```
                .REM    0


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

                  PRODUCT CODE:         AC-B150D-MC

                  PRODUCT TITLE:        CVDVADO DLV11-E OFFLINE TEST

                  PRODUCT DATE:         AUGUST 1984

                  AUTHOR:               ODES CHOATE

                  MAINTAINER:           DIAGNOSTIC ENGINEERING GROUP

                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) 1977,1984 BY DIGITAL EQUIPMENT CORPORATION

              THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                      DIGITAL         PDP             UNIBUS          MASSBUS
                      DEC             DECUS           DECTAPE
```

TABLE OF CONTENTS

```
                                1.0     GENERAL PROGRAM INFORMATION.
                                        --- ---  ------- ------------

                                1.1     PROGRAM PURPOSE (ABSTRACT).

                                        THIS DIAGNOSTIC IS A LOGIC TEST TO VERIFY THE OPERATION OF THE
                                        DLV11-E  SERIAL  LINE INTERFACE.  THE PROGRAM AS SET INITIALLY
                                        DEFAULTS  TO  ALL  OPTIONS,  EXCEPT  PROGRAMMABLE  BAUD  RATE,
                                        ENABLED  AND A WRAP CABLE CONNECTED.  THE USER CAN SELECTIVELY
                                        ENABLE AND DISABLE TESTING OF  THE  OPTIONS  BY  ALTERING  THE
                                        CONTENTS  OF  '$USER'.  THE DIAGNOSTIC IS DESIGNED TO TEST AND
                                        DETECT FAULTS TO THE LOGIC LEVEL  (NOT  TO  THE  CHIP  LEVEL).
                                        THIS TEST OPERATES ON UP TO SIXTEEN(16) IDENTICALLY CONFIGURED
                                        DLV11-E SERIAL LINE INTERFACES.  THE DEFAULT ADDRESSES ARE:

                                                175610 -FIRST SERIAL LINE ADDRESS  OF  16  CONSECUTIVE
                                                        SERIAL LINE DEVICES.


                                                  300 - VECTOR FOR FIRST OF 16 DEVICES.

                                        THIS PROGRAM IS DESIGNED TO RUN  ON  ANY  PDP-11  WITH  4K  OF
                                        MEMORY AND A DLV11-E (LSI-BUS) MODULE.  IT CAN RUN UNDER XXDP,
                                        APT, AND ACT MONITORS, AND  ON  PROCESSORS  WITH  NO  HARDWARE
                                        SWITCH REGISTER.  A POWER FAILURE WILL CAUSE THE DIAG-
                                        NOSTIC TO RESTART.

                                1.2     SYSTEM REQUIREMENTS.

                                     1.         HARDWARE REQUIREMENTS:

                                                ANY PDP-11 FAMILY PROCESSOR
                                                4K MEMORY - MINIMUM
                                                M315 - CABLE TURN AROUND PLUG (OR EQUIVALENT)
                                                MODEM CABLE - BC01V-X OR BC05C X


                                                SOFTWARE REQUIREMENTS:


                                                THIS DIAGNOSTIC IS DESIGNED TO  RUN  IN  ANY  OF  THE
                                         FOLLOWING WAYS:
                                                STAND ALONE
                                                WITH APT MONITOR
                                                WITH ACT MONITOR
                                                WITH XXDP MONITOR (CHAINABLE)
```

```
118                             1.3     RELATED DOCUMENTS AND STANDARDS.
119
120                             DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS        175 003-009-02
121                             APT                                                     MD-11-DZZMA
122                             ACT                                                     AUTOCAT 11-0ZAUB
123                             SYSMAC                                                  MD-11-DZQAC
124
125
126                             1.4     DIAGNOSTIC HIERARCHY PREREQUISITES.
127                             NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN BEFORE THIS, BUT
128                             THE PROCESSOR, MEMORY, AND BUS ARE ASSUMMED TO BE FULLY
129                             OPERATIONAL.
130
131
132                             1.5     ASSUMPTIONS.
133
134
135                             THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED
136                             LOCATION '$USWR' AND '$DEVM' TO THE PROPER VALUES.
137                             THE (H) JUMPER MUST BE REMOVED FROM ALL DLV11-E'S UNDER TEST.
138
139
140                             2.0     OPERATING INSTRUCTIONS.
141                             ---------- ------------
142
143
144                             2.1     LOADING AND STARTING PROCEDURES.
145
146                             USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED
147                             MEDIA.
148
149                             THIS DIAGNOSTIC HAS ONLY ONE (1) STARTING ADDRESS.  200 FOR
150                             START AND RESTART.
151
152                             THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY SETTING
153                             SWITCH 8 IN THE SWITCH REGISTER AND THE TEST NUMBER (IN OCTAL)
154                             IN THE LOWER BYTE.  (NOTE:  ALL TESTS PREVIOUS TO THE SELECTED
155                             ONE ARE EXECUTED WITHOUT ITERATIONS.)
156
157
158                             2.2     SPECIAL ENVIRONMENTS.
159
160                             THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING
161                             UDER APT,ACT,XXDP MONITORS, AS DESCRIBED IN THEIR RESPECTIVE
162                             PROCEDURES MANUAL AND SYSMAC PACKAGE.
163
```

2.3 OPERATIONAL SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

1) TYPE CONTROL G <↑G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE:' SWR=XXXXXX NEW=' (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE TTY:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED). LEADING ZEROS NEED NOT BE TYPED, AND IF MORE THAN 6 DIGITS ARE TYPED THE LAST 6 WILL BE USED. IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U <↑U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 3.

   C) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED FOLLOWED BY A CARRAGE RETURN AND A LINE FEED THEN PROCEED FROM STEP 3 (ERASING ALL PREVIOUS INPUT.)

