.            ;BR IF NO ERROR
026732                                  ERROR                   ;REPORT STACKED ERROR
026732  104460                                                          TRAP    C$ERROR
026734                                  ESCAPE  SUB             ;SKIP TO END OF TEST
026734  104410                                                          TRAP    C$ESCAPE
026736  000126                                                          .WORD   L10054-.

026740  004537  010010                  JSR     R5,TXCTRL       ;LOAD 2ND SYNCH,TX 1ST SYNCH
026744  000001                          TSOM
026746  000004                          4.
026750  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
026754  000000                          000
026756  000000                          0
                                ;-----------------------------------------------------------------
026760  012703  027410          10$:    MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
026764  112337  026774          10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER

026770  004537  007676                  JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
026774  000000          1$:     000                             ;** HOLE FOR NEXT CHARACTER **
026776  000005                          5.
027000  103003                          BCC     .+8.            ;BR IF NO ERROR
027002                                  ERROR                   ;REPORT STACKED ERROR
027002  104460                                                          TRAP    C$ERROR
027004                                  ESCAPE  SUB             ;SKIP TO END OF TEST
027004  104410                                                          TRAP    C$ESCAPE
027006  000056                                                          .WORD   L10054-.

027010  022703  027421                  CMP     #T24TBL+9.,R3   ;ALL CHARACTERS TRANSMITTED ?
027014  001363                          BNE     10$             ; IF NOT, TX ANOTHER ONE
```

```
6551                                    ;*****************************************************************
6552  027016  012703  027410                      MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
6553  027022  112337  027040              40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
6554  027026  142737  000340  027040              BICB    #340,4$         ;MASK OUT UNTPANSMITTED BITS
6555
6556  027034  004537  010110                      JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
6557  027040  000000                      4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
6558  027042  000000                              0
6559  027044  100000                              NOCRDA                  ;NO INITIAL CHECK OF RDA=0
6560  027046  103003                              BCC     .+8.            ;BR IF NO ERROR
6561  027050                                      ERROR                   ;REPORT STACKED ERROR
6562  027050  104460                                                                      TRAP    C$ERROR
6563  027052                                      ESCAPE  SUB             ;SKIP TO END OF TEST
6564  027052  104410                                                                      TRAP    C$ESCAPE
6565  027054  000010                                                                      .WORD   L10054-.
6566
6567  027056  022703  027417                      CMP     #T24TBL+7,R3    ;ALL CHARACTERS CHECKED ?
6568  027062  001357                              BNE     40$             ; IF NOT, CHECK ANOTHER ONE
6569
6570  027064                                  ENDSUB
6571  027064                                                                      L10054:
6572  027064  104403                                                                      TRAP    C$ESUB
6573                                    ;-----------------------------------------------------------------
6574                                    ; SUBROUTINE # 2: 6 BIT CHARACTERS
6575                                    ;-----------------------------------------------------------------
6576  027066                                  BGNSUB
6577  027066                                                                      T20.2:
6578  027066  104402                                                                      TRAP    C$BSUB
6579  027070  004537  007400                      JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
6580  027074  043626                              DDCMP!NOCHK!SYNCH       ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
6581  027076  000306                              BIT7!BIT6!BIT2!BIT1     ;TXCL=RXCL=6 BITS
6582  027100  103003                              BCC     .+8.            ;BR IF NO ERROR
6583  027102                                      ERROR                   ;REPORT STACKED ERROR
6584  027102  104460                                                                      TRAP    C$ERROR
6585  027104                                      ESCAPE  SUB             ;SKIP TO END OF TEST
6586  027104  104410                                                                      TRAP    C$ESCAPE
6587  027106  000126                                                                      .WORD   L10055-.
6588
6589  027110  004537  010010                      JSR     R5,TXCTRL       ;LOAD 2ND SYNCH,TX 1ST SYNCH
6590  027114  000001                              TSOM
6591  027116  000005                              5.
6592  027120  004537  010010                      JSR     R5,TXCTRL       ;CLEAR TSOM
6593  027124  000000                              000
6594  027126  000000                              0
6595                                    ;*****************************************************************
6596  027130  012703  027410                      MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
6597  027134  112337  027144              20$:    MOVB    (R3)+,2$        ;INSTALL NEXT TX CHARACTER
6598
6599  027140  004537  007676                      JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
6600  027144  000000                      2$:     000                     ;** HOLE FOR NEXT CHARACTER **
6601  027146  000006                              6.
6602  027150  103003                              BCC     .+8.            ;BR IF NO ERROR
6603  027152                                      ERROR                   ;REPORT STACKED ERROR
6604  027152  104460                                                                      TRAP    C$ERROR
6605  027154                                      ESCAPE  SUB             ;SKIP TO END OF TEST
6606  027154  104410                                                                      TRAP    C$ESCAPE
```

```
 6607  027156  000056                                                          .WORD   L10055-.
 6608
 6609  027160  022703  027422            CMP     #T24TBL+10.,R3  ;ALL CHARACTERS TRANSMITTED ?
 6610  027164  001363                    BNE     20$             ; IF NOT, TX ANOTHER ONE
 6611                                    ;-------------------------------------------------------------
 6612  027166  012703  027410            MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
 6613  027172  112337  027210    50$:    MOVB    (R3)+,5$        ;INSTALL NEXT EXPECTED RX CHARACTER
 6614  027176  142737  000300  027210    BICB    #300,5$         ;MASK OUT UNTRANSMITTED BITS
 6615
 6616  027204  004537  010110            JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
 6617  027210  000000            5$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
 6618  027212  000000                    0
 6619  027214  100000                    NOCRDA                  ;NO INITIAL CHECK OF RDA=0
 6620  027216  103003                    BCC     .+8.            ;BR IF NO ERROR
 6621  027220                            ERROR                   ;REPORT STACKED ERROR
 6622  027220  104460                                                          TRAP    C$ERROR
 6623  027222                            ESCAPE  SUB             ;SKIP TO END OF TEST
 6624  027222  104410                                                          TRAP    C$ESCAPE
 6625  027224  000010                                                          .WORD   L10055-.
 6626
 6627  027226  022703  027417            CMP     #T24TBL+7,R3    ;ALL CHARACTERS CHECKED ?
 6628  027232  001357                    BNE     50$             ; IF NOT, CHECK ANOTHER ONE
 6629
 6630  027234                          ENDSUB
 6631  027234                                                          L10055:
 6632  027234  104403                                                          TRAP    C$ESUB
 6633                                    ;-------------------------------------------------------------
 6634                                    ; SUBROUTINE # 3: 7 BIT CHARACTERS
 6635                                    ;-------------------------------------------------------------
 6636  027236                            BGNSUB
 6637  027236                                                          T20.3:
 6638  027236  104402                                                          TRAP    C$BSUB
 6639  027240  004537  007400            JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
 6640  027244  043626                    DDCMP!NOCHK!SYNCH       ;SET CHAR MODE,NO ERROR CHECKING,S/AR=226
 6641  027246  000347                    TXDL!RXDL               ;TXCL=RXCL=7 BITS
 6642  027250  103003                    BCC     .+8.            ;BR IF NO ERROR
 6643  027252                            ERROR                   ;REPORT STACKED ERROR
 6644  027252  104460                                                          TRAP    C$ERROR
 6645  027254                            ESCAPE  SUB             ;SKIP TO END OF TEST
 6646  027254  104410                                                          TRAP    C$ESCAPE
 6647  027256  000126                                                          .WORD   L10056-.
 6648
 6649  027260  004537  010010            JSR     R5,TXCTRL       ;LOAD 2ND SYNCH,TX 1ST SYNCH
 6650  027264  000001                    TSOM
 6651  027266  000006                    6.
 6652  027270  004537  010010            JSR     R5,TXCTRL       ;CLEAR TSOM
 6653  027274  000000                    000
 6654  027276  000000                    0
 6655                                    ;-------------------------------------------------------------
 6656  027300  012703  027410            MOV     #T24TBL,R3      ;SET UP DATA TABLE POINTER
 6657  027304  112337  027314    30$:    MOVB    (R3)+,3$        ;INSTALL NEXT TX CHARACTER
 6658
 6659  027310  004537  007676            JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
 6660  027314  000000            3$:     000                     ;** HOLE FOR NEXT CHARACTER **
 6661  027316  000007                    7.
 6662  027320  103003                    BCC     .+8.            ;BR IF NO ERROR
```

```
027322                                  ERROR                           ;REPORT STACKED ERROR
027322  104460                                                                                  TRAP    C$ERROR
027324                                  ESCAPE  SUB                     ;SKIP TO END OF TEST
027324  104410                                                                                  TRAP    C$ESCAPE
027326  000056                                                                                  .WORD   L10056-.

027330  022703  027421                  CMP     #T24TBL+9.,R3           ;ALL CHARACTERS TRANSMITTED ?
027334  001363                          BNE     30$                     ; IF NOT, TX ANOTHER ONE
                                ;***************************************************************
027336  012703  027410                  MOV     #T24TBL,R3              ;SET UP DATA TABLE POINTER
027342  112337  027360          60$:    MOVB    (R3)+,6$                ;INSTALL NEXT EXPECTED RX CHARACTER
027346  142737  000200  027360          BICB    #200,6$                 ;MASK OUT UNTRANSMITTED BITS

027354  004537  010110                  JSR     R5,RXCHAR               ;READ/CHECK NEXT CHARACTER
027360  000000                  6$:     000                             ;** HOLE FOR NEXT EXPECTED CHARACTER
027362  000000                          0
027364  100000                          NOCRDA                          ;NO INITIAL CHECK OF RDA=0
027366  103003                          BCC     .+8.                    ;BR IF NO ERROR
027370                                  ERROR                           ;REPORT STACKED ERROR
027370  104460                                                                                  TRAP    C$ERROR
027372                                  ESCAPE  SUB                     ;SKIP TO END OF TEST
027372  104410                                                                                  TRAP    C$ESCAPE
027374  000010                                                                                  .WORD   L10056-.

027376  022703  027417                  CMP     #T24TBL+7,R3            ;ALL CHARACTERS CHECKED ?
027402  001357                          BNE     60$                     ; IF NOT, CHECK ANOTHER ONE

027404                              ENDSUB
027404                                                                                  L10056:
027404  104403                                                                                  TRAP    C$ESUB
027406                          ENDTST
027406                                                                                  L10053:
027406  104401                                                                                  TRAP    C$ETST

                                ;------------------------------
027410     111                  T24TBL: .BYTE   111     ;D1
027411     222                          .BYTE   222     ;D2
027412     333                          .BYTE   333     ;D3
027413     044                          .BYTE   044     ;D4
027414     155                          .BYTE   155     ;D5
027415     266                          .BYTE   266     ;D6
027416     377                          .BYTE   377     ;D7
027417     000                          .BYTE   000     ;FILLER 1
027420     000                          .BYTE   000     ;FILLER 2
027421     000                          .BYTE   000     ;FILLER 3
                                        .EVEN
                                ;------------------------------
```

```
                                .SBTTL  TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST

                                ;+*********************************************************************
                                ;*
                                ;*      TEST 21 -- BOP TX TABORT/(IDLE = 0) TEST
                                ;*
                                ;*  THE USYRT IS INITIALIZED FOR "BIT-ORIENTED PROTOCOL" (BOP) WITH IDLE
                                ;*  SET TO ZERO. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
                                ;*  OBSERVING TXA -- 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. THE TRANSMITTER IS CLOCKED THROUGH
                                ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
                                ;*  ABORT TO BE TRANSMITTED. 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 ABORT CHARACTER IS SENT (INTO THE BIT
                                ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
                                ;*  THE SEQUENCE.
                                ;*
                                ;--*********************************************************************
                                ;
                                ;       BGNTST
                                                                                T21::
027422  004737  005420          JSR     PC,INIDMV       ;INIT VIA

027426  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
027432  003400                  NOCHK                   ;BOP MODE, NO ERROR CHECK
027434  040000                  NORXEN                  ;NO RECEIVER ENABLE,USE 8 BIT CHARS
027436  103003                  BCC     .+8.            ;BR IF NO ERROR
027440                          ERROR                   ;REPORT STACKED ERROR
027440  104460                                                  TRAP    C$ERROR
027442                          ESCAPE  TST             ;SKIP TO END OF TEST
027442  104410                                                  TRAP    C$ESCAPE
027444  000150                                                  .WORD   L10057-.

027446  004537  007116          JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "FLAG"
027452  000000                  0                       ; CHARACTER (SHOULD BE 0)
027454  103003                  BCC     .+8.            ;BR IF NO ERROR
027456                          ERROR                   ;REPORT STACKED ERROR
027456  104460                                                  TRAP    C$ERROR
027460                          ESCAPE  TST             ;AND EXIT TEST
027460  104410                                                  TRAP    C$ESCAPE
027462  000132                                                  .WORD   L10057-.

027464  004537  007256          JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF "FLAG" CHARACTER
027470  000007                  7.                      ; (OFF OF TSO BIT)
027472  000176                  ^176                    ; EXPECTED BIT SEQUENCE (1111110)
027474  103003                  BCC     .+8.            ;BR IF NO ERROR
027476                          ERROR                   ;REPORT STACKED ERROR
027476  104460                                                  TRAP    C$ERROR
027500                          ESCAPE  TST             ;AND EXIT TEST
027500  104410                                                  TRAP    C$ESCAPE
027502  000112                                                  .WORD   L10057-.

027504  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
027510  000000                  000
027512  000000                  0
```

```
 6765
 6766  027514  004537  003734                JSR     R5,WRITEI       ;LOAD 000 CHARACTER
 6767  027520  120402                        TDSRL
 6768  027522  000000                        000
 6769
 6770  027524  004537  007256                JSR     R5,SERIAL       ;READ FLAG CHARACTER VIA TSO
 6771  027530  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 6772  027532  000176                        176                     ; EXPECTED BIT SEQUENCE (01111110)
 6773  027534  103003                        BCC     .+8.            ;BR IF NO ERROR
 6774  027536                                ERROR                   ;REPORT STACKED ERROR
 6775  027536  104460                                                        TRAP    C$ERROR
 6776  027540                                ESCAPE  TST             ;AND EXIT TEST
 6777  027540  104410                                                        TRAP    C$ESCAPE
 6778  027542  000052                                                        .WORD   L10057-.
 6779
 6780  027544  004537  010010                JSR     R5,TXCTRL       ;SET TXABT BIT
 6781  027550  000004                        TAB
 6782  027552  000000                        0
 6783
 6784  027554  004537  007256                JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
 6785  027560  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 6786  027562  000000                        000                     ; EXPECTED BIT SEQUENCE (00000000)
 6787  027564  103003                        BCC     .+8.            ;BR IF NO ERROR
 6788  027566                                ERROR                   ;REPORT STACKED ERROR
 6789  027566  104460                                                        TRAP    C$ERROR
 6790  027570                                ESCAPE  TST             ;AND EXIT TEST
 6791  027570  104410                                                        TRAP    C$ESCAPE
 6792  027572  000022                                                        .WORD   L10057-.
 6793
 6794  027574  004537  007256                JSR     R5,SERIAL       ;READ ABORT CHARACTER VIA TSO
 6795  027600  000007                        7                       ; 7 BIT CHAR/CLOCK TICKS
 6796  027602  000177                        177                     ; EXPECTED BIT SEQUENCE (1111111)
 6797  027604  103003                        BCC     .+8.            ;BR IF NO ER OR
 6798  027606                                ERROR                   ;REPORT STACKED ERROR
 6799  027606  104460                                                        TRAP    C$ERROR
 6800  027610                                ESCAPE  TST             ;AND EXIT TEST
 6801  027610  104410                                                        TRAP    C$ESCAPE
 6802  027612  000002                                                        .WORD   L10057-.
 6803  027614                        ENDTST
 6804  027614                                                        L10057:
 6805  027614  104401                                                        TRAP    C$ETST
```

```
                                  .SBTTL  TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST

                                  ;*****************************************************************
                                  ;*
                                  ;*      TEST 22 -- BOP TX TABORT/(IDLE = 1) TEST
                                  ;*
                                  ;*  THE USYRT IS INITIALIZED FOR "BIT-ORIENTED PROTOCOL" (BOP) WITH IDLE
                                  ;*  SET TO ONE. TXE AND TSOM IS ASSERTED AND TXC IS MANUALLY STEPPED WHILE
                                  ;*  OBSERVING TXA -- 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. THE TRANSMITTER IS CLOCKED THROUGH
                                  ;*  ONE CHARACTER. WHEN TXBE GOES HIGH  AGAIN, TABORT IS ASSERTED CAUSING
                                  ;*  ABORT TO BE TRANSMITTED. 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 FLAG CHARACTER IS SENT (INTO THE BIT
                                  ;*  BUCKET). ERROR LOOPING WILL DEPEND ON WHERE THE FIRST ERROR OCCURS WITHIN
                                  ;*  THE SEQUENCE.
                                  ;*
                                  ;--****************************************************************
                                  ;
                                  ;       BGNTST
027616                                                                      T22::
027616  004737  005420                  JSR     PC,INIDMV       ;INIT VIA

027622  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
027626  007400                          NOCHK!IDLES             ;BOP MODE, NO ERROR CHECK, IDLE=1
027630  040000                          NORXEN                  ;NO RECEIVER ENABLE,USE 8 BIT CHARS
027632  103003                          BCC     .+8.            ;BR IF NO ERROR
027634                                  ERROR                   ;REPORT STACKED ERROR
027634  104460                                                                  TRAP    C$ERROR
027636                                  ESCAPE  TST             ;SKIP TO END OF TEST
027636  104410                                                                  TRAP    C$ESCAPE
027640  000150                                                                  .WORD   L10060-.

027642  004537  007116                  JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "FLAG"
027646  000000                          0                       ; CHARACTER (SHOULD BE 0)
027650  103003                          BCC     .+8.            ;BR IF NO ERROR
027652                                  ERROR                   ;REPORT STACKED ERROR
027652  104460                                                                  TRAP    C$ERROR
027654                                  ESCAPE  TST             ;AND EXIT TEST
027654  104410                                                                  TRAP    C$ESCAPE
027656  000132                                                                  .WORD   L10060-.

027660  004537  007256                  JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF "FLAG" CHARACTER
027664  000007                          7.                      ; (OFF OF TSO BIT)
027666  000176                          176                     ; EXPECTED BIT SEQUENCE (1111110)
027670  103003                          BCC     .+8.            ;BR IF NO ERROR
027672                                  ERROR                   ;REPORT STACKED ERROR
027672  104460                                                                  TRAP    C$ERROR
027674                                  ESCAPE  TST             ;AND EXIT TEST
027674  104410                                                                  TRAP    C$ESCAPE
027676  000112                                                                  .WORD   L10060-.

027700  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
027704  000000                          000
027706  000000                          0
```

```
6862
6863  027710  004537  003734          JSR     R5,WRITEI       ;LOAD 000 CHARACTER
6864  027714  120402                  TDSRL
6865  027716  000000                  000
6866
6867  027720  004537  007256          JSR     R5,SERIAL       ;READ FLAG CHARACTER VIA TSO
6868  027724  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6869  027726  000176                  176                     ; EXPECTED BIT SEQUENCE (01111110)
6870  027730  103003                  BCC     .+8.            ;BR IF NO ERROR
6871  027732                          ERROR                   ;REPORT STACKED ERROR
6872  027732  104460                                                          TRAP    C$ERROR
6873  027734                          ESCAPE  TST             ;AND EXIT TEST
6874  027734  104410                                                          TRAP    C$ESCAPE
6875  027736  000052                                                          .WORD   L10060-.
6876
6877  027740  004537  010010          JSR     R5,TXCTRL       ;SET TXABT BIT
6878  027744  000004                  TAB
6879  027746  000000                  0
6880
6881  027750  004537  007256          JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
6882  027754  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6883  027756  000000                  000                     ; EXPECTED BIT SEQUENCE (00000000)
6884  027760  103003                  BCC     .+8.            ;BR IF NO ERROR
6885  027762                          ERROR                   ;REPORT STACKED ERROR
6886  027762  104460                                                          TRAP    C$ERROR
6887  027764                          ESCAPE  TST             ;AND EXIT TEST
6888  027764  104410                                                          TRAP    C$ESCAPE
6889  027766  000022                                                          .WORD   L10060-.
6890
6891  027770  004537  007256          JSR     R5,SERIAL       ;READ ABORT (FLAG) CHARACTER VIA TSO
6892  027774  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6893  027776  000176                  176                     ; EXPECTED BIT SEQUENCE (01111110)
6894  030000  103003                  BCC     .+8.            ;BR IF NO ERROR
6895  030002                          ERROR                   ;REPORT STACKED ERROR
6896  030002  104460                                                          TRAP    C$ERROR
6897  030004                          ESCAPE  TST             ;AND EXIT TEST
6898  030004  104410                                                          TRAP    C$ESCAPE
6899  030006  000002                                                          .WORD   L10060-.
6900  030010                  ENDTST
6901  030010                                                          L10060:
6902  030010  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
     030012                                                                              T23::
     030012   004737  005420                      JSR     PC,INIDMV       ;INIT VIA

     030016   004537  007400                      JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
     030022   003400                              NOCHK                   ;BOP MODE, NO ERROR CHECK
     030024   040000                              NORXEN                  ;NO RECEIVER ENABLE,USE 8 BIT CHARS
     030026   103003                              BCC     .+8.            ;BR IF NO ERROR
     030030                                       ERROR                   ;REPORT STACKED ERROR
     030030   104460                                                                          TRAP    C$ERROR
     030032                                       ESCAPE  TST             ;SKIP TO END OF TEST
     030032   104410                                                                          TRAP    C$ESCAPE
     030034   000150                                                                          .WORD   L10061-.

     030036   004537  007116                      JSR     R5,CHKTSO       ;CHECK 1ST BIT OF EXPECTED "FLAG"
     030042   000000                              0                       ; CHARACTER (SHOULD BE 0)
     030044   103003                              BCC     .+8.            ;BR IF NO ERROR
     030046                                       ERROR                   ;REPORT STACKED ERROR
     030046   104460                                                                          TRAP    C$ERROR
     030050                                       ESCAPE  TST             ;AND EXIT TEST
     030050   104410                                                                          TRAP    C$ESCAPE
     030052   000132                                                                          .WORD   L10061-.

     030054   004537  007256                      JSR     R5,SERIAL       ;READ REMAINING 7 BITS OF "FLAG" CHARACTER
     030060   000007                              7.                      ; (OFF OF TSO BIT)
     030062   000176                              176                     ; EXPECTED BIT SEQUENCE (1111110)
     030064   103003                              BCC     .+8.            ;BR IF NO ERROR
     030066                                       ERROR                   ;REPORT STACKED ERROR
     030066   104460                                                                          TRAP    C$ERROR
     030070                                       ESCAPE  TST             ;AND EXIT TEST
     030070   104410                                                                          TRAP    C$ESCAPE
     030072   000112                                                                          .WORD   L10061-.

     030074   004537  010010                      JSR     R5,TXCTRL       ;CLEAR TSOM
     030100   000000                              000
     030102   000000                              0
```

```
6959
6960  030104  004537  003734          JSR     R5,WRITEI       ;LOAD 000 CHARACTER
6961  030110  120402                  TDSRL
6962  030112  000000                  000
6963
6964  030114  004537  007256          JSR     R5,SERIAL       ;READ "FLAG" CHARACTER VIA TSO
6965  030120  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6966  030122  000176                  176                     ; EXPECTED BIT SEQUENCE (01111110)
6967  030124  103003                  BCC     .+8.            ;BR IF NO ERROR
6968  030126                          ERROR                   ;REPORT STACKED ERROR
6969  030126  104460                                                  TRAP    C$ERROR
6970  030130                          ESCAPE  TST             ;AND EXIT TEST
6971  030130  104410                                                  TRAP    C$ESCAPE
6972  030132  000052                                                  .WORD   L10061-.
6973
6974  030134  004537  010010          JSR     R5,TXCTRL       ;SET TX GO-AHEAD AND TEOM
6975  030140  000012                  TGA!TEOM
6976  030142  000000                  0
6977
6978  030144  004537  007256          JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
6979  030150  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6980  030152  000000                  000                     ; EXPECTED BIT SEQUENCE (00000000)
6981  030154  103003                  BCC     .+8.            ;BR IF NO ERROR
6982  030156                          ERROR                   ;REPORT STACKED ERROR
6983  030156  104460                                                  TRAP    C$ERROR
6984  030160                          ESCAPE  TST             ;AND EXIT TEST
6985  030160  104410                                                  TRAP    C$ESCAPE
6986  030162  000022                                                  .WORD   L10061-.
6987
6988  030164  004537  007256          JSR     R5,SERIAL       ;READ GO-AHEAD CHARACTER VIA TSO
6989  030170  000010                  8.                      ; 8 BIT CHAR/CLOCK TICKS
6990  030172  000177                  177                     ; EXPECTED BIT SEQUENCE (01111111)
6991  030174  103003                  BCC     .+8.            ;BR IF NO ERROR
6992  030176                          ERROR                   ;REPORT STACKED ERROR
6993  030176  104460                                                  TRAP    C$ERROR
6994  030200                          ESCAPE  TST             ;AND EXIT TEST
6995  030200  104410                                                  TRAP    C$ESCAPE
6996  030202  000002                                                  .WORD   L10061-.
6997  030204                  ENDTST
6998  030204                                                  L10061:
6999  030204  104401                                                  TRAP    C$ETST
```

```
                                .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::
030206                                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
030206  004737  005420
030212  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030216  003400                          NOCHK                   ;SET BOP MODE,NO CHECK
030220  000000                          0                       ;USE 8 BIT CHARS
030222  103003                          BCC     .+8.            ;BR IF NO ERROR
030224                                  ERROR                   ;REPORT STACKED ERROR
030224  104460                                                                  TRAP    C$ERROR
030226                                  ESCAPE  TST             ;SKIP TO END OF TEST
030226  104410                                                                  TRAP    C$ESCAPE
030230  000332                                                                  .WORD   L10062-.

030232  004537  010010                  JSR     R5,TXCTRL       ;LOAD 2ND FLAG, TX 1ST FLAG
030236  000001                          TSOM
030240  000007                          7.

030242  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
030246  000000                          000
030250  000000                          0

030252  004537  007676                  JSR     R5,TXCHAR       ;LOAD 000, TX 2ND FLAG
030256  000000                          000
030260  000010                          8.
030262  103003                          BCC     .+8.            ;BR IF NO ERROR
030264                                  ERROR                   ;REPORT STACKED ERROR
030264  104460                                                                  TRAP    C$ERROR
030266                                  ESCAPE  TST             ;SKIP TO END OF TEST
030266  104410                                                                  TRAP    C$ESCAPE
030270  000272                                                                  .WORD   L10062-.

030272  004537  007676                  JSR     R5,TXCHAR       ;LOAD 307, TX 000
030276  000307                          307
030300  000010                          8.
030302  103003                          BCC     .+8.            ;BR IF NO ERROR
030304                                  ERROR                   ;REPORT STACKED ERROR
030304  104460                                                                  TRAP    C$ERROR
030306                                  ESCAPE  TST             ;SKIP TO END OF TEST
```

```
 7056  030306  104410                                                          TRAP    C$ESCAPE
 7057  030310  000252                                                          .WORD   L10062-.
 7058
 7059  030312  004537  007676          JSR     R5,TXCHAR       ;LOAD 125; TX 307
 7060  030316  000125                  125
 7061  030320  000010                  8.
 7062  030322  103003                  BCC     .+8.            ;BR IF NO ERROR
 7063  030324                          ERROR                   ;REPORT STACKED ERROR
 7064  030324  104460                                                          TRAP    C$ERROR
 7065  030326                          ESCAPE  TST             ;SKIP TO END OF TEST
 7066  030326  104410                                                          TRAP    C$ESCAPE
 7067  030330  000232                                                          .WORD   L10062-.
 7068
 7069  030332  004537  007676          JSR     R5,TXCHAR       ;LOAD 252
 7070  030336  000252                  252
 7071  030340  000000                  0
 7072  030342  103003                  BCC     .+8.            ;BR IF NO ERROR
 7073  030344                          ERROR                   ;REPORT STACKED ERROR
 7074  030344  104460                                                          TRAP    C$ERROR
 7075  030346                          ESCAPE  TST             ;SKIP TO END OF TEST
 7076  030346  104410                                                          TRAP    C$ESCAPE
 7077  030350  000212                                                          .WORD   L10062-.
 7078
 7079  030352  004537  011364          JSR     R5,RCV1ST       ;CLOCK AND RCV 000
 7080  030356  000000                  0
 7081  030360  103003                  BCC     .+8.            ;BR IF NO ERROR
 7082  030362                          ERROR                   ;REPORT STACKED ERROR
 7083  030362  104460                                                          TRAP    C$ERROR
 7084  030364                          ESCAPE  TST             ;SKIP TO END OF TEST
 7085  030364  104410                                                          TRAP    C$ESCAPE
 7086  030366  000174                                                          .WORD   L10062-.
 7087
 7088  030370  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 000, RCV 307
 7089  030374  000400                  RXSOM!000               ; & CHECK RSOM=1
 7090  030376  000000                  0
 7091  030400  000010                  8.
 7092  030402  103003                  BCC     .+8.            ;BR IF NO ERROR
 7093  030404                          ERROR                   ;REPORT STACKED ERROR
 7094  030404  104460                                                          TRAP    C$ERROR
 7095  030406                          ESCAPE  TST             ;SKIP TO END OF TEST
 7096  030406  104410                                                          TRAP    C$ESCAPE
 7097  030410  000152                                                          .WORD   L10062-.
 7098
 7099  030412  004537  007676          JSR     R5,TXCHAR       ;LOAD 201
 7100  030416  000201                  201
 7101  030420  000000                  0
 7102  030422  103003                  BCC     .+8.            ;BR IF NO ERROR
 7103  030424                          ERROR                   ;REPORT STACKED ERROR
 7104  030424  104460                                                          TRAP    C$ERROR
 7105  030426                          ESCAPE  TST             ;SKIP TO END OF TEST
 7106  030426  104410                                                          TRAP    C$ESCAPE
 7107  030430  000132                                                          .WORD   L10062-.
 7108
 7109  030432  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 307, RCV 125
 7110  030436  000307                  307
 7111  030440  000000                  0
```

```
 7112  030442  000010                        8.
 7113  030444  103003                        BCC     .+8.            ;BR IF NO ERROR
 7114  030446                                ERROR                   ;REPORT STACKED ERROR
 7115  030446  104460                                                                TRAP    C$ERROR
 7116  030450                                ESCAPE  TST             ;SKIP TO END OF TEST
 7117  030450  104410                                                                TRAP    C$ESCAPE
 7118  030452  000110                                                                .WORD   L10062-.
 7119
 7120  030454  004537  010010                JSR     R5,TXCTRL       ;SET TEOM BIT
 7121  030460  000002                        TEOM
 7122  030462  000000                        0
 7123  030464  103003                        BCC     .+8.            ;BR IF NO ERROR
 7124  030466                                ERROR                   ;REPORT STACKED ERROR
 7125  030466  104460                                                                TRAP    C$ERROR
 7126  030470                                ESCAPE  TST             ;SKIP TO END OF TEST
 7127  030470  104410                                                                TRAP    C$ESCAPE
 7128  030472  000070                                                                .WORD   L10062-.
 7129
 7130  030474  004537  010110                JSR     R5,RXCHAR       ;READ & CHK 125, RCV 252
 7131  030500  000125                        125
 7132  030502  000000                        0
 7133  030504  000010                        8.
 7134  030506  103003                        BCC     .+8.            ;BR IF NO ERROR
 7135  030510                                ERROR                   ;REPORT STACKED ERROR
 7136  030510  104460                                                                TRAP    C$ERROR
 7137  030512                                ESCAPE  TST             ;SKIP TO END OF TEST
 7138  030512  104410                                                                TRAP    C$ESCAPE
 7139  030514  000046                                                                .WORD   L10062-.
 7140
 7141  030516  004537  010110                JSR     R5,RXCHAR       ;READ & CHK 252, RCV 201
 7142  030522  000252                        252
 7143  030524  000000                        0
 7144  030526  020010                        NCRACT!8.               ;DON'T CHECK FOR FINAL RXACT=1
 7145  030530  103003                        BCC     .+8.            ;BR IF NO ERROR
 7146  030532                                ERROR                   ;REPORT STACKED ERROR
 7147  030532  104460                                                                TRAP    C$ERROR
 7148  030534                                ESCAPE  TST             ;SKIP TO END OF TEST
 7149  030534  104410                                                                TRAP    C$ESCAPE
 7150  030536  000024                                                                .WORD   L10062-.
 7151
 7152  030540  004537  010110                JSR     R5,RXCHAR       ;READ & CHK 201, RCV FIRST FLAG
 7153  030544  001201                        RXEOM!201               ; & CHECK REOM
 7154  030546  000000                        0
 7155  030550  060000                        NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
 7156  030552  103003                        BCC     .+8.            ;BR IF NO ERROR
 7157  030554                                ERROR                   ;REPORT STACKED ERROR
 7158  030554  104460                                                                TRAP    C$ERROR
 7159  030556                                ESCAPE  TST             ;SKIP TO END OF TEST
 7160  030556  104410                                                                TRAP    C$ESCAPE
 7161  030560  000002                                                                .WORD   L10062-.
 7162  030562                        ENDTST
 7163  030562                                                                L10062:
 7164  030562  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
                                                                                        T25::
030564                                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
030564  004737  005420
                                ;---------------------------------------------------------------------
                                ; SUBROUTINE # 1: 2 BIT CHARACTERS
                                ;---------------------------------------------------------------------
030570                          BGNSUB
030570                                                                          T25.1:
030570  104402                                                                          TRAP    C$BSUB
030572  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
030576  004226                          IDLES!SYNCH             ;SET BOP MODE,CRC-CCITT-1,S/AR=226
030600  000002                          BIT1                    ;INITIALLY: TXCL=8 BITS / RXCL=2 BITS
030602  103003                          BCC     .+8.            ;BR IF NO ERROR
030604                                  ERROR                   ;REPORT STACKED ERROR
030604  104460                                                                  TRAP    C$ERROR
030606                                  ESCAPE  SUB             ;SKIP TO END OF TEST
030606  104410                                                                  TRAP    C$ESCAPE
030610  000344                                                                  .WORD   L10064-.

030612  004537  010010                  JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
030616  000001                          TSOM
030620  000007                          7.
030622  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
030626  000000                          000
030630  000000                          0
030632  004537  007676                  JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
030636  000123                          123
030640  000010                          8.
030642  103003                          BCC     .+8.            ;BR IF NO ERROR
```

```
7221  030644                                ERROR                   ;REPORT STACKED ERROR
7222  030644  104460                                                                TRAP    C$ERROR
7223  030646                                ESCAPE  SUB             ;SKIP TO END OF TEST
7224  030646  104410                                                                TRAP    C$ESCAPE
7225  030650  000304                                                                .WORD   L10064-.
7226
7227  030652  004537  007676                JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7228  030656  000321                        321
7229  030660  000010                        8.
7230  030662  103003                        BCC     .+8.            ;BR IF NO ERROR
7231  030664                                ERROR                   ;REPORT STACKED ERROR
7232  030664  104460                                                                TRAP    C$ERROR
7233  030666                                ESCAPE  SUB             ;SKIP TO END OF TEST
7234  030666  104410                                                                TRAP    C$ESCAPE
7235  030670  000264                                                                .WORD   L10064-.
7236
7237  030672  004537  003734                JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 2 BITS TOO
7238  030676  120407                        PCR
7239  030700  000102                        BIT6!BIT1
7240  030702  004537  007676                JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7241  030706  000111                        111
7242  030710  000010                        8.
7243  030712  103003                        BCC     .+8.            ;BR IF NO ERROR
7244  030714                                ERROR                   ;REPORT STACKED ERROR
7245  030714  104460                                                                TRAP    C$ERROR
7246  030716                                ESCAPE  SUB             ;SKIP TO END OF TEST
7247  030716  104410                                                                TRAP    C$ESCAPE
7248  030720  000234                                                                .WORD   L10064-.
7249                        ;*************************************************************
7250  030722  012703  033477                MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7251  030726  112337  030736        10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7252
7253  030732  004537  007676                JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7254  030736  000000                1$:     000                     ;** HOLE FOR NEXT CHARACTER **
7255  030740  000002                        2.
7256  030742  103003                        BCC     .+8.            ;BR IF NO ERROR
7257  030744                                ERROR                   ;REPORT STACKED ERROR
7258  030744  104460                                                                TRAP    C$ERROR
7259  030746                                ESCAPE  SUB             ;SKIP TO END OF TEST
7260  030746  104410                                                                TRAP    C$ESCAPE
7261  030750  000204                                                                .WORD   L10064-.
7262
7263  030752  022703  033505                CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7264  030756  001363                        BNE     10$             ; IF NOT, TX ANOTHER ONE
7265                        ;*************************************************************
7266  030760  004537  010010                JSR     R5,TXCTRL       ;SET TEOM
7267  030764  000002                        TEOM
7268  030766  000000                        0
7269  030770  004537  011614                JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7270  030774  000044                        36.
7271
7272  030776  004537  010110                JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7273  031002  000523                        RXSOM!123               ; & CHECK FOR RSOM=1
7274  031004  000000                        0
7275  031006  100000                        NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7276  031010  103003                        BCC     .+8.            ;BR IF NO ERROR
```

```
7277  031012                                  ERROR                   ;REPORT STACKED ERROR
7278  031012  104460                                                                  TRAP    C$ERROR
7279  031014                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7280  031014  104410                                                                  TRAP    C$ESCAPE
7281  031016  000136                                                                  .WORD   L10064-.
7282
7283  031020  004537  010110                  JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 001(DATA1)
7284  031024  000321                          321
7285  031026  000000                          0
7286  031030  100000                          NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7287  031032  103003                          BCC     .+8.            ;BR IF NO ERROR
7288  031034                                  ERROR                   ;REPORT STACKED ERROR
7289  031034  104460                                                                  TRAP    C$ERROR
7290  031036                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7291  031036  104410                                                                  TRAP    C$ESCAPE
7292  031040  000114                                                                  .WORD   L10064-.
7293                                  ;-----------------------------------------------------------------
7294  031042  012703  033476                  MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7295  031046  112337  031064          20$:    MOVB    (R3)+,2$        ;INSTALL NEXT EXPECTED RX CHARACTER
7296  031052  142737  000374  031064          BICB    #374,2$         ;MASK OUT UNTRANSMITTED BITS
7297
7298  031060  004537  010110                  JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7299  031064  000000                  2$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
7300  031066  000000                          0
7301  031070  100000                          NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7302  031072  103003                          BCC     .+8.            ;BR IF NO ERROR
7303  031074                                  ERROR                   ;REPORT STACKED ERROR
7304  031074  104460                                                                  TRAP    C$ERROR
7305  031076                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7306  031076  104410                                                                  TRAP    C$ESCAPE
7307  031100  000054                                                                  .WORD   L10064-.
7308
7309  031102  022703  033503                  CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7310  031106  001357                          BNE     20$             ; IF NOT, CHECK ANOTHER ONE
7311                                  ;-----------------------------------------------------------------
7312  031110  004537  010110                  JSR     R5,RXCHAR       ;READ/CHK DATA6(002), RCV DATA7
7313  031114  000002                          0C
7314  031116  000000                          0                       ;NO INITIAL CHECK OF RDA=0
7315  031120  120000                          NCRACT!NOCRDA           ;DON'T CHECK FOR FINAL RXACT=1
7316  031122  103003                          BCC     .+8.            ;BR IF NO ERROR
7317  031124                                  ERROR                   ;REPORT STACKED ERROR
7318  031124  104460                                                                  TRAP    C$ERROR
7319  031126                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7320  031126  104410                                                                  TRAP    C$ESCAPE
7321  031130  000024                                                                  .WORD   L10064-.
7322
7323  031132  004537  010110                  JSR     R5,RXCHAR       ;READ/CHK DATA7(003), RCV FIRST FLAG
7324  031136  001003                          RXEOM!003               ;& CHECK RERR BIT=0 (GOOD CRC) & REOM=1
7325  031140  000001                          RERCHK                  ;NO INITIAL CHECK OF RDA=0
7326  031142  160000                          NOCRDA!NFCRDA!NCRACT    ;DON'T CHECK FOR FINAL RDA=RXACT=1
7327  031144  103003                          BCC     .+8.            ;BR IF NO ERROR
7328  031146                                  ERROR                   ;REPORT STACKED ERROR
7329  031146  104460                                                                  TRAP    C$ERROR
7330  031150                                  ESCAPE  SUB             ;SKIP TO END OF TEST
7331  031150  104410                                                                  TRAP    C$ESCAPE
7332  031152  000002                                                                  .WORD   L10064-.
```

```
                        ENDSUB
                                                                L10064:
                                                                        TRAP    C$ESUB
                        ;-------------------------------------------------------------
                        ; SUBROUTINE # 2: 3 BIT CHARACTERS
                        ;-------------------------------------------------------------
                              BGNSUB
                                                                T25.2:
                                                                        TRAP    C$BSUB
                              JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
                              SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
                              BIT1!BIT0               ;INITIALLY: TXCL=8 BITS / RXCL=3 BITS
                              BCC     .+8.            ;BR IF NO ERROR
                              ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                              ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10065-.

                              JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
                              TSOM
                              7.
                              JSR     R5,TXCTRL       ;CLEAR TSOM
                              000
                              0
                              JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
                              123
                              8.
                              BCC     .+8.            ;BR IF NO ERROR
                              ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                              ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10065-.

                              JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
                              321
                              8.
                              BCC     .+8.            ;BR IF NO ERROR
                              ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                              ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
                                                                        .WORD   L10065-.

                              JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 3 BITS TOO
                              PCR
                              BIT6!BIT5!BIT1!BIT0
                              JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
                              111
                              8.
                              BCC     .+8.            ;BR IF NO ERROR
                              ERROR                   ;REPORT STACKED ERROR
                                                                        TRAP    C$ERROR
                              ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                        TRAP    C$ESCAPE
```

```
031306  000234                                                                       .WORD   L10065-.
                                  ;-----------------------------------------------------------------
031310  012703  033477                   MOV     #T30TBL+1,R3     ;SET UP DATA TABLE POINTER
031314  112337  031324           10$:    MOVB    (R3)+,1$         ;INSTALL NEXT TX CHARACTER

031320  004537  007676                   JSR     R5,TXCHAR        ;TRANSMIT CHARACTER ( ==> RX/FIFO )
031324  000000                   1$:     000                      ;** HOLE FOR NEXT CHARACTER **
031326  000003                           3.
031330  103003                           BCC     .+8.             ;BR IF NO ERROR
031332                                   ERROR                    ;REPORT STACKED ERROR
031332  104460                                                                       TRAP    C$ERROR
031334                                   ESCAPE  SUB              ;SKIP TO END OF TEST
031334  104410                                                                       TRAP    C$ESCAPE
031336  000204                                                                       .WORD   L10065-.

031340  022703  033505                   CMP     #T30TBL+7,R3     ;ALL CHARACTERS TRANSMITTED ?
031344  001363                           BNE     10$              ; IF NOT, TX ANOTHER ONE
                                  ;-----------------------------------------------------------------
031346  004537  010010                   JSR     R5,TXCTRL        ;SET TEOM
031352  000002                           TEOM
031354  000000                           0
031356  004537  011614                   JSR     R5,STEPLU        ; AND TX DATA7 + SOME FLAGS
031362  000044                           36.

031364  004537  010110                   JSR     R5,RXCHAR        ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
031370  000523                           RXSOM!123                ; & CHECK FOR RSOM=1
031372  000000                           0
031374  100000                           NOCRDA                   ;NO INITIAL CHECK OF RDA=0
031376  103003                           BCC     .+8.             ;BR IF NO ERROR
031400                                   ERROR                    ;REPORT STACKED ERROR
031400  104460                                                                       TRAP    C$ERROR
031402                                   ESCAPE  SUB              ;SKIP TO END OF TEST
031402  104410                                                                       TRAP    C$ESCAPE
031404  000136                                                                       .WORD   L10065-.

031406  004537  010110                   JSR     R5,RXCHAR        ;READ/CHECK 321(HEADER2),RCV 001(DATA1)
031412  000321                           321
031414  000000                           0
031416  100000                           NOCRDA                   ;NO INITIAL CHECK OF RDA=0
031420  103003                           BCC     .+8.             ;BR IF NO ERROR
031422                                   ERROR                    ;REPORT STACKED ERROR
031422  104460                                                                       TRAP    C$ERROR
031424                                   ESCAPE  SUB              ;SKIP TO END OF TEST
031424  104410                                                                       TRAP    C$ESCAPE
031426  000114                                                                       .WORD   L10065-.
                                  ;-----------------------------------------------------------------
031430  012703  033476                   MOV     #T30TBL,R3       ;SET UP DATA TABLE POINTER
031434  112337  031452           40$:    MOVB    (R3)+,4$         ;INSTALL NEXT EXPECTED RX CHARACTER
031440  142737  000370  031452           BICB    #370,4$          ;MASK OUT UNTRANSMITTED BITS

031446  004537  010110                   JSR     R5,RXCHAR        ;READ/CHECK NEXT CHARACTER
031452  000000                   4$:     000                      ;** HOLE FOR NEXT EXPECTED CHARACTER
031454  000000                           0
031456  100000                           NOCRDA                   ;NO INITIAL CHECK OF RDA=0
031460  103003                           BCC     .+8.             ;BR IF NO ERROR
031462                                   ERROR                    ;REPORT STACKED ERROR
```

```
031462  104460                                                                        TRAP    C$ERROR
031464                                   ESCAPE  SUB               ;SKIP TO END OF TEST
031464  104410                                                                        TRAP    C$ESCAPE
031466  000054                                                                        .WORD   L10065-.

031470  022703  033503                   CMP     #T30TBL+5,R3      ;ALL CHARACTERS CHECKED ?
031474  001357                           BNE     40$               ; IF NOT, CHECK ANOTHER ONE
                              ;**********************************************************************
031476  004537  010110                   JSR     R5,RXCHAR         ;READ/CHK DATA6(006), RCV DATA7
031502  000006                           006
031504  000000                           0
031506  020000                           NCRACT                    ;DON'T CHECK FOR FINAL RXACT=1
031510  103003                           BCC     .+8.              ;BR IF NO ERROR
031512                                   ERROR                     ;REPORT STACKED ERROR
031512  104460                                                                        TRAP    C$ERROR
031514                                   ESCAPE  TST               ;SKIP TO END OF TEST
031514  104410                                                                        TRAP    C$ESCAPE
031516  001756                                                                        .WORD   L10063-.

031520  004537  010110                   JSR     R5,RXCHAR         ;READ/CHK DATA7(007), RCV FIRST FLAG
031524  001007                           RXEOM!007                 ;AND CHECK FOR REOM=1
031526  000001                           RERCHK                    ; & CHECK RERR BIT=0 (GOOD CRC)
031530  060000                           NFCRDA!NCRACT             ;DON'T CHECK FOR FINAL RDA=RXACT=1
031532  103003                           BCC     .+8.              ;BR IF NO ERROR
031534                                   ERROR                     ;REPORT STACKED ERROR
031534  104460                                                                        TRAP    C$ERROR
031536                                   ESCAPE  TST               ;SKIP TO END OF TEST
031536  104410                                                                        TRAP    C$ESCAPE
031540  001734                                                                        .WORD   L10063-.
031542                           ENDSUB
031542                                                                       L10065:
031542  104403                                                                        TRAP    C$ESUB
                              ;----------------------------------------------------------------------
                              ; SUBROUTINE # 3: 4 BIT CHARACTERS
                              ;----------------------------------------------------------------------
031544                                   BGNSUB
031544                                                                       T25.3:
031544  104402                                                                        TRAP    C$BSUB
031546  004537  007400                   JSR     R5,INITRN         ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
031552  000226                           SYNCH                     ;SET BOP MODE,CRC-CCITT-1,S/AR=226
031554  000004                           BIT2                      ;INITIALLY: TXCL=8 BITS / RXCL=4 BITS
031556  103003                           BCC     .+8.              ;BR IF NO ERROR
031560                                   ERROR                     ;REPORT STACKED ERROR
031560  104460                                                                        TRAP    C$ERROR
031562                                   ESCAPE  SUB               ;SKIP TO END OF TEST
031562  104410                                                                        TRAP    C$ESCAPE
031564  0003[illegible]4                                                                        .WORD   L10066-.

031566  004537  010010                   JSR     R5,TXCTRL         ;LOAD 2ND FLAG,TX 1ST FLAG
031572  000001                           TSOM
031574  000007                           7.
031576  004537  010010                   JSR     R5,TXCTRL         ;CLEAR TSOM
031602  000000                           000
031604  000000                           0
031606  004537  007676                   JSR     R5,TXCHAR         ;LOAD 123(HEADER1), TX 2ND FLAG
031612  000123                           123
```

```
7501  031614  000010                    8.
7502  031616  103003                    BCC     .+8.            ;BR IF NO ERROR
7503  031620                            ERROR                   ;REPORT STACKED ERROR
7504  031620  104460                                                    TRAP    C$ERROR
7505  031622                            ESCAPE  SUB             ;SKIP TO END OF TEST
7506  031622  104410                                                    TRAP    C$ESCAPE
7507  031624  000304                                                    .WORD   L10066-.
7508
7509  031626  004537  007676            JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7510  031632  000321                    321
7511  031634  000010                    8.
7512  031636  103003                    BCC     .+8.            ;BR IF NO ERROR
7513  031640                            ERROR                   ;REPORT STACKED ERROR
7514  031640  104460                                                    TRAP    C$ERROR
7515  031642                            ESCAPE  SUB             ;SKIP TO END OF TEST
7516  031642  104410                                                    TRAP    C$ESCAPE
7517  031644  000264                                                    .WORD   L10066-.
7518
7519  031646  004537  003734            JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 4 BITS TOO
7520  031652  120407                    PCR
7521  031654  000204                    BIT7!BIT2
7522  031656  004537  007676            JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7523  031662  000111                    111
7524  031664  000010                    8.
7525  031666  103003                    BCC     .+8.            ;BR IF NO ERROR
7526  031670                            ERROR                   ;REPORT STACKED ERROR
7527  031670  104460                                                    TRAP    C$ERROR
7528  031672                            ESCAPE  SUB             ;SKIP TO END OF TEST
7529  031672  104410                                                    TRAP    C$ESCAPE
7530  031674  000234                                                    .WORD   L10066-.
7531                            ;-------------------------------------------------------------------
7532  031676  012703  033477            MOV     #T3OTBL+1,R3    ;SET UP DATA TABLE POINTER
7533  031702  112337  031712    10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7534
7535  031706  004537  007676            JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7536  031712  000000    1$:             000                     ;** HOLE FOR NEXT CHARACTER **
7537  031714  000004                    4.
7538  031716  103003                    BCC     .+8.            ;BR IF NO ERROR
7539  031720                            ERROR                   ;REPORT STACKED ERROR
7540  031720  104460                                                    TRAP    C$ERROR
7541  031722                            ESCAPE  SUB             ;SKIP TO END OF TEST
7542  031722  104410                                                    TRAP    C$ESCAPE
7543  031724  000204                                                    .WORD   L10066-.
7544
7545  031726  022703  033505            CMP     #T3OTBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7546  031732  001363                    BNE     10$             ; IF NOT, TX ANOTHER ONE
7547                            ;-------------------------------------------------------------------
7548  031734  004537  010010            JSR     R5,TXCTRL       ;SET TEOM
7549  031740  000002                    TEOM
7550  031742  000000                    0
7551  031744  004537  011614            JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7552  031750  000044                    36.
7553
7554  031752  004537  010110            JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7555  031756  000523                    RXSOM!123               ; & CHECK FOR RSOM=1
7556  031760  000000                    0
```

```
7557  031762  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7558  031764  103003                  BCC     .+8.            ;BR IF NO ERROR
7559  031766                          ERROR                   ;REPORT STACKED ERROR
7560  031766  104460                                                          TRAP    C$ERROR
7561  031770                          ESCAPE  SUB             ;SKIP TO END OF TEST
7562  031770  104410                                                          TRAP    C$ESCAPE
7563  031772  000136                                                          .WORD   L10066-.
7564
7565  031774  004537  010110          JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7566  032000  000321                  321
7567  032002  000000                  0
7568  032004  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7569  032006  103003                  BCC     .+8.            ;BR IF NO ERROR
7570  032010                          ERROR                   ;REPORT STACKED ERROR
7571  032010  104460                                                          TRAP    C$ERROR
7572  032012                          ESCAPE  SUB             ;SKIP TO END OF TEST
7573  032012  104410                                                          TRAP    C$ESCAPE
7574  032014  000114                                                          .WORD   L10066-.
7575                          ;-----------------------------------------------------------------
7576  032016  012703  033476          MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7577  032022  112337  032040  40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
7578  032026  142737  000360  032040  BICB    #360,4$         ;MASK OUT UNTRANSMITTED BITS
7579
7580  032034  004537  010110          JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7581  032040  000000          4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
7582  032042  000000                  0
7583  032044  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7584  032046  103003                  BCC     .+8.            ;BR IF NO ERROR
7585  032050                          ERROR                   ;REPORT STACKED ERROR
7586  032050  104460                                                          TRAP    C$ERROR
7587  032052                          ESCAPE  SUB             ;SKIP TO END OF TEST
7588  032052  104410                                                          TRAP    C$ESCAPE
7589  032054  000054                                                          .WORD   L10066-.
7590
7591  032056  022703  033503          CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7592  032062  001357                  BNE     40$             ; IF NOT, CHECK ANOTHER ONE
7593                          ;-----------------------------------------------------------------
7594  032064  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA6(006), RCV DATA7
7595  032070  000006                  006
7596  032072  000000                  0
7597  032074  020000                  NCRACT                  ;DON'T CHECK FOR FINAL RXACT=1
7598  032076  103003                  BCC     .+8.            ;BR IF NO ERROR
7599  032100                          ERROR                   ;REPORT STACKED ERROR
7600  032100  104460                                                          TRAP    C$ERROR
7601  032102                          ESCAPE  TST             ;SKIP TO END OF TEST
7602  032102  104410                                                          TRAP    C$ESCAPE
7603  032104  001370                                                          .WORD   L10063-.
7604
7605  032106  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA7(017), RCV FIRST FLAG
7606  032112  001017                  RXEOM!017               ; AND CHECK FOR REOM=1
7607  032114  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
7608  032116  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
7609  032120  103003                  BCC     .+8.            ;BR IF NO ERROR
7610  032122                          ERROR                   ;REPORT STACKED ERROR
7611  032122  104460                                                          TRAP    C$ERROR
7612  032124                          ESCAPE  TST             ;SKIP TO END OF TEST
```

```
                                                                                    TRAP    C$ESCAPE
                                                                                    .WORD   L10063-.
                        ENDSUB
                                                                          L10066:
                                                                                    TRAP    C$ESUB
                        ;-------------------------------------------------------------------
                        ; SUBROUTINE # 4: 5 BIT CHARACTERS
                        ;-------------------------------------------------------------------
                                BGNSUB
                                                                          T25.4:
                                                                                    TRAP    C$BSUB
007400                          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
                                SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
                                BIT2!BIT0               .INITIALLY: TXCL=8 BITS / RXCL=5 BITS
                                BCC     .+8.            ;BR IF NO ERROR
                                ERROR                   ;REPORT STACKED ERROR
                                                                                    TRAP    C$ERROR
                                ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                    TRAP    C$ESCAPE
                                                                                    .WORD   L10067-.
010010                          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
                                TSOM
                                7.
010010                          JSR     R5,TXCTRL       ;CLEAR TSOM
                                000
                                0
007676                          JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
                                123
                                8.
                                BCC     .+8.            ;BR IF NO ERROR
                                ERROR                   ;REPORT STACKED ERROR
                                                                                    TRAP    C$ERROR
                                ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                    TRAP    C$ESCAPE
                                                                                    .WORD   L10067-.
007676                          JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
                                321
                                8.
                                BCC     .+8.            ;BR IF NO ERROR
                                ERROR                   ;REPORT STACKED ERROR
                                                                                    TRAP    C$ERROR
                                ESCAPE  SUB             ;SKIP TO END OF TEST
                                                                                    TRAP    C$ESCAPE
                                                                                    .WORD   L10067-.
003734                          JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 5 BITS TOO
                                PCR
                                BIT7!BIT5!BIT2!BIT0
007676                          JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
                                111
                                8.
                                BCC     .+8.            ;BR IF NO ERROR
                                ERROR                   ;REPORT STACKED ERROR
                                                                                    TRAP    C$ERROR
```

```
7669  032260                                ESCAPE  SUB             ;SKIP TO END OF TEST
7670  032260  104410                                                                TRAP    C$ESCAPE
7671  032262  000234                                                                .WORD   L10067-.
7672                                  ;-----------------------------------------------------------------
7673  032264  012703  033477                MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7674  032270  112337  032300          10$:  MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7675
7676  032274  004537  007676                JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7677  032300  000000                  1$:   000                     ;** HOLE FOR NEXT CHARACTER **
7678  032302  000005                        5.
7679  032304  103003                        BCC     .+8.            ;BR IF NO ERROR
7680  032306                                ERROR                   ;REPORT STACKED ERROR
7681  032306  104460                                                                TRAP    C$ERROR
7682  032310                                ESCAPE  SUB             ;SKIP TO END OF TEST
7683  032310  104410                                                                TRAP    C$ESCAPE
7684  032312  000204                                                                .WORD   L10067-.
7685
7686  032314  022703  033505                CMP     #T30TBL+7,R5    ;ALL CHARACTERS TRANSMITTED ?
7687  032320  001363                        BNE     10$             ; IF NOT, TX ANOTHER ONE
7688                                  ;-----------------------------------------------------------------
7689  032322  004537  010010                JSR     R5,TXCTRL       ;SET TEOM
7690  032326  000002                        TEOM
7691  032330  000000                        0
7692  032332  004537  011614                JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7693  032336  000044                        36.
7694
7695  032340  004537  010110                JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
7696  032344  000523                        RXSOM!123               ; & CHECK FOR RSOM=1
7697  032346  000000                        0
7698  032350  100000                        NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7699  032352  103003                        BCC     .+8.            ;BR IF NO ERROR
7700  032354                                ERROR                   ;REPORT STACKED ERROR
7701  032354  104460                                                                TRAP    C$ERROR
7702  032356                                ESCAPE  SUB             ;SKIP TO END OF TEST
7703  032356  104410                                                                TRAP    C$ESCAPE
7704  032360  000136                                                                .WORD   L10067-.
7705
7706  032362  004537  010110                JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
7707  032366  000321                        321
7708  032370  000000                        0
7709  032372  100000                        NOCRDA                  ;NO INITIAL CHECK OF RDA=0
7710  032374  103003                        BCC     .+8.            ;BR IF NO ERROR
7711  032376                                ERROR                   ;REPORT STACKED ERROR
7712  032376  104460                                                                TRAP    C$ERROR
7713  032400                                ESCAPE  SUB             ;SKIP TO END OF TEST
7714  032400  104410                                                                TRAP    C$ESCAPE
7715  032402  000114                                                                .WORD   L10067-.
7716                                  ;-----------------------------------------------------------------
7717  032404  012703  033476                MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
7718  032410  112337  032426          40$:  MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
7719  032414  142737  000340  032426        BICB    #340,4$         ;MASK OUT UNTRANSMITTED BITS
7720
 721  032422  004537  010110                JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
7722  032426  000000                  4$:   000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
7723  032430  000000                        0
7724  032432  100000                        NOCRDA                  ;NO INITIAL CHECK OF RDA=0
```

```
7725  032434  103003                  BCC     .+8.            ;BR IF NO ERROR
7726  032436                          ERROR                   ;REPORT STACKED ERROR
7727  032436  104460                                                  TRAP    C$ERROR
7728  032440                          ESCAPE  SUB             ;SKIP TO END OF TEST
7729  032440  104410                                                  TRAP    C$ESCAPE
7730  032442  000054                                                  .WORD   L10067-.
7731
7732  032444  022703  033503          CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
7733  032450  001357                  BNE     40$             ; IF NOT, CHECK ANOTHER ONE
7734                                ;-----------------------------------------------------------------
7735  032452  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA6(026), RCV DATA7
7736  032456  000026                  026
7737  032460  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
7738  032462  020000                  NCRACT                  ;DON'T CHECK FOR INITIAL RDA=0
7739  032464  103003                  BCC     .+8.            ;BR IF NO ERROR
7740  032466                          ERROR                   ;REPORT STACKED ERROR
7741  032466  104460                                                  TRAP    C$ERROR
7742  032470                          ESCAPE  TST             ;SKIP TO END OF TEST
7743  032470  104410                                                  TRAP    C$ESCAPE
7744  032472  001002                                                  .WORD   L10063-.
7745
7746  032474  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA7(037), RCV FIRST FLAG
7747  032500  001037                  RXEOM!037               ; & CHECK FOR REOM=1
7748  032502  000000                  0
7749  032504  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
7750  032506  103003                  BCC     .+8.            ;BR IF NO ERROR
7751  032510                          ERROR                   ;REPORT STACKED ERROR
7752  032510  104460                                                  TRAP    C$ERROR
7753  032512                          ESCAPE  TST             ;SKIP TO END OF TEST
7754  032512  104410                                                  TRAP    C$ESCAPE
7755  032514  000760                                                  .WORD   L10063-.
7756  032516                        ENDSUB
7757  032516                                                  L10067:
7758  032516  104403                                                  TRAP    C$ESUB
7759                                ;-----------------------------------------------------------------
7760                                ; SUBROUTINE # 5: 6 BIT CHARACTERS
7761                                ;-----------------------------------------------------------------
7762  032520                        BGNSUB
7763  032520                                                  T25.5:
7764  032520  104402                                                  TRAP    C$BSUB
7765  032522  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
7766  032526  000226                  SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
7767  032530  000006                  BIT2!BIT1               ;INITIALLY: TXCL=8 BITS / RXCL=6 BITS
7768  032532  103003                  BCC     .+8.            ;BR IF NO ERROR
7769  032534                          ERROR                   ;REPORT STACKED ERROR
7770  032534  10446C                                                  TRAP    C$ERROR
7771  032536                          ESCAPE  SUB             ;SKIP TO END OF TEST
7772  032536  104410                                                  TRAP    C$ESCAPE
7773  032540  000344                                                  .WORD   L10070-.
7774
7775  032542  004537  010010          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
7776  032546  000001                  TSOM
7777  032550  000007                  7.
7778  032552  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
7779  032556  000000                  000
7780  032560  000000                  0
```

```
7781  032562  004537  007676          JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
7782  032566  000123                  123
7783  032570  000010                  8.
7784  032572  103003                  BCC     .+8.            ;BR IF NO ERROR
7785  032574                          ERROR                   ;REPORT STACKED ERROR
7786  032574  104460                                                  TRAP    C$ERROR
7787  032576                          ESCAPE  SUB             ;SKIP TO END OF TEST
7788  032576  104410                                                  TRAP    C$ESCAPE
7789  032600  000304                                                  .WORD   L10070-.
7790
7791  032602  004537  007676          JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
7792  032606  000321                  321
7793  032610  000010                  8.
7794  032612  103003                  BCC     .+8.            ;BR IF NO ERROR
7795  032614                          ERROR                   ;REPORT STACKED ERROR
7796  032614  104460                                                  TRAP    C$ERROR
7797  032616                          ESCAPE  SUB             ;SKIP TO END OF TEST
7798  032616  104410                                                  TRAP    C$ESCAPE
7799  032620  000264                                                  .WORD   L10070-.
7800
7801  032622  004537  003734          JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 6 BITS TOO
7802  032626  120407                  PCR
7803  032630  000306                  BIT7!BIT6!BIT2!BIT1
7804  032632  004537  007676          JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
7805  032636  000111                  111
7806  032640  000010                  8.
7807  032642  103003                  BCC     .+8.            ;BR IF NO ERROR
7808  032644                          ERROR                   ;REPORT STACKED ERROR
7809  032644  104460                                                  TRAP    C$ERROR
7810  032646                          ESCAPE  SUB             ;SKIP TO END OF TEST
7811  032646  104410                                                  TRAP    C$ESCAPE
7812  032650  000234                                                  .WORD   L10070-.
7813                          ;-----------------------------------------------------------------
7814  032652  012703  033477          MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
7815  032656  112337  032666  10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
7816
7817  032662  004537  007676          JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
7818  032666  000000  1$:             000                     ;** HOLE FOR NEXT CHARACTER **
7819  032670  000006                  6.
7820  032672  103003                  BCC     .+8.            ;BR IF NO ERROR
7821  032674                          ERROR                   ;REPORT STACKED ERROR
7822  032674  104460                                                  TRAP    C$ERROR
7823  032676                          ESCAPE  SUB             ;SKIP TO END OF TEST
7824  032676  104410                                                  TRAP    C$ESCAPE
7825  032700  000204                                                  .WORD   L10070-.
7826
7827  032702  022703  033505          CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
7828  032706  001363                  BNE     10$             ; IF NOT, TX ANOTHER ONE
7829                          ;-----------------------------------------------------------------
7830  032710  004537  010010          JSR     R5,TXCTRL       ;SET TEOM
7831  032714  000002                  TEOM
7832  032716  000000                  0
7833  032720  004537  011614          JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
7834  032724  000044                  36.
7835
7836  032726  004537  010110          JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
```

```
 7837  032732  000523                    RXSOM!123               ; & CHECK FOR RSOM=1
 7838  032734  000000                    0
 7839  032736  100000                    NOCRDA                  ;NO INITIAL CHECK OF RDA=0
 7840  032740  103003                    BCC     .+8.            ;BR IF NO ERROR
 7841  032742                            ERROR                   ;REPORT STACKED ERROR
 7842  032742  104460                                                            TRAP    C$ERROR
 7843  032744                            ESCAPE  SUB             ;SKIP TO END OF TEST
 7844  032744  104410                                                            TRAP    C$ESCAPE
 7845  032746  000136                                                            .WORD   L10070-.
 7846
 7847  032750  004537  010110            JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
 7848  032754  000321                    321
 7849  032756  000000                    0
 7850  032760  100000                    NOCRDA                  ;NO INITIAL CHECK OF RDA=0
 7851  032762  103003                    BCC     .+8.            ;BR IF NO ERROR
 7852  032764                            ERROR                   ;REPORT STACKED ERROR
 7853  032764  104460                                                            TRAP    C$ERROR
 7854  032766                            ESCAPE  SUB             ;SKIP TO END OF TEST
 7855  032766  104410                                                            TRAP    C$ESCAPE
 7856  032770  000114                                                            .WORD   L10070-.
 7857                            ;-----------------------------------------------------------------
 7858  032772  012703  033476            MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
 7859  032776  112337  033014    40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED Rx CHARACTER
 7860  033002  142737  000300  033014    BICB    #300,4$         ;MASK OUT UNTRANSMITTED BITS
 7861
 7862  033010  004537  010110            JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
 7863  033014  000000            4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
 7864  033016  000000                    0
 7865  033020  100000                    NOCRDA                  ;NO INITIAL CHECK OF RDA=0
 7866  033022  103003                    BCC     .+8.            ;BR IF NO ERROR
 7867  033024                            ERROR                   ;REPORT STACKED ERROR
 7868  033024  104460                                                            TRAP    C$ERROR
 7869  033026                            ESCAPE  SUB             ;SKIP TO END OF TEST
 7870  033026  104410                                                            TRAP    C$ESCAPE
 7871  033030  000054                                                            .WORD   L10070-.
 7872
 7873  033032  022703  033503            CMP     #T30TBL+5,R3    ;ALL CHARACTERS CHECKED ?
 7874  033036  001357                    BNE     40$             ; IF NOT, CHECK ANOTHER ONE
 7875                            ;-----------------------------------------------------------------
 7876  033040  004537  010110            JSR     R5,RXCHAR       ;READ/CHK DATA6(066), RCV DATA7
 7877  033044  000066                    066
 7878  033046  000001                    RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
 7879  033050  020000                    NCRACT                  ;DON'T CHECK FOR INITIAL RDA=0
 7880  033052  103003                    BCC     .+8.            ;BR IF NO ERROR
 7881  033054                            ERROR                   ;REPORT STACKED ERROR
 7882  033054  104460                                                            TRAP    C$ERROR
 7883  033056                            ESCAPE  TST             ;SKIP TO END OF TEST
 7884  033056  104410                                                            TRAP    C$ESCAPE
 7885  033060  000414                                                            .WORD   L10063-.
 7886
 7887  033062  004537  010110            JSR     R5,RXCHAR       ;READ/CHK DATA7(077), RCV FIRST FLAG
 7888  033066  001077                    RXEOM!077               ; & CHECK FOR REOM=1
 7889  033070  000000                    0
 7890  033072  060000                    NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
 7891  033074  103003                    BCC     .+8.            ;BR IF NO ERROR
 7892  033076                            ERROR                   ;REPORT STACKED ERROR
```

```
033076  104460                                                                  TRAP     C$ERROR
033100                          ESCAPE  TST             ;SKIP TO END OF TEST
033100  104410                                                                  TRAP     C$ESCAPE
033102  000372                                                                  .WORD    L10063-.
033104                  ENDSUB
033104                                                                  L10070:
033104  104403                                                                  TRAP     C$ESUB
                ;--------------------------------------------------------------------
                ; SUBROUTINE # 6: 7 BIT CHARACTERS
                ;--------------------------------------------------------------------
033106                          BGNSUB
033106                                                                  T25.6:
033106  104402                                                                  TRAP     C$BSUB
033110  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
033114  000226                  SYNCH                   ;SET BOP MODE,CRC-CCITT-1,S/AR=226
033116  000007                  RXDL                    ;INITIALLY: TXCL=8 BITS / RXCL=7 BITS
033120  103003                  BCC     .+8.            ;BR IF NO ERROR
033122                          ERROR                   ;REPORT STACKED ERROR
033122  104460                                                                  TRAP     C$ERROR
033124                          ESCAPE  SUB             ;SKIP TO END OF TEST
033124  104410                                                                  TRAP     C$ESCAPE
033126  000344                                                                  .WORD    L10071-.

033130  004537  010010          JSR     R5,TXCTRL       ;LOAD 2ND FLAG,TX 1ST FLAG
033134  000001                  TSOM
033136  000007                  7.
033140  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
033144  000000                  000
033146  000000                  0
033150  004537  007676          JSR     R5,TXCHAR       ;LOAD 123(HEADER1), TX 2ND FLAG
033154  000123                  123
033156  000010                  8.
033160  103003                  BCC     .+8.            ;BR IF NO ERROR
033162                          ERROR                   ;REPORT STACKED ERROR
033162  104460                                                                  TRAP     C$ERROR
033164                          ESCAPE  SUB             ;SKIP TO END OF TEST
033164  104410                                                                  TRAP     C$ESCAPE
033166  000304                                                                  .WORD    L10071-.

033170  004537  007676          JSR     R5,TXCHAR       ;LOAD 321(HEADER2), TX 123(HEADER1)
033174  000321                  321
033176  000010                  8.
033200  103003                  BCC     .+8.            ;BR IF NO ERROR
033202                          ERROR                   ;REPORT STACKED ERROR
033202  104460                                                                  TRAP     C$ERROR
033204                          ESCAPE  SUB             ;SKIP TO END OF TEST
033204  104410                                                                  TRAP     C$ESCAPE
033206  000264                                                                  .WORD    L10071-.

033210  004537  003734          JSR     R5,WRITEI       ;NOW CHANGE TXCL TO 6 BITS TOO
033214  120407                  PCR
033216  000347                  TXDL!RXDL
033220  004537  007676          JSR     R5,TXCHAR       ;LOAD 111(DATA1), TX 321(HEADER2)
033224  000111                  111
033226  000010                  8.
033230  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
 7949  033232                                 ERROR                   ;REPORT STACKED ERROR
 7950  033232  104460                                                                  TRAP    C$ERROR
 7951  033234                                 ESCAPE  SUB             ;SKIP TO END OF TEST
 7952  033234  104410                                                                  TRAP    C$ESCAPE
 7953  033236  000234                                                                  .WORD   L10071-.
 7954                                 ;-----------------------------------------------------------------
 7955  033240  012703  033477                MOV     #T30TBL+1,R3    ;SET UP DATA TABLE POINTER
 7956  033244  112337  033254        10$:    MOVB    (R3)+,1$        ;INSTALL NEXT TX CHARACTER
 7957
 7958  033250  004537  007676                JSR     R5,TXCHAR       ;TRANSMIT CHARACTER ( ==> RX/FIFO )
 7959  033254  000000                 1$:     000                     ;** HOLE FOR NEXT CHARACTER **
 7960  033256  000007                         7.
 7961  033260  103003                         BCC     .+8.            ;BR IF NO ERROR
 7962  033262                                 ERROR                   ;REPORT STACKED ERROR
 7963  033262  104460                                                                  TRAP    C$ERROR
 7964  033264                                 ESCAPE  SUB             ;SKIP TO END OF TEST
 7965  033264  104410                                                                  TRAP    C$ESCAPE
 7966  033266  000204                                                                  .WORD   L10071-.
 7967
 7968  033270  022703  033505                CMP     #T30TBL+7,R3    ;ALL CHARACTERS TRANSMITTED ?
 7969  033274  001363                         BNE     10$             ; IF NOT, TX ANOTHER ONE
 7970                                 ;-----------------------------------------------------------------
 7971  033276  004537  010010                JSR     R5,TXCTRL       ;SET TEOM
 7972  033302  000002                         TEOM
 7973  033304  000000                         0
 7974  033306  004537  011614                JSR     R5,STEPLU       ; AND TX DATA7 + SOME FLAGS
 7975  033312  000044                         36.
 7976
 7977  033314  004537  010110                JSR     R5,RXCHAR       ;READ & CHK 123(HEADER1), RCV 321(HEADER2)
 7978  033320  000523                         RXSOM!123               ; & CHECK FOR RSOM=1
 7979  033322  000000                         0
 7980  033324  100000                         NOCRDA                  ;NO INITIAL CHECK OF RDA=0
 7981  033326  103003                         BCC     .+8.            ;BR IF NO ERROR
 7982  033330                                 ERROR                   ;REPORT STACKED ERROR
 7983  033330  104460                                                                  TRAP    C$ERROR
 7984  033332                                 ESCAPE  SUB             ;SKIP TO END OF TEST
 7985  033332  104410                                                                  TRAP    C$ESCAPE
 7986  033334  000136                                                                  .WORD   L10071-.
 7987
 7988  033336  004537  010110                JSR     R5,RXCHAR       ;READ/CHECK 321(HEADER2),RCV 011(DATA1)
 7989  033342  000321                         321
 7990  033344  000000                         0
 7991  033346  100000                         NOCRDA                  ;NO INITIAL CHECK OF RDA=0
 7992  033350  103003                         BCC     .+8.            ;BR IF NO ERROR
 7993  033352                                 ERROR                   ;REPORT STACKED ERROR
 7994  033352  104460                                                                  TRAP    C$ERROR
 7995  033354                                 ESCAPE  SUB             ;SKIP TO END OF TEST
 7996  033354  104410                                                                  TRAP    C$ESCAPE
 7997  033356  000114                                                                  .WORD   L10071-.
 7998                                 ;-----------------------------------------------------------------
 7999  033360  012703  033476                MOV     #T30TBL,R3      ;SET UP DATA TABLE POINTER
 8000  033364  112337  033402        40$:    MOVB    (R3)+,4$        ;INSTALL NEXT EXPECTED RX CHARACTER
 8001  033370  142737  000200  033402        BICB    #200,4$         ;MASK OUT UNTRANSMITTED BITS
 8002
 8003  033376  004537  010110                JSR     R5,RXCHAR       ;READ/CHECK NEXT CHARACTER
 8004  033402  000000                 4$:     000                     ;** HOLE FOR NEXT EXPECTED CHARACTER
```

```
8005  033404  000000                  0
8006  033406  100000                  NOCRDA                  ;NO INITIAL CHECK OF RDA=0
8007  033410  103003                  BCC     .+8.            ;BR IF NO ERROR
8008  033412                          ERROR                   ;REPORT STACKED ERROR
8009  033412  104460                                                          TRAP    C$ERROR
8010  033414                          ESCAPE  SUB             ;SKIP TO END OF TEST
8011  033414  104410                                                          TRAP    C$ESCAPE
8012  033416  000054                                                          .WORD   L10071-.
8013
8014  033420  022703  033503          CMP     @T3OTBL+5,R3    ;ALL CHARACTERS CHECKED ?
8015  033424  001357                  BNE     40$             ; IF NOT, CHECK ANOTHER ONE
8016                                ;-----------------------------------------------------------
8017  033426  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA6(066), RCV DATA7
8018  033432  000066                  066
8019  033434  000001                  RERCHK                  ; & CHECK RERR BIT=0 (GOOD CRC)
8020  033436  020000                  NCRACT                  ;DON'T CHECK FOR INITIAL RDA=0
8021  033440  103003                  BCC     .+8.            ;BR IF NO ERROR
8022  033442                          ERROR                   ;REPORT STACKED ERROR
8023  033442  104460                                                          TRAP    C$ERROR
8024  033444                          ESCAPE  TST             ;SKIP TO END OF TEST
8025  033444  104410                                                          TRAP    C$ESCAPE
8026  033446  000026                                                          .WORD   L10063-.
8027
8028  033450  004537  010110          JSR     R5,RXCHAR       ;READ/CHK DATA7(177), RCV FIRST FLAG
8029  033454  001177                  RXEOM!177               ; & CHECK FOR REOM=1
8030  033456  000000                  0
8031  033460  060000                  NFCRDA!NCRACT           ;DON'T CHECK FOR FINAL RDA=RXACT=1
8032  033462  103003                  BCC     .+8.            ;BR IF NO ERROR
8033  033464                          ERROR                   ;REPORT STACKED ERROR
8034  033464  104460                                                          TRAP    C$ERROR
8035  033466                          ESCAPE  TST             ;SKIP TO END OF TEST
8036  033466  104410                                                          TRAP    C$ESCAPE
8037  033470  000004                                                          .WORD   L10063-.
8038  033472
8039  033472                            ENDSUB                                L10071:
8040  033472  104403                                                          TRAP    C$ESUB
8041  033474
8042  033474                          ENDTST                                  L10063:
8043  033474  104401                                                          TRAP    C$ETST
8044
8045  033476     111                  ;-------------------------------
8046  033477     222                  T3OTBL: .BYTE   111     ;D1
8047  033500     333                          .BYTE   222     ;D2
8048  033501     044                          .BYTE   333     ;D3
8049  033502     155                          .BYTE   044     ;D4
8050  033503     266                          .BYTE   155     ;D5
8051  033504     377                          .BYTE   266     ;D6
8052           033506                         .BYTE   377     ;D7
8053                                          .EVEN
                                    ;-------------------------------
```