   DYNAMIC SWITCH REGISTER

BIT 15 - HALT ON ERROR
    14 - LOOP ON TEST
    13 - INHIBIT ERROR TYPEOUTS
    12 - (UNUSED)
    11 - INHIBIT ITERATIONS
    10 - BELL ON ERROR
     9 - LOOP ON ERROR
     8 - LOOP ON TEST IN SWR<7:0>
   7:0 - TEST NUMBER TO LOOP ON (USED WITH BIT 8)

```
218                             2.4     PROGRAM OPTIONS.
219
220                             THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE  DLV11-E'S.   IT
221                             REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '$BASE') AND
222                             ITS INTERRUPT VECTOR (STORED AT '$VECT1 );  AND WILL  BE  ABLE
223                             TO  ADDRESS ANY DLV11-E STARTING AT THE SPECIFIED BASE ADDRESS
224                             UP TO 16 CONSECUTIVE DEVICES.
225                             EXAMPLES:       $BASE:  175610
226                                             $VECT1:    300
227
228                             THE PROGRAM WILL BE  ABLE  TO  TEST  ANY  DLV11-E  WITHIN  THE
229                             ADDRESS RANGE 175610 --> 176000
230
231                             $BASE AND $VECT1 DEFAULT TO 175610 AND 300 RESPECTIVELY.
232                             THE PROGRAM ASSOCIATES UNIT NUMBERS AS FOLLOWS:   (NUMBERS  IN
233                             PARENTHESIS ARE OCTAL)
234                             UNIT#0 -- BASE ADDRESS STORED AT '$BASE'
235                                       ASSOCIATED BASE VECTOR STORED AT '$VECT1'
236                             UNIT#1 -- BASE ADDRESS + (10)
237                                       BASE VECTOR  + (10)
238                                     !
239                                     !    UP TO
240                                     !
241                             UNIT#15 -- BASE ADDRESS + (170)
242                                        BASE VECTOR  + (170)
243
244                             LOCATION '$DEVM' IS USED AS A BIT MAP TO INDICATE  WHICH  UNIT
245                             NUMBERS ARE PRESENT AND WILL BE TESTED.
246
247                             BIT 15                  BIT 1    BIT 0
248                             ---------------------------------------
249                             !UNIT!                 !UNIT  !   UNIT!
250                             ! 15 !                 ! #1   !    #0 !
251                             ---------------------------------------
252
253                             A BIT MAP CAN BE ENTERED AT  '$DEVM'  PRIOR  TO  STARTING  THE
254                             PROGRAM.
255
256                             EXAMPLE:
257                                     $BASE:  175610
258                                     $VECTOR:   300
259                                     $DEVM:      13
260
261                             THE PROGRAM WILL TEST-
262
263                                     UNIT#0  175610  300
264                                     UNIT#1  175620  310
265                                     UNIT#3  175640  330
266
```

OPTIONS

LOCATION $USWR CONTAINS ALL THE USER SELECTABLE OPTIONS. THE VALUES IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD CONFIGURATION.
THE DEFAULT VALUE OF $USWR IS AS FOLLOWS:

BIT POSITION	DEFINITION	DEFAULT VALUE
0-3	#OF DATA BITS	10(8) = 8
4	PARITY ENABLED	0 = NO
5	EVEN ODD PARITY	0 = ODD
6	COMMON SPEED	1 = YES
7	PROGRAMMABLE BAUD RATE	0 = NO
8-11	BAUD RATE OFFSET (SEE FOLLOWING NOTE)	05(8) = 110 BAUD
12	BREAK GENERATION ENABLED	1 = YES
13	CABLE TERMINATED (H315)	1 = YES
14	(-FR) AND (-FD) JUMPERS IN	1 = YES
15	(NOT DEFINED)	

NOTE

THIS DIAGNOSTIC DOES NOT TEST THE PARITY LOGIC.

WHEN THE PROGRAMABLE BAUD RATE OPTION IS ENABLED THE PROGRAMABLE BAUD RATE TEST WILL EXIT WITH THE BAUD RATE SET TO THE SELECTED VALUE. TO CHANGE THE DEFAULT VALUE OF 110 BAUD REPLACE BITS <11:8> WITH THE OFFSET INDICATED IN THE TABLE AT THE END OF THE PBR TEST.

DLV11-E INDIVIDUAL TEST REQUIREMENTS

	TESTS NOT EXECUTED	IF BIT =
APT TEST (BIT 1 OF $ENV)	T25,T37	1
PROGAMMABLE BAUD RATE (BIT 7 OF $USWR)	T32	1
BREAK GENERATION CIRCUIT (BIT 12 OF $USWR)	T2,T40	0
CABLE TERMINATED (BIT 13 OF $USWR)	T15,T16,T17,T20,T21	1
	T22,T23,PARTS OF T34, T36	1
(-FR) & (-FD) JUMPERS IN (BIT 14 OF $USWR)	T5,T6,T16,T17,T20,T22, T23	1

ALL OTHER TESTS WILL RUN REGARDLESS OF ANY BIT SETTINGS.

2.5 EXECUTION TIMES.

EXECUTION TIMES ARE FOR AN LSI-11 PROCESSOR WITH ALL OPTIONS ENABLED ON THE DLV11-E (EXECPT FOR PROGRAMMABLE BAUD RATE), AT 110 BAUD.

FIRST PASS-	90 SECONDS
ADDITIONAL PASSES	95 SECONDS
ADDITIONAL DEVICES	95 SECONDS

THE TEST TIME IS BAUD RATE DEPENDANT; HIGHER BAUD GIVES SHORTER PASS TIMES.

```
                        3.0     ERROR INFORMATION.
                                --- -----------

                        3.1     ERROR REPORTING PROCEDURE.
                                --- --------- ---------- --

                                SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 4 K OF MEMORY THE
                                ERROR  TYPEOUT IS VERY BRIEF.  THE FORMAT OF THE ERROR TYPEOUT
                                IS AS FOLLOWS:

                                TEST#_____,ERROR#_____,PC=_____,ADDRESS=______,VECTOR=

                                WHERE ALL VALUES TYPED ARE OCTAL.
                                THE ADDRESS AND VECTOR REFER TO THE FAILING DLV11-E.
                                FOR FURTHER INFORMATION THE LISTING MUST  BE  CONSULTED.
                                BITS  15,13,10  AND  9  OF  THE  SWITCH  REGISTER  CONTROL THE
                                SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.


                                        BIT 15 - CAUSES THE PROGRAM  TO  HALT  IN  THE  ERROR
                                                 ROUTINE.  CONTINUEING THE PROGRAM CAUSES IT TO
                                                 PROCEED.

                                        BIT 13 - DISABLES THE PRINTING OF THE ERROR MESSAGE.

                                        BIT 10 - CAUSES THE BELL TO RING ON ERROR.

                                        BIT  9 - CAUSES THE DIAGNOSTIC TO LOOP FROM BEGINNING
                                                 OF TEST TO ERROR.

                                       THE ERROR ROUTINE SUPPORTS THE CONTROL G FUNCTION.

                        3.2     ERROR HALTS.
                                ----- ------

                                THE ONLY HALT IN THIS DIAGNOSTIC IS IN THE ERROR ROUTINE.  AND
                                IS  EXECUTED  ONLY  IF  BIT 15 OF THE SWITCH REGISTER IS A ONE
                                WHEN AN ERROR OCCURS.

                        4.0     PERFORMANCE AND PROGRESS REPORTS.
                                ----------- --- -------- --------

                        4.1     PERFORMANCE REPORTS.

                                AS EACH DEVICE  COMPLETES  ONE  PASS  OF  THE  DIAGNOSTIC  THE
                                FOLLOWING WILL BE TYPED:
                                   CSR:_____,VECTOR:_____,ERRORS:_____
```

```
                        WHERE.'CSR:_____' IS THE DEVICE CSR UNDER TEST
                              'VECTOR:__' IS THE ASSOCIATED VECTOR
                        AND   'ERRORS:  ' IS THE TOTAL NUMBER OF ERRORS ON THIS DEVICE
                                          ON THIS PASS.


                                                NOTE
                            THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.

                        AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS
                        TYPED.
                                "ENDPASS@_____."
```

5.0 DEVICE INFORMATION TABLES.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCSR	DATA INT	RING	CLR SEND	CAR DET	RCVR ACT	REC REC			RCVR DONE	RCVR IE	DATA IE		SEC XMIT	REQ SEND	DTR	
RBUF	ERRO R	OR ERR	FR ERR	P ERR					RECEIVED DATA BUFFER							
TCSR	PROGRAMMABLE BAUD RATE SELECT				PBR ENAB				XMIT RDY	XMIT IE			MAIN T			BREA K
TBUF	TRANSMITTER DATA BUFFER															

NOTE

BLANK BOXES INDICATE UNUSED AND RESERVED
BIT POSITIONS. SEE THE LISTING FOR AN
EXPLANATION OF THE BITS.

```
426                             6.0     SUMMARY OF TESTS AND SPECIAL SUBROUTINES.
427                                     --- -- -  --- - --- ------  ----------
428
429
430                                     TEST 1 ADDRESSABILITY
431                                     ---- -
432
433                                             THIS TEST VERIFIES THAT THE ADDRESS AS PLACED  IN  THE
434                                             HARDWARE   P-TABLE  TO  BE  CORRECT  AND  THE  DLV11-E
435                                             RESPONDS TO THAT ADDRESS SPACE.
436
437                                     THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS
438
439
440
441                                     TEST 2 BREAK - TCSR0 SET, CLEAR, RESET
442                                     ---- -
443
444
445                                     TEST 3 MAINT - TCSR2 SET, CLEAR, RESET
446                                     ---- -
447
448
449                                     TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET
450                                     ---- -
451
452                                     TEST 5 DTR - RCSR1 SET, CLEAR
453                                     ---- -
454
455
456                                                             NOTE
457
458                                             RESET DOES NOT CLEAR THIS BIT.  WE CANNOT TEST
459                                             FOR  AN  INITIAL  CONDITION  AS  THIS  BIT  IS
460                                             UNDEFINED  UPON  POWER  UP  AND  INIT  DOESN'T
461                                             AFFECT IT.
462
463
464
465
466                                     TEST 6 REQSEND - RCSR2 SET, CLEAR, RESET
467                                     ---- -
468
469                                             THIS TEST ASSUMES THAT JUMPER FR IS IN.
470
471
```

```
                                        TEST 7 SECXMIT - RCSR3 SET, CLEAR, RESET
                                        ----

                                        TEST 10 DATAIE - RCSR5 SET, CLEAR, RESET
                                        ---- --

                                        TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET
                                        -- - --
```

```
                        THE FOLLOWING 4 TESTS VERIFY  THAT  RESET  (INIT)  INITIALIZES
                        READ ONLY BITS.


                        TEST 12 RCVRDONE - RCSR 7 - IS CLEARED BY INIT
                          -- --


                        TEST 13 XMITRDY - TCSR 7 - IS SET BY INIT
                        ---- --


                        TEST 14 DATAINT - RCSR 15 - IS CLEARED BY INIT.
                        ---- --


                        TEST 15 RCVRACT - RCSR 11 - 15 CLEARED BY INIT
                        ---- --

                        THE FOLLOWING 4 TESTS VERIFY  THAT  THE  EIA  SIGNALS  CAN  BE
                        TRANSMITTED AND RECEIVED THROUGH THE CABLE.

                        TEST 16 CARDET SETS AND CLEARS AS DTR SETS AND CLEARS
                        ---- --


                        TEST 17 CLRSEND SETS AND CLEARS AS DTR SETS AND CLEARS
                        ---- --


                        TEST 20 RING SETS AND CLEARS AS REQSEND SETS AND CLEARS
                        ---- --


                        TEST 21 SECREC SETS AND CLEARS AS SECXMIT SETS AND CLEARS
                        ---- --


                        TEST 22 DATAINT (RCSR-15) SETS WHEN DTR CHANGES STATE AND THAT
                        ---- --
                                DATAINT IS CLEARED AFTER READING RCSR


                                                    NOTE

                                    DTR IS TIED TO BOTH CARDET AND CLRSEND BY  THE
                                    H315.
```

```
531
532                                     TEST 23 DATAINT SETS WHEN RING SETS AND  THAT  DATAINT
533                                     ---- -
534                                         DOES NOT SET WHEN RING CLEARS
535
536
537                                     TEST 24 DATAINT SETS WHEN SECREC CHANGES STA
538                                     ---- -
539
540
```

```
                        TEST 25 XMIT RDY - TCSR 7   CLEARS WHEN TBUF IS LOADED
                        ---- --
                            WITH  A  CHARACTER  AND  THAT IT SETS WITHIN A
                            REASONABLE AMOUNT OF TIME.


                        TEST 26 OUTPUTTING A CHAR FROM TBUF (WITH  MAINT  SET)
                        ---- --
                            RESULTS   IN   RCVRDONE   SETTING   WITHIN   A
                            REASONABLE  AMOUNT  OF  TIME  AND  THAT  RESET
                            CLEARS THE BIT.


                        TEST 27 RCVRDONE IS CLEARED BY READING RBUF
                        ---- --

                        TEST 30 RCVRACT - RCSR 11 - SETS WHEN A START  BIT  IS
                        ---- --
                            RECEIVED  AND  CLEARS WHEN RCVRDONE - RCSR 7 -
                            SETS


                        TEST 31 OVERRUN BIT   RBUF 14
                        ---- --


                        TEST 32 PROGRAMMABLE BAUD RATE TEST TEST AT ALL SPEEDS
                        ---- --
                            AVAILABLE  A COMPARISON WILL BE MADE TO SEE IF
                            NEW TIME IS LESS THAN PREVIOUS.


                        TEST 33 TRANSMITTER INTERRUPT LOGIC TEST
                        ---- --
                            LOGICALLLY THIS IS 4 SEPARATE TESTS
                            A) DOES TRANSMITTER INTERRUPT LOGIC WORK
                            B) AT PRIORITY OF 0
                            C) AND ONLY ONCE
                            D) BUT NOT WITH INTERRUPT ENABLE CLEAR


                        TEST 34 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL
                        ---- --
                                OF  THE RECEIVER SIDE OF THE INTERRUPT LOGIC, BOTH
                                DATASET AND CHARACTER MODES.


                        TEST  35  TEST  ACTUAL   DATA   TRANSFERED   NON-INTERRUPT
                        ----  --
                                MAINTENANCE BIT SET
```

```
                TEST 36 TEST DATA THROUGH CABLE
                ---- --


                TEST 37 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE
                - -  --
                        MODE.


                TEST 40 TEST BREAK GENERATION LOGIC  TRANSMIT  KNOWN  CHAR
                ---- --
                        WITH BREAK SET AND COMPARE RECEIVED WITH 0.


                TEST 41 NOT A TEST - SEND BACK TO LOOP
                ---- --

                                        NOTE

                            FOR ALL OF THE FOLLOWING ROUTINES THE  USE
                            OF  (R5)  IS PART OF THE LINKAGE MECHANISM
                            BETWEEN THE CALLER AND THE CALLED.


                ROUTINE:TIMER
                ------------

                            THIS ROUTINE IS USED TO TEST THE STATUS OF
                            ANY BIT IN ANY REGISTER.
                INPUTS:
                        HOWLONG     THE  MAXIMUM  AMOUNT  OF  TIME  TO
                                    SPEND IN THIS ROUTINE.
                        WHICHBIT    A MASK WITH THE  BIT(S)  SET  THAT
                                    ARE TO BE CHECKED
                        REG         A POINTER TO THE  REGISTER  TO  BE
                                    CHECKED
                        SETCLR      THE DESIRED RESULTS -- EITHER  SET
                                    OR  CLEAR
                OUTPUT:
                        THE 'C' BIT IS SET TO INDICATE AN ERROR BUT IT
                        IS TESTED BY THE IF.ERROR STATEMENT.
```

```
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
```

```
ROUTINE:DATLNG
 ----   ------
        THIS ROUTINE SETS UP A MASK FOR DATA, WITH

INPUT:  NOTHING IS PASSED TO THIS ROUTINE  BUT  GLOBAL
        INFORMATION IS ASSUMED TO EXIST:
        $USWR-- THE WORD FOR SOFTWARE PARAMETERS
        DATA-- A MASK FOR THE LOCATION  OF  THE  OCTAL
        NUMBER OF DATA BITS

OUTPUT----
        MASK-- A MASK OF BINARY ZEROS  RIGHT-JUSTIFIED
        THE NUMBER OF WHICH IS DEFINED IN $USWR WORD.


ROUTINE:WAIT
------------
        THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
        MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
        THIS  IS  ACCOMPLISHED  BY  INCREMENTING  A
        REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
        TO APPROXIMATE 1 MILLI SEC.


SERVICE ROUTINE:  INTSRV
------- --------  ------
        THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT


        'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
        THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
        TO LOOK FOR.


ROUTINE:CYCLE
------------
        THIS ROUTINE  CAUSES  ADRS  TO  POINT  TO  THE