```
                                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
033506                                                                  T26::
033506  004737  005420                JSR     PC,INIDMV       ;INIT VIA
                                ;-------------------------------------------------------------------
                                ; SET UP USYRT AND VIA REGISTERS
                                ;-------------------------------------------------------------------
033512  004537  003734                JSR     R5,WRITEI       ;SET ACR FOR T1 ONE-SHOT MODE
033516  120013                        VIAACR
033520  000200                        200
033522  004537  003734                JSR     R5,WRITEI       ;LOAD VIA T1L-L
033526  120006                        VIAT1C
033530  000300                        300
033532  004537  003734                JSR     R5,WRITEI       ;LOAD VIA T1L-H
033536  120007                        VIAT1D
033540  000000                        000

033542  004537  003734                JSR     R5,WRITEI       ;RESET THE USYRT
033546  120000                        VIAORB
033550  000031                        RTSND!DTR!PRESET
033552  004537  003734                JSR     R5,WRITEI       ;CLEAR USYRT RESET BIT
033556  120000                        VIAORB
033560  000030                        RTSND!DTR

033562  004537  003734                JSR     R5,WRITEI       ;LOAD USYRT PCSARL
033566  120404                        PCSARL
033570  000000                        000
033572  004537  003734                JSR     R5,WRITEI       ;LOAD USYRT PCSARH
033576  120405                        PCSARH
033600  000007                        XYZ                     ;(NO ERROR CHECKING)

033602  004537  003734                JSR     R5,WRITEI       ;SET UP USYRT
033606  120000                        VIAORB
033610  000172                        RTSND!TXEN!RXEN!DTR!TTLOOP

033612  004537  006036                JSR     R5,CKTBMT       ;CHK FOR TBMT = 1
033616  000001                        1
033620  103003                        BCC     .+8.            ;IF NO ERROR, PROCEED
033622                                ERROR                   ;ELSE, REPORT IT AND
033622  104460                                                                TRAP    C$ERROR
033624                                ESCAPE  TST             ;       EXIT THIS TEST
033624  104410                                                                TRAP    C$ESCAPE
033626  000234                                                                .WORD   L10072-.
                                ;-------------------------------------------------------------------
```

```
8110                                    ; INIT SPACING SEQUENCE BY SETTING TSOM AND TEOM
8111                                    ;-----------------------------------------------------------------
8112  033630  004537  010010            6$:     JSR     R5,TXCTRL       ;SET TSOM AND TEOM
8113  033634  000003                            TSOM!TEOM               ; ("SPACING SEQUENCE")
8114  033636  000000                            0
8115
8116  033640  004537  011614                    JSR     R5,STEPLU       ;1 TICK TO START SPACE SEQUENCE
8117  033644  000001                            1
8118
8119  033646  004537  007256                    JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
8120  033652  000010                            8.                      ; 8 BIT CHAR/CLOCK TICKS
8121  033654  000000                            000                     ; EXPECTED BIT SEQUENCE (0000000)
8122  033656  103003                            BCC     .+8.            ;BR IF NO ERROR
8123  033660                                    ERROR                   ;REPORT STACKED ERROR
8124  033660  104460                                                                    TRAP    C$ERROR
8125  033662                                    ESCAPE  TST             ;AND EXIT TEST
8126  033662  104410                                                                    TRAP    C$ESCAPE
8127  033664  000176                                                                    .WORD   L10072-.
8128
8129  033666  004537  006036                    JSR     R5,CKTBMT       ;CHK FOR TBMT = 1
8130  033672  000001                            1
8131  033674  103003                            BCC     .+8.            ;IF NO ERROR, PROCEED
8132  033676                                    ERROR                   ;ELSE, REPORT IT AND
8133  033676  104460                                                                    TRAP    C$ERROR
8134  033700                                    ESCAPE  TST             ;       EXIT THIS TEST
8135  033700  104410                                                                    TRAP    C$ESCAPE
8136  033702  000160                                                                    .WORD   L10072-.
8137                                    ;-----------------------------------------------------------------
8138  033704  004537  010010                    JSR     R5,TXCTRL       ;CLEAR TEOM
8139  033710  000001                            TSOM
8140  033712  000000                            0
8141
8142  033714  004537  007256                    JSR     R5,SERIAL       ;READ 000 CHARACTER VIA TSO
8143  033720  000010                            8.                      ; 8 BIT CHAR/CLOCK TICKS
8144  033722  000000                            000                     ; EXPECTED BIT SEQUENCE (00000000)
8145  033724  103003                            BCC     .+8.            ;BR IF NO ERROR
8146  033726                                    ERROR                   ;REPORT STACKED ERROR
8147  033726  104460                                                                    TRAP    C$ERROR
8148  033730                                    ESCAPE  TST             ;AND EXIT TEST
8149  033730  104410                                                                    TRAP    C$ESCAPE
8150  033732  000130                                                                    .WORD   L10072-.
8151
8152  033734  004537  006036                    JSR     R5,CKTBMT       ;CHK FOR TBMT = 1
8153  033740  000001                            1
8154  033742  103003                            BCC     .+8.            ;IF NO ERROR, PROCEED
8155  033744                                    ERROR                   ;ELSE, REPORT IT AND
8156  033744  104460                                                                    TRAP    C$ERROR
8157  033746                                    ESCAPE  TST             ;       EXIT THIS TEST
8158  033746  104410                                                                    TRAP    C$ESCAPE
8159  033750  000112                                                                    .WORD   L10072-.
8160                                    ;-----------------------------------------------------------------
8161                                    ; SEND "SHORT MESSAGE" (125,252)
8162                                    ;-----------------------------------------------------------------
8163  033752  004537  003734                    JSR     R5,WRITEI       ;LOAD 125 CHARACTER
8164  033756  120402                            TDSRL
8165  033760  000125                            125
```

```
 8166  033762  004537  010010                JSR     R5,TXCTRL       ;CLEAR TSOM
 8167  033766  000000                        000
 8168  033770  000000                        0
 8169
 8170  033772  004537  007256                JSR     R5,SERIAL       ;READ FLAG CHARACTER (176) VIA TSO
 8171  033776  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 8172  034000  000176                        176                     ; EXPECTED BIT SEQUENCE (01111110)
 8173  034002  103003                        BCC     .+8.            ;BR IF NO ERROR
 8174  034004                                ERROR                   ;REPORT STACKED ERROR
 8175  034004  104460                                                                TRAP    C$ERROR
 8176  034006                                ESCAPE  TST             ;AND EXIT TEST
 8177  034006  104410                                                                TRAP    C$ESCAPE
 8178  034010  000052                                                                .WORD   L10072-.
 8179
 8180  034012  004537  003734                JSR     R5,WRITEI       ;LOAD 252 CHARACTER
 8181  034016  120402                        TDSRL
 8182  034020  000252                        252
 8183
 8184  034022  004537  007256                JSR     R5,SERIAL       ;READ 1ST DATA CHARACTER (125) VIA TSO
 8185  034026  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 8186  034030  000252                        252                     ; EXPECTED BIT SEQUENCE (01010110)
 8187  034032  103003                        BCC     .+8.            ;BR IF NO ERROR
 8188  034034                                ERROR                   ;REPORT STACKED ERROR
 8189  034034  104460                                                                TRAP    C$ERROR
 8190  034036                                ESCAPE  TST             ;AND EXIT TEST
 8191  034036  104410                                                                TRAP    C$ESCAPE
 8192  034040  000022                                                                .WORD   L10072-.
 8193
 8194  034042  004537  007256                JSR     R5,SERIAL       ;READ 2ND DATA CHARACTER (252) VIA TSO
 8195  034046  000010                        8.                      ; 8 BIT CHAR/CLOCK TICKS
 8196  034050  000125                        125                     ; EXPECTED BIT SEQUENCE (10101010)
 8197  034052  103003                        BCC     .+8.            ;BR IF NO ERROR
 8198  034054                                ERROR                   ;REPORT STACKED ERROR
 8199  034054  104460                                                                TRAP    C$ERROR
 8200  034056                                ESCAPE  TST             ;AND EXIT TEST
 8201  034056  104410                                                                TRAP    C$ESCAPE
 8202  034060  000002                                                                .WORD   L10072-.
 8203  034062                        ENDTST
 8204  034062                                                                L10072:
 8205  034062  104401                                                                TRAP    C$ETST
```

```
                                .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
                                                                                T27::
034064                                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
034064  004737  005420
034070  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
034074  043626                          DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
034076  000000                          0                       ;USE 8 BIT CHARS
034100  103003                          BCC     .+8.            ;BR IF NO ERROR
034102                                  ERROR                   ;REPORT STACKED ERROR
                                                                                TRAP    C$ERROR
034102  104460
034104                                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
034104  104410
034106  000324                                                                  .WORD   L10073-.
034110  004537  010010                  JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
034114  000001                          TSOM                    ;
034116  000007                          7.

034120  004537  010010                  JSR     R5,TXCTRL       ;CLEAR TSOM
034124  000000                          000
034126  000000                          0

                                ;-----------------------------------------------------------
                                ; FILL THE FIFO WITH CHARACTERS AND FORCE OVERRUN
                                ;-----------------------------------------------------------

034130  012702  034434                  MOV     #TXTBL4,R2      ;SET UP TABLE POINTER
034134  112237  034144          5$:     MOVB    (R2)+,10$       ;SETUP TRANSMIT CHARACTER

034140  004537  007676                  JSR     R5,TXCHAR       ;TRANSMIT A CHARACTER
034144  000000                  10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
034146  100010                          NCTBMT+256.!8.          ;NO CHECK OF INITIAL TBMT=0
034150  103003                          BCC     .+8.            ;BR IF NO ERROR
034152                                  ERROR                   ;REPORT STACKED ERROR
                                                                                TRAP    C$ERROR
034152  104460
034154                                  ESCAPE  TST             ;SKIP TO END OF TEST
                                                                                TRAP    C$ESCAPE
034154  104410
034156  000254                                                                  .WORD   L10073-.

```

```
 8262  034160  022702  03444/          CMP     @TXEND4,R2      ;CONTINUE TRANSMITTING UNTIL
 8263  034164  001363                  BNE     5$              ;OVERRUN.
 8264
 8265  034166  004537  006476          JSR     R5,CKROR        ;VERIFY RECEIVER OVERRUN OCCURED
 8266  034172  000001                  1
 8267  034174  103003                  BCC     .+8.            ;BR IF OK
 8268  034176                          ERROR                   ;REPORT STACKED ERROR
 8269  034176  104460                                                          TRAP    C$ERROR
 8270  034200                          ESCAPE  TST             ;SKIP TO END OF TEST
 8271  034200  104410                                                          TRAP    C$ESCAPE
 8272  034202  000230                                                          .WORD   L10073-.
 8273                                  ;-------------------------------------------------------
 8274                                  ; THE RECEIVER SHOULD HAVE OVERRUN. VERIFY THIS AND READ/VERIFY
 8275                                  ; THE REMAINING FIFO CHARACTERS (ESPECIALLY THE LAST ONE).
 8276                                  ;-------------------------------------------------------
 8277  034204  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
 8278  034210  000226                  SYNCH                   ; (THIS CHARACTER ISN'T AFFECTED)
 8279  034212  000000                  0
 8280  034214  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
 8281  034216  103003                  BCC     .+8.            ;BR IF NO ERROR
 8282  034220                          ERROR                   ;REPORT STACKED ERROR
 8283  034220  104460                                                          TRAP    C$ERROR
 8284  034222                          ESCAPE  TST             ;SKIP TO END OF TEST
 8285  034222  104410                                                          TRAP    C$ESCAPE
 8286  034224  000206                                                          .WORD   L10073-.
 8287
 8288  034226  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
 8289  034232  000000                  000                     ; (THIS IS THE OVERWRITTEN CHARACTER)
 8290  034234  000000                  0                       ; ...WAS A "SYNCH" BEFORE OVERWRITTEN
 8291  034236  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
 8292  034240  103003                  BCC     .+8.            ;BR IF NO ERROR
 8293  034242                          ERROR                   ;REPORT STACKED ERROR
 8294  034242  104460                                                          TRAP    C$ERROR
 8295  034244                          ESCAPE  TST             ;SKIP TO END OF TEST
 8296  034244  104410                                                          TRAP    C$ESCAPE
 8297  034246  000164                                                          .WORD   L10073-.
 8298
 8299  034250  012702  034437          MOV     @TXTBL4+3,R2    ;SET UP TABLE POINTER
 8300  034254  112237  034264  15$:    MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
 8301
 8302  034260  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
 8303  034264  000000  20$:    000                     ;** HOLE FOR EXPECTED RECEIVE CHAR.
 8304  034266  000000                  0                       ;
 8305  034270  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
 8306  034272  103003                  BCC     .+8.            ;BR IF NO ERROR
 8307  034274                          ERROR                   ;REPORT STACKED ERROR
 8308  034274  104460                                                          TRAP    C$ERROR
 8309  034276                          ESCAPE  TST             ;SKIP TO END OF TEST
 8310  034276  104410                                                          TRAP    C$ESCAPE
 8311  034300  000132                                                          .WORD   L10073-.
 8312  034302  022702  034444          CMP     @TXEND4-3,R2
 8313  034306  001362                  BNE     15$
 8314
 8315  034310  012702  000004          MOV     #4,R2           ;TRANSMIT 4 EXTRA CHARACTERS TO AVOID
 8316  034314  004537  007676  30$:    JSR     R5,TXCHAR       ;TRANSMITTER UNDERRUN.
 8317  034320  000333                  333                     ;FILLER CHARACTER
```

```
 8318  034322  100010                  NCTBMT=256.!8.          ;NO CHECK OF INITIAL TBMT=0
 8319  034324  077205                  SOB     R2,.-2
 8320
 8321  034326  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
 8322  034332  000303                  303
 8323  034334  000000                  0
 8324  034336  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
 8325  034340  103003                  BCC     .+8.            ;BR IF NO ERROR
 8326  034342                          ERROR                   ;REPORT STACKED ERROR
 8327  034342  104460                                                          TRAP    C$ERROR
 8328  034344                          ESCAPE  TST             ;SKIP TO END OF TEST
 8329  034344  104410                                                          TRAP    C$ESCAPE
 8330  034346  000064                                                          .WORD   L10073-.
 8331
 8332  034350  004537  010110          JSR     R5,RXCHAR       ;READ & CHK CHARACTER
 8333  034354  000074                  074
 8334  034356  000000                  0
 8335  034360  100000                  NOCRDA          ;NO INITIAL CHECK OF RDA=0
 8336  034362  103003                  BCC     .+8.            ;BR IF NO ERROR
 8337  034364                          ERROR                   ;REPORT STACKED ERROR
 8338  034364  104460                                                          TRAP    C$ERROR
 8339  034366                          ESCAPE  TST             ;SKIP TO END OF TEST
 8340  034366  104410                                                          TRAP    C$ESCAPE
 8341  034370  000042                                                          .WORD   L10073-.
 8342
 8343  034372  004537  010110          JSR     R5,RXCHAR       ;* READ & CHK FINAL CHARACTER
 8344  034376  000125                  125                     ;*
 8345  034400  000000                  0                       ;*
 8346  034402  100000                  NOCRDA          ;* NO INITIAL CHECK OF RDA=0
 8347  034404  103003                  BCC     .+8.            ;* BR IF NO ERROR
 8348  034406                          ERROR                   ;* REPORT STACKED ERROR
 8349  034406  104460                                                          TRAP    C$ERROR
 8350  034410                          ESCAPE  TST             ;* SKIP TO END OF TEST
 8351  034410  104410                                                          TRAP    C$ESCAPE
 8352  034412  000020                                                          .WORD   L10073-.
 8353
 8354  034414  004537  011532          JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
 8355  034420  000011                  9.
 8356  034422  103003                  BCC     .+8.            ;BR IF NO ERROR
 8357  034424                          ERROR                   ;REPORT STACKED ERROR
 8358  034424  104460                                                          TRAP    C$ERROR
 8359  034426                          ESCAPE  TST             ;SKIP TO END OF TEST
 8360  034426  104410                                                          TRAP    C$ESCAPE
 8361  034430  000002                                                          .WORD   L10073-.
 8362  034432                  ENDTST
 8363  034432                                                          L10073:
 8364  034432  104401                                                          TRAP    C$ETST
 8365                          ;------------------------------
 8366  034434     226          TXTBL4: .BYTE   226
 8367  034435     226                  .BYTE   226
 8368  034436     000                  .BYTE   000
 8369  034437     377                  .BYTE   377
 8370  034440     125                  .BYTE   125
 8371  034441     252                  .BYTE   252
 8372  034442     347                  .BYTE   347
 8373  034443     030                  .BYTE   030
```

```
 8374  034444  303                     .BYTE   303
 8375  034445  074                     .BYTE   074
 8376  034446  125                     .BYTE   125
 8377  034447  000             TXEND4: .BYTE   000
 8378                                   .EVEN
 8379                            ;------------------------------
```

```
                                .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:
                                                                                        TRAP    C$BSUB
8410  034450  104402
8411  034452  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
8412
8413  034456  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
8414  034462  043626                  DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
8415  034464  000000                  0                       ;USE 8 BIT CHARS
8416  034466  103003                  BCC     .+8.            ;BR IF NO ERROR
8417  034470                          ERROR                   ;REPORT STACKED ERROR
8418  034470  104460                                                        TRAP    C$ERROR
8419  034472                          ESCAPE  SUB             ;SKIP TO END OF TEST
8420  034472  104410                                                        TRAP    C$ESCAPE
8421  034474  000112                                                        .WORD   L10075-.
8422
8423  034476  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
8424  034502  000001                  TSOM                    ;
8425  034504  000007                  7.
8426  034506  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
8427  034512  000001                  TSOM                    ;
8428  034514  000010                  8.
8429  034516  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
8430  034522  000000                  000
8431  034524  000000                  0
8432
8433  034526  004537  006476          JSR     R5,CKROR        ;CHECK FOR RXOR = 0
8434  034532  000000                  0
8435  034534  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
8436  034536                                ERROR                   ;REPORT STACKED ERROR
8437  034536  104460                                                                TRAP    C$ERROR
8438  034540                                ESCAPE  SUB             ;SKIP TO END OF TEST
8439  034540  104410                                                                TRAP    C$ESCAPE
8440  034542  000044                                                                .WORD   L10075-.
8441
8442  034544  004537  011614               JSR     R5,STEPLU       ;FORCE RECEIVER OVERRUN
8443  034550  000116                       78.
8444
8445  034552  004537  006476               JSR     R5,CKROR        ;CHECK FOR RXOR = 1
8446  034556  000001                       1
8447  034560  103003                       BCC     .+8.            ;BR IF NO ERROR
8448  034562                               ERROR                   ;REPORT STACKED ERROR
8449  034562  104460                                                                TRAP    C$ERROR
8450  034564                               ESCAPE  TST             ;SKIP TO END OF TEST
8451  034564  104410                                                                TRAP    C$ESCAPE
8452  034566  000262                                                                .WORD   L10074-.
8453
8454  034570  004537  006476               JSR     R5,CKROR        ;CHECK FOR RXOR = 0
8455  034574  000000                       0
8456  034576  103003                       BCC     .+8.            ;BR IF NO ERROR
8457  034600                               ERROR                   ;REPORT STACKED ERROR
8458  034600  104460                                                                TRAP    C$ERROR
8459  034602                               ESCAPE  SUB             ;SKIP TO END OF TEST
8460  034602  104410                                                                TRAP    C$ESCAPE
8461  034604  000002                                                                .WORD   L10075-.
8462  034606                               ENDSUB
8463  034606                                                                L10075:
8464  034606  104403                                                                TRAP    C$ESUB
8465                        ;*************** SUBTEST #2 ***************
8466  034610                               BGNSUB
8467  034610                                                                T28.2:
8468  034610  104402                                                                TRAP    C$BSUB
8469  034612  004737  005420       JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
8470
8471  034616  004537  007400       JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
8472  034622  043626               DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
8473  034624  000000               0                       ;USE 8 BIT CHARS
8474  034626  103003               BCC     .+8.            ;BR IF NO ERROR
8475  034630                       ERROR                   ;REPORT STACKED ERROR
8476  034630  104460                                                                TRAP    C$ERROR
8477  034632                       ESCAPE  SUB             ;SKIP TO END OF TEST
8478  034632  104410                                                                TRAP    C$ESCAPE
8479  034634  000054                                                                .WORD   L10076-.
8480
8481  034636  004537  010010       JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
8482  034642  000001               TSOM                    ;
8483  034644  000007               7.
8484  034646  004537  010010       JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
8485  034652  000001               TSOM                    ;
8486  034654  000010               8.
8487  034656  004537  010010       JSR     R5,TXCTRL       ;CLEAR TSOM
8488  034662  000000               000
8489  034664  000000               0
8490  034666  004537  011614       JSR     R5,STEPLU       ;FORCE RECEIVER OVERRUN
8491  034672  000116               78.
```

```
8492
8493  034674  004537  005126          JSR     R5,RSTCHK       ;RESET USYRT/VERIFY SAME
8494  034700  103003                  BCC     .+8.            ;BR IF NO ERROR
8495  034702                          ERROR                   ;REPORT STACKED ERROR
8496  034702  104460                                                  TRAP    C$ERROR
8497  034704                          ESCAPE TST              ;SKIP TO END OF TEST
8498  034704  104410                                                  TRAP    C$ESCAPE
8499  034706  000142                                                  .WORD   L10074-.
8500  034710                          ENDSUB
8501  034710                                                      L10076:
8502  034710  104403                                                  TRAP    C$ESUB
8503                          ;*************** SUBTEST #3 ***************
8504  034712                          BGNSUB
8505  034712                                                      T28.3:
8506  034712  104402                                                  TRAP    C$BSUB
8507  034714  004737  005420          JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
8508
8509  034720  004537  007400          JSR     R5,INITRN       ;LOAD 1 SOM, CLK T  NTIL ACTIVE
8510  034724  043626                  DDCMP!NOCHK!SYNCH ;SET DDCMP, NO ERR CHECKING, SYNCH=226
8511  034726  000000                  0                       ;USE 8 BIT CHARS
8512  034730  103003                  BCC     .+8.            ;BR IF NO ERROR
8513  034732                          ERROR                   ;REPORT STACKED ERROR
8514  034732  104460                                                  TRAP    C$ERROR
8515  034734                          ESCAPE  SUB             ;SKIP TO END OF TEST
8516  034734  104410                                                  TRAP    C$ESCAPE
8517  034736  000110                                                  .WORD   L10077-.
8518
8519  034740  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTERS
8520  034744  000001                  TSOM                    ;
8521  034746  000007                  7.
8522  034750  004537  010010          JSR     R5,TXCTRL       ;OUTPUT 2ND SYNC CHARACTER
8523  034754  000001                  TSOM                    ;
8524  034756  000010                  8.
8525  034760  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
8526  034764  000000                  000
8527  034766  000000                  0
8528  034770  004537  011614          JSR     R5,STEPLU       ;FORCE RECEIVER OVERRUN
8529  034774  000116                  78.
8530
8531  034776  004537  003734          JSR     R5,WRITEI       ;DROP RECEIVER ENABLE (RXEN)
8532  035002  120000                  VIAORB                  ; (RDA SHOULD ALSO DROP, WHICH WILL
8533  035004  000042                  TXEN!TTLOOP             ;  CAUSE FIFO TO FLUSH ITSELF).
8534
8535  035006  012701  000050          MOV     #50,R1          ;DELAY FOR NNN SEC. TO ALLOW FIFO TIME
8536  035012  004737  005312  10$:    JSR     PC,WAIT50       ;TO FLUSH ITSELF.
8537  035016  077103                  SOB     R1,10$
8538
8539  035020  004537  003734          JSR     R5,WRITEI       ;TURN ON RECEIVER ENABLE (RXEN)
8540  035024  120000                  VIAORB
8541  035026  000142                  TXEN!RXEN!TTLOOP
8542
8543  035030  004537  006476          JSR     R5,CKROR        ;VERIFY CLEARING OF RECEIVER OVERRUN
8544  035034  000000                  0
8545  035036  103003                  BCC     .+8.            ;BR IF NO ERROR
8546  035040                          ERROR                   ;REPORT STACKED ERROR
8547  035040  104460                                                  TRAP    C$ERROR
```

```
 8548  035042                          ESCAPE  SUB             ;SKIP TO END OF TEST
 8549  035042  104410                                                  TRAP    C$ESCAPE
 8550  035044  000002                                                  .WORD   L10077-.
 8551  035046                          ENDSUB
 8552  035046                                                  L10077:
 8553  035046  104403                                                  TRAP    C$ESUB
 8554  035050                  ENDTST
 8555  035050                                                  L10074:
 8556  035050  104401                                                  TRAP    C$E;ST
```

```
                                .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
035052                                                                  T2$::

035052  004737  005420                JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP

035056  004537  007400                JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
035062  043626                        DDCMP!NOCHK!SYNCH       ;SET DDCMP,NO CHECK,SYNCH=226
035064  040000                        NORXEN                  ;USE 8 BIT CHARS; LEAVE RXEN=0
035066  103003                        BCC     .+8.            ;BR IF NO ERROR
035070                                ERROR                   ;REPORT STACKED ERROR
035070  104460                                                                TRAP    C$ERROR
035072                                ESCAPE  TST             ;SKIP TO END OF TEST
035072  104410                                                                TRAP    C$ESCAPE
035074  000414                                                                .WORD   L10100-.

035076  004537  010010                JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
035102  000001                        TSOM                    ; (IGNORED BY RECEIVER)
035104  000007                        7.
035106  004537  010010                JSR     R5,TXCTRL       ;CLEAR TSOM
035112  000000                        000
035114  000000                        0
035116  004537  003734                JSR     R5,WRITEI       ;ENABLE RECEIVER (RXEN => 1)
035122  120000                        VIAORB
035124  000142                        RXEN!TXEN!TTLOOP
                                ;---------------------------------------------------------
                                ; THE RECEIVER SHOULD IGNORE THE NEXT STRING OF CHARACTERS
                                ;---------------------------------------------------------
035126  004537  007676                JSR     R5,TXCHAR       ;LOAD 000, TX 2ND SYNCH
035132  000000                        000
035134  000010                        8.
035136  103003                        BCC     .+8.            ;BR IF NO ERROR
035140                                ERROR                   ;REPORT STACKED ERROR
035140  104460                                                                TRAP    C$ERROR
035142                                ESCAPE  TST             ;SKIP TO END OF TEST
035142  104410                                                                TRAP    C$ESCAPE
035144  000344                                                                .WORD   L10100-.

035146  004537  007676                JSR     R5,TXCHAR       ;LOAD 377, TX 000
```

```
 8613  035152  000377                    377
 8614  035154  000010                    8.
 8615  035156  103003                    BCC     .+8.            ;BR IF NO ERROR
 8616  035160                            ERROR                   ;REPORT STACKED ERROR
 8617  035160  104460                                                    TRAP    C$ERROR
 8618  035162                            ESCAPE  TST             ;SKIP TO END OF TEST
 8619  035162  104410                                                    TRAP    C$ESCAPE
 8620  035164  000324                                                    .WORD   L10100-.
 8621
 8622  035166  004537  007676            JSR     R5,TXCHAR       ;LOAD 125, TX 377
 8623  035172  000125                    125
 8624  035174  000010                    8.
 8625  035176  103003                    BCC     .+8.            ;BR IF NO ERROR
 8626  035200                            ERROR                   ;REPORT STACKED ERROR
 8627  035200  104460                                                    TRAP    C$ERROR
 8628  035202                            ESCAPE  TST             ;SKIP TO END OF TEST
 8629  035202  104410                                                    TRAP    C$ESCAPE
 8630  035204  000304                                                    .WORD   L10100-.
 8631
 8632  035206  004537  007676            JSR     R5,TXCHAR       ;TX 125, LOAD SYNCH
 8633  035212  000226                    SYNCH
 8634  035214  000010                    8.
 8635  035216  103003                    BCC     .+8.            ;BR IF NO ERROR
 8636  035220                            ERROR                   ;REPORT STACKED ERROR
 8637  035220  104460                                                    TRAP    C$ERROR
 8638  035222                            ESCAPE  TST             ;SKIP TO END OF TEST
 8639  035222  104410                                                    TRAP    C$ESCAPE
 8640  035224  000264                                                    .WORD   L10100-.
 8641
 8642  035226  004537  006176            JSR     R5,CKRDA        ;CHECK RECEIVE DATA AVAILABLE
 8643  035232  000000                    0                       ;   (NO DATA EXPECTED)
 8644  035234  103003                    BCC     .+8.            ;BR IF NO ERROR
 8645  035236                            ERROR                   ;REPORT STACKED ERROR
 8646  035236  104460                                                    TRAP    C$ERROR
 8647  035240                            ESCAPE  TST             ;SKIP TO END OF TEST
 8648  035240  104410                                                    TRAP    C$ESCAPE
 8649  035242  000246                                                    .WORD   L10100-.
 8650                            ;-------  ------------------------------------------------------------
 8651                            ; THE RECEIVER SHOULD SYNCHRONIZE ON THE NEXT TWO SYNC CHARACTERS
 8652                            ; AND THEN READ THE REMAINING ONES.
 8653                            ;-------------------------------------------------------------------
 8654  035244  004537  007676            JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
 8655  035250  000226                    SYNCH
 8656  035252  000010                    8.                      ;
 8657  035254  103003                    BCC     .+8.            ;BR IF NO ERROR
 8658  035256                            ERROR                   ;REPORT STACKED ERROR
 8659  035256  104460                                                    TRAP    C$ERROR
 8660  035260                            ESCAPE  TST             ;SKIP TO END OF TEST
 8661  035260  104410                                                    TRAP    C$ESCAPE
 8662  035262  000226                                                    .WORD   L10100-.
 8663
 8664  035264  004537  007676            JSR     R5,TXCHAR       ;LOAD 3ND SYNCH, TX 2ND SYNCH
 8665  035270  000226                    SYNCH
 8666  035272  000010                    8.                      ;
 8667  035274  103003                    BCC     .+8.            ;BR IF NO ERROR
 8668  035276                            ERROR                   ;REPORT STACKED ERROR
```

```
035276  104460                                                                          TRAP    C$ERROR
035300                                  ESCAPE  TST             ;SKIP TO END OF TEST
035300  104410                                                                          TRAP    C$ESCAPE
035302  000206                                                                          .WORD   L10100-.

035304  004537  007676                  JSR     R5,TXCHAR       ;LOAD 4TH SYNCH, TX 3RD SYNCH
035310  000226                          SYNCH
035312  000010                          8.                      ;
035314  103003                          BCC     .+8.            ;BR IF NO ERROR
035316                                  ERROR                   ;REPORT STACKED ERROR
035316  104460                                                                          TRAP    C$ERROR
035320                                  ESCAPE  TST             ;SKIP TO END OF TEST
035320  104410                                                                          TRAP    C$ESCAPE
035322  000166                                                                          .WORD   L10100-.

035324  004537  007676                  JSR     R5,TXCHAR       ;LOAD 5TH SYNCH
035330  000226                          SYNCH
035332  000000                          0                       ;
035334  103003                          BCC     .+8.            ;BR IF NO ERROR
035336                                  ERROR                   ;REPORT STACKED ERROR
035336  104460                                                                          TRAP    C$ERROR
035340                                  ESCAPE  TST             ;SKIP TO END OF TEST
035340  104410                                                                          TRAP    C$ESCAPE
035342  000146                                                                          .WORD   L10100-.

035344  004537  011364                  JSR     R5,RCV1ST       ;CLOCK AND RECEIVE 3RD SYNCH
035350  000000                          0
035352  103003                          BCC     .+8.            ;BR IF NO ERROR
035354                                  ERROR                   ;REPORT STACKED ERROR
035354  104460                                                                          TRAP    C$ERROR
035356                                  ESCAPE  TST             ;SKIP TO END OF TEST
035356  104410                                                                          TRAP    C$ESCAPE
035360  000130                                                                          .WORD   L10100-.

035362  004537  010110                  JSR     R5,RXCHAR       ;RECEIVE/CHECK SYNCH CHARACTER
035366  000226                          SYNCH                   ;
035370  000000                          0
035372  000010                          8.
035374  103003                          BCC     .+8.            ;BR IF NO ERROR
035376                                  ERROR                   ;REPORT STACKED ERROR
035376  104460                                                                          TRAP    C$ERROR
035400                                  ESCAPE  TST             ;SKIP TO END OF TEST
035400  104410                                                                          TRAP    C$ESCAPE
035402  000106                                                                          .WORD   L10100-.
                                ;-----------------------------------------------------------------
035404  012702  035512                  MOV     #TXTBL1,R2      ;SET UP TABLE POINTER
035410  112237  035446          5$:     MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
035414  116237  000001  035426          MOVB    1(R2),10$       ;SETUP TRANSMIT CHARACTER