        ADDRESS  OF  DLV11-E  UNDER  TEST,  ADRS +2 TO
        POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
        IT  KEEPS  TRACK OF THE CURRENT DEVICE AND BIT
        MASKS.
```

```
                        .
                        .TITLE  MAINDEC-ZZ-CVDVA-D
                        ;*COPYRIGHT (C) 1977
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY ODES CHOATE
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                        ;*
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
                        ;*
                        ;*      SWITCH                  USE
                        ;*      ------          --------------------
                        ;*        15            HALT ON ERROR
                        ;*        14            LOOP ON TEST
                        ;*        13            INHIBIT ERROR TYPEOUTS
                        ;*        11            INHIBIT ITERATIONS
                        ;*        10            BELL ON ERROR
                        ;*         9            LOOP ON ERROR
                        ;*         8            LOOP ON TEST IN SWR<7:0>

                        .SBTTL  BASIC DEFINITIONS

                        ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                        ;*MISCELLANEOUS DEFINITIONS
000011                  HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                  LF=     12              ;;CODE FOR LINE FEED
000015                  CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                  CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                  PS=     177776          ;;PROCESSOR STATUS WORD
                        .EQUIV  PS,PSW
177774                  STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                  PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                  DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                  DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER

                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
```

```
                        ;
                        PR1=    40              ;;PRIORITY LEVEL 1
                        PR2=    100             ;;PRIORITY LEVEL 2
                        PR3=    140             ;;PRIORITY LEVEL 3
                        PR4=    200             ;;PRIORITY LEVEL 4
                        PR5=    240             ;;PRIORITY LEVEL 5
                        PR6=    300             ;;PRIORITY LEVEL 6
                        PR7=    340             ;;PRIORITY LEVEL 7
                        ;
                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                        SW15=   100000
                        SW14=   40000
                        SW13=   20000
                        SW12=   10000
                        SW11=   4000
                        SW10=   2000
                        SW09=   1000
                        SW08=   400
                        SW07=   200
                        SW06=   100
                        SW05=   40
                        SW04=   20
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0
                        ;
                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        BIT15=  100000
                        BIT14=  40000
                        BIT13=  20000
                        BIT12=  10000
                        BIT11=  4000
                        BIT10=  2000
                        BIT09=  1000
                        BIT08=  400
                        BIT07=  200
                        BIT06=  100
                        BIT05=  40
                        BIT04=  20
                        BIT03=  10
                        BIT02=  4
                        BIT01=  2
                        BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
```

```
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
000004                          ILLMEM= 4
000001                          ADRS=   P1
000002                          GOOD=   R2
000003                          BAD=    R3
000001                          REGISTER=R1
000002                          BIT=    R2
000003                          FUNCT=  R3
000002                          LEAD=   R2
000004                          FOLLOW= R4
175610                          DLADDR= 175610

                                ; THE FOLLOWING DEFINITIONS APPLY TO THE GLOBAL SUBS
177777                          SET=    -1
000000                          CLR=    0


                        ;;*****************************************************************
                                ; RCSR REGISTER BIT NAMES
                        ;;*****************************************************************
100000                          DATAINT=        BIT15           ; DATASET INTERRUPT
040000                          RING=           BIT14           ; RINGING SIGNAL INDICATOR
020000                          CLRSEND=        BIT13           ; CLEAR TO SEND FROM DATASET
010000                          CARDET=         BIT12           ; CARRIER DETECT
004000                          RCVRACT=        BIT11           ; RECEIVER ACTIVE INDICATOR
002000                          SECREC=         BIT10           ; SECONDARY RECEIVE
                                ; UNUSED        BIT09
                                ; UNUSED        BIT08
000200                          RCVRDONE=       BIT07           ; RECEIVER DONE
000100                          RCVRIE=         BIT06           ; RECEIVER INTERRUPT ENABLE
000040                          DATAIE=         BIT05           ; DATASET INTERRUPT ENABLE
                                ; UNUSED        BIT04
000010                          SECXMIT=        BIT03           ; SECONDARY TRANSMIT DATA
000004                          REQSEND=        BIT02           ; REQUEST TO SEND
000002                          DTR=            BIT01           ; DATA TERMINAL READY
                                ; UNUSED        BIT00
```

```
                        ;;*****************************************************************
                        ; RBUF REGISTER BIT NAMES
                        ;;*****************************************************************
100000                  ERROR=      BIT15       ; ERROR INDICATOR
040000                  ORERR=      BIT14       ; OVERRUN ERROR
020000                  FRERR=      BIT13       ; FRAMING ERROR
010000                  PERR=       BIT12       ; PARITY ERROR
                        ; UNUSED    BIT11
                        ; UNUSED    BIT10
                        ; UNUSED    BIT09
                        ; UNUSED    BIT08
000200                  RDATA7=     BIT07       ; \
000100                  RDATA6=     BIT06       ;  !
000040                  RDATA5=     BIT05       ;  !
000020                  RDATA4=     PIT04       ;   \   RECEIVED DATA BITS
000010                  RDATA3=     BIT03       ;   /
000004                  RDATA2=     BIT02       ;  !
000002                  RDATA1=     BIT01       ;  !
000001                  RDATA0=     BIT00       ; /

                        ;;*****************************************************************
                        ; TCSR REGISTER BIT NAMES
                        ;;*****************************************************************
100000                  PBAUD3=     BIT15       ;  \
040000                  PBAUD2=     BIT14       ;   \   PROGRAMMABLE BAUD
020000                  PBAUD1=     BIT13       ;   /   RATE  BITS
010000                  PBAUD0=     BIT12       ;  /
004000                  PBAUDSET=   BIT11       ; ENABLE SETTING OF
                                                ; PROGRAMMABLE BAUDE RATE
                        ; UNUSED    BIT10
                        ; UNUSED    BIT09
                        ; UNUSED    BIT08
000200                  XMITRDY=    BIT07       ; TRANSMITTER READY
000100                  XMITIE=     BIT06       ; TRANSMITTER INTERRUPT ENABLE
                        ; UNUSED    BIT05
                        ; UNUSED    BIT04
                        ; UNUSED    BIT03
000004                  MAINT=      BIT02       ; MAINTENANCE SET BIT
                        ; UNUSED    BIT01
000001                  BREAK=      BIT00       ; SEND BREAK (CONTINUOUS SPACE)


                        ;;*****************************************************************
                        ; TBUF REGISTER BIT NAMES
                        ;;*****************************************************************
                        ; UNUSED    BIT15
```

```
                                        ; UNUSED        BIT14
                                        ; UNUSED        BIT13
                                        ; UNUSED        BIT12
                                        ; UNUSED        BIT11
                                        ; UNUSED        BIT10
                                        ; UNUSED        BIT09
                                        ; UNUSED        BIT08
000200                                  TDATA7=         BIT07           ; \
000100                                  TDATA6=         BIT06           ; !
000040                                  TDATA5=         BIT05           ; !
000020                                  TDATA4=         BIT04           ;  \ TRANSMITTER DATA BUFFER
000010                                  TDATA3=         BIT03           ;  /