035422  004537  007676                  JSR     R5,TXCHAR       ;LOAD A CHARACTER
035426  000000                  10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
035430  000000                          0
035432  103003                          BCC     .+8.            ;BR IF NO ERROR
035434                                  ERROR                   ;REPORT STACKED ERROR
035434  104460                                                                          TRAP    C$ERROR
035436                                  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
                ;------------------------------
```

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

                        .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 SYNCR NIZATION, IGNORE THE NEXT
                        ;*  THREE  SYNC CHARACTERS, AND PASS THE REST OF THE MESSAGE (4 SYNC AND
                        ;*  6 DATA CHARACTERS) THROUGH RXDB REGISTER.
                        ;*
                        ;--*********************************************************************
                        ;
035530                  ;       BGNTST                                          T30::
035530  004737  005420            JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
035534  004537  007400            JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
035540  063626                    DDCMP!STRIPS!NOCHK!SYNCH        ;SET DDCMP,NO CHECK,SYNCH=226
035542  040000                    NORXEN                  ;USE 8 BIT CHARS; LEAVE RXEN=0
035544  103003                    BCC     .+8.            ;BR IF NO ERROR
035546                            ERROR                   ;REPORT STACKED ERROR
035546  104460                                                            TRAP    C$ERROR
035550                            ESCAPE  TST             ;SKIP TO END OF TEST
035550  104410                                                            TRAP    C$ESCAPE
035552  000474                                                            .WORD   L10101-.
035554  004537  010010            JSR     R5,TXCTRL       ;OUTPUT 1ST SYNC CHARACTER
035560  000001                    TSOM                    ; (IGNORED BY RECEIVER)
035562  000007                    7.
035564  004537  010010            JSR     R5,TXCTRL       ;CLEAR TSOM
035570  000000                    000
035572  000000                    0
035574  004537  003734            JSR     R5,WRITEI       ;ENABLE RECEIVER (RXEN => 1)
035600  120000                    V2AORB
035602  000142                    RXEN!TXEN!TTLOOP
                        ;-----------------------------------------------------------
                        ; THE RECEIVER SHOULD IGNORE THE NEXT STRING OF CHARACTERS
                        ;-----------------------------------------------------------
035604  004537  007676            JSR     R5,TXCHAR       ;LOAD 000, TX 2ND SYNCH
035610  000000                    000
035612  000010                    8.
035614  103003                    BCC     .+8.            ;BR IF NO ERROR
035616                            ERROR                   ;REPORT STACKED ERROR
035616  104460                                                            TRAP    C$ERROR
035620                            ESCAPE  TST             ;SKIP TO END OF TEST
035620  104410                                                            TRAP    C$ESCAPE
035622  000424                                                            .WORD   L10101-.
```

```
8827  035624  004537  007676          JSR     R5,TXCHAR       ;LOAD 377, TX 000
8828  035630  000377                  377
8829  035632  000010                  8.
8830  035634  103003                  BCC     .+8.            ;BR IF NO ERROR
8831  035636                          ERROR                   ;REPORT STACKED ERROR
8832  035636  104460                                                  TRAP    C$ERROR
8833  035640                          ESCAPE  TST             ;SKIP TO END OF TEST
8834  035640  104410                                                  TRAP    C$ESCAPE
8835  035642  000404                                                  .WORD   L10101-.
8836
8837  035644  004537  007676          JSR     R5,TXCHAR       ;LOAD 125, TX 377
8838  035650  000125                  125
8839  035652  000010                  8.
8840  035654  103003                  BCC     .+8.            ;BR IF NO ERROR
8841  035656                          ERROR                   ;REPORT STACKED ERROR
8842  035656  104460                                                  TRAP    C$ERROR
8843  035660                          ESCAPE  TST             ;SKIP TO END OF TEST
8844  035660  104410                                                  TRAP    C$ESCAPE
8845  035662  000364                                                  .WORD   L10101-.
8846
8847  035664  004537  007676          JSR     R5,TXCHAR       ;LOAD SYNCH, TX 125
8848  035670  000226                  SYNCH
8849  035672  000010                  8.
8850  035674  103003                  BCC     .+8.            ;BR IF NO ERROR
8851  035676                          ERROR                   ;REPORT STACKED ERROR
8852  035676  104460                                                  TRAP    C$ERROR
8853  035700                          ESCAPE  TST             ;SKIP TO END OF TEST
8854  035700  104410                                                  TRAP    C$ESCAPE
8855  035702  000344                                                  .WORD   L10101-.
8856
8857  035704  004537  006176          JSR     R5,CKRDA        ;CHECK RECEIVE DATA AVAILABLE
8858  035710  000000                  0                       ;   (NO DATA EXPECTED)
8859  035712  103003                  BCC     .+8.            ;BR IF NO ERROR
8860  035714                          ERROR                   ;REPORT STACKED ERROR
8861  035714  104460                                                  TRAP    C$ERROR
8862  035716                          ESCAPE  TST             ;SKIP TO END OF TEST
8863  035716  104410                                                  TRAP    C$ESCAPE
8864  035720  000326                                                  .WORD   L10101-.
8865                          ;-----------------------------------------------------------
8866                          ; THE RECEIVER SHOULD SYNCHRONIZE ON THE NEXT TWO SYNC CHARACTERS.
8867                          ; STRIP THE NEXT THREE, AND THEN READ THE REMAINING ONES.
8868                          ;-----------------------------------------------------------
8869  035722  004537  007676          JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
8870  035726  000226                  SYNCH
8871  035730  000010                  8.                      ;
8872  035732  103003                  BCC     .+8.            ;BR IF NO ERROR
8873  035734                          ERROR                   ;REPORT STACKED ERROR
8874  035734  104460                                                  TRAP    C$ERROR
8875  035736                          ESCAPE  TST             ;SKIP TO END OF TEST
8876  035736  104410                                                  TRAP    C$ESCAPE
8877  035740  000306                                                  .WORD   L10101-.
8878
8879  035742  004537  007676          JSR     R5,TXCHAR       ;LOAD 3ND SYNCH, TX 2ND SYNCH
8880  035746  000226                  SYNCH
8881  035750  000010                  8.                      ;
8882  035752  103003                  BCC     .+8.            ;BR IF NO ERROR
```

```
8883  035754                                ERROR                   ;REPORT STACKED ERROR
8884  035754  104460                                                                TRAP    C$ERROR
8885  035756                                ESCAPE  TST             ;SKIP TO END OF TEST
8886  035756  104410                                                                TRAP    C$ESCAPE
8887  035760  000266                                                                .WORD   L10101-.
8888
8889  035762  004537  007676              JSR     R5,TXCHAR       ;LOAD 4TH SYNCH, TX 3RD SYNCH
8890  035766  000226                      SYNCH
8891  035770  000010                      8.                      ;
8892  035772  103003                      BCC     .+8.            ;BR IF NO ERROR
8893  035774                              ERROR                   ;REPORT STACKED ERROR
8894  035774  104460                                                                TRAP    C$ERROR
8895  035776                              ESCAPE  TST             ;SKIP TO END OF TEST
8896  035776  104410                                                                TRAP    C$ESCAPE
8897  036000  000246                                                                .WORD   L10101-.
8898
8899  036002  004537  007676              JSR     R5,TXCHAR       ;LOAD 5TH SYNCH, TX 4TH SYNCH
8900  036006  000226                      SYNCH
8901  036010  000010                      8.                      ;
8902  036012  103003                      BCC     .+8.            ;BR IF NO ERROR
8903  036014                              ERROR                   ;REPORT STACKED ERROR
8904  036014  104460                                                                TRAP    C$ERROR
8905  036016                              ESCAPE  TST             ;SKIP TO END OF TEST
8906  036016  104410                                                                TRAP    C$ESCAPE
8907  036020  000226                                                                .WORD   L10101-.
8908
8909  036022  004537  007676              JSR     R5,TXCHAR       ;LOAD DATA1 ,TX 5TH SYNCH
8910  036026  000252                      252
8911  036030  000010                      8.                      ;
8912  036032  103003                      BCC     .+8.            ;BR IF NO ERROR
8913  036034                              ERROR                   ;REPORT STACKED ERROR
8914  036034  104460                                                                TRAP    C$ERROR
8915  036036                              ESCAPE  TST             ;SKIP TO END OF TEST
8916  036036  104410                                                                TRAP    C$ESCAPE
8917  036040  000206                                                                .WORD   L10101-.
8918
8919  036042  004537  007676              JSR     R5,TXCHAR       ;LOAD DATA2 ,TX DATA1
8920  036046  000347                      347
8921  036050  000010                      8.                      ;
8922  036052  103003                      BCC     .+8.            ;BR IF NO ERROR
8923  036054                              ERROR                   ;REPORT STACKED ERROR
8924  036054  104460                                                                TRAP    C$ERROR
8925  036056                              ESCAPE  TST             ;SKIP TO END OF TEST
8926  036056  104410                                                                TRAP    C$ESCAPE
8927  036060  000166                                                                .WORD   L10101-.
8928
8929  036062  004537  007676              JSR     R5,TXCHAR       ;LOAD DATA3
8930  036066  000030                      030
8931  036070  000000                      0                       ;
8932  036072  103003                      BCC     .+8.            ;BR IF NO ERROR
8933  036074                              ERROR                   ;REPORT STACKED ERROR
8934  036074  104460                                                                TRAP    C$ERROR
8935  036076                              ESCAPE  TST             ;SKIP TO END OF TEST
8936  036076  104410                                                                TRAP    C$ESCAPE
8937  036100  000146                                                                .WORD   L10101-.
8938
```

```
8939  036102  004537  011364              JSR     R5,RCV1ST       ;CLOCK AND RECEIVE DATA1
8940  036106  000000                      0
8941  036110  103003                      BCC     .+8.            ;BR IF NO ERROR
8942  036112                              ERROR                   ;REPORT STACKED ERROR
8943  036112  104460                                                              TRAP    C$ERROR
8944  036114                              ESCAPE  TST             ;SKIP TO END OF TEST
8945  036114  104410                                                              TRAP    C$ESCAPE
8946  036116  000130                                                              .WORD   L10101-.
8947
8948  036120  004537  010110              JSR     R5,RXCHAR       ;RECEIVE/CHECK DATA1 (252)
8949  036124  000252                      252                     ;
8950  036126  000000                      0
8951  036130  000010                      8.
8952  036132  103003                      BCC     .+8.            ;BR IF NO ERROR
8953  036134                              ERROR                   ;REPORT STACKED ERROR
8954  036134  104460                                                              TRAP    C$ERROR
8955  036136                              ESCAPE  TST             ;SKIP TO END OF TEST
8956  036136  104410                                                              TRAP    C$ESCAPE
8957  036140  000106                                                              .WORD   L10101-.
8958                          ;-------------------------------------------------------------
8959  036142  012702  036250              MOV     #TXTBL2,R2      ;SET UP TABLE POINTER
8960  036146  112237  036204      5$:     MOVB    (R2)+,20$       ;SETUP EXPECTED CHARACTER
8961  036152  116237  000001  036164      MOVB    1(R2),10$       ;SETUP TRANSMIT CHARACTER
8962
8963  036160  004537  007676              JSR     R5,TXCHAR       ;LOAD A CHARACTER
8964  036164  000000              10$:    000                     ;** HOLE FOR NEXT TX CHARACTER
8965  036166  000000                      0
8966  036170  103003                      BCC     .+8.            ;BR IF NO ERROR
8967  036172                              ERROR                   ;REPORT STACKED ERROR
8968  036172  104460                                                              TRAP    C$ERROR
8969  036174                              ESCAPE  TST             ;SKIP TO END OF TEST
8970  036174  104410                                                              TRAP    C$ESCAPE
8971  036176  000050                                                              .WORD   L10101-.
8972
8973  036200  004537  010110              JSR     R5,RXCHAR       ;CLOCK/RECEIVE/CHECK PREVIOUS CHARACTER
8974  036204  000000              20$:    000                     ;** HOLE FOR EXPECTED CHARACTER
8975  036206  000000                      0
8976  036210  000010                      8.
8977  036212  103003                      BCC     .+8.            ;BR IF NO ERROR
8978  036214                              ERROR                   ;REPORT STACKED ERROR
8979  036214  104460                                                              TRAP    C$ERROR
8980  036216                              ESCAPE  TST             ;SKIP TO END OF TEST
8981  036216  104410                                                              TRAP    C$ESCAPE
8982  036220  000026                                                              .WORD   L10101-.
8983
8984  036222  022702  036262              CMP     #TXEND2,R2
8985  036226  001347                      BNE     5$
8986                          ;-------------------------------------------------------------
8987  036230  004537  011532              JSR     R5,ENDTRN       ;SHUT DOWN TRANSMITTER, RECEIVER
8988  036234  000010                      8.
8989  036236  103003                      BCC     .+8.            ;BR IF NO ERROR
8990  036240                              ERROR                   ;REPORT STACKED ERROR
8991  036240  104460                                                              TRAP    C$ERROR
8992  036242                              ESCAPE  TST             ;SKIP TO END OF TEST
8993  036242  104410                                                              TRAP    C$ESCAPE
8994  036244  000002                                                              .WORD   L10101-.
```

```
 8995  036246                          ENDTST
 8996  036246                                                          L10101:
 8997  036246  104401                                                          TRAP    C$ETST
 8998                                  ;------------------------------
 8999  036250     347                  TXTBL2: .BYTE   347
 9000  036251     030                          .BYTE   030
 9001  036252     226                          .BYTE   226     ;SYNCH
 9002  036253     226                          .BYTE   226     ;SYNCH
 9003  036254     252                          .BYTE   252
 9004  036255     101                          .BYTE   101
 9005  036256     202                          .BYTE   202
 9006  036257     226                          .BYTE   226     ;SYNCH
 9007  036260     226                          .BYTE   226     ;SYNCH
 9008  036261     000                          .BYTE   000
 9009  036262     000                  TXEND2: .BYTE   000
 9010          036264                          .EVEN
 9011                                  ;------------------------------
```

```
                                .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::
036264                                  JSR     PC,INIDMV       ;INIT DMV-11, ENTER M-LOOP
036264  004737  005420
                                
036270  004537  007400                  JSR     R5,INITRN       ;LOAD 1 SOM, CLK TX UNTIL ACTIVE
036274  065626                          DDCMP!STRIPS!IDLES!CRC16!SYNCH ;SET DDCMP, STRIP,IDLE,CRC-16, SYNCH=226
036276  000000                          0                       ;USE 8 BIT CHARS
036300  103003                          BCC     .+8.            ;BR IF NO ERROR
036302                                  ERROR                   ;REPORT STACKED ERROR
036302  104460                                                                  TRAP    C$ERROR
036304                                  ESCAPE  TST             ;SKIP TO END OF TEST
036304  104410                                                                  TRAP    C$ESCAPE
036306  000334                                                                  .WORD   L10102-.

036310  004537  007676                  JSR     R5,TXCHAR       ;LOAD 2ND SYNCH, TX 1ST SYNCH
036314  000226                          SYNCH
036316  000007                          7.
036320  103003                          BCC     .+8.            ;BR IF NO ERROR
036322                                  ERROR                   ;REPORT STACKED ERROR
036322  104460                                                                  TRAP    C$ERROR
036324                                  ESCAPE  TST             ;SKIP TO END OF TEST
036324  104410                                                                  TRAP    C$ESCAPE
036326  000314                                                                  .WORD   L10102-.

036330  004537  010010          JSR     R5,TXCTRL       ;CLEAR TSOM
036334  000000                  000
036336  000000                  0
036340  004537  007676          JSR     R5,TXCHAR       ;LOAD 000, TX 2ND SYNCH
036344  000000                  000
036346  000010                  8.
036350  103003                  BCC     .+8.            ;BR IF NO ERROR
036352                          ERROR                   ;REPORT STACKED ERROR
036352  104460                                                          TRAP    C$ERROR
036354                          ESCAPE  TST             ;SKIP TO END OF TEST
036354  104410                                                          TRAP    C$ESCAPE
036356  000264                                                          .WORD   L10102-.

036360  004537  007676          JSR     R5,TXCHAR       ;LOAD 125, TX 000
036364  000125                  125
036366  000010                  8.
036370  103003                  BCC     .+8.            ;BR IF NO ERROR
036372                          ERROR                   ;REPORT STACKED ERROR
036372  104460                                                          TRAP    C$ERROR
```

```
9068  036374                                ESCAPE  TST             ;SKIP TO END OF TEST
9069  036374  104410                                                                TRAP    C$ESCAPE
9070  036376  000244                                                                .WORD   L10102-.
9071
9072  036400  004537  007676               JSR     R5,TXCHAR       ;LOAD 252, TX 125
9073  036404  000252                       252
9074  036406  000010                       8.
9075  036410  103003                       BCC     .+8.            ;BR IF NO ERROR
9076  036412                               ERROR                   ;REPORT STACKED ERROR
9077  036412  104460                                                                TRAP    C$ERROR
9078  036414                               ESCAPE  TST             ;SKIP TO END OF TEST
9079  036414  104410                                                                TRAP    C$ESCAPE
9080  036416  000224                                                                .WORD   L10102-.
9081
9082  036420  004537  007676               JSR     R5,TXCHAR       ;LOAD 377, TX 252
9083  036424  000377                       377
9084  036426  000010                       8.
9085  036430  103003                       BCC     .+8.            ;BR IF NO ERROR
9086  036432                               ERROR                   ;REPORT STACKED ERROR
9087  036432  104460                                                                TRAP    C$ERROR
9088  036434                               ESCAPE  TST             ;SKIP TO END OF TEST
9089  036434  104410                                                                TRAP    C$ESCAPE
9090  036436  000204                                                                .WORD   L10102-.
9091
9092  036440  004537  007676               JSR     R5,TXCHAR       ;LOAD 000
9093  036444  000000                       000
9094  036446  000000                       0
9095  036450  103003                       BCC     .+8.            ;BR IF NO ERROR
9096  036452                               ERROR                   ;REPORT STACKED ERROR
9097  036452  104460                                                                TRAP    C$ERROR
9098  036454                               ESCAPE  TST             ;SKIP TO END OF TEST
9099  036454  104410                                                                TRAP    C$ESCAPE
9100  036456  000164                                                                .WORD   L10102-.
9101
9102  036460  004537  011364               JSR     R5,RCV1ST       ;CLOCK AND RCV 000
9103  036464  000000                       0
9104  036466  103003                       BCC     .+8.            ;BR IF NO ERROR
9105  036470                               ERROR                   ;REPORT STACKED ERROR
9106  036470  104460                                                                TRAP    C$ERROR
9107  036472                               ESCAPE  TST             ;SKIP TO END OF TEST
9108  036472  104410                                                                TRAP    C$ESCAPE
9109  036474  000146                                                                .WORD   L10102-.
9110
9111  036476  004537  010110               JSR     R5,RXCHAR       ;READ & CHK 000, RCV 125
9112  036502  000000                       000
9113  036504  000000                       0
9114  036506  000010                       8.
9115  036510  103003                       BCC     .+8.            ;BR IF NO ERROR
9116  036512                               ERROR                   ;REPORT STACKED ERROR
9117  036512  104460                                                                TRAP    C$ERROR
9118  036514                               ESCAPE  TST             ;SKIP TO END OF TEST
9119  036514  104410                                                                TRAP    C$ESCAPE
9120  036516  000124                                                                .WORD   L10102-.
9121                                ;---------- ------------------------------------------------------------
9122                                ; TX AND RX NOW SYNC'D. ONE CHARACTER HAS BEEN READ/VERIFIED....
9123                                ; NOW CLEAR RXEN (RECEIVER ENABLE).
```

```
                                       ;----------------------------------------------------------------------
036520  004537  003476                         JSR     R5,READ           ;READ VALUE OF VIAORB REGISTER
036524  120000                                 VIAORB                    ; AND PUT IT IN LOCATION PROVIDED
036526  036554                                 CGORB                     ;BELOW (CGORB).
036530  103003                                 BCC     .+8.              ;BR IF NO ERROR
036532                                         ERROR                     ;REPORT ERROR
036532  104460                                                                               TRAP    C$ERROR
036534                                         ESCAPE  TST
036534  104410                                                                               TRAP    C$ESCAPE
036536  000104                                                                               .WORD   L10102-.

036540  042737  000100  036554                 BIC     #RXEN,CGORB       ;CLEAR RXEN BIT OF ORB STATUS WORD

036546  004537  003734                         JSR     R5,WRITEI         ;WRITE NEW STATUS WORD (W/RXEN=0)
036552  120000                                 VIAORB                    ;BACK INTO VIAORB
036554  000000                         CGORB:  .WORD   0                 ;ACTUAL DATA WRITTEN TO VIAORB
036556  103003                                 BCC     .+8.              ;BR IF NO ERROR
036560                                         ERROR                     ;REPORT ERROR
036560  104460                                                                               TRAP    C$ERROR
036562                                         ESCAPE  TST
036562  104410                                                                               TRAP    C$ESCAPE
036564  000056                                                                               .WORD   L10102-.
                                       ;----------------------------------------------------------------------
                                       ; RXEN NOW CLEARED. CHECK USYRT STATUS, TX ANOTHER CHARACTER, AND
                                       ; CHECK USYRT STATUS AGAIN.
                                       ;----------------------------------------------------------------------
036566  004537  005432                         JSR     R5,CKUSTS         ;CHK USYRT STATUS FOR PROPER STATE
036572  000104                                 TXACT!TBMT                ;TXACT = 1, TBMT = 1
036574  103003                                 BCC     .+8.              ;BR IF NO ERROR
036576                                         ERROR
036576  104460                                                                               TRAP    C$ERROR
036600                                         ESCAPE  TST
036600  104410                                                                               TRAP    C$ESCAPE
036602  000040                                                                               .WORD   L10102-.

036604  004537  007676                         JSR     R5,TXCHAR         ;LOAD/TRANSMIT 303
036610  000303                                 303
036612  100010                                 NCTBMT*256.!8.
036614  103003                                 BCC     .+8.              ;BR IF NO ERROR
036616                                         ERROR                     ;REPORT STACKED ERROR
036616  104460                                                                               TRAP    C$ERROR
036620                                         ESCAPE  TST               ;SKIP TO END OF TEST
036620  104410                                                                               TRAP    C$ESCAPE
036622  000020                                                                               .WORD   L10102-.

036624  004537  005432                         JSR     R5,CKUSTS         ;CHK USYRT STATUS FOR PROPER STATE
036630  000114                                 TXACT!TSO!TBMT            ;TXACT = 1
036632  103003                                 BCC     .+8.              ;BR IF NO ERROR
036634                                         ERROR
036634  104460                                                                               TRAP    C$ERROR
036636                                         ESCAPE  TST
036636  104410                                                                               TRAP    C$ESCAPE
036640  000002                                                                               .WORD   L10102-.
036642                                 ENDTST
036642                                                                               L10102:
036642  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.
                                ;////////////////////////////////////////////////////////////////////////

036644                                  BGNHRD
036644  000041                                                          .WORD L10103-L$HARD/2
036646                                                          L$HARD::

036646                                  GPRMA   ADDRES,0,0,160020,177776,YES
036646  000031                                                          .WORD   T$CODE
036650  036750                                                          .WORD   ADDRES
036652  160020                                                          .WORD   T$LOLIM
036654  177776                                                          .WORD   T$HILIM
036656                                  GPRMA   VECTOR,2,0,0,674,YES
036656  001031                                                          .WORD   T$CODE
036660  036776                                                          .WORD   VECTOR
036662  000000                                                          .WORD   T$LOLIM
036664  000674                                                          .WORD   T$HILIM
036666                                  GPRMD   PRIRTY,4,0,7000,4,7,YES
036666  002032                                                          .WORD   T$CODE
036670  037027                                                          .WORD   PRIRTY
036672  007000                                                          .WORD   7000
036674  000004                                                          .WORD   T$LOLIM
036676  000007                                                          .WORD   T$HILIM
036700                                  GPRMD   SW1.M,6,0,377,0,377,YES
036700  003032                                                          .WORD   T$CODE
036702  037060                                                          .WORD   SW1.M
036704  000377                                                          .WORD   377
036706  000000                                                          .WORD   T$LOLIM
036710  000377                                                          .WORD   T$HILIM
036712                                  GPRMD   SW2.M,10,0,377,0,377,YES
036712  004032                                                          .WORD   T$CODE
036714  037121                                                          .WORD   SW2.M
036716  000377                                                          .WORD   377
036720  000000                                                          .WORD   T$LOLIM
036722  000377                                                          .WORD   T$HILIM
036724                                  GPRMD   BDTY.M,12,0,7,0,2,YES
036724  005032                                                          .WORD   T$CODE
036726  037163                                                          .WORD   BDTY.M
036730  000007                                                          .WORD   7
036732  000000                                                          .WORD   T$LOLIM
036734  000002                                                          .WORD   T$HILIM
036736                                  GPRMD   BR.M,16,0,17,0,1,YES
036736  007032                                                          .WORD   T$CODE
036740  037246                                                          .WORD   BR.M
036742  000017                                                          .WORD   17
036744  000000                                                          .WORD   T$LOLIM
```

```
 9236  036746  000001                                                          .WORD   T$HILIM
 9237
 9238  036750                                ENDHRD
 9239                                                                          .EVEN
 9240  036750                                                          L10103:
 9241
 9242                                .NLIST  BEX
       036750  042504  044526  042503  ADDRES: .ASCIZ  /DEVICE CSR ADDRESS : /
       036776  042504  044526  042503  VECTOR: .ASCIZ  /DEVICE VECTOR ADDRESS : /
       037027     104  053105  041511  PRIRTY: .ASCIZ  /DEVICE PRIORITY LEVEL : /
       037060  053523  052111  044103  SW1.M:  .ASCIZ  /SWITCH PACK # 1 (BOOT ADDRESS): /
       037121     123  044527  041524  SW2.M:  .ASCIZ  /SWITCH PACK # 2 (DDCMP ADDRESS): /
       037163     102  040517  042122  BDTY.M: .ASCIZ  /BOARD TYPE (0=M8064, 1=M8053-V.35, 2=M8053-EIA) : /
       037246  040502  042125  051040  BR.M:   .ASCIZ  /BAUD RATE (0=LOW (19.2K), 1=HIGH (56K)): /
                                .LIST   BEX
 9243                            .EVEN
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION


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

037320                                  BGNSFT
037320  000000                                                  .WORD L10104-L$SOFT/2
037322                                                  L$SOFT::

037322                                  ENDSFT
                                                                .EVEN
037322                                                  L10104:
```

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

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

037322                          PATCH:
        037422                          .=.+100
037422  000240                          NOP
037424  000240                          NOP
037426  000240                          NOP
                                ;**********************************

                                .SBTTL  "ENDMOD" STATEMENT

037430                          ENDMOD

                                .SBTTL  "LASTAD" STATEMENT & END OF PROGRAM
037430                          LASTAD
                                                                        .EVEN
037430  000000                                                          .WORD   0
037432  000000                                                          .WORD   0
037434                          L$LAST::

        000001                  .END
```

```
ABC    = 000160          1632#
ADDRES  036750           9199    9242#
ADR    = 000020 G        1564#
AD.HIT  021540           4601    4716#
AD.OK   021534           4705    4710#
APA    = 000200          1688#
APAD   = 100000          1697#
ASSEMB= 000010           1300
BADDAT  002442           2082#
BDATA   002402           2065#   2690*   2693    2696    2971*   3074*   3075*   3506*   3765*   3766*   3783*   3784*   4153
                         4182    4231    4276    4340    4365    4825*   5040*   5155*   5384    5386*   5387    5593*   5594*
                         5619*   5620*   5645*   5646*
BDRATE  002550           2129#   4649*   4910
BDTY.M  037163           9227    9242#
BIT0   = 000001 G        1537#   1582    1594    1636    1645    1664    1693    1718    1734    1752    1768    1910    1924
                         1946    1999    3105    3150    3195    3240    3285    3328    3375    3442    4615    6219    6521
                         7344    7380    7626    7662
BIT00  = 000001 G        1526#   1537
BIT01  = 000002 G        1525#   1536
BIT02  = 000004 G        1524#   1535
BIT03  = 000010 G        1523#   1534
BIT04  = 000020 G        1522#   1533
BIT05  = 000040 G        1521#   1532
BIT06  = 000100 G        1520#   1531
BIT07  = 000200 G        1519#   1530
BIT08  = 000400 G        1518#   1529
BIT09  = 001000 G        1517#   1528
BIT1   = 000002 G        1536#   1635    1663    1693    1718    1733    1751    1767    1909    1923    1945    1998    3399
                         6219    6581    7203    7239    7344    7380    7767    7803
BIT10  = 002000 G        1516#   1641    1669    1703    1704    1705
BIT11  = 004000 G        1515#   1640    1668    1701
BIT12  = 010000 G        1514#   1700
BIT13  = 020000 G        1513#   1699    2008
BIT14  = 040000 G        1512#   1698    2001    2007
BIT15  = 100000 G        1511#   1639    1667    1697    1995    2006
BIT2   = 000004 G        1535#   1634    1662    1693    1718    1732    1750    1766    1898    1923    1944    6219    6521
                         6581    7485    7521    7626    7662    7767    7803
BIT3   = 000010 G        1534#   1633    1661    1692    1717    1731    1749    1765    1898    1923    1943
BIT4   = 000020 G        1533#   1581    1632    1691    1716    1730    1748    1764    1898    1922    1942
BIT5   = 000040 G        1532#   1632    1690    1715    1729    1747    1763    1893    1921    1941    3907    6521    7380
                         7662
BIT6   = 000100 G        1531#   1593    1632    1689    1715    1728    1746    1762    1883    1921    1940    3927    6219
                         6581    7239    7380    7803
BIT7   = 000200 G        1530#   1592    1600    1631    1660    1688    1715    1727    1745    1761    1883    1921    1934
                         1956    2004    3744    3745    3916    5936    5975    6052    6164    6203    6521    6581    7521
                         7662    7803
BIT8   = 000400 G        1529#   1643    1671    1702    1703    1704
BIT9   = 001000 G        1528#   1642    1670    1702    1703    2002
BOE    = 000400 G        1568#
BRDTYP  002544           2127#   4647*   4894
BR.M    037246           9233    9242#
BSEL0   002472           2096#   2632    4688    4694
BSEL1   002474           2099#   2428*   2633
BSEL10  002512           2109#   2640
BSEL11  002514           2111#   2641
BSEL12  002516           2112#   2642
```

```
BSEL13  002520          2114#  2643
BSEL14  002522          2115#  2644
BSEL15  002524          2117#  2645
BSEL16  002526          2118#  2646
BSEL17  002530          2120#  2647  4637
BSEL2   002476          2100#  2435  2471*  2478  2518*  2525  2590*  2597  2634  4803
BSEL3   002500          2102#  2635
BSEL4   002502          2103#  2636
BSEL5   002504          2105#  2637
BSEL6   002506          2106#  2493  2638
BSEL7   002510          2108#  2639
BSR0    002256          2032#  2632*
BSR1    002260          2034#  2633*
BSR10   002276          2047#  2640*
BSR11   002300          2048#  2641*
BSR12   002302          2049#  2642*
BSR13   002304          2050#  2643*
BSR14   002306          2051#  2644*
BSR15   002310          2052#  2645*
BSR16   002312          2053#  2646*
BSR17   002314          2054#  2647*
BSR2    002262          2036#  2634*
BSR3    002264          2038#  2635*
BSR4    002266          2040#  2636*
BSR5    002270          2042#  2637*
BSR6    002272          2044#  2638*
BSR7    002274          2046#  2639*
CARIER= 000100          1762#
CA1CTL= 000001          1924#
CA2CTL= 000016          1923#
CB1CTL= 000020          1922#
CB2CTL= 000340          1921#
CGORB   036554          9127   9135*  9139#
CHKTSO  007116          3437#  3495   5885  5989  6113  6221  6743  6840  6937
CHPTYP  002454          2087#  4564*
CXRACT  005676          3145#  3909   3962  3989  4016  4027
CXRDA   006176          3235#  3918   3929  3965  3981  3986  8642  8857
CXROR   006476          3323#  8265   8433  8445  8454  8543
CXRSA   006336          3280#
CXSEOM  006636          3370#  3968
CXTACT  005536          3100#  3606   3642  3694  4013  4024
CXTBMT  006036          3190#  3589   3603  3651  3658  3699  3706  8101  8129  8152
CXUSTS  005432          3065#  3572   4787  5924  5963  6040  6152  6191  9150  9169
CRC16 = 001400          1702#  9030
CTS   = 003010          1765#
C$AU  = 000052          1300#  4760
C$AUTO= 000061          1300#  4714
C$BRK = 000022          1300#
C$BSEG= 000004          1300#
C$BSUB= 000002          1300#  4863  4893  5207  5259  5313  5373  5439  6518  6578  6638  7200  7341
                        7482   7623  7764  7905  8410  8468  8506
C$CEFG= 000045          1300#
C$CLCK= 000062          1300#
C$CLEA= 000012          1300#  4732
C$CLOS= 000035          1300#
C$CLP1= 000006          1300#  4884
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
C$CVEC= 000036   13000#  4703
C$DCLN= 000044   13000#
C$DODU= 000051   13000#  4708
C$DRPT= 000024   13000#
C$DU  = 000053   13000#  4747
C$EDIT= 000003   13000#  1372
C$ERDF= 000055   13000#  4810  4829  4879  4915  5044  5159  5393  5598  5624  5650  5940  5979
                 6056    6168  6207
C$ERNR= 000056   13000#
C$ERFJ= 000060   13000#  4782  4791  4901  5096  5196  5215  5248  5267  5302  5321  5362  5428
                 5447    5480  5522  5695  5716  5727  5735  5782  5792  5802  5812  5821  5832
                 5840    5880  5889  5899  5917  5928  5955  5967  5993  6003  6019  6029  6044
                 6071    6108  6117  6127  6145  6156  6183  6195  6225  6235  6248  6256  6290
                 6308    6318  6334  6345  6356  6392  6414  6437  6454  6465  6474  6524  6544
                 6562    6584  6604  6622  6644  6664  6682  6738  6747  6757  6775  6789  6799
                 6835    6844  6854  6872  6886  6896  6932  6941  6951  6969  6983  6993  7026
                 7044    7054  7064  7074  7083  7094  7104  7115  7125  7136  7147  7158  7206
                 7222    7232  7245  7258  7278  7289  7304  7318  7329  7347  7363  7373  7386
                 7399    7419  7430  7445  7459  7470  7488  7504  7514  7527  7540  7560  7571
                 7586    7600  7611  7629  7645  7655  7668  7681  7701  7712  7727  7741  7752
                 7770    7786  7796  7809  7822  7842  7853  7868  7882  7893  7911  7927  7937
                 7950    7963  7983  7994  8009  8023  8034  8105  8124  8133  8147  8156  8175
                 8189    8199  8235  8257  8269  8283  8294  8308  8327  8338  8349  8358  8418
                 8437    8449  8458  8476  8496  8514  8547  8585  8607  8617  8627  8637  8646
                 8659    8669  8679  8689  8698  8709  8723  8734  8746  8800  8822  8832  8842
                 8852    8861  8874  8884  8894  8904  8914  8924  8934  8943  8954  8968  8979
                 8991    9034  9044  9057  9067  9077  9087  9097  9106  9117  9130  9142  9154
                 9164    9173
C$ERSF= 000054   13000#
C$ERSO= 000057   13000#
C$ESCA= 000010   13000#  4784  4793  4815  4903  5049  5098  5164  5198  5217  5250  5269  5304
                 5323    5364  5378  5430  5449  5482  5524  5603  5629  5697  5718  5729  5737
                 5784    5794  5804  5814  5823  5834  5842  5882  5891  5901  5919  5930  5945
                 5957    5969  5984  5995  6005  6021  6031  6046  6061  6073  6110  6119  6129
                 6147    6158  6173  6185  6197  6212  6227  6237  6250  6258  6292  6310  6320
                 6336    6347  6358  6394  6416  6439  6456  6467  6476  6526  6546  6564  6586
                 6606    6624  6646  6666  6684  6740  6749  6759  6777  6791  6801  6837  6846
                 6856    6874  6888  6898  6934  6943  6953  6971  6985  6995  7028  7046  7056
                 7066    7076  7085  7096  7106  7117  7127  7138  7149  7160  7208  7224  7234
                 7247    7260  7280  7291  7306  7320  7331  7349  7365  7375  7388  7401  7421
                 7432    7447  7461  7472  7490  7506  7516  7529  7542  7562  7573  7588  7602
                 7613    7631  7647  7657  7670  7683  7703  7714  7729  7743  7754  7772  7788
                 7798    7811  7824  7844  7855  7870  7884  7895  7913  7929  7939  7952  7965
                 7985    7996  8011  8025  8036  8107  8126  8135  8149  8158  8177  8191  8201
                 8237    8259  8271  8285  8296  8310  8329  8340  8351  8360  8420  8439  8451
                 8460    8478  8498  8516  8549  8587  8609  8619  8629  8639  8648  8661  8671
                 8681    8691  8700  8711  8725  8736  8748  8802  8824  8834  8844  8854  8863
                 8876    8886  8896  8906  8916  8926  8936  8945  8956  8970  8981  8993  9036
                 9046    9059  9069  9079  9089  9099  9108  9119  9132  9144  9156  9166  9175
C$ESEG= 000005   13000#
C$ESUB= 000003   13000#  4889  4921  5222  5274  5328  5403  5454  6572  6632  6692  7335  7476
                 7617    7758  7899  8040  8464  8502  8553
C$ETST= 000001   13000#  4836  4925  5056  5104  5176  5228  5280  5334  5409  5460  5502  5544
                 5657    5742  5846  6077  6262  6362  6480  6695  6805  6902  6999  7164  8043
                 8205    8364  8556  8753  8997  9179
C$EXIT= 000032   13000#
```

```
C$GETB= 000026           13004#
C$GETW= 000027           13004#
C$GMAN= 000043           13004#
C$GPHR= 000042           13004#  4623
C$GPLO= 000030           13004#
C$GPRI= 000040           13004#
C$INIT= 000011           13004#  4659
C$INLP= 000020           13004#
C$MANI= 000050           13004#
C$MEM = 000031           13004#
C$MSG = 000023           13004#  4136  4163  4197  4241  4286  4321  4350  4973  4988
C$OPEN= 000034           13004#
C$PNTB= 000014           13004#  4148  4158  4177  4187  4193  4210  4216  4226  4236  4255  4261  4271
                         4281    4300  4306  4316  4335  4345  4414  4933  4969  4983
C$PNTF= 000017           13004#
C$PNTS= 000016           13004#
C$PNTX= 000015           13004#  4113  4129  4381  4391  4398  4408  4430  4440  4447  4457  4464  4474
                         4481    4491  4507  4517  4524  4534  4939  4945  4953  4963
C$QIO = 000377           13004#
C$RDBU= 000007           13004#
C$REFG= 000047           13004#  4581  4588  4595  4602
C$RESE= 000033           13004#  4652  4744
C$REVI= 000003           13004#  1371
C$RFLA= 000021           13004#
C$RPT = 000025           13004#
C$SEFG= 000046           13004#
C$SPRI= 000041           13004#
C$SVEC= 000037           13004#  4684
C$TPRI= 000013           13004#
DDCMP = 040000           16984#  5691  5778  5876  6104  6286  6388  6520  6580  6640  8231  8414  8472
                         8510    8581  8796  9030
DEVMAP  002460           20894#  4611*  4629*
DEVPTR  002462           20904#  4615*  4627*  4629  4630*
DFPTBL  002224 G         14704#
DIAGMC= 000000           1300
DOTBMT= 000007           16104#  4796  4807
DTR   = 000020           17484#  2953  2956  3029  3545  3548  8085  8088  8099
DTRL  = 000000           17534#
D.BUG = 000000           20104#
EF.CON= 000036 G         15444#  4601
EF.NEW= 000035 G         15454#  4594
EF.PWR= 000034 G         15464#
EF.RES= 000037 G         15434#  4587
EF.STA= 000040 G         15424#  4580
EIAV35= 000002           19984#
EM1     013103           40684#  5046
EM100   014425           3451    40684#
EM101   014445           3463    40684#
EM102   014471           3512    40684#
EM14    013332           3831    40684#
EM16    013346           3821    40684#
EM2     013142           2977    40684#
EM25    013366           2704    40684#  5395
EM26    013414           40684#
EM27    013446           40684#
EM28    013477           3395    3890  40684#
```

CVDMCCO DMV11 LINE UNIT DIAG1 MACY11 30A(1052) 23-JUL-84 14:02 PAGE 213
CVDMCC.P11 12-JUL-84 16:09 CROSS REFERENCE TABLE -- USER SYMBOLS

```
EM29    013520            3384    3900    40680
EM3     013211            2443    40680
EM30    013535            3419    3867    40680
EM31    013556            3408    3877    40680
EM34    013573            3772    40680
EM35    013621            3798    40680
EM36    013642            3808    40680
EM39    013657            3844    40680
EM4     013235            2487    2534    2605    40680   4812
EM40    013701            3854    40680
EM5     013252            40680   5600    5652
EM54    013717            3759    40680
EM6     013302            40680   5626
EM66    013741            40680   5161
EM68    013760            3081    40680
EM69    014007            3114    40680
EM70    014025            3125    40680
EM71    014047            3159    40680
EM72    014065            3170    40680
EM73    014107            3204    40680
EM74    014124            3215    40680
EM75    014145            3249    40680
EM76    014161            3260    40680
EM77    014201            3294    40680
EM78    014215            3305    40680
EM90    014235            3337    40680
EM91    014266            3348    40680
EM92    014323            40680   4831
EM98    014357            40680   5942    6058    6170
EM99    014404            40680   5981    6209
ENDEMB  011640            40680   4190    4411
ENDIT   021420            4604    46560
ENDPAT  003106            23600
ENDTRN  011532            40110   6470    8354    8742    8987
ERRBLK  002254 G          20260   2444*   2488*   2535*   2606*   2705*   2978*   3082*   3115*   3126*   3160*   3171*   3205*
                          3216*   3250*   3261*   3295*   3306*   3338*   3349*   3385*   3396*   3409*   3420*   3452*   3464*
                          3513*   3760*   3773*   3799*   3809*   3822*   3832*   3845*   3855*   3868*   3878*   3891*   3901*
ERRFLG  002424            20750
ERRMSG  002252 G          20250   2443*   2487*   2534*   2605*   2704*   2977*   3081*   3114*   3125*   3159*   3170*   3204*
                          3215*   3249*   3260*   3294*   3305*   3337*   3348*   3384*   3395*   3408*   3419*   3451*   3463*
                          3512*   3759*   3772*   3798*   3808*   3821*   3831*   3844*   3854*   3867*   3877*   3890*   3900*
ERRNBR  002250 G          20240   2442*   2486*   2533*   2604*   2703*   2976*   3080*   3113*   3124*   3158*   3169*   3203*
                          3214*   3248*   3259*   3293*   3304*   3336*   3347*   3383*   3394*   3407*   3418*   3450*   3462*
                          3511*   3758*   3771*   3797*   3807*   3820*   3830*   3843*   3853*   3866*   3876*   3889*   3899*
ERROR1  002452            20860   4565*
ERRTYP  002246 G          20230   2441*   2485*   2532*   2603*   2702*   2975*   3079*   3112*   3123*   3157*   3168*   3202*
                          3213*   3247*   3258*   3292*   3303*   3335*   3346*   3382*   3393*   3406*   3417*   3449*   3461*
                          3510*   3757*   3770*   3796*   3806*   3819*   3829*   3841*   3852*   3865*   3875*   3888*   3898*
ERR10   017372 G          2978    3082    3773    42030   5047    5162
ERR11   017546 G          42480   5601    5627    5653
ERR11$  020356            4283    44230   4985
ERR12   017722 G          3115    3126    3160    3171    3205    3216    3250    3261    3295    3306    3338    3349    3385
                          3396    3409    3420    3452    3464    3760    3799    3809    3822    3832    3845    3855    3868
                          3878    3891    3901    42930   5943    5982    6059    6171    6210
ERR12$  020710            4238    4318    4347    45000
ERR13   020036 G          3513    43280
```