000004                                  TDATA2=         BIT02           ; !
000002                                  TDATA1=         BIT01           ; !
000001                                  TDATA0=         BIT00           ; /


                                ;;*****************************************************************
                                        ; FLAG BITS TO BE USE OR CLEARED IN $USWR.

000017                                      DATA =          17
000020                                      PARITY =        20
000040                                      EVENODD =       40
000100                                      COMSPD =        100
000200                                      PBR =           200

                                        ; BAUDE MUST BE ON THE UPPER
                                        ;  BYTE BOUNDRY OF $USWR.--4 BITS
007400                                      BAUD =          7400
010000                                      BRK =           10000
020000                                      CABLE =         20000
040000                                      FRFD =          40000
                                ;;*****************************************************************
                                .SBTTL  TRAP CATCHER

000000                                  .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
000174                                  .=174
000174  000000                  DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000                  SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
                                .SBTTL  STARTING ADDRESS(ES)
000200  000137  001336                  JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
                                .SBTTL  ACT11 HOOKS

                                ;;*****************************************************************
                                ;HOOKS REQUIRED BY ACT11
000204                                  $SVPC=.                 ;SAVE PC
000046                                  .=46
000046  012452                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                                  .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
000204                                  .=$SVPC                 ;; RESTORE PC
001000                          .=1000
                                .SBTTL  APT PARAMETER BLOCK
```

```
                                ;;**********************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**********************************************************************
001000                                  .$X=.   ;;SAVE CURRENT LOCATION
000024                                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                         200     ;;FOR APT START UP
000044                                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                         $APTHDR ;;POINT TO APT HEADER BLOCK
001000                                  .=.$X   ;;RESET LOCATION COUNTER
                                ;;**********************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

001000                          $APTHD:
001000  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001174                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000005                  $TSTM:  .WORD   5       ;;RUN TIM OF LONGEST TEST
001006  000055                  $PASTM: .WORD   45.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000036                  $UNITM: .WORD   30.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000030                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
MAINDEC-ZZ CVDVA D      MACY11 30A(1052)  12-JUL-84  09:52  PAGE 27
CVDVAD.P11    12-JUL-84 05:04           COMMON TAGS

                                        .SBTTL  COMMON TAGS

                                        ;;**********************************************************************
                                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                        ;*USED IN THE PROGRAM.

                001100                          .=1100
001100                                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                                  .WORD   0
001102  000                             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000                             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000                          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000                          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000                          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000                          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000                             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001                             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000                          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000                          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000                          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000                          $GDDAT:  WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000                          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                                  .WORD   0
001134  000                             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000                             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                                  .WORD   0
001140  177570                          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570                          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560                          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562                          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564                          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566                          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000                             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002                             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012                             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001157  000                             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001160  000000                          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001162  000000                          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001164  177607  000377                  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001170  077                             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001171  015                             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001172  000012                          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                        ;;**********************************************************************
                                        .SBTTL  APT MAILBOX-ETABLE

                                        ;;**********************************************************************
                                        .EVEN
001174                                  $MAIL:                          ;;APT MAILBOX
001174  000000                          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001176  000000                          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001200  000000                          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001202  000000                          $PASS:  .WORD   APASS           ;;PASS COUNT
001204  000000                          $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001206  000000                          $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001210  000000                          $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
```

```
1026  001212  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1027  001214                        $ETABLE:                ;;APT ENVIRONMENT TABLE
1028  001214     000                $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1029  001215     000                $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1030  001216  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1031  001220  071110                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1032  001222  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1033                                ;*                      BITS 15-11=CPU TYPE
1034                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1035                                ;*                              11/70=06,PDQ=07,Q=10
1036                                ;*                      BIT 10=REAL TIME CLOCK
1037                                ;*                      BIT  9=FLOATING POINT PROCESSOR
1038                                ;*                      BIT  8=MEMORY MANAGEMENT
1039  001224     000                $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1040  001225     000                $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1041                                ;*                      MEM.TYPE P TE   --  (HIGH BYTE)