```
ERR4     017030 G          2444    2488    2535    2606    4102#   4813
ERR5     017154 G          4141#   4832
ERR5$    020160            4133    4160    4374#
ERR7A    017252 G          2705    4168#   5396
EVL    = 000004 G          1562#
EVRC   = 002400            1704#
EXADD  = 000020            1716#
EXCON  = 000010            1717#
EXECUT = 000005            1609#
E$END  = 002100            1300#
E$LOAD = 000035            1300#   1395
FLGCA1 = 000002            1945#
FLGCA2 = 000001            1946#
FLGCB1 = 000020            1942#
FLGCB2 = 000010            1943#
FLGIRQ = 000200            1934#
FLGSR  = 000004            1944#
FLGT1  = 000100            1940#
FLGT2  = 000040            1941#
FMT10    012252            4068#   4155
FMT10A   012326            4068#
FMT11    012347            4068#   4388    4405
FMT15    012366            4068#   4174
FMT15A   012420            4068#
FMT19    012472            4068#
FMT2     011650            4068#   4145
FMT21    012523            4068#   4207    4252    4297    4332    4980
FMT22    012533            4068#   4213    4258    4303
FMT23    012555            4068#   4233    4278    4342
FMT24    012634            4068#   4427    4504
FMT25    012647            4068#   4437    4471    4514
FMT26    012677            4068#   4454    4488    4531
FMT27    012732            4068#   4223    4268    4313
FMT29    012741            4068#   4444    4461    4478    4521
FMT3     011705            4068#   4184
FMT30    012746            4068#
FMT31    013003            4068#
FMT32    013051            4068#
FMT4     011771            4068#   4378
FMT4A    012027            4068#
FMT4B    012062            4068#   4395
FMT4C    012067            4068#
FMT5     012122            4068#   4110
FMT5A    012165            4068#   4126
FRSTIM   002444            2083#   4567    4576*
F$AU   = 000015            1300#   4757    4759
F$AUTO = 000020            1300#   4677    4713
F$BGN  = 000040            1300#   1303    4102    4141    4168    4203    4248    4293    4328    4547    4560    4677    4727
                           4741    4757    4777    4784    4793    4815    4835    4858    4862    4888    4892    4903    4920
                           4924    4928    4976    5015    5049    5055    5080    5098    5103    5126    5164    5175    5192
                           5198    5206    5217    5221    5227    5244    5250    5258    5269    5273    5279    5298    5304
                           5312    5323    5327    5333    5358    5364    5372    5398    5402    5408    5424    5430    5438
                           5449    5453    5459    5476    5482    5501    5518    5524    5543    5571    5603    5629    5656
                           5686    5697    5718    5729    5737    5741    5773    5784    5794    5804    5814    5823    5834
                           5842    5845    5872    5882    5891    5901    5919    5930    5945    5957    5969    5984    5995
                           6005    6021    6031    6046    6061    6073    6076    6100    6110    6119    6129    6147    6158
```

CROSS REFERENCE TABLE -- USER SYMBOLS

6173 6185 6197 6212 6227 6237 6250 6258 6261 6281 6292 6310 6320
6336 6347 6358 6361 6384 6394 6416 6439 6456 6467 6476 6479 6511
6517 6526 6546 6564 6571 6577 6586 6606 6624 6631 6637 6646 6666
6684 6691 6694 6730 6740 6749 6759 6777 6791 6801 6804 6827 6837
6846 6856 6874 6888 6898 6901 6924 6934 6943 6953 6971 6985 6995
6998 7018 7028 7046 7056 7066 7076 7085 7096 7106 7117 7127 7138
7149 7160 7163 7193 7199 7208 7224 7234 7247 7260 7280 7291 7306
7320 7331 7334 7340 7349 7365 7375 7388 7401 7421 7432 7447 7461
7472 7475 7481 7490 7506 7516 7529 7542 7562 7573 7588 7602 7613
7616 7622 7631 7647 7657 7670 7683 7703 7714 7729 7743 7754 7757
7763 7772 7788 7798 7811 7824 7844 7855 7870 7884 7895 7898 7904
7913 7929 7939 7952 7965 7985 7996 8011 8025 8036 8039 8042 8068
8107 8126 8135 8149 8158 8177 8191 8201 8204 8227 8237 8259 8271
8285 8296 8310 8329 8340 8351 8360 8363 8406 8409 8420 8439 8451
8460 8463 8467 8478 8498 8501 8505 8516 8549 8552 8555 8576 8587
8609 8619 8629 8639 8648 8661 8671 8681 8691 8700 8711 8725 8736
8748 8752 8792 8802 8824 8834 8844 8854 8863 8876 8886 8896 8906
8916 8926 8936 8945 8956 8970 8981 8993 9006 9026 9036 9046 9059
9069 9079 9089 9099 9108 9119 9132 9144 9156 9166 9175 9178 9194
9257 9275

F$CLEA= 000007 13000 4727 4731
F$DU = 000016 13000 4741 4746
F$END = 000041 13000 1303 4137 4164 4198 4242 4287 4322 4351 4660 4715 4733 4748
4761 4777 4784 4793 4815 4835 4837 4858 4862 4888 4890 4892 4903
4920 4922 4924 4926 4974 4989 5015 5049 5055 5057 5080 5098 5103
5105 5126 5164 5175 5177 5192 5198 5206 5217 5221 5223 5227 5229
5244 5250 5258 5269 5273 5275 5279 5281 5298 5304 5312 5323 5327
5329 5333 5335 5358 5364 5372 5398 5402 5404 5408 5410 5424 5430
5438 5449 5453 5455 5459 5461 5476 5482 5501 5503 5518 5524 5543
5545 5571 5603 5629 5656 5658 5686 5697 5718 5729 5737 5741 5743
5773 5784 5794 5804 5814 5823 5834 5842 5845 5847 5872 5882 5891
5901 5919 5930 5945 5957 5969 5984 5995 6005 6021 6031 6046 6061
6073 6076 6078 6100 6110 6119 6129 6147 6158 6173 6185 6197 6212
6227 6237 6250 6258 6261 6263 6281 6292 6310 6320 6336 6347 6358
6361 6363 6384 6394 6416 6439 6456 6467 6476 6479 6481 6511 6517
6526 6546 6564 6571 6573 6577 6586 6606 6624 6631 6633 6637 6646
6666 6684 6691 6693 6694 6696 6730 6740 6749 6759 6777 6791 6801
6804 6806 6827 6837 6846 6856 6874 6888 6898 6901 6903 6924 6934
6943 6953 6971 6985 6995 6998 7000 7018 7028 7046 7056 7066 7076
7085 7096 7106 7117 7127 7138 7149 7160 7163 7165 7193 7199 7208
7224 7234 7247 7260 7280 7291 7306 7320 7331 7334 7336 7340 7349
7365 7375 7388 7401 7421 7432 7447 7461 7472 7475 7477 7481 7490
7506 7516 7529 7542 7562 7573 7588 7602 7613 7616 7618 7622 7631
7647 7657 7670 7683 7703 7714 7729 7743 7754 7757 7759 7763 7772
7788 7798 7811 7824 7844 7855 7870 7884 7895 7898 7900 7904 7913
7929 7939 7952 7965 7985 7996 8011 8025 8036 8039 8041 8042 8044
8068 8107 8126 8135 8149 8158 8177 8191 8201 8204 8206 8227 8237
8259 8271 8285 8296 8310 8329 8340 8351 8360 8363 8365 8406 8409
8420 8439 8451 8460 8463 8465 8467 8478 8498 8501 8503 8505 8516
8549 8552 8554 8555 8557 8576 8587 8609 8619 8629 8639 8648 8661
8671 8681 8691 8700 8711 8725 8736 8748 8752 8754 8792 8802 8824
8834 8844 8854 8863 8876 8886 8896 8906 8916 8926 8936 8945 8956
8970 8981 8993 8996 8998 9026 9036 9046 9059 9069 9079 9089 9099
9108 9119 9132 9144 9156 9166 9175 9178 9180 9241 9263 9275

F$HARD= 000004 13000 9194 9239
F$HW = 000013 13000 1468 1484

```
F$INIT= 000006            1300#  4560   4658
F$JMP  = 000050            1300#
F$MOD  = 000000            1300#  1303   9275
F$MSG  = 000011            1300#  4102   4135   4141   4162   4168   4196   4203   4240   4248   4285   4293   4320
                           4328   4349   4928   4972   4976   4987
F$PROT= 000021             1300#  4547   4552
F$PWR  = 000017            1300#
F$RPT  = 000012            1300#
F$SEG  = 000003            1300#
F$SOFT= 000005             1300#  9257   9261
F$SRV  = 000010            1300#
F$SUB  = 000002            1300#  4863   4888   4893   4920   5207   5221   5259   5273   5313   5327   5373   5402
                           5439   5453   6518   6571   6578   6631   6638   6691   7200   7334   7341   7475   7482
                           7616   7623   7757   7764   7898   7905   8039   8410   8463   8468   8501   8506   8552
F$SW   = 000014            1300#  1493   1498
F$TEST= 000001             1300#  4778   4835   4859   4924   5016   5055   5081   5103   5127   5175   5193   5227
                           5245   5279   5299   5333   5359   5408   5425   5459   5477   5501   5519   5543   5572
                           5656   5687   5741   5774   5845   5873   6076   6101   6261   6282   6361   6385   6479
                           6512   6694   6731   6804   6828   6901   6925   6998   7019   7163   7194   8042   8069
                           8204   8228   8363   8407   8555   8577   8752   8793   8996   9027   9178
GDATA   002400             2064#  2438*  2482*  2529*  2600*  2696   2969*  2970*  3073*  3505*  3763*  3764*  3781*
                           3782*  4103*  4105   4125   4154   4183   4232   4277   4341   4364   4807*  4824*  5038*
                           5039*  5153*  5154*  5210*  5262*  5316*  5376*  5387   5442*  5592*  5618*  5644*
GETBSR  004046             2632#
GETPRM  021264             4605   4619#  4628
GETURS  004402             2729#  4500
GETVRS  004502             2760#  4423
GETWSR  004210             2437   2481   2528   2599   2653#  4806   4818
GOODAT  002440             2081#
G$CNTO= 000200             1300#
G$DELM= 000372             1300#
G$DISP= 000003             1300#
G$EXCP= 000400             1300#
G$HILI= 000002             1300#
G$LOLI= 000001             1300#
G$NO   = 000000            1300#
G$OFFS= 000400             1300#  9198   9203   9208   9214   9220   9226   9232
G$OFSI= 000376             1300#  9198   9203   9208   9214   9220   9226   9232
G$PRMA= 000001             1300#  9198   9203
G$PRMD= 000002             1300#  9208   9214   9220   9226   9232
G$PRML= 000000             1300#
G$RADA= 000140             1300#
G$RADB= 000000             1300#
G$RADD= 000040             1300#
G$RADL= 000120             1300#
G$RADO= 000020             1300#  9198   9203   9208   9214   9220   9226   9232
G$XFER= 000004             1300#
G$YES  = 000010            1300#  9198   9203   9208   9214   9220   9226   9232
HDX    = 000004            1750#
HELP   = 000000            1291#  1325   1413   1459   2407
HOE    = 100000 G          1575#
IBE    = 010000 G          1572#
IDLE   = 000010            1692#
IDLES  = 004000            1701#  5778   5876   6104   6831   7202   9030
IDU    = 000040 G          1565#
IEI    = 000001            1582#
```

```
IE0    = 000020            1581#
IER    = 020000 G          1573#
INIDMV  005420             3051#  4859  5016  5027  5081  5127  5193  5245  5299  5359  5425  5477  5519
                           5572   5688  5775  5873  6101  6283  6385  6512  6731  6828  6925  7019  7194
                           8069   8228  8411  8469  8507  8578  8793  9027
INITRN  007400             3541#  5690  5777  5875  6103  6285  6387  6519  6579  6639  6733  6830  6927
                           7021   7201  7342  7483  7624  7765  7906  8230  8413  8471  8509  8580  8795
                           9029
INITT1  004556             2798#
INITT2  004756             2885#
INTFLG  002422             2073#
INTGRL= 000001             1999#
INTSC  = 000200            1956#
ISR    = 000100 G          1566#
IXE    = 004000 G          1571#
I$AU   = 000041            1300#  4757#  4761#
I$AUTO= 000041             1300#  4677#  4715#
I$CLN  = 000041            1300#  4727#  4733#
I$DU   = 000041            1300#  4741#  4748#
I$HRD  = 000041            9194#  9241#
I$INIT= 000041             1300#  4560#  4660#
I$MOD  = 000041            1300#  1302#  9275#
I$MSG  = 000041            1300#  4102#  4137#  4141#  4164#  4168#  4198#  4203#  4242#  4248#  4287#  4293#  4322#
                           4328#  4351#  4928#  4974#  4976#  4989#
I$PROT= 000040             1300#  4547#
I$PTAB= 000041             1300#
I$PWR  = 000041            1300#
I$RPT  = 000041            1300#
I$SEG  = 000041            1300#  4777  4858  4862  4892  5015  5080  5126  5192  5206  5244  5258  5298
                           5312   5358  5372  5424  5438  5476  5518  5571  5686  5773  5872  6100  6281
                           6384   6511  6517  6577  6637  6730  6827  6924  7018  7193  7199  7340  7481
                           7622   7763  7904  8068  8227  8406  8409  8467  8505  8576  8792  9026
I$SETU= 000041             1300#
I$SFT  = 000041            9257#  9263#
I$SRV  = 000041            1300#
I$SUB  = 000041            1300#  4777  4858  4862#  4888#  4890#  4892#  4903  4920#  4922#  5015  5080  5126
                           5192   5206#  5221#  5223#  5244  5258#  5273#  5275#  5298  5312#  5327#  5329#  5358
                           5372#  5402#  5404#  5424  5438#  5453#  5455#  5476  5518  5571  5686  5773  5872
                           6100   6281  6384  6511  6517#  6526  6546  6564  6571#  6573#  6577#  6586  6606
                           6624   6631#  6633#  6637#  6646  6666  6684  6691#  6693#  6730  6827  6924  7018
                           7193   7199#  7208  7224  7234  7247  7260  7280  7291  7306  7320  7331  7334#
                           7336#  7340#  7349  7365  7375  7388  7401  7421  7432  7447  7475#  7477#  7481#
                           7490   7506  7516  7529  7542  7562  7573  7588  7616#  7618#  7622#  7631  7647
                           7657   7670  7683  7703  7714  7729  7757#  7759#  7763#  7772  7788  7798  7811
                           7824   7844  7855  7870  7898#  7900#  7904#  7913  7929  7939  7952  7965  7985
                           7996   8011  8039#  8041#  8068  8227  8406  8409#  8420  8439  8460  8463#  8465#
                           8467#  8478  8501#  8503#  8505#  8516  8549  8552#  8554#  8576  8792  9026
I$TST  = 000041            1300#  4777#  4784  4793  4815  4835#  4837#  4858#  4862  4892  4924#  4926#  5015#
                           5049   5055#  5057#  5080#  5098  5103#  5105#  5126#  5164  5175#  5177#  5192#  5198
                           5206   5217  5227#  5229#  5244#  5250  5258  5269  5279#  5281#  5298#  5304  5312
                           5323   5333#  5335#  5358#  5364  5372  5398  5408#  5410#  5424#  5430  5438  5449
                           5459#  5461#  5476#  5482  5501#  5503#  5518#  5524  5543#  5545#  5571#  5603  5629
                           5656#  5658#  5686#  5697  5718  5729  5737  5741#  5743#  5773#  5784  5794  5804
                           5814   5823  5834  5842  5845#  5847#  5872#  5882  5891  5901  5919  5930  5945
                           5957   5969  5984  5995  6005  6021  6031  6046  6061  6073  6076#  6078#  6100#
                           6110   6119  6129  6147  6158  6173  6185  6197  6212  6227  6237  6250  6258
```

```
                    6261#  6263#  6281#  6292   6310   6320   6336   6347   6358   6361#  6363#  6384#  6394
                    6416   6439   6456   6467   6476   6479#  6481#  6511#  6517   6577   6637   6694#  6696#
                    6730#  6740   6749   6759   6777   6791   6801   6804#  6806#  6827#  6837   6846   6856
                    6874   6888   6898   6901#  6903#  6924#  6934   6943   6953   6971   6985   6995   6998#
                    7000#  7018#  7028   7046   7056   7066   7076   7085   7096   7106   7117   7127   7138
                    7149   7160   7163#  7165#  7193#  7199   7340   7461   7472   7481   7602   7613   7622
                    7743   7754   7763   7884   7895   7904   8025   8036   8042#  8044#  8068#  8107   8126
                    8135   8149   8158   8177   8191   8201   8204#  8206#  8227#  8237   8259   8271   8285
                    8296   8310   8329   8340   8351   8360   8363#  8365#  8406#  8409   8451   8467   8498
                    8505   8555#  8557#  8576#  8587   8609   8619   8629   8639   8648   8661   8671   8681
                    8691   8700   8711   8725   8736   8748   8752#  8754#  8792#  8802   8824   8834   8844
                    8854   8863   8876   8886   8896   8906   8916   8926   8936   8945   8956   8970   8981
                    8993   8996#  8998#  9026#  9036   9046   9059   9069   9079   9089   9099   9108   9119
                    9132   9144   9156   9166   9175   9178#  9180#
J$JMP = 000167       1300#
LOADAT  002436       2080#
LOE   = 040000 G     1574#
LOGDEV  002410       2068#  4613*  4620*  4622   4707
LOT   = 000010 G     1563#
LUSWI1  002540       2125#  4645*  4874   4948
LUSWI2  002542       2126#  4646*  4871   4949
LU1MOD  002000 G     1303#
L$ACP   002110 G     1402#
L$APT   002036 G     1360#
L$AU    021554 G     1387   4757#
L$AUT   002070 G     1386#
L$AUTO  021422 G     1403   4677#
L$CCP   002106 G     1400#
L$CLEA  021546 G     1401   4727#
L$CO    002032 G     1356#
L$DEPO  002011 G     1338#
L$DESC  003326 G     1393   2395#
L$DESP  002076 G     1392#
L$DEVP  002060 G     1378#
L$DISP  002124 G     1363   1426#
L$DLY   002116 G     1408#
L$DTP   002040 G     1362#
L$DTYP  002034 G     1358#
L$DU    021550 G     1389   4741#
L$DUT   002072 G     1388#
L$DVTY  003306 G     1379   2383#
L$EF    002052 G     1373#
L$ENVI  002044 G     1366#
L$ERRT  002246 G     1397   2022#
L$ETP   002102 G     1396#
L$EXP1  002046 G     1366#
L$EXP4  002064 G     1382#
L$EXP5  002066 G     1384#
L$HARD  036646 G     1345   9194   9195#
L$HIME  002120 G     1410#
L$HPCP  002016 G     1344#
L$HPTP  002022 G     1348#
L$HW    002224 G     1349   1468   1469#
L$ICP   002104 G     1398#
L$INIT  021100 G     1399   4560#
L$LADP  002026 G     1352#
```

```
L$LAST  037434 G        1353    9281#
L$LOAD  002100 G        1394#
L$LUN   002074 G        1390#
L$MREV  002050 G        1370#
L$NAME  002000 G        1327#
L$PRIO  002042 G        1364#
L$PROT  021072 G        1405    4547#
L$PRT   002112 G        1404#
L$REPP  002062 G        1380#
L$REV   002010 G        1336#
L$SOFT  037322 G        9257    9258#
L$SPC   002056 G        1376#
L$SPCP  002020 G        1346#
L$SPTP  002024 G        1350#
L$STA   002030 G        1354#
L$SW    002245 G        1493    1494#
L$TEST  002114 G        1406#
L$TIML  002014 G        1342#
L$UNIT  002012 G        1340#
L10000  002244          1468    1484#
L10001  002246          1493    1498#
L10002  017152          4135#
L10003  017250          4162#
L10004  017370          4196#
L10005  017544          4240#
L10006  017720          4285#
L10007  020034          4320#
L10010  020132          4349#
L10012  021420          4658#
L10013  021536          4713#
L10014  021546          4731#
L10015  021552          4746#
L10016  021554          4759#
L10017  021734          4785    4794    4816    4835#
L10020  022122          4924#
L10021  022020          4888#
L10022  022120          4904    4920#
L10023  022314          4972#
L10024  022352          4987#
L10025  023120          5050    5055#
L10026  023204          5099    5103#
L10027  023430          5165    5175#
L10030  023522          5199    5218    5227#
L10031  023516          5221#
L10032  023614          5251    5270    5279#
L10033  023610          5273#
L10034  023706          5305    5324    5333#
L10035  023702          5327#
L10036  024036          5365    5399    5408#
L10037  024032          5402#
L10040  024130          5431    5450    5459#
L 0041  024124          5453#
L10042  024204          5483    )1#
L10043  024260          5525    )43#
L10044  024602          5604    5630    5656#
L10045  024734          5698    5719    5730    5738    5741#
```

```
L10046  025116  5785  5795  5805  5815  5824  5835  5843  5845#
L10047  025624  5883  5892  5902  5920  5931  5946  5958  5970  5985  5996  6006  6022  6032
                6047  6062  6074  6076#
L10050  026224  6111  6120  6130  6148  6159  6174  6186  6198  6213  6228  6238  6251  6259
                6261#
L10051  026432  6293  6311  6321  6337  6348  6359  6361#
L10052  026674  6395  6417  6440  6457  6468  6477  6479#
L10053  027406  6694#
L10054  027064  6527  6547  6565  6571#
L10055  027234  6587  6607  6625  6631#
L10056  027404  6647  6667  6685  6691#
L10057  027614  6741  6750  6760  6778  6792  6802  6804#
L10060  030010  6838  6847  6857  6875  6889  6899  6901#
L10061  030204  6935  6944  6954  6972  6986  6996  6998#
L10062  030562  7029  7047  7057  7067  7077  7086  7097  7107  7118  7128  7139  7150  7161
                7163#
L10063  033474  7462  7473  7603  7614  7744  7755  7885  7896  8026  8037  8042#
L10064  031154  7209  7225  7235  7248  7261  7281  7292  7307  7321  7332  7334#
L10065  031542  7350  7366  7376  7389  7402  7422  7433  7448  7475#
L10066  032130  7491  7507  7517  7530  7543  7563  7574  7589  7616#
L10067  032516  7632  7648  7658  7671  7684  7704  7715  7730  7757#
L10070  033104  7773  7789  7799  7812  7825  7845  7856  7871  7898#
L10071  033472  7914  7930  7940  7953  7966  7986  7997  8012  8039#
L10072  034062  8108  8127  8136  8150  8159  8178  8192  8202  8204#
L10073  034432  8238  8260  8272  8286  8297  8311  8330  8341  8352  8361  8363#
L10074  035050  8452  8499  8555#
L10075  034606  8421  8440  8461  8463#
L10076  034710  8479  8501#
L10077  035046  8517  8550  8552#
L10100  035510  8588  8610  8620  8630  8640  8649  8662  8672  8682  8692  8701  8712  8726
                8737  8749  8752#
L10101  036246  8803  8825  8835  8845  8855  8864  8877  8887  8897  8907  8917  8927  8937
                8946  8957  8971  8982  8994  8996#
L10102  036642  9037  9047  9060  9070  9080  9090  9100  9109  9120  9133  9145  9157  9167
                9176  9178#
L10103  036750  9194  9240#
L10104  037322  9257  9262#
MCLR  = 000100  1593#  1595  2428  2438
MDMRDY= 000040  1763#
MLWRI   003744  2565  2590#
MPCSR   002472  2098#  4205  4250  4295  4330  4633  4978
MPIVEC  002532  2122#  4641*
MPOVEC  002534  2123#  4642*  4643*
MPRIOR  002536  2124#  4644*
MRDY  = 000200  1600#  2435  2478  2525  2597  4803
MREQ  = 000001  1594#  1595  2428  2438
MSTCLR  003374  2428#  3051  4779
NCRACT= 020000  2008#  7144  7155  7315  7326  7456  7467  7597  7608  7738  7749  7879  7890
                8020  8031
NCTBMT= 000200  2004#  3648  6411  8254  8318  9161
NEWLIN  011645  4068#  4966
NEWST   021244  4597  4612#
NFCRDA= 040000  2007#  7155  7326  7467  7608  7749  7890  8031
NOCHK = 003400  1703#  5691  5778  5876  6104  6286  6388  6520  6580  6640  6734  6831  6928
                7022  8231  8414  8472  8510  8581  8796
NOCRDA= 100000  2006#  6434  6451  6462  6559  6619  6679  7275  7286  7301  7315  7326  7416
```

```
                        7427    7442    7557    7568    7583    7698    7709    7724    7839    7850    7865    7980    7991
                        8006    8280    8291    8305    8324    8335    8346
NOLOOP= 001000          20020
NORXEN= 040000          20010   5877    6105    6735    6832    6929    8582    8797
NULCLK= 000200          17450   5586    5592    5612    5638    5644
OVRC  = 002000          17050
O$APTS= 000000          13000   1354
C$AU  = 000001          13000   13240   1386
O$BGNR= 000000          13000   1380
O$BGNS= 000000          13000   1346
O$DU  = 000001          13000   13240   1388
O$ERRT= 000001          13000   13240   1396
O$GNSW= 000000          13000   1350
O$POIN= 000001          13000   13240   1412
O$SETU= 000000          13000   1340    9279
PALENB= 000001          19100
PATCH   037322          92650
PATE    002652          21780   5018    5083
PATF    002662          21890   2959    5029    5128
PATG    002672          22000   5131    5201    5202
PATH    002722          22270   5202    5253    5254    5433    5434    5485    5486    5527    5528
PATI    002753          22550   5254    5307    5308    5434    5486    5528
PATJ    003014          22910   5136    5308
PATK    003024          23020   5367    5368
PATL    003045          23220   5368    5369
PATQ    003066          23420
PATQB   003076          23520
PBLENB= 000002          19090
PCR   = 120407          17110   3561    5317    7238    7379    7520    7661    7802    7943
PCSARH= 120405          16840   3555    5211    6011    6218    8094
PCSARL= 120404          16770   3551    5263    8091
PNT   = 001000 G        15690
PRESET= 000001          17520   2953    3545    8085
PRI   = 002000 G        15700
PRIOR   002416          20710
PRIRTY  037027          9209    92420
PRI00 = 000000 G        15580
PRI01 = 000040 G        15570
PRI02 = 000100 G        15560
PRI03 = 000140 G        15550
PRI04 = 000200 G        15540
PRI05 = 000240 G        15530
PRI06 = 000300 G        15520
PRI07 = 000340 G        15510
PROTO = 000100          16890   6012
PSTACK  002414          20700   4562*
RABGA = 000004          16340   3835    3837    3847
RAMADR= 001000          19960
RCVBUF  003106          23660
RCVDAT= 000002          17670
RCV1ST  011364          39560   5712    5817    6330    7079    8694    8939    9102
RDA   = 000200          17270   3242    3253    3977
RDSRH = 120401          16280   3326    3373    3735    5533    5537
RDSRL = 120400          16220   3738    5491    5495
READ    003476          24700   2691    2733    2738    2763    5382    9125
READI   003610          25160   2820    2907    2961    3066    3102    3147    3192    3237    3282    3325    3372    3439
```

```
                        3595    3734    3737    3748    3974    4864    4867    4896    5031    5147    5494    5536    5583
                        5609    5635    5933    5972    6049    6161    6200
REDBYT  002432          2078#
REDDAT  002572          2143#   5129    5137*   5140*   5150    5154
REDLOC= 000001          1605#   2471    2482    2518    2529
REDPAG= 000003          1607#
REGNUM  002412          2069#   2967*   2968*   3072*   3101*   3146*   3191*   3236*   3281*   3324*   3371*   3438*   3753*
                        3762*   3780*   3785*   4143    4169    4218    4263    4308    4826*   5036*   5037*   5152*   5203*
                        5255*   5309*   5390*   5435*   5591*   5617*   5643*
REG0    002602          2146#   5582*   5588*   5608*   5614*   5634*   5640*
REG1    002604          2147#
REG2    002606          2148#
REG3    002610          2149#
REG4    002612          2150#
REG5    002614          2151#
REG6    002616          2152#
REG7    002620          2153#
REOM  = 000002          1635#   3401    3412    3858    3860    3870
RERCHK= 000001          1645#   3786    7325    7466    7607    7737    7878    8019
RERR  = 000200          1631#   3789    3791    3801
RETADR  002430          2077#
RING  = 000200          1761#
ROR   = 000010          1633#   3330    3341    3812    3814    3824
RSA   = 000020          1730#   3287    3298
RSOM  = 000001          1636#   3377    3388    3881    3883    3893
RSTCHK  005126          2947#   5093    5732    5837    6253    8493
RTSND = 000010          1749#   2953    2956    3029    3545    3548    8085    8088    8099
RUN   = 000200          1592#   1595    2428    2438
RXABGA= 002000          1641#
RXACT = 000040          1729#   3152    3163
RXCHAR  010110          3731#   5721    5826    6339    6350    6431    6448    6459    6556    6616    6676    7088    7109
                        7130    7141    7152    7272    7283    7298    7312    7323    7413    7424    7439    7453    7464
                        7554    7565    7580    7594    7605    7695    7706    7721    7735    7746    7836    7847    7862
                        7876    7887    7977    7988    8003    8017    8028    8277    8288    8302    8321    8332    8343
                        8703    8728    8948    8973    9111
RXDL  = 000007          1718#   3742    6641    7908    7944
RXEN  = 000100          1746#   3581    3611    6325    8099    8541    8598    8813    9135
RXEOM = 001000          1642#   7153    7324    7465    7606    7747    7888    8029
RXERR = 100000          1639#
RXOR  = 004000          1640#
RXSOM = 000400          1643#   7089    7273    7414    7555    7696    7837    7978
SAVE4   002446          2084#   4569*   4573
SAVE6   002450          2085#   4570*   4574
SAVLEN  002456          2088#   3558*   3559*   3742    4566*
SCRACH  002406          2067#
SECAD = 000020          1691#
SECADR= 010000          1700#
SEL0    002472          2097#   2653
SEL10   002512          2110#   2657
SEL12   002516          2113#   2658
SEL14   002522          2116#   2659
SEL16   002526          2119#   2660
SEL2    002476          2101#   2654    4796*
SEL4    002502          2104#   2470*   2517*   2563*   2588*   2655
SEL6    002506          2107#   2540    2564*   2589*   2656
SERIAL  007256          3483#   5894    5912    5950    5998    6014    6024    6066    6122    6140    6178    6230    6243
```

```
                    6752   6770   6784   6794   6849   6867   6881   6891   6946   6964   6978   6988   8119
                    8142   8170   8184   8194
SETVIA  005326      30170  3052
SFPTBL  002246 G    14950
SFR     000001      17340
SPEED   000020      17640
SRMODE  000034      18980
STALL   004400      27160
STARES  002466      20920  4608*  4614*
STARST  021234      4583   4590   4607#
STEPLU  011614      3492   3593   3654   3702   3922   3971   4022   40450  5960   6188   6327   6422   6445
                    7269   7410   7551   7692   7833   7974   8116   8442   8490   8528
STRIP   000040      16900
STRIPS  020000      16990  8796   9030
STRTML  000301      15950
STUREG  004272      26820  5212   5264   5318   5444
SUBRFC  002420      20720  4563*
SVCGBL  000000      13000  1303   13100  1327   1336   1338   1340   1342   1344   1346   1348   1350   1352
                    1354   1356   1358   1360   1362   1364   1366   1368   1370   1373   1376   1378   1380
                    1382   1384   1386   1388   1390   1392   1394   1396   1398   1400   1402   1404   1406
                    1408   1410   1426   1469   1470   1494   1495   2022   2383   2395   4102   4141   4168
                    4203   4248   4293   4328   4547   4560   4677   4727   4741   4757   4928   4976   9195
                    9258   92810  9282
SVCINS  000001      13000  13070  1328   1329   1330   1331   1332   1333   1334   1335   1337   1339   1341
                    1343   1345   1347   1349   1351   1353   1355   1357   1359   1361   1363   1365   1367
                    1369   1371   1372   1374   1375   1377   1379   1381   1383   1385   1387   1389   1391
                    1393   1395   1397   1399   1401   1403   1405   1407   1409   1411   1425   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   1456   1457   1468   1493   2384   2387   2396   2403   2441   2442   2443   2444
                    2485   2486   2487   2488   2532   2533   2534   2535   2603   2604   2605   2606   2702
                    2703   2704   2705   2975   2976   2977   2978   3079   3080   3081   3082   3112   3113
                    3114   3115   3123   3124   3125   3126   3157   3158   3159   3160   3168   3169   3170
                    3171   3202   3203   3204   3205   3213   3214   3215   3216   3247   3248   3249   3250
                    3258   3259   3260   3261   3292   3293   3294   3295   3303   3304   3305   3306   3335
                    3336   3337   3338   3346   3347   3348   3349   3382   3383   3384   3385   3393   3394
                    3395   3396   3406   3407   3408   3409   3417   3418   3419   3420   3449   3450   3451
                    3452   3461   3462   3463   3464   3510   3511   3512   3513   3757   3758   3759   3760
                    3770   3771   3772   3773   3796   3797   3798   3799   3806   3807   3808   3809   3819
                    3820   3821   3822   3829   3830   3831   3832   3842   3843   3844   3845   3852   3853
                    3854   3855   3865   3866   3867   3868   3875   3876   3877   3878   3888   3889   3890
                    3891   3898   3899   3900   3901   4109   4110   4111   4112   4113   4114   4124   4125
                    4126   4127   4128   4129   4130   4136   4143   4144   4145   4146   4147   4148   4149
                    4152   4153   4154   4155   4156   4157   4158   4159   4163   4172   4173   4174   4175
                    4176   4177   4178   4181   4182   4183   4184   4185   4186   4187   4188   4190   4191
                    4192   4193   4194   4197   4205   4206   4207   4208   4209   4210   4211   4213   4214
                    4215   4216   4217   4221   4222   4223   4224   4225   4226   4227   4230   4231   4232
                    4233   4234   4235   4236   4237   4241   4250   4251   4252   4253   4254   4255   4256
                    4258   4259   4260   4261   4262   4266   4267   4268   4269   4270   4271   4272   4275
                    4276   4277   4278   4279   4280   4281   4282   4286   4295   4296   4297   4298   4299
                    4300   4301   4303   4304   4305   4306   4307   4311   4312   4313   4314   4315   4316
                    4317   4321   4330   4331   4332   4333   4334   4335   4336   4339   4340   4341   4342
                    4343   4344   4345   4346   4350   4376   4377   4378   4379   4380   4381   4382   4384
                    4385   4386   4387   4388   4389   4390   4391   4392   4394   4395   4396   4397   4398
                    4399   4401   4402   4403   4404   4405   4406   4407   4408   4409   4411   4412   4413
                    4414   4415   4425   4426   4427   4428   4429   4430   4431   4433   4434   4435   4436
```

C2

4437	4438	4439	4440	4441	4443	4444	44[illegible]	4446	4447	4448	4450	4451
4452	4453	4454	4455	4456	4457	4458	446	4461	4462	4463	4464	4465
4467	4468	4469	4470	4471	4472	4473	4474	4475	4477	4478	4479	4480
4481	4482	4484	4485	4486	4487	4488	4489	4490	4491	4492	4502	4503
4504	4505	4506	4507	4508	4510	4511	45[illegible]	4513	4514	4515	4516	4517
4518	4520	4521	4522	4523	4524	4525	4527	4528	4529	4530	4531	4532
4533	4534	4535	4580	4581	4583	4587	4588	4590	4594	4595	4597	4601
4602	4604	4622	4623	4624	4626	4652	4659	4680	4681	4682	4683	4684
4685	4702	4703	4707	4708	4714	4732	4744	4747	4760	4782	4784	4785
4791	4793	4794	4810	4811	4812	4813	4815	4816	4829	4830	4831	4832
4836	4863	4879	4880	4881	4882	4884	4889	4893	4901	4903	4904	4915
4916	4917	4918	4921	4925	4930	4931	4932	4933	4934	4936	4937	4938
4939	4940	4942	4943	4944	4945	4946	4948	4949	4950	4951	4952	4953
4954	4958	4959	4960	4961	4962	4963	4964	4966	4967	4968	4969	4970
4973	4978	4979	4980	4981	4982	4983	4984	4988	5044	5045	5046	5047
5049	5050	5056	5096	5098	5099	5104	5159	5160	5161	5162	5164	5165
5176	5196	5198	5199	5207	5215	5217	5218	5222	5228	5248	5250	5251
5259	5267	5269	5270	5274	5280	5302	5304	5305	5313	5321	5323	5324
5328	5334	5362	5364	5365	5373	5393	5394	5395	5396	5398	5399	5403
5409	5428	5430	5431	5439	5447	5449	5450	5454	5460	5480	5482	5483
5502	5522	5524	5525	5544	5598	5599	5600	5601	5603	5604	5624	5625
5626	5627	5629	5630	5650	5651	5652	5653	5657	5695	5697	5698	5716
5718	5719	5727	5729	5730	5735	5737	5738	5742	5782	5784	5785	5792
5794	5795	5802	5804	5805	5812	5814	5815	5821	5823	5824	5832	5834
5835	5840	5842	5843	5846	5880	5882	5883	5889	5891	5892	5895	5901
5902	5917	5919	5920	5928	5930	5931	5940	5941	5942	5943	5945	5946
5955	5957	5958	5967	5969	5970	5979	5980	5981	5982	5984	5985	5993
5995	5996	6003	6005	6006	6019	6021	6022	6029	6031	6032	6044	6046
6047	6056	6057	6058	6059	6061	6062	6071	6073	6074	6077	6108	6110
6111	6117	6119	6120	6127	6129	6130	6145	6147	6148	6156	6158	6159
6168	6169	6170	6171	6173	6174	6183	6185	6186	6195	6197	6198	6207
6208	6209	6210	6212	6213	6225	6227	6228	6235	6237	6238	6248	6250
6251	6256	6258	6259	6262	6290	6292	6293	6308	6310	6311	6318	6320
6321	6334	6336	6337	6345	6347	6348	6356	6358	6359	6362	6392	6394
6395	6414	6416	6417	6437	6439	6440	6454	6456	6457	6465	6467	6468
6474	6476	6477	6480	6518	6524	6526	6527	6544	6546	6547	6562	6564
6565	6572	6578	6584	6586	6587	6604	6606	6607	6622	6624	6625	6632
6638	6644	6646	6647	6664	6666	6667	6682	6684	6685	6692	6695	6738
6740	6741	6747	6749	6750	6757	6759	6760	6775	6777	6778	6789	6791
6792	6799	6801	6802	6805	6835	6837	6838	6844	6846	6847	6854	6856
6857	6872	6874	6875	6886	6888	6889	6896	6898	6899	6902	6932	6934
6935	6941	6943	6944	6951	6953	6954	6969	6971	6972	6983	6985	6986
6993	6995	6996	6999	7026	7028	7029	7044	7046	7047	7054	7056	7057
7064	7066	7067	7074	7076	7077	7083	7085	7086	7094	7096	7097	7104
7106	7107	7115	7117	7118	7125	7127	7128	7136	7138	7139	7147	7149
7150	7158	7160	7161	7164	7200	7206	7208	7209	7222	7224	7225	7232
7234	7235	7245	7247	7248	7258	7260	7261	7278	7280	7281	7289	7291
7292	7304	7306	7307	7318	7320	7321	7329	7331	7332	7335	7341	7347
7349	7350	7363	7365	7366	7373	7375	7376	7386	7388	7389	7399	7401
7402	7419	7421	7422	7430	7432	7433	7445	7447	7448	7459	7461	7462
7470	7472	7473	7476	7482	7488	7490	7491	7504	7506	7507	7514	7516
7517	7527	7529	7530	7540	7542	7543	7560	7562	7563	7571	7573	7574
7586	7588	7589	7600	7602	7603	7611	7613	7614	7617	7623	7629	7631
7632	7645	7647	7648	7655	7657	7658	7668	7670	7671	7681	7683	7684
7701	7703	7704	7712	7714	7715	7727	7729	7730	7741	7743	7744	7752
7754	7755	7758	7764	7770	7772	7773	7786	7788	7789	7796	7798	7799

D2