1042                                ;*                              900 NSEC CORE=001
1043                                ;*                              300 NSEC BIPOLAR=002
1044                                ;*                              500 NSEC MOS=003
1045  001226  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1046                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1047  001230     000                $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1048  001231     000                $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1049  001232  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1050  001234     000                $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1051  001235     000                $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1052  001236  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1053  001240     000                $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1054  001241     000                $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1055  001242  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1056  001244  000300                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1057  001246  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1058  001250  175610                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1059  001252  000001                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1060  001254                        $ETEND:
1061                                .MEXIT
```

```
1062                                    .SBTTL  ERROR POINTER TABLE
1063
1064                                    ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
1065                                    ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
1066                                    ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
1067                                    ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
1068                                    ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
1069
1070                                    ;*      EM              ;;POINTS TO THE ERROR MESSAGE
1071                                    ;*      DH              ;;POINTS TO THE DATA HEADER
1072                                    ;*      DT              ;;POINTS TO THE DATA
1073                                    ;*      DF              ;;POINTS TO THE DATA FORMAT
1074
1075
1076  001254                            $ERRTB:
1077                                    ;;          GLOBAL DATA
1078  001254  175610                        DLADD:  DLADDR
1079  001256  000300                        DLVEC:  300
1080  001260  175610                        RCSR:   DLADDR  + 0
1081  001262  175612                        RBUF:   DLADDR  + 2
1082  001264  175614                        TCSR:   DLADDR  + 4
1083  001266  175615                        TCSRHI: DLADDR  + 5
1084  001270  175616                        TBUF:   DLADDR  + 6
1085  001272  000000                        I:      0
1086  001274  000020                        .BLKW 20        ;FOR R5 STACK
1087  001334  000000                        R5STACK:        .WORD 0
1088  001336                            START:
1089                                    .SBTTL  INITIALIZE THE COMMON TAGS
1090                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1091  001336  012706  001100                    MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1092  001342  005026                            CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1093  001344  022706  001140                    CMP     #SWR,R6 ;;DONE?
1094  001350  001374                            BNE     .-6             ;;LOOP BACK IF NO
1095  001352  012706  001100                    MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1096                                    ;;INITIALIZE A FEW VECTORS
1097  001356  012737  014374  000020            MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1098  001364  012737  000340  000022            MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1099  001372  012737  014174  000030            MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1100  001400  012737  000340  000032            MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1101  001406  012737  015326  000034            MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1102  001414  012737  000340  000036            MOV     #340,@#TRAPVEC+2;LEVEL 7
1103  001422  012737  012506  000024            MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1104  001430  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1105  001436  016767  010756  010746            MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1106  001444  005067  177510                   CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1107  001450  005067  177506                   CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1108  001454  112767  000001  177433           MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1109  001462  012767  001462  177416           MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCO
1110  001470  012767  001470  177412           MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1111                                    ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1112                                    ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1113  001476  013746  000004                   MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1114  001502  012737  001536  000004            MOV     #64$,@#EPRVEC   ;;SET UP ERROR VECTOR
1115  001510  012767  177570  177422           MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1116  001516  012767  177570  177416           MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1117  001524  022777  177777  177406           CMP     # 1,@SWR         ;;TRY TO REFERENCE HARDWARE SWR
```

```
1118  001532  001012                        BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1119                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1120  001534  000403                        BR      65$             ;;BRANCH IF NO TIMEOUT
1121  001536  012716  001544        64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1122  001542  000002                        RTI
1123  001544  012767  000176  177366 65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1124  001552  012767  000174  177362        MOV     #DISPREG,DISPLAY
1125  001560  012637  000004        66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1126
1127  001564  005067  177412                CLR     $PASS           ;;CLEAR PASS COUNT
1128  001570  132767  000200  177417        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1129  001576  001403                        BEQ     67$             ;;YES,USE NON-APT SWITCH
1130  001600  012767  001216  177332        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1131  001606                        67$:
                                    .SBTTL  TYPE PROGRAM NAME
1132
1133                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1134  001606  005227  177777                INC     #-1             ;;FIRST TIME?
1135  001612  001051                        BNE     68$             ;;BRANCH IF NO
1136  001614  022737  012452  000042        CMP     #$ENDAD,@#42    ;;ACT-11?
1137  001622  001445                        BEQ     68$             ;;BRANCH IF YES
1138  001624  104401  001672                TYPE    ,69$            ;;TYPE ASCIZ STRING
1139                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1140  001630  005737  000042                TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1141  001634  001012                        BNE     70$             ;;BRANCH IF YES
1142  001636  126727  177352  000001        CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1143  001644  001406                        BEQ     70$             ;;BRANCH IF YES
1144  001646  026727  177266  000176        CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1145  001654  001005                        BNE     71$             ;;BRANCH IF NO
1146  001656  104406                        GTSWR                   ;;GET SOFT-SWR SETTINGS
1147  001660  000403                        BR      71$
1148  001662  112767  000001  177244 70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1149  001670                        71$:
1150  001670  000422                        BR      68$             ;;GET OVER THE ASCIZ
1151                                ;;69$:  .ASCIZ  <CRLF>+ZZ-CVDVA-D  DLV11-E OFFLINE TEST+<CRLF>
1152  001736                        68$:
1153  001736                                LET INITFLAG := #1
1154  001736  012767  000001  010410        MOV     #1,INITFLAG
1155  001744                        LOOP:
1156  001744                                CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
1157  001744  004767  010254                JSR     PC,CYCLE
1158                                                                ;               ADDRS+2  > NEXT VECTOR
1159                                                                ;GET UNIT ADDRESS
1160  001750                                                                LET     DLADD := (ADRS)+
1161  001750  012167  177300                MOV     (ADRS)+,DLADD
1162                                                                ;GET UNIT VECTOR