```
                    7809  7811  7812  7822  7824  7825  7842  7844  7845  7853  7855  7856  7868
                    7870  7871  7882  7884  7885  7893  7895  7896  7899  7905  7911  7913  7914
                    7927  7929  7930  7937  7939  7940  7950  7952  7953  7963  7965  7966  7983
                    7985  7986  7994  7996  7997  8009  8011  8012  8023  8025  8026  8034  8036
                    8037  8040  8043  8105  8107  8108  8124  8126  8127  8133  8135  8136  8147
                    8149  8150  8156  8158  8159  8175  8177  8178  8189  8191  8192  8199  8201
                    8202  8205  8235  8237  8238  8257  8259  8260  8269  8271  8272  8283  8285
                    8286  8294  8296  8297  8308  8310  8311  8327  8329  8330  8338  8340  8341
                    8349  8351  8352  8358  8360  8361  8364  8410  8418  8420  8421  8437  8439
                    8440  8449  8451  8452  8458  8460  8461  8464  8468  8476  8478  8479  8496
                    8498  8499  8502  8506  8514  8516  8517  8547  8549  8550  8553  8556  8585
                    8587  8588  8607  8609  8610  8617  8619  8620  8627  8629  8630  8637  8639
                    8640  8646  8648  8649  8659  8661  8662  8669  8671  8672  8679  8681  8682
                    8689  8691  8692  8698  8700  8701  8709  8711  8712  8723  8725  8726  8734
                    8736  8737  8746  8748  8749  8753  8800  8802  8803  8822  8824  8825  8832
                    8834  8835  8842  8844  8845  8852  8854  8855  8861  8863  8864  8874  8876
                    8877  8884  8886  8887  8894  8896  8897  8904  8906  8907  8914  8916  8917
                    8924  8926  8927  8934  8936  8937  8943  8945  8946  8954  8956  8957  8968
                    8970  8971  8979  8981  8982  8991  8993  8994  8997  9034  9036  9037  9044
                    9046  9047  9057  9059  9060  9067  9069  9070  9077  9079  9080  9087  9089
                    9090  9097  9099  9100  9106  9108  9109  9117  9119  9120  9130  9132  9133
                    9142  9144  9145  9154  9156  9157  9164  9166  9167  9173  9175  9176  9179
                    9194  9198  9199  9200  9201  9203  9204  9205  9206  9208  9209  9210  9211
                    9212  9214  9215  9216  9217  9218  9220  9221  9222  9223  9224  9226  9227
                    9228  9229  9230  9232  9233  9234  9235  9236  9239  9257  9261  9278  9279
                    9280
SVCSUB= 000001       13000 13090  4862  4892  5206  5258  5312  5372  5438  6517  6577  6637  7199
                     7340  7481  7622  7763  7904  8409  8467  8505
SVCTAG= 000001       13000 13110  1418  1422  1484  1498  4135  4162  4196  4240  4285  4320  4349
                     4658  4713  4731  4746  4759  4835  4888  4920  4924  4972  4987  5055  5103
                     5175  5221  5227  5273  5279  5327  5333  5402  5408  5453  5459  5501  5543
                     5656  5741  5845  6076  6261  6361  6479  6571  6631  6691  6694  6804  6901
                     6998  7163  7334  7475  7616  7757  7898  8039  8042  8204  8363  8463  8501
                     8552  8555  8752  8996  9178  9240  9262
SVCTST= 000001       13000 13080  4777  4858  5015  5080  5126  5192  5244  5298  5358  5424  5476
                     5518  5571  5686  5773  5872  6100  6281  6384  6511  6730  6827  6924  7018
                     7193  8068  8227  8406  8576  8792  9026
SWPBOT= 121000       19900  4868
SWPDDC= 121400       19910  4865
SW1.M   037060       9215  92420
SW2.M   037121       9221  92420
SYNCH = 000226       16780  3588  5778  5788  5876  6104  6286  6388  6520  6580  6640  7202  7343
                     7484  7625  7766  7907  8231  8278  8414  8472  8510  8581  8633  8655  8665
                     8675  8685  8704  8796  8848  8870  8880  8890  8900  9030  9040
S$LSYM= 010000       13006 14850  14990 41360 41630 41970 42410 42860 43210 43500 46590 47140 47320
                     47470 47600 48360 48890 49210 49250 49730 49880 50560 51040 51760 52220 52280
                     52740 52800 53280 53340 54030 54090 54540 54600 55020 55440 56570 57420 58460
                     60770 62620 63620 64800 65720 66320 66920 66950 68050 69020 69990 71640 73350
                     74760 76170 77580 78990 80400 80430 82050 83640 84640 85020 85530 85560 87530
                     89970 91790 92410 92630
TAB   = 000004       16620  6781  6878
TBMT  = 000100       17280  3197  3208  3598  5925  5964  6153  6192  9151  9170
TCCHEK= 100000       19950
TDATA   002376       20630  2687  5209* 5261* 5315* 5375* 5380  5441*
TDSRH = 120403       16570  3584  3689  5379  5383  5934  5973  6050  6162  6201
TDSRL = 120402       16510  3587  3636  5138  5443  5905  6133  6767  6864  6961  8164  8181
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
TDSRNR  002651          21750
TEOM  = 000002          16630   6975    7121    7267    7408    7549    7690    7831    7972    8113
TERR  = 000200          16600
TGA   = 000010          16610   6975
TIMFLG  002426          20760
TM    = 000004          17660
TMP0    002622          21560   4686*   4704    4716*
TMP1    002624          21570
TMP2    002626          21580
TMP3    002630          21590
TMP4    002632          21600
TMP5    002634          21610
TMP6    002636          21620
TMP7    002640          21630
TSO   = 000010          17310   3444    3456    5925    5964    6041    6153    6192    9170
TSOM  = 000001          16640   3585    5701    5705    5709    6035    6296    6398    6530    6590    6650    7032    7212
                        7353    7494    7635    7776    7917    8113    8139    8240    8424    8427    8482    8485    8520
                        8523    8591    8806
TSTCON  002546          21280   4648*
TSTNUM  002470          20930
TTLOOP= 000002          17510   3581    3611    4021    6325    8099    8533    8541    8598    8813
TXA8  = 002000          16690
TXACT = 000004          17320   3107    3118    5925    5964    6041    6153    6192    9151    9170
TXCHAR  007676          36310   5787    5797    5807    6303    6313    6409    6539    6599    6659    7039    7049    7059
                        7069    7099    7217    7227    7240    7253    7358    7368    7381    7394    7499    7509    7522
                        7535    7640    7650    7663    7676    7781    7791    7804    7817    7922    7932    7945    7958
                        8252    8316    8602    8612    8622    8632    8654    8664    8674    8684    8718    8817    8827
                        8837    8847    8869    8879    8889    8899    8909    8919    8929    8963    9039    9052    9062
                        9072    9082    9092    9159
TXCTRL  010010          36840   5700    5704    5708    5908    6034    6136    6295    6299    6397    6400    6529    6532
                        6589    6592    6649    6652    6762    6780    6859    6877    6956    6974    7031    7035    7120
                        7211    7214    7266    7352    7355    7407    7493    7496    7548    7634    7637    7689    7775
                        7778    7830    7916    7919    7971    8112    8138    8166    8239    8243    8423    8426    8429
                        8481    8484    8487    8519    8522    8525    8590    8593    8805    8808    9049
TXDL  = 000340          17150   3742    6641    7944
TXEN  = 000040          17470   3581    3611    8099    8533    8541    8598    8813
TXEND1  035527          8739    87650
TXEND2  036262          8984    90090
TXEND3  026710          6419    6442    64920
TXEND4  034447          8262    8312    83770
TXEOM = 001000          16700
TXERR = 100000          16670
TXGA  = 004000          16680
TXSOM = 000400          16710
TXTBL1  035512          8714    87550
TXTBL2  036250          8959    89990
TXTBL3  026676          6406    6428    64820
TXTBL4  034434          8249    8299    83660
TXTMLT  016702          40770   4124
TXTML0  016113          40630   4077
TXTML1  016117          40680   4077
TXTML2  016133          40680   4077
TXTML3  016150          40680   4077
TXTML4  016172          40680   4077
TXTML5  016213          40680   4077
TXTML6  016243          40680   4077
```

F2

TXTML7	016255	4068#	4077				
TXTNP	016437	4068#	4089				
TXTNPT	016766	4090#					
TXTNP0	016444	4068#	4090				
TXTNP1	016454	4068#	4090				
TXTNP2	016464	4068#	4090				
TXTNP3	016474	4068#	4090				
TXTNP4	016511	4068#	4090				
TXTNP5	016526	4068#	4090				
TXTNP6	016543	4068#	4090				
TXTNP7	016557	4068#	4090				
TXTNP8	016573	4068#	4090				
TXTNUL	016111	4068#					
TXTUR	016607	4068#	4173	4222	4312		
TXTURT	017010	4093#	4172	4221	4311		
TXTUR0	016622	4068#	4093				
TXTUR1	016630	4068#	4093				
TXTUR2	016636	4068#	4093				
TXTUR3	016644	4068#	4093				
TXTUR4	016652	4068#	4093				
TXTUR5	016661	4068#	4093				
TXTUR6	016670	4068#	4093				
TXTUR7	016674	4068#	4093				
TXTVR	016312	4068#	4081	4267			
TXTVRA	016410	4068#	4085				
TXTVRB	016413	4068#	4085				
TXTVRC	016417	4068#	4085				
TXTVRD	016423	4068#	4085				
TXTVRE	016427	4068#	4085				
TXTVRF	016433	4068#	4085				
TXTVRT	016724	4082#	4266				
TXTVR0	016330	4068#	4082				
TXTVR1	016334	4068#	4082				
TXTVR2	016340	4068#	4082				
TXTVR3	016345	4068#	4082				
TXTVR4	016352	4068#	4082				
TXTVR5	016357	4068#	4082				
TXTVR6	016364	4068#	4082				
TXTVR7	016371	4068#	4082				
TXTVR8	016376	4068#	4085				
TXTVR9	016403	4068#	4085				
TXT1	014541	4068#					
TXT10	015341	4068#					
TXT11	015361	4068#					
TXT11A	015433	4068#					
TXT11B	015471	4068#					
TXT12	015541	4068#	4206	4251	4296	4331	4979
TXT13	015567	4068#	4503				
TXT14	015604	4068#	4502				
TXT15	015642	4068#	4520				
TXT16	015704	4068#	4426				
TXT17	015717	4068#	4425				
TXT18	015754	4068#	4443				
TXT19	016015	4068#	4460				
TXT2	014577	4068#					
TXT2A	014641	4068#					

```
TXT2B   014700          40680#
TXT2O   016051          40680#  4477
TXT3    014743          40680#
TXT4    014773          40680#  4376
TXT4A   015033          40680#  4394
TXT5    015074          40680#  4144
TXT6    015076          40680#  4377
TXT7    015121          4068#
TXT7A   015211          40680#
TXT8    015301          40680#
TXT9    015321          40680#
TXU   = 000002          17330#  3751    5964    6192
T$ARGC= 000003          13280#  13290#  13300#  13310#  1332#  13330#  4109#  4114  4124#  4130  4143#  4149  4152#
                        4159  4172#  4178  4181#  4188  4190#  4194  4205#  4211  4213#  4217  4221#  4227
                        4230#  4237  4250#  4256  4258#  4262  4266#  4272  4275#  4282  4295#  4301  4303#
                        4307  4311#  4317  4330#  4336  4339#  4346  4376#  4382  4384#  4392  4394#  4399
                        4401#  4409  4411#  4415  4425#  4431  4433#  4441  4443#  4448  4450#  4458  4460#
                        4465  4467#  4475  4477#  4482  4484#  4492  4502#  4508  4510#  4518  4520#  4525
                        4527#  4535  4930#  4934  4936#  4940  4942#  4946  4948#  4954  4958#  4964  4966#
                        4970  4978#  4984
T$CODE= 007032          9198#  9203#  9208#  9214#  9220#  9226#  9232#
T$ERRN= 000065          1300#  4811#  4830#  4880#  4916#  5045#  5160#  5394#  5599#  5625#  5651#  5941#  5980#
                        6057#  6169#  6208#
T$EXCP= 000000          9198#  92 #  9203#  9207  9208#  9213  9214#  9219  9220#  9225  9226#  9231  9232#
                        9237
T$FLAG= 000040          4784#  4793#  4815#  4903#  5049#  5098#  5164#  5198#  5217#  5250#  5269#  5304#  5323#
                        5364#  5398#  5430#  5449#  5482#  5524#  5603#  5629#  5697#  5718#  5729#  5737#  5784#
                        5794#  5804#  5814#  5823#  5834#  5842#  5882#  5891#  5901#  5919#  5930#  5945#  5957#
                        5969#  5984#  5995#  6005#  6021#  6031#  6046#  6061#  6073#  6110#  6119#  6129#  6147#
                        6158#  6173#  6185#  6197#  6212#  6227#  6237#  6250#  6258#  6292#  6310#  6320#  6336#
                        6347#  6358#  6394#  6416#  6439#  6456#  6467#  6476#  6526#  6546#  6564#  6586#  6606#
                        6624#  6646#  6666#  6684#  6740#  6749#  6759#  6777#  6791#  6801#  6837#  6846#  6856#
                        6874#  6888#  6898#  6934#  6943#  6953#  6971#  6985#  6995#  7028#  7046#  7056#  7066#
                        7076#  7085#  7096#  7106#  7117#  7127#  7138#  7149#  7160#  7208#  7224#  7234#  7247#
                        7260#  7280#  7291#  7306#  7320#  7331#  7349#  7365#  7375#  7388#  7401#  7421#  7432#
                        7447#  7461#  7472#  7490#  7506#  7516#  7529#  7542#  7562#  7573#  7588#  7602#  7613#
                        7631#  7647#  7657#  7670#  7683#  7703#  7714#  7729#  7743#  7754#  7772#  7788#  7798#
                        7811#  7824#  7844#  7855#  7870#  7884#  7895#  7913#  7929#  7939#  7952#  7965#  7985#
                        7996#  8011#  8025#  8036#  8107#  8126#  8135#  8149#  8158#  8177#  8191#  8201#  8237#
                        8259#  8271#  8285#  8296#  8310#  8329#  8340#  8351#  8360#  8420#  8439#  8451#  8460#
                        8478#  8498#  8516#  8549#  8587#  8609#  8619#  8629#  8639#  8648#  8661#  8671#  8681#
                        8691#  8760#  8711#  8725#  8736#  8748#  8802#  8824#  8834#  8844#  8854#  8863#  8876#
                        8886#  8896#  8906#  8916#  8926#  8936#  8945#  8956#  8970#  8981#  8993#  9036#  9046#
                        9059#  9069#  9079#  9089#  9099#  9108#  9119#  9132#  9144#  9156#  9166#  9175#
T$GMAN= 000000          1300#
T$HILI= 000001          9198#  9201  9203#  9206  9208#  9212  9214#  9218  9220#  9224  9226#  9230  9232#
                        9236
T$LAST= 000001          1300#  9279#
T$LOLI= 000000          9198#  9200  9203#  9205  9208#  9211  9214#  9217  9220#  9223  9226#  9229  9232#
                        9235
T$LSYM= 010000          1300#  1485  1499  4136  4163  4197  4241  4286  4321  4350  4659  4714  4732
                        4747  4760  4836  4889  4921  4925  4973  4988  5056  5104  5176  5222  5228
                        5274  5280  5328  5334  5403  5409  5454  5460  5502  5544  5657  5742  5846
                        6077  6262  6362  6480  6572  6632  6692  6695  6805  6902  6999  7164  7335
                        7476  7617  7758  7899  8040  8043  8205  8364  8464  8502  8553  8556  8753
                        8997  9179  9241  9263
```

```
T$LTNO= 000057          9282#
T$NEST= 177777          13000#  13030#  14680#  14840#  14930#  14980#  41020#  41350#  41410#  41620#  41680#  41960#  42030#
                        42400#  42480#  42850#  42930#  43200#  43280#  43490#  45470#  45520#  45600#  46580#  46770#  47130#
                        47270#  47310#  47410#  47460#  47570#  47590#  47780#  48350#  48590#  48630#  48880#  48930#  49200#
                        49240#  49280#  49720#  49760#  49870#  50160#  50550#  50810#  51030#  51270#  51750#  51930#  52070#
                        52210#  52270#  52450#  52590#  52730#  52790#  52990#  53130#  53270#  53330#  53590#  53730#  54020#
                        54080#  54250#  54390#  54530#  54590#  54770#  55010#  55190#  55430#  55720#  56560#  56870#  57410#
                        57740#  58450#  58730#  60760#  61010#  62610#  62820#  63610#  63850#  64790#  65120#  65180#  65710#
                        65780#  66310#  66380#  66910#  66940#  67310#  68040#  68280#  69010#  69250#  69980#  70190#  71630#
                        71940#  72000#  73340#  73410#  74750#  74820#  76160#  76230#  77570#  77640#  78980#  79050#  80390#
                        80420#  80690#  82040#  82280#  83630#  84070#  84100#  84630#  84680#  85010#  85060#  85520#  85550#
                        85770#  87520#  87930#  89960#  90270#  91780#  91940#  92390#  92570#  92610#  92750#
T$NS0 = 000000          13030#  9275
T$NS1 = 000005          14680#  1484    14930#  1498    41020#  4135    41410#  4162    41680#  4196    42030#  4240    42480#
                        4285    42930#  4320    43280#  4349    45470#  4552    45600#  4658    46770#  4713    47270#  4731
                        47410#  4746    47570#  4759    47780#  4835    48590#  4924    49280#  4972    49760#  4987    50160#
                        5055    50810#  5103    51270#  5175    51930#  5227    52450#  5279    52990#  5333    53590#  5408
                        54250#  5459    54770#  5501    55190#  5543    55720#  5656    56870#  5741    57740#  5845    58730#
                        6076    61010#  6261    62820#  6361    63850#  6479    65120#  6694    67310#  6804    68280#  6901
                        69250#  6998    70190#  7163    71940#  8042    80690#  8204    82280#  8363    84070#  8555    85770#
                        8752    87930#  8996    90270#  9178    91940#  9239    92570#  9261
T$NS2 = 000002          48630#  4888    48930#  4920    52070#  5221    52590#  5273    53130#  5327    53730#  5402    54390#
                        5453    65180#  6571    65780#  6631    66380#  6691    72000#  7334    73410#  7475    74820#  7616
                        76230#  7757    77640#  7898    79050#  8039    84100#  8463    84680#  8501    85060#  8552
T$PTNU= 000000          13000#
T$SAVL= 177777          13000#
T$SEGL= 177777          13000#
T$SUBN= 000000          13000#  47770#  48580#  48620#  48920#  50150#  50300#  51260#  51920#  52060#  52440#  52580#  52980#
                        53120#  53580#  53720#  54240#  54380#  54760#  55180#  55710#  56860#  57730#  58720#  61000#  62810#
                        63840#  65110#  65170#  65770#  66370#  67300#  68270#  69240#  70180#  71930#  71990#  73400#  74810#
                        76220#  77630#  79040#  80680#  82270#  84060#  84090#  84670#  85050#  85760#  87920#  90260#
T$TAGL= 177777          13000#
T$TAGN= 010105          13000#  14680#  14930#  41020#  41410#  41680#  42030#  42480#  42930#  43280#  45470#  45600#  46770#
                        47270#  47410#  47570#  47780#  48590#  48630#  48930#  49280#  49760#  50160#  50810#  51270#  51930#
                        52070#  52450#  52590#  52990#  53130#  53590#  53730#  54250#  54390#  54770#  55190#  55720#  56870#
                        57740#  58730#  61010#  62820#  63850#  65120#  65180#  65780#  66380#  67310#  68280#  69250#  70190#
                        71940#  72000#  73410#  74820#  76230#  77640#  79050#  80690#  82280#  84070#  84100#  84680#  85060#
                        85770#  87930#  90270#  91940#  92570#
T$TEMP= 000000          14180#  14220#  14270#  14280#  14290#  14300#  14310#  14320#  14330#  14340#  14350#  14360#  14370#
                        14380#  14390#  14400#  14410#  14420#  14430#  14440#  14450#  14460#  14470#  14480#  14490#  14500#
                        14510#  14520#  14530#  14540#  14550#  14560#  14570#  14580#  14840#  14980#  41350#  41620#  41960#
                        42400#  42850#  43200#  43490#  45520#  46580#  47130#  47310#  47460#  47590#  47840#  4785    47930#
                        4794    48150#  4816    48350#  48880#  49030#  4904    49200#  49240#  49720#  49870#  50490#  5050
                        50550#  50980#  5099    51030#  51640#  5165    51750#  51980#  5199    52170#  5218    52210#  52270#
                        52500#  5251    52690#  5270    52730#  52790#  53040#  5305    53230#  5324    53270#  53330#  53640#
                        5365    53980#  5399    54020#  54080#  54300#  5431    54490#  5450    54530#  54590#  54820#  5483
                        55010#  55240#  5525    55430#  56030#  5604    56290#  5630    56560#  56970#  5698    57180#  5719
                        57290#  5730    57370#  5738    57410#  57840#  5785    57940#  5795    58040#  5805    58140#  5815
                        58230#  5824    58340#  5835    58420#  5843    58450#  58820#  5883    58910#  5892    59010#  5902
                        59190#  5920    59300#  5931    59450#  5946    59570#  5958    59690#  5970    59840#  5985    59950#
                        5996    60050#  6006    60210#  6022    60310#  6032    60460#  6047    60610#  6062    60730#  6074
                        60760#  61100#  6111    61190#  6120    61290#  6130    61470#  6148    61580#  6159    61730#  6174
                        61850#  6186    61970#  6198    62120#  6213    62270#  6228    62370#  6238    62500#  6251    62580#
                        6259    62610#  62920#  6293    63100#  6311    63200#  6321    63360#  6337    63470#  6348    63580#
                        6359    63610#  63940#  6395    64160#  6417    64390#  6440    64560#  6457    64670#  6468    64760#
                        6477    64790#  65260#  6527    65460#  6547    65640#  6565    65710#  65860#  6587    66060#  6607
```

```
                    6624#  6625   6631#  6646#  6647   6666#  6667   6684#  6685   6691#  6694#  6740#  6741
                    6749#  6750   6759#  6760   6777#  6778   6791#  6792   6801#  6802   6804#  6837#  6838
                    6846#  6847   6856#  6857   6874#  6875   6888#  6889   6898#  6899   6901#  6934#  6935
                    6943#  6944   6953#  6954   6971#  6972   6985#  6986   6995#  6996   6998#  7028#  7029
                    7046#  7047   7056#  7057   7066#  7067   7076#  7077   7085#  7086   7096#  7097   7106#
                    7107   7117#  7118   7127#  7128   7138#  7139   7149#  7150   7160#  7161   7163#  7208#
                    7209   7224#  7225   7234#  7235   7247#  7248   7260#  7261   7280#  7281   7291#  7292
                    7306#  7307   7320#  7321   7331#  7332   7334#  7349#  7350   7365#  7366   7375#  7376
                    7388#  7389   7401#  7402   7421#  7422   7432#  7433   7447#  7448   7461#  7462   7472#
                    7473   7475#  7490#  7491   7506#  7507   7516#  7517   7529#  7530   7542#  7543   7562#
                    7563   7573#  7574   7588#  7589   7602#  7603   7613#  7614   7616#  7631#  7632   7647#
                    7648   7657#  7658   7670#  7671   7683#  7684   7703#  7704   7714#  7715   7729#  7730
                    7743#  7744   7754#  7755   7757#  7772#  7773   7788#  7789   7798#  7799   7811#  7812
                    7824#  7825   7844#  7845   7855#  7856   7870#  7871   7884#  7885   7895#  7896   7898#
                    7913#  7914   7929#  7930   7939#  7940   7952#  7953   7965#  7966   7985#  7986   7996#
                    7997   8011#  8012   8025#  8026   8036#  8037   8039#  8042#  8107#  8108   8126#  8127
                    8135#  8136   8149#  8150   8158#  8159   8177#  8178   8191#  8192   8201#  8202   8204#
                    8237#  8238   8259#  8260   8271#  8272   8285#  8286   8296#  8297   8310#  8311   8329#
                    8330   8340#  8341   8351#  8352   8360#  8361   8363#  8420#  8421   8439#  8440   8451#
                    8452   8460#  8461   8463#  8478#  8479   8498#  8499   8501#  8516#  8517   8549#  8550
                    8552#  8555#  8587#  8588   8609#  8610   8619#  8620   8629#  8630   8639#  8640   8648#
                    8649   8661#  8662   8671#  8672   8681#  8682   8691#  8692   8700#  8701   8711#  8712
                    8725#  8726   8736#  8737   8748#  8749   8752#  8802#  8803   8824#  8825   8834#  8835
                    8844#  8845   8854#  8855   8863#  8864   8876#  8877   8886#  8887   8896#  8897   8906#
                    8907   8916#  8917   8926#  8927   8936#  8937   8945#  8946   8956#  8957   8970#  8971
                    8981#  8982   8993#  8994   8996#  9036#  9037   9046#  9047   9059#  9060   9069#  9070
                    9079#  9080   9089#  9090   9099#  9100   9108#  9109   9119#  9120   9132#  9133   9144#
                    9145   9156#  9157   9166#  9167   9175#  9176   9178#  9198#  9203#  9208#  9214#  9220#
                    9226#  9232#  9239#  9261#  9275#
T$TEST= 000037       1300#  4777#  4858#  4862   4892   5015#  5080#  5126#  5192#  5206   5244#  5258   5298#
                    5312   5358#  5372   5424#  5438   5476#  5518#  5571#  5686#  5773#  5872#  6100#  6281#
                    6384#  6511#  6517   6577   6637   6730#  6827#  6924#  7018#  7193#  7199   7340   7481
                    7622   7763   7904   8068#  8227#  8406#  8409   8467   8505   8576#  8792#  9026#  9282
T$TSTM= 177777       1300#  4113   4129   4136   4148   4158   4163   4177   4187   4193   4197   4210   4216
                    4226   4236   4241   4255   4261   4271   4281   4286   4300   4306   4316   4321   4335
                    4345   4350   4381   4391   4398   4408   4414   4430   4440   4447   4457   4464   4474
                    4481   4491   4507   4517   4524   4534   4581   4588   4595   4602   4623   4652   4659
                    4684   4703   4708   4714   4732   4744   4747   4760   4782   4784   4791   4793   4810
                    4815   4829   4836   4863   4879   4884   4889   4893   4901   4903   4915   4921   4925
                    4933   4939   4945   4953   4963   4969   4973   4983   4988   5044   5049   5056   5096
                    5098   5104   5159   5164   5176   5196   5198   5207   5215   5217   5222   5228   5248
                    5250   5259   5267   5269   5274   5280   5302   5304   5313   5321   5323   5328   5334
                    5362   5364   5373   5393   5398   5403   5409   5428   5430   5439   5447   5449   5454
                    5460   5480   5482   5502   5522   5524   5544   5598   5603   5624   5629   5650   5657
                    5695   5697   5716   5718   5727   5729   5735   5737   5742   5782   5784   5792   5794
                    5802   5804   5812   5814   5821   5823   5832   5834   5840   5842   5846   5880   5882
                    5889   5891   5899   5901   5917   5919   5928   5930   5940   5945   5955   5957   5967
                    5969   5979   5984   5993   5995   6003   6005   6019   6021   6029   6031   6044   6046
                    6056   6061   6071   6073   6077   6108   6110   6117   6119   6127   6129   6145   6147
                    6156   6158   6168   6173   6183   6185   6195   6197   6207   6212   6225   6227   6235
                    6237   6248   6250   6256   6258   6262   6290   6292   6308   6310   6318   6320   6334
                    6336   6345   6347   6356   6358   6362   6392   6394   6414   6416   6437   6439   6454
                    6456   6465   6467   6474   6476   6480   6518   6524   6526   6544   6546   6562   6564
                    6572   6578   6584   6586   6604   6606   6622   6624   6632   6638   6644   6646   6664
                    6666   6682   6684   6692   6695   6738   6740   6747   6749   6757   6759   6775   6777
                    6789   6791   6799   6801   6805   6835   6837   6844   6846   6854   6856   6872   6874
```

```
                    6886  6888  6896  6898  6902  6932  6934  6941  6943  6951  6953  6969  6971
                    6983  6985  6993  6995  6999  7026  7028  7044  7046  7054  7056  7064  7066
                    7074  7076  7083  7085  7094  7096  7104  7106  7115  7117  7125  7127  7136
                    7138  7147  7149  7158  7160  7164  7200  7206  7208  7222  7224  7232  7234
                    7245  7247  7258  7260  7278  7280  7289  7291  7304  7306  7318  7320  7329
                    7331  7335  7341  7347  7349  7363  7365  7373  7375  7386  7388  7399  7401
                    7419  7421  7430  7432  7445  7447  7459  7461  7470  7472  7476  7482  7488
                    7490  7504  7506  7514  7516  7527  7529  7540  7542  7560  7562  7571  7573
                    7586  7588  7600  7602  7611  7613  7617  7623  7629  7631  7645  7647  7655
                    7657  7668  7670  7681  7683  7701  7703  7712  7714  7727  7729  7741  7743
                    7752  7754  7758  7764  7770  7772  7786  7788  7796  7798  7809  7811  7822
                    7824  7842  7844  7853  7855  7868  7870  7882  7884  7893  7895  7899  7905
                    7911  7913  7927  7929  7937  7939  7950  7952  7963  7965  7983  7985  7994
                    7996  8009  8011  8023  8025  8034  8036  8040  8043  8105  8107  8124  8126
                    8133  8135  8147  8149  8156  8158  8175  8177  8189  8191  8199  8201  8205
                    8235  8237  8257  8259  8269  8271  8283  8285  8294  8296  8308  8310  8327
                    8329  8338  8340  8349  8351  8358  8360  8364  8410  8418  8420  8437  8439
                    8449  8451  8458  8460  8464  8468  8476  8478  8496  8498  8502  8506  8514
                    8516  8547  8549  8553  8556  8585  8587  8607  8609  8617  8619  8627  8629
                    8637  8639  8646  8648  8659  8661  8669  8671  8679  8681  8689  8691  8698
                    8700  8709  8711  8723  8725  8734  8736  8746  8748  8753  8800  8802  8822
                    8824  8832  8834  8842  8844  8852  8854  8861  8863  8874  8876  8884  8886
                    8894  8896  8904  8906  8914  8916  8924  8926  8934  8936  8943  8945  8954
                    8956  8968  8970  8979  8981  3991  8993  8997  9034  9036  9044  9046  9057
                    9059  9067  9069  9077  9079  9087  9089  9097  9099  9106  9108  9117  9119
                    9130  9132  9142  9144  9154  9156  9164  9166  9173  9175  9179
T$TSTS= 000001       13000  4778#  4859#  5016#  5081#  5127#  5193#  5245#  5299#  5359#  5425#  5477#  5519#
                    5572#  5687#  5774#  5873#  6101#  6282#  6385#  6512#  6731#  6828#  6925#  7019#  7194#
                    8069#  8228#  8407#  8577#  8793#  9027#
T$$AU = 010016       4757#  4759
T$$AUT= 010013       4677#  4713
T$$CLE= 010014       4727#  4731
T$$DU = 010015       4741#  4746
T$$HAR= 010103       9194#  9240
T$$HW = 010000       1468#  1484
T$$INI= 010012       4560#  4658
T$$MSG= 010024       4102#  4135  4141#  4162  4168#  4196  4203#  4240  4248#  4285  4293#  4320  4328#
                    4349  4928#  4972  4976#  4987
T$$PRO= 010011       4547#
T$$SOF= 010104       9257#  9262
T$$SUB= 010077       4863#  4888  4893#  4903  4920  5207#  5221  5259#  5273  5313#  5327  5373#  5402
                    5439#  5453  6518#  6526  6546  6564  6571  6578#  6586  6606  6624  6631  6638#
                    6646  6666  6684  6691  7200#  7208  7224  7234  7247  7260  7280  7291  7306
                    7320  7331  7334  7341#  7349  7365  7375  7388  7401  7421  7432  7447  7475
                    7482#  7490  7506  7516  7529  7542  7562  7573  7588  7616  7623#  7631  7647
                    7657  7670  7683  7703  7714  7729  7757  7764#  7772  7788  7798  7811  7824
                    7844  7855  7870  7898  7905#  7913  7929  7939  7952  7965  7985  7996  8011
                    8039  8410#  8420  8439  8460  8463  8468#  8478  8501  8506#  8516  8549  8552
T$$SW = 010001       1493#  1498
T$$TES= 010102       4778#  4784  4793  4815  4835  4859#  4924  5016#  5049  5055  5081#  5098  5103
                    5127#  5164  5175  5193#  5198  5217  5227  5245#  5250  5269  5279  5299#  5304
                    5323  5333  5359#  5364  5398  5408  5425#  5430  5449  5459  5477#  5482  5501
                    5519#  5524  5543  5572#  5603  5629  5656  5687#  5697  5718  5729  5737  5741
                    5774#  5784  5794  5804  5814  5823  5834  5842  5845  5873#  5882  5891  5901
                    5919  5930  5945  5957  5969  5984  5995  6005  6021  6031  6046  6061  6073
                    6076  6101#  6110  6119  6129  6147  6158  6173  6185  6197  6212  6227  6237
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
                        6250    6258    6261    6282#   6292    6310    6320    6336    6347    6358    6361    6385#   6394
                        6416    6439    6456    6467    6476    6479    6512#   6694    6731#   6740    6749    6759    6777
                        6791    6801    6804    6828#   6837    6846    6856    6874    6888    6898    6901    6925#   6934
                        6943    6953    6971    6985    6995    6998    7019#   7028    7046    7056    7066    7076    7085
                        7096    7106    7117    7127    7138    7149    7160    7163    7194#   7461    7472    7602    7613
                        7743    7754    7884    7895    8025    8036    8042    8069#   8107    8126    8135    8149    8158
                        8177    8191    8201    8204    8228#   8237    8259    8271    8285    8296    8310    8329    8340
                        8351    8360    8363    8407#   8451    8498    8555    8577#   8587    8609    8619    8629    8639
                        8648    8661    8671    8681    8691    8700    8711    8725    8736    8748    8752    8793#   8802
                        8824    8834    8844    8854    8863    8876    8886    8896    8906    8916    8926    8936    8945
                        8956    8970    8981    8993    8996    9027#   9036    9046    9059    9069    9079    9089    9099
                        9108    9119    9132    9144    9156    9166    9175    9178
T.EDF  = 000001         2010#   2441    2485    2532    2603    2702    2975    3079    3112    3123    3157    3168    3202
                        3213    3247    3258    3292    3303    3335    3346    3382    3393    3406    3417    3449    3461
                        3510    3757    3770    3796    3806    3819    3829    3842    3852    3865    3875    3888    3898
T.EHRD= 000002          2010#
T.ESF  = 000000         2010#
T.ESFT= 000003          2010#
T1      021556 G        1427    4777#
T1MODE= 000300          1883#
T10     024040 G        1436    5424#
T10.LP  023740          5371#   5405
T10.1   024072          5438#
T11     024132 G        1437    5476#
T11.LP  024072          5437#   5456
T12     024206 G        1438    5518#
T13     024262 G        1439    5571#
T14     024604 G        1440    5686#
T15     024736 C        1441    5773#
T16     025120 G        1442    5872#
T17     025626 G        1443    6100#
T18     026226 G        1444    6281#
T19     026434 G        1445    6384#
T2      021736 G        1428    4858#
T2MODE= 000040          1893#
T2.1    021742          4862#
T2.2    022022          4892#
T20     026712 G        1446    6511#
T20.1   026716          6517#
T20.2   027066          6577#
T20.3   027236          6637#
T21     027422 G        1447    6730#
T22     027616 G        1448    6827#
T23     030012 G        1449    6924#
T24     030206 G        1450    7018#
T24TBL  027410          6536    6549    6552    6567    6596    6609    6612    6627    6656    6669    6672    6687    6697#
T25     030564 G        1451    7193#
T25.1   030570          7199#
T25.2   031156          7340#
T25.3   031544          7481#
T25.4   032132          7622#
T25.5   032520          7763#
T25.6   033106          7904#
T26     033506 G        1452    8068#
T27     034064 G        1453    8227#
T28     034450 G        1454    8406#
```

```
T28.1   034450           8409#
T28.2   034610           8467#
T28.3   034712           8505#
T29     035052 G         1455    8576#
T3      022736 G         1429    5015#
T3.EM0  022354           4881    4990#
T3.EM1  022124 G         4882    4928#
T3.EM2  022316 G         4918    4976#
T3.END  022122           4923#
T3.ERR  022004           4872    4877#
T3.OK   022016           4875    4886#
T3.SW1  021752           4866#   4871    4955*   4959
T3.SW2  021762           4869#   4874    4956*   4958
T3.1    022400           4930    4990#
T3.2    022465           4936    4990#
T3.3    022514           4942    4990#
T3.4    022545           4950    4990#
T3.5    022610           4960    4990#
T3.6    022652           4917    4990#
T30     035530 G         1456    8792#
T30TBL  033476           7250    7263    7294    7309    7391    7404    7435    7450    7532    7545    7576    7591    7673
                         7686    7717    7732    7814    7827    7858    7873    7955    7968    7999    8014    8045#
T31     036264 G         1457    9026#
T4      023122 G         1430    5080#
T5      023206 G         1431    5126#
T6      023432 G         1432    5192#
T6.1    023464           5206#
T7      023524 G         1433    5244#
T7.LP   023464           5205#   5224
T7.1    023556           5258#
T8      023616 G         1434    5298#
T8.LP   023556           5257#   5276
T8.1    023650           5312#
T9      023710 G         1435    5358#
T9.LP   023650           5311#   5330
T9.1    023740           5372#
UAM   = 000200 G         1567#
UMAINT= 000001           1768#
UNIT    002464           2091#
UPBITS  002642           2166#
UREGS   002316           2056#   2729    2732*   2735    4510    4511    4512    4513    4527    4528    4529    4530
USTATR= 122000           1723#   2734    3067    3103    3148    3193    3238    3283    3440    3596    3749    3975
USYREG  002552           2133#   2960    5019    5030    5084    5135    5146
USYRT = 120400           1616#   2730    2743    2747
VECTOR  036776           9204    9242#
VIA   = 120000           1737#   2761    2768
VIAACR= 120013           1881#   2821    2830    2908    2917    3031    3564    8074
VIADPA= 120003           1787#   3022
VIADPB= 120002           1775#   3019
VIAIER= 120016           1954#   2817    2904    3037
VIAIFR= 120015           1932#
VIAMS = 120001           1759#
VIAORA= 120017           1974#   3025    4897
VIAORB= 120000           1743#   2952    2955    3028    3544    3547    3580    4020    5584    5610    5636    6324    8084
                         8087    8098    8532    8540    8597    8812    9126    9138
VIAPCR= 120014           1919#   3034
```