1163  001754                                                                LET     DLVEC := (ADRS)
1164  001754  011167  177276                MOV     (ADRS),DLVEC
1165  001760                                                                LET     ADRS  := DLADD
1166  001760  016701  177270                MOV     DLADD,ADRS
1167                                                                ;RCSR = DLADD  +  0
1168  001764                                                                LET     RCSR  := DLADD
1169  001764  016767  177264  177266        MOV     DLADD,RCSR
1170  001772                                                                LET     RBUF  := DLADD + #2
1171  001772  016767  177256  177262        MOV     DLADD,RBUF
1172  002000  062767  000002  177254        ADD     #2,RBUF
1173  002006                                                                LET     TCSR  := DLADD + #4
```

```
 1174  002006  016767  177242  177250          MOV     DLADD,TCSR
 1175  002014  062767  000004  177242          ADD     #4,TCSR
 1176  002022                                                      LET     TCSRHI := DLADD + #5
 1177  002022  016767  177226  177236          MOV     DLADD,TCSRHI
 1178  002030  062767  000005  177230          ADD     #5,TCSRHI
 1179  002036                                                      LET     TBUF   := DLADD + #6
 1180  002036  016767  177212  177224          MOV     DLADD,TBUF
 1181  002044  062767  000006  177216          ADD     #6,TBUF
 1182  002052                                                      LET     R5     := #R5STACK
 1183  002052  012705  001334                  MOV     #R5STACK,R5
 1184                                                              ;;BRESET
 1185  002056  000005                          RESET
```

```
1186                                    ;;*****************************************************************
1187                                    ;*TEST 1         ADDRESSABILITY
1188                                    ;*              THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
1189                                    ;*              THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS
1190                                    ;*              TO THAT ADDRESS SPACE
1191                                    ;;*****************************************************************
1192  002060  000004                   TST1:   SCOPE
1193  002062  012767  000002  177070           MOV     #2,$TIMES       ;;DO 2 ITERATIONS
1194  002070  012767  000001  177102           MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1195  002076                                                                   LET     ADRS := DLADD
1196  002076  016701  177152                   MOV     DLADD,ADRS
1197                                                                             ; SET UP INTERRUPT
1198  002102                                                                   SETVEC  #ILLMEM,#INTSRV,#PR7
1199  002102  010146                           MOV     R1,-(SP)
1200  002104  012701  000004                   MOV     #ILLMEM,R1
1201  002110  012721  012032                   MOV     #INTSRV,(R1)+
1202  002114  012711  000340                   MOV     #PR7,(R1)
1203  002120  012601                           MOV     (SP)+,R1
1204  002122                                                                   LET     I := #0
1205  002122  005067  177144                   CLR     I
1206  002126                                                                   REPEAT
1207  002126                                  $1:
1208  002126                                                                       BGNSUB
1209  002126  012767  002134  176754           MOV     #64$,$LPERR
1210                                                                                 ;CLEAR FLAG
1211  002134                                                                         LET INTFLAG := #0
1212  002134  005067  007700                   CLR     INTFLAG
1213
1214                                                                                 ;READ FLAG
1215  002140  005711                           TST @ADRS
1216  002142                                                                         IF INTFLAG NE #0 THEN
1217  002142  005767  007672                   TST     INTFLAG
1218  002146  001401                           BEQ     $2
1219                                                                                 ; FATAL ERROR
1220  002150                                                                             ERRDF 1,,NODL
1221  002150  104001                           ERROR   1
1222  002152                                                                         ENDIF
1223  002152                                  $2:
1224  002152                                                                       ENDSUB
1225  002152                                                                       LET    I := I + #2
1226  002152  062767  000002  177112           ADD     #2,I
1227  002160                                                                       LET    ADRS := DLADD + I
1228  002160  016701  177070                   MOV     DLADD,ADRS
1229  002164  066701  177102                   ADD     I,ADRS
1230  002170                                                                   UNTIL I EQ #8.
1231  002170  026727  177076  000010           CMP     I,#8.
1232  002176  001353                           BNE     $1
1233  002200                                                                   CLRVEC ILLMEM
1234  002200  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1235  002202  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
1236  002204  012701  000004                   MOV     #ILLMEM,R1
1237  002210  010102                           MOV     R1,R2
1238  002212  062702  000002                   ADD     #2,R2
1239  002216  010221                           MOV     R2,(R1)+
1240  002220  005011                           CLR     (R1)
1241  002222  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
1242  002224  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
1243                                                                  ;END OF TEST
1244  002226                                                  ENDTST
1245                                  ;;**********************************************************************
1246                                  ;* THE FOLLOWING 8 TESTS  TEST ALL 'READ WRITE' BITS
1247                                  ;;**********************************************************************
1248
1249
```

```
1250                                    ;;*****************************************************************
1251                                    ;*TEST 2        BREAK - TCSR0 SET, CLEAR, RESET
1252                                    ;*      NOTE:  THE (H) JUMPER MUST BE REMOVED FOR THIS
1253                                    ;*             TEST TO FUNCTION PROPERLY.
1254                                    ;;*****************************************************************
1255  002226  000004                   TST2:   SCOPE
1256  002230  012767  000010  176722            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1257  002236  012767  000002  176734            MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1258
1259  002244                                                              IF #BRK NOTSETIN $USWR THEN
1260  002244  032767  010000  176746            BIT     #BRK,$USWR
1261  002252  001004                           BNE     $3
1262  002254                                                                  EXIT TST
1263  002254  012767  000001  176676            MOV     #1,$TIMES
1264  002262  000452                           BR      TST3            ;;;EXIT THIS TEST
1265  002264                                                              ENDIF
1266  002264                                $3:
1267                                                      ; SEE IF IT IS CLEAR
1268  002264                                                            BGNSUB
1269  002264  012767  002272  176616            MOV     #64$,$LPERR
1270
1271  002272                                                              IF      #BREAK SETIN @TCSR THEN
1272  002272  032777  000001  176764            BIT     #BREAK,@TCSR
1273  002300  001401                           BEQ     $4
1274                                                                        ; BREAK DID NOT RESET IN TCSR
1275  002302                                                               ERRHRD 2,,DIDNOT
1276  002302  104002                           ERROR   2
1277  002304                                                              ENDIF
1278  002304                                $4:
1279  002304                                                          ENDSUB
1280
1281                                                      ; TRY TO SET BREAK BIT
1282  002304                                                            BGNSUB
1283  002304  012767  002312  176576            MOV     #64$,$LPERR
1284  002312                                                              LET     @TCSR := @TCSR SET.BY #BREAK
1285  002312  052777  000001  176744            BIS     #BREAK,@TCSR
1286                                                                      ; STUCK TO 0