```
VIASR = 120012          1871#
VIAT1A= 120004          1801#  2843   5574
VIAT1B= 120005          1813#  2848   4049   5577
VIAT1C= 120006          1826#  2834   3567   8077
VIAT1D= 120007          1837#  2839   3570   8080
VIAT2A= 120010          1848#  2921
VIAT2B= 120011          1860#  2926
VREGS   002336          2059#  2760   4433   4434   4435   4436   4450   4451   4452   4453   4467   4468   4469
                        4470   4484   4485   4486   4487
WAITSO  005312          2994#  8536
WRIBYT  002434          2079#
WRILOC= 000002          1606#  2590   2600
WRIPAG= 000004          1608#
WRITE   003722          2563#  2685   5378
WRITEI  003734          2587#  2816   2829   2833   2838   2842   2847   2903   2916   2920   2925   2951   2954
                        3018   3021   3024   3027   3030   3033   3036   3543   3546   3550   3554   3560   3563
                        3566   3569   3579   3583   3586   3635   3688   4019   4048   5021   5086   5141   5490
                        5532   5573   5576   5904   6010   6132   6217   6323   6766   6863   6960   7237   7378
                        7519   7660   7801   7942   8073   8076   8079   8083   8086   8090   8093   8097   8163
                        8180   8531   8539   8596   8811   9137
WSR0    002256          2031#  2653*  4387
WSR10   002266          2039#  2657*  4404
WSR12   002270          2041#  2658*  4403
WSR14   002272          2043#  2659*  4402
WSR16   002274          2045#  2660*  4401
WSR2    002260          2033#  2654*  4386   4819   4825
WSR4    002262          2035#  2655*  4385
WSR6    002264          2037#  2656*  4384
XDATA   002404          2066#  4152   4181   4230   4275   4339   4365*  4366*
XORGB   020134          4150   4179   4228   4273   4337   4363#
XYZ   = 000007          1693#  6012   8095
X$ALWA= 000000          1300#
X$FALS= 000040          1300#
X$OFFS= 000400          1300#
X$TRUE= 000020          1300#
$E    = 000065          2010#  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#
                        4809#  4828#  4878#  4914#  5043#  5158#  5392#  5597#  5623#  5649#  5939#  5978#  6055#
                        6167#  6206#
$LSTIN= 000001          1305#
$LSTTA= 000001          1306#
$T    = 000037          2010#  4762#  4838#  4992#  5057#  5105#  5177#  5229#  5281#  5335#  5410#  5461#  5503#
                        5545#  5658#  5743#  5848#  6078#  6263#  6363#  6495#  6709#  6806#  6903#  7000#  7165#
                        8054#  8206#  8380#  8557#  8771#  9012#
.     = 037434          1296#  2056#  2059#  2143#  2226#  2366#  2387#  2403#  3515   4780   4785   4789   4794
                        4816   4899   4904   5050   5094   5099   5165   5199   5218   5251   5270   5305   5324
                        5365   5399   5431   5450   5483   5525   5604   5630   5693   5698   5714   5719   5725
                        5730   5733   5738   5780   5785   5790   5795   5800   5805   5810   5815   5819   5824
                        5830   5835   5838   5843   5878   5883   5887   5892   5897   5902   5915   5920   5926
                        5931   5946   5953   5958   5965   5970   5985   5991   5996   6001   6006   6017   6022
                        6027   6032   6042   6047   6062   6069   6074   6106   6111   6115   6120   6125   6130
                        6143   6148   6154   6159   6174   6181   6186   6193   6198   6213   6223   6228   6233
                        6238   6246   6251   6254   6259   6288   6293   6306   6311   6316   6321   6332   6337
                        6343   6348   6354   6359   6390   6395   6412   6417   6435   6440   6452   6457   6463
                        6468   6472   6477   6493#  6522   6527   6542   6547   6560   6565   6582   6587   6602
```

6607	6620	6625	6642	6647	6662	6667	6680	6685	6736	6741	6745	6750
6755	6760	6773	6778	6787	6[illegible]2	6797	6802	6833	6838	6842	6847	6852
6857	6870	6875	6884	6889	6o94	6899	6930	6935	6939	6944	6949	6954
6967	6972	6981	6986	6991	6996	7024	7029	7042	7047	7052	7057	7062
7067	7072	7077	7081	7086	7092	7097	7102	7107	7113	7118	7123	7128
7134	7139	7145	7150	7156	7161	7204	7209	7220	7225	7230	7235	7243
7248	7256	7261	7276	7281	7287	7292	7302	7307	7316	7321	7327	7332
7345	7350	7361	7366	7371	7376	7384	7389	7397	7402	7417	7422	7428
7433	7443	7448	7457	7462	7468	7473	7486	7491	7502	7507	7512	7517
7525	7530	7538	7543	7558	7563	7569	7574	7584	7589	7598	7603	7609
7614	7627	7632	7643	7648	7653	7658	7666	7671	7679	7684	7699	7704
7710	7715	7725	7730	7739	7744	7750	7755	7768	7773	7784	7789	7794
7799	7807	7812	7820	7825	7840	7845	7851	7856	7866	7871	7880	7885
7891	7896	7909	7914	7925	7930	7935	7940	7948	7953	7961	7966	7981
7986	7992	7997	8007	8012	8021	8026	8032	8037	8052#	8103	8108	8122
812[illegible]	8131	8136	8145	8150	8154	8159	8173	8178	8187	8192	8197	8202
8233	8238	8255	8260	8267	8272	8281	8286	8292	8297	8306	8311	8325
8330	8336	8341	8347	8352	8356	8361	8416	8421	8435	8440	8447	8452
8456	8461	8474	8479	8494	8499	8512	8517	8545	8550	8583	8588	8605
8610	861[illegible]	8620	8625	8630	8635	8640	8644	8649	8657	8662	8667	8672
8677	868[illegible]	8687	8692	8696	8701	8707	8712	8721	8726	8732	8737	8744
8749	879[illegible]	8803	8820	8825	8830	8835	8840	8845	8850	8855	8859	8864
8872	8[illegible]77	8882	8887	8892	8897	8902	8907	8912	8917	8922	8927	8932
8937	8941	8946	8952	8957	8966	8971	8977	8982	8989	8994	9010#	9032
9037	9042	9047	9055	9060	9065	9070	9075	9080	9085	9090	9095	9100
9104	9109	9115	9120	9128	9133	9140	9145	9152	9157	9162	9167	9171
9176	9266#											

```
BCOMPL      1#   13000#  4582   4589   4596   4603   4625
BERROR      1#   13000#
BGNAU       1#   13000#  4756
BGNAUT      1#   13000#  4676
BGNCLN      1#   13000#  4726
BGNDU       1#   13000#  4740
BGNHRD      1#   13000#  9193
BGNHW       1#   13000#  1467
BGNINI      1#   13000#  4559
BGNMOD      1#   13000#  1302
BGNMSG      1#   13000#  4101   4140   4167   4202   4247   4292   4327   4927   4975
BGNPRO      1#   13000#  4546
BGNPTA      1#   13000#
BGNRPT      1#   13000#
BGNSEG      1#   13000#
BGNSET      1#   13000#
BGNSFT      1#   13000#  9256
BGNSRV      1#   13000#
BGNSUB      1#   13000#  4861   4891   5205   5257   5311   5371   5437   6516   6576   6636   7198   7339   7480
          7621   7762    7903   8408   8466   8504
BGNSW       1#   13000#  1492
BGNTST      1#   13000#  4777   4858   5015   5080   5126   5192   5244   5298   5358   5424   5476   5518   5571
          5686   5773    5872   6100   6281   6384   6511   6730   6827   6924   7018   7193   8068   8227   8406
          8576   8792    9026
BNCOMP      1#   13000#
BNERRO      1#   13000#
BREAK       1#   13000#
BRESET      1#   13000#  4651   4743
CKLOOP      1#   13000#  4883
CLOCK       1#   13000#
CLOSE       1#   13000#
CLRVEC      1#   13000#  4701
COMMEN      1#   13000#
DELAY       1#   13000#
DESCRI      1#   13000#  2394
DEVTYP      1#   13000#  2382
DISPAT      1#   13000#  1424
DISPLA      1#   13000#
DOCLN       1#   13000#
DODU        1#   13000#  4706
DORPT       1#   13000#
ENDAU       1#   13000#  4758
ENDAUT      1#   13000#  4712
ENDCLN      1#   13000#  4730
ENDCOM      1#   13000#
ENDDU       1#   13000#  4745
ENDHRD      1#   13000#  9238
ENDHW       1#   13000#  1483
ENDINI      1#   13000#  4657
ENDMOD      1#   13000#  9274
ENDMSG      1#   13000#  4134   4161   4195   4239   4284   4319   4348   4971   4986
ENDPRO      1#   13000#  4551
ENDPTA      1#   13000#
ENDRPT      1#   13000#
ENDSEG      1#   13000#
ENDSET      1#   13000#
```

ENDSFT 1# 13004 9260
ENDSRV 1# 13004
ENDSUB 1# 13004 4887 4919 5220 5272 5326 5401 5452 6570 6630 6690 7333 7474 76 5
7756 7897 8038 8462 8500 8551
ENDSW 1# 13004 1497
ENDTST 1# 13004 4834 4923 5054 5102 5174 5226 5278 5332 5407 5458 5500 5542 5655
5740 5844 6075 6260 6360 6478 6693 6803 6900 6997 7162 8041 8203 8362 8554
8751 8995 9177
EQUALS 1# 13004 1507
ERRDF 1# 13004 4810 4829 4879 4915 5044 5159 5393 5598 5624 5650 5940 5979 6056
6168 6207
ERRHRD 1# 13004
ERROR 1# 13004 4781 4790 4900 5095 5195 5214 5247 5266 5301 5320 5361 5427 5446
5479 5521 5694 5715 5726 5734 5781 5791 5801 5811 5820 5831 5839 5879 5888
5898 5916 5927 5954 5966 5992 6002 6018 6028 6043 6070 6107 6116 6126 6144
6155 6182 6194 6224 6234 6247 6255 6289 6307 6317 6333 6344 6355 6391 6413
6436 6453 6464 6473 6523 6543 6561 6583 6603 6621 6643 6663 6681 6737 6746
6756 6774 6788 6798 6834 6843 6853 6871 6885 6895 6931 6940 6950 6968 6982
6992 7025 7043 7053 7063 7073 7082 7093 7103 7114 7124 7135 7146 7157 7205
7221 7231 7244 7257 7277 7288 7303 7317 7328 7346 7362 7372 7385 7398 7418
7429 7444 7458 7469 7487 7503 7513 7526 7539 7559 7570 7585 7599 7610 7628
7644 7654 7667 7680 7700 7711 7726 7740 7751 7769 7785 7795 7808 7821 7841
7852 7867 7881 7892 7910 7926 7936 7949 7962 7982 7993 8008 8022 8033 8104
8123 8132 8146 8155 8174 8188 8198 8234 8256 8268 8282 8293 8307 8326 8337
8348 8357 8417 8436 8448 8457 8475 8495 8513 8546 8584 8606 8616 8626 8636
8645 8658 8668 8678 8688 8697 8708 8722 8733 8745 8799 8821 8831 8841 8851
8860 8873 8883 8893 8903 8913 8923 8933 8942 8953 8967 8978 8990 9033 9043
9056 9066 9076 9086 9096 9105 9116 9129 9141 9153 9163 9172
ERRSF 1# 13004
ERRSOF 1# 13004
ERRTBL 1# 13004 2021
ESCAPE 1# 13004 4783 4792 4814 4902 5048 5097 5163 5197 5216 5249 5268 5303 5322
5363 5397 5429 5448 5481 5523 5602 5628 5696 5717 5728 5736 5783 5793 5803
5813 5822 5833 5841 5881 5890 5900 5918 5929 5944 5956 5968 5983 5994 6004
6020 6030 6045 6060 6072 6109 6118 6128 6146 6157 6172 6184 6196 6211 6226
6236 6249 6257 6291 6309 6319 6335 6346 6357 6393 6415 6438 6455 6466 6475
6525 6545 6563 6585 6605 6623 6645 6665 6683 6739 6748 6758 6776 6790 6800
6836 6845 6855 6873 6887 6897 6933 6942 6952 6970 6984 6994 7027 7045 7055
7065 7075 7084 7095 7105 7116 7126 7137 7148 7159 7207 7223 7233 7246 7259
7279 7290 7305 7319 7330 7348 7364 7374 7387 7400 7420 7431 7446 7460 7471
7489 7505 7515 7528 7541 7561 7572 7587 7601 7612 7630 7646 7656 7669 7682
7702 7713 7728 7742 7753 7771 7787 7797 7810 7823 7843 7854 7869 7883 7894
7912 7928 7938 7951 7964 7984 7995 8010 8024 8035 8106 8125 8134 8148 8157
8176 8190 8200 8236 8258 8270 8284 8295 8309 8328 8339 8350 8359 8419 8438
8450 8459 8477 8497 8515 8548 8586 8608 8618 8628 8638 8647 8660 8670 8680
8690 8699 8710 8724 8735 8747 8801 8823 8833 8843 8853 8862 8875 8885 8895
8905 8915 8925 8935 8944 8955 8969 8980 8992 9035 9045 9058 9068 9078 9088
9098 9107 9118 9131 9143 9155 9165 9174
EXIT 1# 13004
FEQUAL 1# 13004
GEDF 20104 4808 4827 4877 4913 5042 5157 5391 5596 5622 5648 5938 5977 6054 6166
6205
GEHRD 20104
GESF 20104
GESFT 20104
GETBYT 1# 13004

```
GETPRI      1#  1300#
GETWOR      1#  1300#
GMANIA      1#  1300#
GMANID      1#  1300#
GMVNIL      1#  1300#
GP  D       1#  1300#  4621
GPRMA       1#  1300#  9197  9202
GPRMD       1#  1300#  9207  9213  9219  9225  9231
GPRML       1#  1300#
GTDF      2010#  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
GTHRD     2010#
GTSF      2010#
GTSFT     2010#
HEADER      1#  1300#  1326
INLOOP      1#  1300#
IOSETU      1#  1300#
IOSTAR      1#  1300#
KT11        1#  1300#
LASTAD      1#  1300#  9277
MANUAL      1#  1300#
MEMORY      1#  1300#
MSG       4762#  4768  4838#  4844  4992#  4998  5057#  5063  5105#  5111  5177#  5183  5229#  5235  5281#
          5287  5335#  5341  5410#  5416  5461#  5467  5503#  5509  5545#  5551  5658#  5664  5743#  5749
          5848#  5854  6078#  6084  6263#  6269  6363#  6369  6495#  6501  6709#  6715  6806#  6812  6903#
          6909  7000#  7006  7165#  7171  8054#  8060  8206#  8212  8380#  8386  8557#  8563  8771#  8777
          9012#  9018
M$BYTE      1#  1300#  1327#  1333  1334  1335
M$CHEC      1#  1300#
M$CNTO      1#  1300#  9198#  9203#  9208#  9214#  9220#  9226#  9232#
M$COUN      1#  1300#  4109#  4124#  4143#  4152#  4172#  4181#  4190#  4205#  4213#  4221#  4230#  4250#  4258#
          4266#  4275#  4295#  4303#  4311#  4330#  4339#  4376#  4384#  4394#  4401#  4411#  4425#  4433#  4443#
          4450#  4460#  4467#  4477#  4484#  4502#  4510#  4520#  4527#  4930#  4936#  4942#  4948#  4958#  4966#
          4978#
M$DATA      1#  1300#  1327#  1336  1338  1340  1342  1344  1346  1348  1350  1352  1354  1356  1358
          1360  1362  1364  1366#  1368  1370  1373  1376  1378  1380  1382  1384  1386  1388  1390
          1392  1394  1396  1398  1400  1402  1404  1406  1408  1410  2383#  2395#
M$DECR      1#  1300#  1484#  1498#  4135#  4162#  4196#  4240#  4285#  4320#  4349#  4552#  4658#  4713#  4731#
          4746#  4759#  4835#  4888#  4920#  4924#  4972#  4987#  5055#  5103#  5175#  5221#  5227#  5273#  5279#
          5327#  5333#  5402#  5408#  5453#  5459#  5501#  5543#  5656#  5741#  5845#  6076#  6261#  6361#  6479#
          6571#  6631#  6691#  6694#  6804#  6901#  6998#  7163#  7334#  7475#  7616#  7757#  7898#  8039#  8042#
          8204#  8363#  8463#  8501#  8552#  8555#  8752#  8996#  9178#  9239#  9261#  9275#
M$DEFA      1#  1300#  9198#  9203#  9208#  9214#  9220#  9226#  9232#
M$ENDE      1#  1300#  1484#  1498#  4135#  4162#  4196#  4240#  4285#  4320#  4349#  4658#  4713#  4731#  4746#
          4759#  4835#  4888#  4920#  4924#  4972#  4987#  5055#  5103#  5175#  5221#  5227#  5273#  5279#  5327#
          5333#  5402#  5408#  5453#  5459#  5501#  5543#  5656#  5741#  5845#  6076#  6261#  6361#  6479#  6571#
          6631#  6691#  6694#  6804#  6901#  6998#  7163#  7334#  7475#  7616#  7757#  7898#  8039#  8042#  8204#
          8363#  8463#  8501#  8552#  8555#  8752#  8996#  9178#  9239#  9261#  9275#
M$ERRI      1#  1300#  4810#  4829#  4879#  4915#  5044#  5159#  5393#  5598#  5624#  5650#  5940#  5979#  6056#
          6168#  6207#
M$ESCA      1#  1300#  4784#  4785  4793#  4794  4815#  4816  4903#  4904  5049#  5050  5098#  5099  5164#
          5165  5198#  5199  5217#  5218  5250#  5251  5269#  5270  5304#  5305  5323#  5324  5364#  5365
          5398#  5399  5430#  5431  5449#  5450  5482#  5483  5524#  5525  5603#  5604  5629#  5630  5697#
          5698  5718#  5719  5729#  5730  5737#  5738  5784#  5785  5794#  5795  5804#  5805  5814#  5815
          5823#  5824  5834#  5835  5842#  5843  5882#  5883  5891#  5892  5901#  5902  5919#  5920  5930#
```

	5931	5945#	5946	5957#	5958	5969#	5970	5984#	5985	5995#	5996	6005#	6006	6021#	6022
	6031#	6032	6046#	6047	6061#	6062	6073#	6074	6110#	6111	6119#	6120	6129#	6130	6147#
	6148	6158#	6159	6173#	6174	6185#	6186	6197#	6198	6212#	6213	6227#	6228	6237#	6238
	6250#	6251	6258#	6259	6292#	6293	6310#	6311	6320#	6321	6336#	6337	6347#	6348	6358#
	6359	6394#	6395	6416#	6417	6439#	6440	6456#	6457	6467#	6468	6476#	6477	6526#	6527
	6546#	6547	6564#	6565	6586#	6587	6606#	6607	6624#	6625	6646#	6647	6666#	6667	6684#
	6685	6740#	6741	6749#	6750	6759#	6760	6777#	6778	6791#	6792	6801#	6802	6837#	6838
	6846#	6847	6856#	6857	6874#	6875	6888#	6889	6898#	6899	6934#	6935	6943#	6944	6953#
	6954	6971#	6972	6985#	6986	6995#	6996	7028#	7029	7046#	7047	7056#	7057	7066#	7067
	7076#	7077	7085#	7086	7096#	7097	7106#	7107	7117#	7118	7127#	7128	7138#	7139	7149#
	7150	7160#	7161	7208#	7209	7224#	7225	7234#	7235	7247#	7248	7260#	7261	7280#	7281
	7291#	7292	7306#	7307	7320#	7321	7331#	7332	7349#	7350	7365#	7366	7375#	7376	7388#
	7389	7401#	7402	7421#	7422	7432#	7433	7447#	7448	7461#	7462	7472#	7473	7490#	7491
	7506#	7507	7516#	7517	7529#	7530	7542#	7543	7562#	7563	7573#	7574	7588#	7589	7602#
	7603	7613#	7614	7631#	7632	7647#	7648	7657#	7658	7670#	7671	7683#	7684	7703#	7704
	7714#	7715	7729#	7730	7743#	7744	7754#	7755	7772#	7773	7788#	7789	7798#	7799	7811#
	7812	7824#	7825	7844#	7845	7855#	7856	7870#	7871	7884#	7885	7895#	7896	7913#	7914
	7929#	7930	7939#	7940	7952#	7953	7965#	7966	7985#	7986	7996#	7997	8011#	8012	8025#
	8026	8036#	8037	8107#	8108	8126#	8127	8133#	8136	8149#	8150	8158#	8159	8177#	8178
	8191#	8192	8201#	8202	8237#	8238	8259#	8260	8271#	8272	8285#	8286	8296#	8297	8310#
	8311	8329#	8330	8340#	8341	8351#	8352	8360#	8361	8420#	8421	8439#	8440	8451#	8452
	8460#	8461	8478#	8479	8498#	8499	8516#	8517	8549#	8550	8587#	8588	8609#	8610	8619#
	8620	8629#	8630	8639#	8640	8648#	8649	8661#	8662	8671#	8672	8681#	8682	8691#	8692
	8700#	8701	8711#	8712	8725#	8726	8736#	8737	8748#	8749	8802#	8803	8824#	8825	8834#
	8835	8844#	8845	8854#	8855	8863#	8864	8876#	8877	8886#	8887	8896#	8897	8906#	8907
	8916#	8917	8926#	8927	8936#	893?	8945#	8946	8956#	8957	8970#	8971	8981#	8982	8993#
	8994	9036#	9037	9046#	9047	9059#	9060	9069#	9070	9079#	9080	9089#	9090	9099#	9100
	9108#	9109	9119#	9120	9132#	9133	9144#	9145	9156#	9157	9166#	9167	9175#	9176	
M$ESCS	1#	1300#	4784#	4793#	4815#	4903#	5049#	5098#	5164#	5198#	5217#	5250#	5269#	5304#	5323#
	5364#	5398#	5430#	5449#	5482#	5524#	5603#	5629#	5697#	5718#	5729#	5737#	5784#	5794#	5804#
	5814#	5823#	5834#	5842#	5882#	5891#	5901#	5919#	5930#	5945#	5957#	5969#	5984#	5995#	6005#
	6021#	6031#	6046#	6061#	6073#	6110#	6119#	6129#	6147#	6158#	6173#	6185#	6197#	6212#	6227#
	6237#	6250#	6258#	6292#	6310#	6320#	6336#	6347#	6358#	6394#	6416#	6439#	6456#	6467#	6476#
	6526#	6546#	6564#	6586#	6606#	6624#	6646#	6666#	6684#	6740#	6749#	6759#	6777#	6791#	6801#
	6837#	6846#	6856#	6874#	6888#	6898#	6934#	6943#	6953#	6971#	6985#	6995#	7028#	7046#	7056#
	7066#	7076#	7085#	7096#	7106#	7117#	7127#	7138#	7149#	7160#	7208#	7224#	7234#	7247#	7260#
	7280#	7291#	7306#	7320#	7331#	7349#	7365#	7375#	7388#	7401#	7421#	7432#	7447#	7461#	7472#
	7490#	7506#	7516#	7529#	7542#	7562#	7573#	7588#	7602#	7613#	7631#	7647#	7657#	7670#	7683#
	7703#	7714#	7729#	7743#	7754#	7772#	7788#	7798#	7811#	7824#	7844#	7855#	7870#	7884#	7895#
	7913#	7929#	7939#	7952#	7965#	7985#	7996#	8011#	8025#	8036#	8107#	8126#	8135#	8149#	8158#
	8177#	8191#	8201#	8237#	8259#	8271#	8285#	8296#	8310#	8329#	8340#	8351#	8360#	8420#	8439#
	8451#	8460#	8478#	8498#	8516#	8549#	8587#	8609#	8619#	8629#	8639#	8648#	8661#	8671#	8681#
	8691#	8700#	8711#	8725#	8736#	8748#	8802#	8824#	8834#	8844#	8854#	8863#	8876#	8886#	8896#
	8906#	8916#	8926#	8936#	8945#	8956#	8970#	8981#	8993#	9036#	9046#	9059#	9069#	9079#	9089#
	9099#	9108#	9119#	9132#	9144#	9156#	9166#	9175#							
M$EXCP	1#	1300#	9198#	9203#	9208#	9214#	9220#	9226#	9232#						
M$EXIT	1#	1300#													
M$EXSE	1#	1300#													
M$EXTJ	1#	1300#													
M$GEN	1#	1300#	1303#	1327#	1336#	1338#	1340#	1342#	1344#	1346#	1348#	1350#	1352#	1354#	1356#
	1358#	1360#	1362#	1364#	1366#	1368#	1370#	1373#	1376#	1378#	1380#	1382#	1384#	1386#	1388#
	1390#	1392#	1394#	1396#	1398#	1400#	1402#	1404#	1406#	1408#	1410#	1426#	1469#	1470#	1484#
	1494#	1495#	1498#	2022#	2383#	2395#	4102#	4135#	4141#	4162#	4168#	4196#	4203#	4240#	4248#
	4285#	4293#	4320#	4328#	4349#	4547#	4560#	4658#	4677#	4713#	4727#	4731#	4741#	4746#	4757#
	4759#	4777#	4835#	4858#	4862#	4888#	4892#	4920#	4924#	4928#	4972#	4976#	4987#	5015#	5055#
	5080#	5103#	5126#	5175#	5192#	5206#	5221#	5227#	5244#	5258#	5273#	5279#	5298#	5312#	5327#

```
                5333#  5358#  5372#  5402#  5408#  5424#  5438#  5453#  5459#  5476#  5501#  5518#  5543#  5571#  5656#
                5686#  5741#  5773#  5845#  5872#  6076#  6100#  6261#  6281#  6361#  6384#  6479#  6511#  6517#  6571#
                6577#  6631#  6637#  6691#  6694#  6730#  6804#  6827#  6901#  6924#  6998#  7018#  7163#  7193#  7199#
                7334#  7340#  7475#  7481#  7616#  7622#  7757#  7763#  7898#  7904#  8039#  8042#  8068#  8204#  8227#
                8363#  8406#  8409#  8463#  8467#  8501#  8505#  8552#  8555#  8576#  8752#  8792#  8996#  9026#  9178#
                9195#  9240#  9258#  9262#  9281#
M$GENB             1#  1300#
M$GETS             1#  1300#  1484#  1498#  4135#  4162#  4196#  4240#  4285#  4320#  4349#  4552#  4658#  4713#  4731#
                4746#  4759#  4835#  4888#  4920#  4924#  4972#  4987#  5055#  5103#  5175#  5221#  5227#  5273#  5279#
                5327#  5333#  5402#  5408#  5453#  5459#  5501#  5543#  5656#  5741#  5845#  6076#  6261#  6361#  6479#
                6571#  6631#  6691#  6694#  6804#  6901#  6998#  7163#  7334#  7475#  7616#  7757#  7898#  8039#  8042#
                8204#  8363#  8463#  8501#  8552#  8555#  8752#  8996#  9178#  9239#  9261#  9275#
M$GETT             1#  1300#  4784#  4793#  4815#  4903#  5049#  5098#  5164#  5198#  5217#  5250#  5269#  5304#  5323#
                5364#  5398#  5430#  5449#  5482#  5524#  5603#  5629#  5697#  5718#  5729#  5737#  5784#  5794#  5804#
                5814#  5823#  5834#  5842#  5882#  5891#  5901#  5919#  5930#  5945#  5957#  5969#  5984#  5995#  6005#
                6021#  6031#  6046#  6061#  6073#  6110#  6119#  6129#  6147#  6158#  6173#  6185#  6197#  6212#  6227#
                6237#  6250#  6258#  6292#  6310#  6320#  6336#  6347#  6358#  6394#  6416#  6439#  6456#  6467#  6476#
                6526#  6546#  6564#  6586#  6606#  6624#  6646#  6666#  6684#  6740#  6749#  6759#  6777#  6791#  6801#
                6837#  6846#  6856#  6874#  6888#  6898#  6934#  6943#  6953#  6971#  6985#  6995#  7028#  7046#  7056#
                7066#  7076#  7085#  7096#  7106#  7117#  7127#  7138#  7149#  7160#  7208#  7224#  7234#  7247#  7260#
                7280#  7291#  7306#  7320#  7331#  7349#  7365#  7375#  7388#  7401#  7421#  7432#  7447#  7461#  7472#
                7490#  7506#  7516#  7529#  7542#  7562#  7573#  7588#  7602#  7613#  7631#  7647#  7657#  7670#  7683#
                7703#  7714#  7729#  7743#  7754#  7772#  7788#  7798#  7811#  7824#  7844#  7855#  7870#  7884#  7895#
                7913#  7929#  7939#  7952#  7965#  7985#  7996#  8011#  8025#  8036#  8107#  8126#  8135#  8149#  8158#
                8177#  8191#  8201#  8237#  8259#  8271#  8285#  8296#  8310#  8329#  8340#  8351#  8360#  8420#  8439#
                8451#  8460#  8478#  8498#  8516#  8549#  8587#  8609#  8619#  8629#  8639#  8648#  8661#  8671#  8681#
                8691#  8700#  8711#  8725#  8736#  8748#  8802#  8824#  8834#  8844#  8854#  8863#  8876#  8886#  8896#
                8906#  8916#  8926#  8936#  8945#  8956#  8970#  8981#  8993#  9036#  9046#  9059#  9069#  9079#  9089#
                9099#  9108#  9119#  9132#  9144#  9156#  9166#  9175#
M$GNGB             1#  1300#  1303#  1327#  1336#  1338#  1340#  1342#  1344#  1346#  1348#  1350#  1352#  1354#  1356#
                1358#  1360#  1362#  1364#  1366#  1368#  1370#  1373#  1376#  1378#  1380#  1382#  1384#  1386#  1388#
                1390#  1392#  1394#  1396#  1398#  1400#  1402#  1404#  1406#  1408#  1410#  1425#  1426  1468#  1469
                1470  1493#  1494  1495  2022#  2383#  2395#  4102#  4141#  4168#  4203#  4248#  4293#  4328#  4547#
                4560#  4677#  4727#  4741#  4757#  4928#  4976#  9194#  9195  9257#  9258  9278#  9281
M$GNIN             1#  1300#  1327#  1328  1329  1330  1331  1332  1333#  1334#  1335#  1336#  1337  1338#  1339
                1340#  1341  1342#  1343  1344#  1345  1346#  1347  1348#  1349  1350#  1351  1352#  1353  1354#
                1355  1356#  1357  1358#  1359  1360#  1361  1362#  1363  1364#  1365  1366#  1367  1368#  1369
                1370#  1371  1372  1373#  1374  1375#  1376#  1377  1378#  1379  1380#  1381  1382#  1383  1384#
                1385  1386#  1387  1388#  1389  1390#  1391  1392#  1393  1394#  1395  1396#  1397  1398#  1399
                1400#  1401  1402#  1403  1404#  1405  1406#  1407  1408#  1409  1410#  1411  1425#  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#  1456#  1457#  1468#
                1493#  2383#  2384  2387  2395#  2396  2403  2441  2442  2443  2444  2485  2486  2487  2488
                2532  2533  2534  2535  2603  2604  2605  2606  2702  2703  2704  2705  2975  2976  2977
                2978  3079  3080  3081  3082  3112  3113  3114  3115  3123  3124  3125  3126  3157  3158
                3159  3160  3168  3169  3170  3171  3202  3203  3204  3205  3213  3214  3215  3216  3247
                3248  3249  3250  3258  3259  3260  3261  3292  3293  3294  3295  3303  3304  3305  3306
                3335  3336  3337  3338  3346  3347  3348  3349  3382  3383  3384  3385  3393  3394  3395
                3396  3406  3407  3408  3409  3417  3418  3419  3420  3449  3450  3451  3452  3461  3462
                3463  3464  3510  3511  3512  3513  3757  3758  3759  3760  3770  3771  3772  3773  3796
                3797  3798  3799  3806  3807  3808  3809  3819  3820  3821  3822  3829  3830  3831  3832
                3842  3843  3844  3845  3852  3853  3854  3855  3865  3866  3867  3868  3875  3876  3877
                3878  3888  3889  3890  3891  3898  3899  3900  3901  4109#  4110#  4111#  4112  4113#  4114
                4124#  4125#  4126#  4127#  4128  4129#  4130  4136#  4143#  4144#  4145#  4146#  4147  4148#  4149
                4152#  4153#  4154#  4155#  4156#  4157  4158#  4159  4163#  4172#  4173#  4174#  4175#  4176  4177#
                4178  4181#  4182#  4183#  4184#  4185#  4186  4187#  4188  4190#  4191#  4192  4193#  4194  4197#
```

42050	42060	42070	42080	4209	42100	4211	42130	42140	4215	42160	4217	42210	42220	42230
42240	4225	42260	4227	42300	42310	42320	42330	42340	4235	42360	4237	42410	42500	42510
42520	42530	4254	42550	4256	42580	42590	4260	42610	4262	42660	42670	42680	42690	4270
42710	4272	42750	42760	42770	42780	42790	4280	42810	4282	42860	42950	42960	42970	42980
4299	43000	4301	43030	43040	4305	43060	4307	43110	43120	43130	43140	4315	43160	4317
43210	43300	43310	43320	43330	4334	43350	4336	43390	43400	43410	43420	43430	4344	43450
4346	43500	43760	43770	43780	43790	4380	43810	4382	43840	43850	43860	43870	43880	43890
4390	43910	4392	43940	43950	43960	4397	43980	4399	44010	44020	44030	44040	44050	44060
4407	44080	4409	44110	44120	4413	44140	4415	44250	44260	44270	44280	4429	44300	4431
44330	44340	44350	44360	44370	44380	4439	44400	4441	44430	44440	44450	4446	44470	4448
44500	44510	44520	44530	44540	44550	4456	44570	4458	44600	44610	44620	4463	44640	4465
44670	44680	44690	44700	44710	44720	4473	44740	4475	44770	44780	44790	4480	44810	4482
44840	44850	44860	44870	44880	44890	4490	44910	4492	45020	45030	45040	45050	4506	45070
4508	45100	45110	45120	45130	45140	45150	4516	45170	4518	45200	45210	45220	4523	45240
4525	45270	45280	45290	45300	45310	45320	4533	45340	4535	45800	45810	45830	45870	45880
45900	45940	45950	45970	46010	46020	46040	46220	46230	46240	46260	46520	46590	46800	46810
46820	46830	46840	4685	47020	47030	47070	47080	47140	47320	47440	47470	47600	47820	47840
47850	47910	47930	47940	48100	48110	48120	48130	48150	48160	48290	48300	48310	48320	48360
48630	48790	48800	48810	48820	48840	48890	48930	49010	49030	49040	49150	49160	49170	49180
49210	49250	49300	49310	4932	49330	4934	49360	49370	4938	49390	4940	49420	49430	4944
49450	4946	49480	49490	49500	49510	4952	49530	4954	49580	49590	49600	49610	4962	49630
4964	49660	49670	4968	49690	4970	49730	49780	49790	49800	49810	4982	49830	4984	49880
50440	50450	50460	50470	50490	50500	50560	50960	50980	50990	51040	51590	51600	51610	51620
51640	51650	51760	51960	51980	51990	52070	52150	52170	52180	52220	52280	52480	52500	52510
52590	52670	52690	52700	52740	52800	53020	53040	53050	53130	53210	53230	53240	53280	53340
53620	53640	53650	53730	53930	53940	53950	53960	53980	53990	54030	54090	54280	54300	54310
54390	54470	54490	54500	54540	54600	54800	54820	54830	55020	55220	55240	55250	55440	55980
55990	56000	56010	56030	56040	56240	56250	56260	56270	56290	56300	56500	56510	56520	56530
56570	56950	56970	56980	57160	57180	57190	57270	57290	57300	57350	57370	57380	57420	57820
57840	57850	57920	57940	57950	58020	58040	58050	58120	58140	58150	58210	58230	58240	58320
58340	58350	58400	58420	58430	58460	58800	58820	58830	58890	58910	58920	58990	59010	59020
59170	59190	59200	59280	59300	59310	59400	59410	59420	59430	59450	59460	59550	59570	59580
59670	59690	59700	59790	59800	59810	59820	59840	59850	59930	59950	59960	60030	60050	60060
60190	60210	60220	60290	60310	60320	60440	60460	60470	60560	60570	60580	60590	60610	60620
60710	60730	60740	60770	61080	61100	61110	61170	61190	61200	61270	61290	61300	61450	61470
61480	61560	61580	61590	61680	61690	61700	61710	61730	61740	61830	61850	61860	61950	61970
61980	62070	62080	62090	62100	62120	62130	62250	62270	62280	62350	62370	62380	62480	62500
62510	62560	62580	62590	62620	62900	62920	62930	63080	63100	63110	63180	63200	63210	63340
63360	63370	63450	63470	63480	63560	63580	63590	63620	63920	63940	63950	64140	64160	64170
64370	64390	64400	64540	64560	64570	64650	64670	64680	64740	64760	64770	64800	65180	65240
65260	65270	65440	65460	65470	65620	65640	65650	65720	65780	65840	65860	65870	66040	66060
66070	66220	66240	66250	66320	66380	66440	66460	66470	66640	66660	66670	66820	66840	66850
66920	66950	67380	67400	67410	67470	67490	67500	67570	67590	67600	67750	67770	67780	67890
67910	67920	67990	68010	68020	68050	68350	68370	68380	68440	68460	68470	68540	68560	68570
68720	68740	68750	68860	68880	68890	68960	68980	68990	69020	69320	69340	69350	69410	69430
69440	69510	69530	69540	69690	69710	69720	69830	69850	69860	69930	69950	69960	69990	70260
70280	70290	70440	70460	70470	70540	70560	70570	70640	70660	70670	70740	70760	70770	70830
70850	70860	70940	70960	70970	71040	71060	71070	71150	71170	71180	71250	71270	71280	71360
71380	71390	71470	71490	71500	71580	71600	71610	71640	72000	72060	72080	72090	72220	72240
72250	72320	72340	72350	72450	72470	72480	72580	72600	72610	72780	72800	72810	72890	72910
72920	73040	73060	73070	73180	73200	73210	73290	73310	73320	73350	73410	73470	73490	73500
73630	73650	73660	73730	73750	73760	73860	73880	73890	73990	74010	74020	74190	74210	74220
74300	74320	74330	74450	74470	74480	74590	74610	74620	74700	74720	74730	74760	74820	74880
74900	74910	75040	75060	75070	75140	75160	75170	75270	75296	75300	75400	75420	75430	75600
75620	75630	75710	75730	75740	75860	75880	75890	76000	76020	76030	76110	76130	76140	76170
76230	76290	76310	76320	76450	76470	76480	76550	76570	76580	76680	76700	76710	76810	76830

```
          7684#  7701#  7703#  7704#  7712#  7714#  7715#  7727#  7729#  7730#  7741#  7743#  7744#  7752#  7754#
          7755#  7758#  7764#  7770#  7772#  7773#  7786#  7788#  7789#  7796#  7798#  7799#  7809#  7811#  7812#
          7822#  7824#  7825#  7842#  7844#  7845#  7853#  7855#  7856#  7868#  7870#  7871#  7882#  7884#  7885#
          7893#  7895#  7896#  7899#  7905#  7911#  7913#  7914#  7927#  7929#  7930#  7937#  7939#  7940#  7950#
          7952#  7953#  7963#  7965#  7966#  7983#  7985#  7986#  7994#  7996#  7997#  8009#  8011#  8012#  8023#
          8025#  8026#  8034#  8036#  8037#  8040#  8043#  8105#  8107#  8108#  8124#  8126#  8127#  8133#  8135#
          8136#  8147#  8149#  8150#  8156#  8158#  8159#  8175#  8177#  8178#  8189#  8191#  8192#  8199#  8201#
          8202#  8205#  8235#  8237#  8238#  8257#  8259#  8260#  8269#  8271#  8272#  8283#  8285#  8286#  8294#
          8296#  8297#  8308#  8310#  8311#  8327#  8329#  8330#  8338#  8340#  8341#  8349#  8351#  8352#  8358#
          8360#  8361#  8364#  8410#  8418#  8420#  8421#  8437#  8439#  8440#  8449#  8451#  8452#  8458#  8460#
          8461#  8464#  8468#  8476#  8478#  8479#  8496#  8498#  8499#  8502#  8506#  8514#  8516#  8517#  8547#
          8549#  8550#  8553#  8556#  8585#  8587#  8588#  8607#  8609#  8610#  8617#  8619#  8620#  8627#  8629#
          8630#  8637#  8639#  8640#  8646#  8648#  8649#  8659#  8661#  8662#  8669#  8671#  8672#  8679#  8681#
          8682#  8689#  8691#  8692#  8698#  8700#  8701#  8709#  8711#  8712#  8723#  8725#  8726#  8734#  8736#
          8737#  8746#  8748#  8749#  8753#  8800#  8802#  8803#  8822#  8824#  8825#  8832#  8834#  8835#  8842#
          8844#  8845#  8852#  8854#  8855#  8861#  8863#  8864#  8874#  8876#  8877#  8884#  8886#  8887#  8894#
          8896#  8897#  8904#  8906#  8907#  8914#  8916#  8917#  8924#  8926#  8927#  8934#  8936#  8937#  8943#
          8945#  8946#  8954#  8956#  8957#  8968#  8970#  8971#  8979#  8981#  8982#  8991#  8993#  8994#  8997#
          9034#  9036#  9037#  9044#  9046#  9047#  9057#  9059#  9060#  9067#  9069#  9070#  9077#  9079#  9080#
          9087#  9089#  9090#  9097#  9099#  9100#  9106#  9108#  9109#  9117#  9119#  9120#  9130#  9132#  9133#
          9142#  9144#  9145#  9154#  9156#  9157#  9164#  9166#  9167#  9173#  9175#  9176#  9179#  9194#  9198#
          9199   9200   9201   9203#  9204   9205   9206   9208#  9209   9210   9211   9212   9214#  9215   9216
          9217   9218   9220#  9221   9222   9223   9224   9226#  9227   9228   9229   9230   9232#  9233   9234
          9235   9236   9239#  9257#  9261#  9278#  9279#  9280#
M$GNLS       1#  1300#
M$GNSU       1#  1300#  4862#  4892#  5206#  5258#  5312#  5372#  5438#  6517#  6577#  6637#  7199#  7340#  7481#
          7622#  7763#  7904#  8409#  8467#  8505#
M$GNTA       1#  1300#  1484#  1498#  4135#  4162#  4196#  4240#  4285#  4320#  4349#  4658#  4713#  4731#  4746#
          4759#  4835#  4888#  4920#  4924#  4972#  4987#  5055#  5103#  5175#  5221#  5227#  5273#  5279#  5327#
          5333#  5402#  5408#  5453#  5459#  5501#  5543#  5656#  5741#  5845#  6076#  6261#  6361#  6479#  6571#
          6631#  6691#  6694#  6804#  6901#  6998#  7163#  7334#  7475#  7616#  7757#  7898#  8039#  8042#  8204#
          8363#  8463#  8501#  8552#  8555#  8752#  8996#  9178#  9239#  9240   9261#  9262
M$GNTE       1#  1300#  4777#  4858#  5015#  5080#  5126#  5192#  5244#  5298#  5358#  5424#  5476#  5518#  5571#
          5686#  5773#  5872#  6100#  6281#  6384#  6511#  6730#  6827#  6924#  7018#  7193#  8068#  8227#  8406#
          8576#  8792#  9026#
M$HAPT       1#  1300#  1327#
M$HNAP       1#  1300#  1327#  1366
M$INCR       1#  1300#  1303#  1468#  1493#  4102#  4113#  4129#  4136#  4141#  4148#  4158#  4163#  4168#  4177#
          4187#  4193#  4197#  4203#  4210#  4216#  4226#  4236#  4241#  4248#  4255#  4261#  4271#  4281#  4286#
          4293#  4300#  4306#  4316#  4321#  4328#  4335#  4345#  4350#  4381#  4391#  4398#  4408#  4414#  4430#
          4440#  4447#  4457#  4464#  4474#  4481#  4491#  4507#  4517#  4524#  4534#  4547#  4560#  4581#  4588#
          4595#  4602#  4623#  4652#  4659#  4677#  4684#  4703#  4708#  4714#  4727#  4732#  4741#  4744#  4747#
          4757#  4760#  4777#  4778#  4782#  4784#  4791#  4793#  4810#  4815#  4829#  4836#  4858#  4859#  4862#
          4863#  4879#  4884#  4889#  4892#  4893#  4901#  4903#  4915#  4921#  4925#  4928#  4933#  4939#  4945#
          4953#  4963#  4969#  4973#  4976#  4983#  4988#  5015#  5016#  5044#  5049#  5056#  5080#  5081#  5096#
          5098#  5104#  5126#  5127#  5159#  5164#  5176#  5192#  5193#  5196#  5198#  5206#  5207#  5215#  5217#
          5222#  5228#  5244#  5245#  5248#  5250#  5258#  5259#  5267#  5269#  5274#  5280#  5298#  5299#  5302#
          5304#  5312#  5313#  5321#  5323#  5328#  5334#  5358#  5359#  5362#  5364#  5372#  5373#  5393#  5398#
          5403#  5409#  5424#  5425#  5428#  5430#  5438#  5439#  5447#  5449#  5454#  5460#  5476#  5477#  5480#
          5482#  5502#  5518#  5519#  5522#  5524#  5544#  5571#  5572#  5598#  5603#  5624#  5629#  5650#  5657#
          5686#  5687#  5695#  5697#  5716#  5718#  5727#  5729#  5735#  5737#  5742#  5773#  5774#  5782#  5784#
          5792#  5794#  5802#  5804#  5812#  5814#  5821#  5823#  5832#  5834#  5840#  5842#  5846#  5872#  5873#
          5880#  5882#  5889#  5891#  5899#  5901#  5917#  5919#  5928#  5930#  5940#  5945#  5955#  5957#  5967#
          5969#  5979#  5984#  5993#  5995#  6003#  6005#  6019#  6021#  6029#  6031#  6044#  6046#  6056#  6061#
          6071#  6073#  6077#  6100#  6101#  6108#  6110#  6117#  6119#  6127#  6129#  6145#  6147#  6156#  6158#
          6168#  6173#  6183#  6185#  6195#  6197#  6207#  6212#  6225#  6227#  6235#  6237#  6248#  6250#  6256#
```

```
                6258#  6262#  6281#  6282#  6290#  6292#  6308#  6310#  6318#  6320#  6334#  6336#  6345#  6347#  6356#
                6358#  6362#  6384#  6385#  6392#  6394#  6414#  6416#  6437#  6439#  6454#  6456#  6465#  6467#  6474#
                6476#  6480#  6511#  6512#  6517#  6518#  6524#  6526#  6544#  6546#  6562#  6564#  6572#  6577#  6578#
                6584#  6586#  6604#  6606#  6622#  6624#  6632#  6637#  6638#  6644#  6646#  6664#  6666#  6682#  6684#
                6692#  6695#  6730#  6731#  6738#  6740#  6747#  6749#  6757#  6759#  6775#  6777#  6789#  6791#  6799#
                6801#  6805#  6827#  6828#  6835#  6837#  6844#  6846#  6854#  6856#  6872#  6874#  6886#  6888#  6896#
                6898#  6902#  6924#  6925#  6932#  6934#  6941#  6943#  6951#  6953#  6969#  6971#  6983#  6985#  6993#
                6995#  6999#  7018#  7019#  7026#  7028#  7044#  7046#  7054#  7056#  7064#  7066#  7074#  7076#  7083#
                7085#  7094#  7096#  7104#  7106#  7115#  7117#  7125#  7127#  7136#  7138#  7147#  7149#  7158#  7160#
                7164#  7193#  7194#  7199#  7200#  7206#  7208#  7222#  7224#  7232#  7234#  7245#  7247#  7258#  7260#
                7278#  7280#  7289#  7291#  7304#  7306#  7318#  7320#  7329#  7331#  7335#  7340#  7341#  7347#  7349#
                7363#  7365#  7373#  7375#  7386#  7388#  7399#  7401#  7419#  7421#  7430#  7432#  7445#  7447#  7459#
                7461#  7470#  7472#  7476#  7481#  7482#  7488#  7490#  7504#  7506#  7514#  7516#  7527#  7529#  7540#
                7542#  7560#  7562#  7571#  7573#  7586#  7588#  7600#  7602#  7611#  7613#  7617#  7622#  7623#  7629#
                7631#  7645#  7647#  7655#  7657#  7668#  7670#  7681#  7683#  7701#  7703#  7712#  7714#  7727#  7729#
                7741#  7743#  7752#  7754#  7758#  7763#  7764#  7770#  7772#  7786#  7788#  7796#  7798#  7809#  7811#
                7822#  7824#  7842#  7844#  7853#  7855#  7868#  7870#  7882#  7884#  7893#  7895#  7899#  7904#  7905#
                7911#  7913#  7927#  7929#  7937#  7939#  7950#  7952#  7963#  7965#  7983#  7985#  7994#  7996#  8009#
                8011#  8023#  8025#  8034#  8036#  8040#  8043#  8068#  8069#  8105#  8107#  8124#  8126#  8133#  8135#
                8147#  8149#  8156#  8158#  8175#  8177#  8189#  8191#  8199#  8201#  8205#  8227#  8228#  8235#  8237#
                8257#  8259#  8269#  8271#  8283#  8285#  8294#  8296#  8308#  8310#  8327#  8329#  8338#  8340#  8349#
                8351#  8358#  8360#  8364#  8406#  8407#  8409#  8410#  8418#  8420#  8437#  8439#  8449#  8451#  8458#
                8460#  8464#  8467#  8468#  8476#  8478#  8496#  8498#  8502#  8505#  8506#  8514#  8516#  8547#  8549#
                8553#  8556#  8576#  8577#  8585#  8587#  8607#  8609#  8617#  8619#  8627#  8629#  8637#  8639#  8646#
                8648#  8659#  8661#  8669#  8671#  8679#  8681#  8689#  8691#  8698#  8700#  8709#  8711#  8723#  8725#
                8734#  8736#  8746#  8748#  8753#  8792#  8793#  8800#  8802#  8822#  8824#  8832#  8834#  8842#  8844#
                8852#  8854#  8861#  8863#  8874#  8876#  8884#  8886#  8894#  8896#  8904#  8906#  8914#  8916#  8924#
                8926#  8934#  8936#  8943#  8945#  8954#  8956#  8968#  8970#  8979#  8981#  8991#  8993#  8997#  9026#
                9027#  9034#  9036#  9044#  9046#  9057#  9059#  9067#  9069#  9077#  9079#  9087#  9089#  9097#  9099#
                9106#  9108#  9117#  9119#  9130#  9132#  9142#  9144#  9154#  9156#  9164#  9166#  9173#  9175#  9179#
                9194#  9257#
M$IOSE             1#  1300#
M$LDRO             1#  1300#  4580#  4587#  4594#  4601#  4622#  4702#  4707#
M$MASK             1#  1300#
M$MCHI             1#  1300#
M$MCLO             1#  1300#
M$MSK1             1#  1300#
M$POP              1#  1300#  1484#  1498#  4135#  4162#  4196#  4240#  4285#  4320#  4349#  4552#  4658#  4713#  4731#
                4746#  4759#  4835#  4888#  4920#  4924#  4972#  4987#  5055#  5103#  5175#  5221#  5227#  5273#  5279#
                5327#  5333#  5402#  5408#  5453#  5459#  5501#  5543#  5656#  5741#  5845#  6076#  6261#  6361#  6479#
                6571#  6631#  6691#  6694#  6804#  6901#  6998#  7163#  7334#  7475#  7616#  7757#  7898#  8039#  8042#
                8204#  8363#  8463#  8501#  8552#  8555#  8752#  8996#  9178#  9239#  9261#  9275#
M$PRIN             1#  1300#  4109#  4124#  4143#  4152#  4172#  4181#  4190#  4205#  4213#  4221#  4230#  4250#  4258#
                4266#  4275#  4295#  4303#  4311#  4330#  4339#  4376#  4384#  4394#  4401#  4411#  4425#  4433#  4443#
                4450#  4460#  4467#  4477#  4484#  4502#  4510#  4520#  4527#  4930#  4936#  4942#  4948#  4958#  4966#
                4978#
M$PUSH             1#  1300#  1303#  1468#  1493#  4102#  4141#  4168#  4203#  4248#  4293#  4328#  4547#  4560#  4677#
                4727#  4741#  4757#  4777#  4778   4858#  4859   4862#  4863   4892#  4893   4928#  4976#  5015#  5016
                5080#  5081   5126#  5127   5192#  5193   5206#  5207   5244#  5245   5258#  5259   5298#  5299   5312#
                5313   5358#  5359   5372#  5373   5424#  5425   5438#  5439   5476#  5477   5518#  5519   5571#  5572
                5686#  5687   5773#  5774   5872#  5873   6100#  6101   6281#  6282   6384#  5385   6511#  6512   6517#
                6518   6577#  6578   6637#  6638   6730#  6731   6827#  6828   6924#  6925   7018#  7019   7193#  7194
                7199#  7200   7340#  7341   7481#  7482   7622#  7623   7763#  7764   7904#  7905   8068#  8069   8227#
                8228   8406#  8407   8409#  8410   8467#  8468   8505#  8506   8576#  8577   8792#  8793   9026#  9027
                9194#  9257#
M$PUT              1#  1300#  4109#  4124#  4143#  4152#  4172#  4181#  4190#  4205#  4213#  4221#  4230#  4250#  4258#
```

```
                4266#  4275#  4295#  4303#  4311#  4330#  4339#  4376#  4384#  4394#  4401#  4411#  4425#  4433#  4443#
                4450#  4460#  4467#  4477#  4484#  4502#  4510#  4520#  4527#  4680#  4930#  4936#  4942#  4948#  4958#
                4966#  4978#
M$PUT1             1#  1300#  4109#  4110   4111   4124#  4125   4126   4127   4143#  4144   4145   4146   4152#  4153
                4154   4155   4156   4172#  4173   4174   4175   4181#  4182   4183   4184   4185   4190#  4191   4205#
                4206   4207   4208   4213#  4214   4221#  4222   4223   4224   4230#  4231   4232   4233   4234   4250#
                4251   4252   4253   4258#  4259   4266#  4267   4268   4269   4275#  4276   4277   4278   4279   4295#
                4296   4297   4298   4303#  4304   4311#  4312   4313   4314   4330#  4331   4332   4333   4339#  4340
                4341   4342   4343   4376#  4377   4378   4379   4384#  4385   4386   4387   4388   4389   4394#  4395
                4396   4401#  4402   4403   4404   4405   4406   4411#  4412   4425#  4426   4427   4428   4433#  4434
                4435   4436   4437   4438   4443#  4444   4445   4450#  4451   4452   4453   4454   4455   4460#  4461
                4462   4467#  4468   4469   4470   4471   4472   4477#  4478   4479   4484#  4485   4486   4487   4488
                4489   4502#  4503   4504   4505   4510#  4511   4512   4513   4514   4515   4520#  4521   4522   4527#
                4528   4529   4530   4531   4532   4680#  4681   4682   4683   4930#  4931   4936#  4937   4942#  4943
                4948#  4949   4950   4951   4958#  4959   4960   4961   4966#  4967   4978#  4979   4980   4981
M$RADI             1#  1300#  9198#  9203#  9208#  9214#  9220#  9226#  9232#
M$RBRO             1#  1300#
M$RNRO             1#  1300#  4622#  4624
M$SETS             1#  1300#  1303#  1468#  1493#  4102#  4141#  4168#  4203#  4248#  4293#  4328#  4547#  4560#  4677#
                4727#  4741#  4757#  4778#  4859#  4863#  4893#  4928#  4976#  5016#  5081#  5127#  5193#  5207#  5245#
                5259#  5299#  5313#  5359#  5373#  5425#  5439#  5477#  5519#  5572#  5687#  5774#  5873#  6101#  6282#
                6385#  6512#  6518#  6578#  6638#  6731#  6828#  6925#  7019#  7194#  7200#  7341#  7482#  7623#  7764#
                7905#  8069#  8228#  8407#  8410#  8468#  8506#  8577#  8793#  9027#  9194#  9257#
M$STAR             1#  1300#
M$SVC              1#  1300#  4109#  4113   4124#  4129   4135#  4136   4143#  4148   4152#  4158   4162#  4163   4172#
                4177   4181#  4187   4190#  4193   4196#  4197   4205#  4210   4213#  4216   4221#  4226   4230#  4236
                4240#  4241   4250#  4255   4258#  4261   4266#  4271   4275#  4281   4285#  4286   4295#  4300   4303#
                4306   4311#  4316   4320#  4321   4330#  4335   4339#  4345   4349#  4350   4376#  4381   4384#  4391
                4394#  4398   4401#  4408   4411#  4414   4425#  4430   4433#  444^   4443#  4447   4450#  4457   4460#
                4464   4467#  4474   4477#  4481   4484#  4491   4502#  4507   4510#  4517   4520#  4524   4527#  4534
                4580#  4581   4587#  4588   4594#  4595   4601#  4602   4622#  4623   4652#  4658#  4659   4680#  4684
                4702#  4703   4707#  4708   4713#  4714   4731#  4732   4744#  4746#  4747   4759#  4760   4782#  4784#
                4791#  4793#  4810   4815#  4829   4835#  4836   4862#  4863   4879   4884#  4888#  4889   4892#  4893
                4901#  4903#  4915   4920#  4921   4924#  4925   4930#  4933   4936#  4939   4942#  4945   4948#  4953
                4958#  4963   4966#  4969   4972#  4973   4978#  4983   4987#  4988   5044   5049#  5055#  5056   5096#
                5098#  5103#  5104   5159   5164#  5175#  5176   5196#  5198#  5206#  5207   5215#  5217#  5221#  5222
                5227#  5228   5248#  5250#  5258#  5259   5267#  5269#  5273#  5274   5279#  5280   5302#  5304#  5312#
                5313   5321#  5323#  5327#  5328   5333#  5334   5362#  5364#  5372#  5373   5393   5398#  5402#  5403
                5408#  5409   5428#  5430#  5438#  5439   5447#  5449#  5453#  5454   5459#  5460   5480#  5482#  5501#
                5552   5522#  5524#  5543#  5544   5598   5603#  5624   5629#  5650   5656#  5657   5695#  5697#  5716#
                5718#  5727#  5729#  5735#  5737#  5741#  5742   5782#  5784#  5792#  5794#  5802#  5804#  5812#  5814#
                5821#  5823#  5832#  5834#  5840#  5842#  5845#  5846   5880#  5882#  5898#  5891#  5899#  5901#  5917#
                5919#  5928#  5930#  5940   5945#  5955#  5957#  5967#  5969#  5979   5994#  5993#  5995#  6003#  6005#
                6019#  6021#  6029#  6031#  6044#  6046#  6056   6061#  6071#  6073#  6076#  6077   6108#  6110#  6117#
                6119#  6127#  6129#  6145#  6147#  6156#  6158#  6168   6173#  6183#  6185#  6195#  6197#  6207   6212#
                6225#  6227#  6235#  6237#  6248#  6250#  6256#  6258#  6261#  6262   6290#  6292#  6308#  6310#  6318#
                6320#  6334#  6336#  6345#  6347#  6356#  6358#  6361#  6362   6392#  6394#  6414#  6416#  6437#  6439#
                6454#  6456#  6465#  6467#  6474#  6476#  6479#  6480   6517#  6518   6524#  6526#  6544#  6546#  6562#
                6564#  6571#  6572   6577#  6578   6584#  6586#  6604#  6606#  6622#  6624#  6631#  6632   6637#  6638
                6644#  6646#  6664#  6666#  6682#  6684#  6691#  6692   6694#  6695   6738#  6740#  6747#  6749#  6757#
                6759#  6775#  6777#  6789#  6791#  6799#  6801#  6804#  6805   6835#  6837#  6844#  6846#  6854#  6856#
                6872#  6874#  6886#  6888#  6896#  6898#  6901#  6902   6932#  6934#  6941#  6943#  6951#  6953#  6969#
                6971#  6983#  6985#  6993#  6995#  6998#  6999   7026#  7028#  7044#  7046#  7054#  7056#  7064#  7066#
                7074#  7076#  7083#  7085#  7094#  7096#  7104#  7106#  7115#  7117#  7125#  7127#  7136#  7138#  7147#
                7149#  7158#  7160#  7163#  7164   7199#  7200   7206#  7208#  7222#  7224#  7232#  7234#  7245#  7247#
                7258#  7260#  7278#  7280#  7289#  7291#  7304#  7306#  7318#  7320#  7329#  7331#  7334#  7335   7340#
```

7341 7347# 7349# 7363# 7365# 7373# 7375# 7386# 7388# 7399# 7401# 7419# 7421# 7430# 7432#
7445# 7447# 7459# 7461# 7470# 7472# 7475# 7476 7481# 7482 7488# 7490# 7504# 7506# 7514#
7516# 7527# 7529# 7540# 7542# 7560# 7562# 7571# 7573# 7586# 7588# 7600# 7602# 7611# 7613#
7616# 7617 7622# 7623 7629# 7631# 7645# 7647# 7655# 7657# 7668# 7670# 7681# 7683# 7701#
7703# 7712# 7714# 7727# 7729# 7741# 7743# 7752# 7754# 7757# 7758 7763# 7764 7770# 7772#
7786# 7788# 7796# 7798# 7809# 7811# 7822# 7824# 7842# 7844# 7853# 7855# 7868# 7870# 7882#
7884# 7893# 7895# 7898# 7899 7904# 7905 7911# 7913# 7927# 7929# 7937# 7939# 7950# 7952#
7963# 7965# 7983# 7985# 7994# 7996# 8009# 8011# 8023# 8025# 8034# 8036# 8039# 8040 8042#
8043 8105# 8107# 8124# 8126# 8133# 8135# 8147# 8149# 8156# 8158# 8175# 8177# 8189# 8191#
8199# 8201# 8204# 8205 8235# 8237# 8257# 8259# 8269# 8271# 8283# 8285# 8294# 8296# 8308#
8310# 8327# 8329# 8338# 8340# 8349# 8351# 8358# 8360# 8363# 8364 8409# 8410 8418# 8420#
8437# 8439# 8449# 8451# 8458# 8460# 8463# 8464 8467# 8468 8476# 8478# 8496# 8498# 8501#
8502 8505# 8506 8514# 8516# 8547# 8549# 8552# 8553 8555# 8556 8585# 8587# 8607# 8609#
8617# 8619# 8627# 8629# 8637# 8639# 8646# 8648# 8659# 8661# 8669# 8671# 8679# 8681# 8689#
8691# 8698# 8700# 8709# 8711# 8723# 8725# 8734# 8736# 8746# 8748# 8752# 8753 8800# 8802#
8822# 8824# 8832# 8834# 8842# 8844# 8852# 8854# 8861# 8863# 8874# 8876# 8884# 8886# 8894#
8896# 8904# 8906# 8914# 8916# 8924# 8926# 8934# 8936# 8943# 8945# 8954# 8956# 8968# 8970#
8979# 8981# 8991# 8993# 8996# 8997 9034# 9036# 9044# 9046# 9057# 9059# 9067# 9069# 9077#
9079# 9087# 9089# 9097# 9099# 9106# 9108# 9117# 9119# 9130# 9132# 9142# 9144# 9154# 9156#
9164# 9166# 9173# 9175# 9178# 9179
M$TLAB 1# 1300# 4113# 4129# 4136# 4148# 4158# 4163# 4177# 4187# 4193# 4197# 4210# 4216# 4226#
4236# 4241# 4255# 4261# 4271# 4281# 4286# 4300# 4306# 4316# 4321# 4335# 4345# 4350# 4381#
4391# 4398# 4408# 4414# 4430# 4440# 4447# 4457# 4464# 4474# 4481# 4491# 4507# 4517# 4524#
4534# 4581# 4588# 4595# 4602# 4623# 4652# 4659# 4684# 4703# 4708# 4714# 4732# 4744# 4747#
4760# 4782# 4784# 4791# 4793# 4810# 4815# 4829# 4836# 4863# 4879# 4884# 4889# 4893# 4901#
4903# 4915# 4921# 4925# 4933# 4939# 4945# 4953# 4963# 4969# 4973# 4983# 4988# 5044# 5049#
5056# 5096# 5098# 5104# 5159# 5164# 5176# 5196# 5198# 5207# 5215# 5217# 5222# 5228# 5248#
5250# 5259# 5267# 5269# 5274# 5280# 5302# 5304# 5313# 5321# 5323# 5328# 5334# 5362# 5364#
5373# 5393# 5398# 5403# 5409# 5428# 5430# 5439# 5447# 5449# 5454# 5460# 5480# 5482# 5502#
5522# 5524# 5544# 5598# 5603# 5624# 5629# 5650# 5657# 5695# 5697# 5716# 5718# 5727# 5729#
5735# 5737# 5742# 5782# 5784# 5792# 5794# 5802# 5804# 5812# 5814# 5821# 5823# 5832# 5834#
5840# 5842# 5846# 5880# 5882# 5889# 5891# 5899# 5901# 5917# 5919# 5928# 5930# 5940# 5945#
5955# 5957# 5967# 5969# 5979# 5984# 5993# 5995# 6003# 6005# 6019# 6021# 6029# 6031# 6044#
6046# 6056# 6061# 6071# 6073# 6077# 6108# 6110# 6117# 6119# 6127# 6129# 6145# 6147# 6156#
6158# 6168# 6173# 6183# 6185# 6195# 6197# 6207# 6212# 6225# 6227# 6235# 6237# 6248# 6250#
6256# 6258# 6262# 6290# 6292# 6308# 6310# 6318# 6320# 6334# 6336# 6345# 6347# 6356# 6358#
6362# 6392# 6394# 6414# 6416# 6437# 6439# 6454# 6456# 6465# 6467# 6474# 6476# 6480# 6518#
6524# 6526# 6544# 6546# 6562# 6564# 6572# 6578# 6584# 6586# 6604# 6606# 6622# 6624# 6632#
6638# 6644# 6646# 6664# 6666# 6682# 6684# 6692# 6695# 6738# 6740# 6747# 6749# 6757# 6759#
6775# 6777# 6789# 6791# 6799# 6801# 6805# 6835# 6837# 6844# 6846# 6854# 6856# 6872# 6874#
6886# 6888# 6896# 6898# 6902# 6932# 6934# 6941# 6943# 6951# 6953# 6969# 6971# 6983# 6985#
6993# 6995# 6999# 7026# 7028# 7044# 7046# 7054# 7056# 7064# 7066# 7074# 7076# 7083# 7085#
7094# 7096# 7104# 7106# 7115# 7117# 7125# 7127# 7136# 7138# 7147# 7149# 7158# 7160# 7164#
7200# 7206# 7208# 7222# 7224# 7232# 7234# 7245# 7247# 7258# 7260# 7278# 7280# 7289# 7291#
7304# 7306# 7318# 7320# 7329# 7331# 7335# 7341# 7347# 7349# 7363# 7365# 7373# 7375# 7386#
7388# 7399# 7401# 7419# 7421# 7430# 7432# 7445# 7447# 7459# 7461# 7470# 7472# 7476# 7482#
7488# 7490# 7504# 7506# 7514# 7516# 7527# 7529# 7540# 7542# 7560# 7562# 7571# 7573# 7586#
7588# 7600# 7602# 7611# 7613# 7617# 7623# 7629# 7631# 7645# 7647# 7655# 7657# 7668# 7670#
7681# 7683# 7701# 7703# 7712# 7714# 7727# 7729# 7741# 7743# 7752# 7754# 7758# 7764# 7770#
7772# 7786# 7788# 7796# 7798# 7809# 7811# 7822# 7824# 7842# 7844# 7853# 7855# 7868# 7870#
7882# 7884# 7893# 7895# 7899# 7905# 7911# 7913# 7927# 7929# 7937# 7939# 7950# 7952# 7963#
7965# 7983# 7985# 7994# 7996# 8009# 8011# 8023# 8025# 8034# 8036# 8040# 8043# 8105# 8107#
8124# 8126# 8133# 8135# 8147# 8149# 8156# 8158# 8175# 8177# 8189# 8191# 8199# 8201# 8205#
8235# 8237# 8257# 8259# 8269# 8271# 8283# 8285# 8294# 8296# 8308# 8310# 8327# 8329# 8338#
8340# 8349# 8351# 8358# 8360# 8364# 8410# 8418# 8420# 8437# 8439# 8449# 8451# 8458# 8460#
8464# 8468# 8476# 8478# 8496# 8498# 8502# 8506# 8514# 8516# 8547# 8549# 8553# 8556# 8585#

```
                8587#   8607#   8609#   8617#   8619#   8627#   8629#   8637#   8639#   8646#   8648#   8659#   8661#   8669#   8671#
                8679#   8681#   8689#   8691#   8698#   8700#   8709#   8711#   8723#   8725#   8734#   8736#   8746#   8748#   8753#
                8800#   8802#   8822#   8824#   8832#   8834#   8842#   8844#   8852#   8854#   8861#   8863#   8874#   8876#   8884#
                8886#   8894#   8896#   8904#   8906#   8914#   8916#   8924#   8926#   8934#   8936#   8943#   8945#   8954#   8956#
                8968#   8970#   8979#   8981#   8991#   8993#   8997#   9034#   9036#   9044#   9046#   9057#   9059#   9067#   9069#
                9077#   9079#   9087#   9089#   9097#   9099#   9106#   9108#   9117#   9119#   9130#   9132#   9142#   9144#   9154#
                9156#   9164#   9166#   9173#   9175#   9179#
M$TSTL             1#  13004   4113#   4129#   4136#   4148#   4158#   4163#   4177#   4187#   4193#   4197#   4210#   4216#   4226#
                4236#   4241#   4255#   4261#   4271#   4281#   4286#   4300#   4306#   4316#   4321#   4335#   4345#   4350#   4381#
                4391#   4398#   4408#   4414#   4430#   4440#   4447#   4457#   4464#   4474#   4481#   4491#   4507#   4517#   4524#
                4534#   4581#   4588#   4595#   4602#   4623#   4652#   4659#   4684#   4703#   4708#   4714#   4732#   4744#   4747#
                4760#   4782#   4784#   4791#   4793#   4810#   4815#   4829#   4836#   4863#   4879#   4884#   4889#   4893#   4901#
                4903#   4915#   4921#   4925#   4933#   4939#   4945#   4953#   4963#   4969#   4973#   4983#   4988#   5044#   5049#
                5056#   5096#   5098#   5104#   5159#   5164#   5176#   5196#   5198#   5207#   5215#   5217#   5222#   5228#   5248#
                5250#   5259#   5267#   5269#   5274#   5280#   5302#   5304#   5313#   5321#   5323#   5328#   5334#   5362#   5364#
                5373#   5393#   5398#   5403#   5409#   5428#   5430#   5439#   5447#   5449#   5454#   5460#   5480#   5482#   5502#
                5522#   5524#   5544#   5598#   5603#   5624#   5629#   5650#   5657#   5695#   5697#   5716#   5718#   5727#   5729#
                5735#   5737#   5742#   5782#   5784#   5792#   5794#   5802#   5804#   5812#   5814#   5821#   5823#   5832#   5834#
                5840#   5842#   5846#   5880#   5882#   5889#   5891#   5899#   5901#   5917#   5919#   5928#   5930#   5940#   5945#
                5955#   5957#   5967#   5969#   5979#   5984#   5993#   5995#   6003#   6005#   6019#   6021#   6029#   6031#   6044#
                6046#   6056#   6061#   6071#   6073#   6077#   6108#   6110#   6117#   6119#   6127#   6129#   6145#   6147#   6156#
                6158#   6168#   6173#   6183#   6185#   6195#   6197#   6207#   6212#   6225#   6227#   6235#   6237#   6248#   6250#
                6256#   6258#   6262#   6290#   6292#   6308#   6310#   6318#   6320#   6334#   6336#   6345#   6347#   6356#   6358#
                6362#   6392#   6394#   6414#   6416#   6437#   6439#   6454#   6456#   6465#   6467#   6474#   6476#   6480#   6518#
                6524#   6526#   6544#   6546#   6562#   6564#   6572#   6578#   6584#   6586#   6604#   6606#   6622#   6624#   6632#
                6638#   6644#   6646#   6664#   6666#   6682#   6684#   6692#   6695#   6738#   6740#   6747#   6749#   6757#   6759#
                6775#   6777#   6789#   6791#   6799#   6801#   6805#   6835#   6837#   6844#   6846#   6854#   6856#   6872#   6874#
                6886#   6888#   6896#   6898#   6902#   6932#   6934#   6941#   6943#   6951#   6953#   6969#   6971#   6983#   6985#
                6993#   6995#   6999#   7026#   7028#   7044#   7046#   7054#   7056#   7064#   7066#   7074#   7076#   7083#   7085#
                7094#   7096#   7104#   7106#   7115#   7117#   7125#   7127#   7136#   7138#   7147#   7149#   7158#   7160#   7164#
                7200#   7206#   7208#   7222#   7224#   7232#   7234#   7245#   7247#   7258#   7260#   7278#   7280#   7289#   7291#
                7304#   7306#   7318#   7320#   7329#   7331#   7335#   7341#   7347#   7349#   7363#   7365#   7373#   7375#   7386#
                7388#   7399#   7401#   7419#   7421#   7430#   7432#   7445#   7447#   7459#   7461#   7470#   7472#   7476#   7482#
                7488#   7490#   7504#   7506#   7514#   7516#   7527#   7529#   7540#   7542#   7560#   7562#   7571#   7573#   7586#
                7588#   7600#   7602#   7611#   7613#   7617#   7623#   7629#   7631#   7645#   7647#   7655#   7657#   7668#   7670#
                7681#   7683#   7701#   7703#   7712#   7714#   7727#   7729#   7741#   7743#   7752#   7754#   7758#   7764#   7770#
                7772#   7786#   7788#   7796#   7798#   7809#   7811#   7822#   7824#   7842#   7844#   7853#   7855#   7868#   7870#
                7882#   7884#   7893#   7895#   7899#   7905#   7911#   7913#   7927#   7929#   7937#   7939#   7950#   7952#   7963#
                7965#   7983#   7985#   7994#   7996#   8009#   8011#   8023#   8025#   8034#   8036#   8040#   8043#   8105#   8107#
                8124#   8126#   8133#   8135#   8147#   8149#   8156#   8158#   8175#   8177#   8189#   8191#   8199#   8201#   8205#
                8235#   8237#   8257#   8259#   8269#   8271#   8283#   8285#   8294#   8296#   8308#   8310#   8327#   8329#   8338#
                8340#   8349#   8351#   8358#   8360#   8364#   8410#   8418#   8420#   8437#   8439#   8449#   8451#   8458#   8460#
                8464#   8468#   8476#   8478#   8496#   8498#   8502#   8506#   8514#   8516#   8547#   8549#   8553#   8556#   8585#
                8587#   8607#   8609#   8617#   8619#   8627#   8629#   8637#   8639#   8646#   8648#   8659#   8661#   8669#   8671#
                8679#   8681#   8689#   8691#   8698#   8700#   8709#   8711#   8723#   8725#   8734#   8736#   8746#   8748#   8753#
                8800#   8802#   8822#   8824#   8832#   8834#   8842#   8844#   8852#   8854#   8861#   8863#   8874#   8876#   8884#
                8886#   8894#   8896#   8904#   8906#   8914#   8916#   8924#   8926#   8934#   8936#   8943#   8945#   8954#   8956#
                8968#   8970#   8979#   8981#   8991#   8993#   8997#   9034#   9036#   9044#   9046#   9057#   9059#   9067#   9069#
                9077#   9079#   9087#   9089#   9097#   9099#   9106#   9108#   9117#   9119#   9130#   9132#   9142#   9144#   9154#
                9156#   9164#   9166#   9173#   9175#   9179#
M$WORD             1#  13004   1366#   1375    1425#   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    1456    1457    4810#   4811    4812    4813    4829#   4830    4831    4832    4879#
                4880    4881    4882    4915#   4916    4917    4918    5044#   5045    5046    5047    5159#   5160    5161    5162
                5393#   5394    5395    5396    5598#   5599    5600    5601    5624#   5625    5626    5627    5650#   5651    5652
                5653    5940#   5941    5942    5943    5979#   5980    5981    5982    6056#   6057    6058    6059    6168#   6169
```

	6170	6171	6207#	6208	6209	6210	9198#	9203#	9208#	9214#	9220#	9226#	9232#	9279	9280
M$XFER	1#	13004#													
NEWTST	2010#	4762	4838	4992	5057	5105	5177	5229	5281	5335	5410	5461	5503	5545	5658
	5743	5848	6078	6263	6363	6495	6709	6806	6903	7000	7165	8054	8206	8380	8557
	8771	9012													
NTST	2010#	4762	4838	4992	5057	5105	5177	5229	5281	5335	5410	5461	5503	5545	5658
	5743	5848	6078	6263	6363	6495	6709	6806	6903	7000	7165	8054	8206	8380	8557
	8771	9012													
OPEN	1#	13004#													
POINTE	1#	13004#	1323												
PRINTB	1#	13004#	4142	4151	4171	4180	4189	4204	4212	4220	4229	4249	4257	4265	4274
	4294	4302	4310	4329	4338	4410	4929	4965	4977						
PRINTF	1#	13004#													
PRINTS	1#	13004#													
PRINTX	1#	13004#	4108	4123	4375	4383	4393	4400	4424	4432	4442	4449	4459	4466	4476
	4483	4501	4509	4519	4526	4935	4941	4947	4957						
READBU	1#	13004#													
READEF	1#	13004#	4579	4586	4593	4600									
RFLAGS	1#	13004#													
SETDF	2010#	2441	2485	2532	2603	2702	2975	3079	3112	3123	3157	3168	3202	3213	3247
	3258	3292	3303	3335	3346	3382	3393	3406	3417	3449	3461	3510	3757	3770	3796
	3806	3819	3829	3842	3852	3865	3875	3888	3898						
SETHRD	2010#														
SETPRI	1#	13004#													
SETSF	2010#														
SETSFT	2010#														
SETVEC	1#	13004#	4679												
SLASH	1#	13004#	1417	1421											
STARS	1#	13004#													
SVC	1#	12984#	1299												
T$GEN	2010#	2441	2485	2532	2603	2702	2975	3079	3112	3123	3157	3168	3202	3213	3247
	3258	3292	3303	3335	3346	3382	3393	3406	3417	3449	3461	3510	3757	3770	3796
	3806	3819	3829	3842	3852	3865	3875	3888	3898						
XFER	1#	13004#													
XFERF	1#	13004#													
XFERT	1#	13004#													
$GEDF	2010#	4809	4828	4878	4914	5043	5158	5392	5597	5623	5649	5939	5978	6055	6167
	6206														
$GEHRD	2010#														
$GESF	2010#														
$GESFT	2010#														
$GTDF	2010#	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						

```
$GTHRD   2010#
$GTSF    2010#
$GTSFT   2010#


. ABS.  037434     000


 ERRORS DETECTED:  0

 CVDMCC,CVDMCC/CRF/SOL=SVC34R.MLB,CVDMCC.P11
 RUN-TIME: 41 52 6 SECONDS
 RUN-TIME RATIO: 144/100=1.4
 CORE USED:  23K  (45 PAGES)
```