1287  002320                                                              IF      #BREAK NOTSETIN @TCSR THEN
1288  002320  032777  000001  176736            BIT     #BREAK,@TCSR
1289  002326  001001                           BNE     $5
1290                                                                        ; BREAK DID NOT SET IN TCSR
1291  002330                                                               ERRHRD 3,,DIDNOT
1292  002330  104003                           ERROR   3
1293  002332                                                              ENDIF
1294  002332                                $5:
1295  002332                                                          ENDSUB
1296
1297                                                      ; TRY TO CLEAR A SET BIT
1298  002332                                                            BGNSUB
1299  002332  012767  002340  176550            MOV     #64$,$LPERR
1300
1301  002340                                                              LET     @TCSR := @TCSR CLR.BY #BREAK
1302  002340  042777  000001  176716            BIC     #BREAK,@TCSR
1303                                                                      ; SHOULD HAVE CLEARED
1304  002346                                                              IF      #BREAK SETIN @TCSR THEN
1305  002346  032777  000001  176710            BIT     #BREAK,@TCSR
```

```
                        T2      BREAK - TCSRO SET, CLEAR, RESET

1306  002354  001401                  BEQ     $6
1307                                                    ; BREAK  DID NOT CLEAR IN TCSR
1308  002356                                            ERRHRD 4,,DIDNOT
1309  002356  104004                  ERROR   4
1310  002360                                          ENDIF
1311  002360                        $6:
1312  002360                                        ENDSUB
1313
1314                                          ; NOW SEE IF RESET CLEARS IT
1315  002360                                        BGNSUB
1316  002360  012767  002366  176522  MOV     #64$,$LPERR
1317
1318  002366                                          LET     @TCSR := @TCSR SET.BY #BREAK
1319  002366  052777  000001  176670  BIS     #BREAK,@TCSR
1320                                                  ; ISSUE BUS RESET
1321  002374                                          BRESET
1322  002374  000005                  RESET
1323  002376                                          IF      #BREAK SETIN @TCSR THEN
1324  002376  032777  000001  176660  BIT     #BREAK,@TCSR
1325  002404  001401                  BEQ     $7
1326                                                  ; BREAK DID NOT RESET IN TCSR
1327  002406                                          ERRHRD 5,,DIDNOT
1328  002406  104005                  ERROR   5
1329  002410                                          ENDIF
1330  002410                        $7:
1331  002410                                        ENDSUB
1332  002410                                      ENDTST
1333
1334
1335                            ;;*****************************************************************
```

```
                        ;;*****************************************************************
                        ;*TEST 3          MAINT - TCSR2 SET, CLEAR, RESET
                        ;;*****************************************************************
002410  000004          TST3:   SCOPE
002412  012767  000010  176540          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002420  012767  000003  176552          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

                                        ; SEE IF IT IS CLEAR
                                                BGNSUB
002426  012767  002434  176454          MOV     #64$,$LPERR
                                                    IF      #MAINT SETIN @TCSR THEN
002434  032777  000004  176622          BIT     #MAINT,@TCSR
002442  001401                          BEQ     $10
                                                    ; MAINT DID NOT RESET IN TCSR
                                                    ERRHRD 6,,DIDNOT
002444  104006                          ERROR   6
                                                    ENDIF
002446                          $10:
                                                ENDSUB

                                        ; TRY TO SET MAINT BIT
                                                BGNSUB
002446  012767  002454  176434          MOV     #64$,$LPERR
                                                    LET     @TCSR := @TCSR SET.BY #MAINT
002454  052777  000004  176602          BIS     #MAINT,@TCSR
                                                    ; STUCK TO 0
                                                    IF      #MAINT NOTSETIN @TCSR THEN
002462  032777  000004  176574          BIT     #MAINT,@TCSR
002470  001001                          BNE     $11
                                                    ; MAINT DID NOT SET IN TCSR
                                                    ERRHRD 7,,DIDNOT
002472  104007                          ERROR   7
                                                    ENDIF
002474                          $11:
                                                ENDSUB

                                        ; TRY TO CLEAR A SET BIT
                                                BGNSUB
002474  012767  002502  176406          MOV     #64$,$LPERR
                                                    LET     @TCSR := @TCSR CLR.BY #MAINT
002502  042777  000004  176554          BIC     #MAINT,@TCSR
                                                    ; SHOULD HAVE CLEARED
                                                    IF      #MAINT SETIN @TCSR THEN
002510  032777  000004  176546          BIT     #MAINT,@TCSR
002516  001401                          BEQ     $12
                                                    ; MAINT DIT NOT CLEAR INTCSR
                                                    ERRHRD 10.,DIDNOT
002520  104010                          ERROR   10
                                                    ENDIF
002522                          $12:
                                                ENDSUB

                                        ; NOW SEE IF RESET CLEARS IT
002522                                          BGNSUB
```

```
 1392  002522  012767  002530  176360          MOV     #64$,$LPERR
 1393
 1394  002530                                                          LET     @TCSR := @TCSR SET.BY #MAINT
 1395  002530  052777  000004  176526          BIS     #MAINT,@TCSR
 1396                                                                      ; ISSUE BUS RESET
 1397  002536                                                          BRESET
 1398  002536  000005                          RESET
 1399  002540                                                          IF      #MAINT SET IN @TCSR THEN
 1400  002540  032777  000004  176516          BIT     #MAINT,@TCSR
 1401  002546  001401                          BEQ     $13
 1402
 1403  002550                                                              ; MAINT DID NOT RESET IN TCSR
 1404  002550  104011                          ERROR   11                  ERRHRD 11,,DIDNOT
 1405  002552
 1406  002552                          $13:                            ENDIF
 1407  002552                                                      ENDSUB
 1408  002552                                                  ENDTST
 1409
 1410
 1411
 1412                            ;;*****************************************************************
```

```
1413                                    ;;*****************************************************************
1414                                    ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
1415                                    ;;*****************************************************************
1416  002552  000004                   TST4:   SCOPE
1417  002554  012767  000010  176376           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1418  002562  012767  000004  176410           MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1419                                                                    ; USE PRIORITY OF 7
1420  002570  012746  000340                   MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
1421  002574  012746  002602                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1422  002600  000002                           RTI                     ;;POP NEW PC AND PS
1423  002602                           64$:
1424
1425                                                    ; SEE IF IT IS CLEAR
1426  002602                                                  BGNSUB
1427  002602  012767  002610  176300           MOV     #65$,$LPERR
1428
1429  002610                                                      IF      #XMITIE SETIN @TCSR THEN
1430  002610  032777  000100  176446           BIT     #XMITIE,@TCSR
1431  002616  001401                           BEQ     $14
1432
1433  002620                                                      ; XMITIE DID NOT RESET IN TCSR
1434  002620  104012                           ERROR   12          ERRHRD 12,,DIDNOT
1435  002622                                                      ENDIF
1436  002622                           $14:
1437  002622                                                  ENDSUB
1438
1439                                                    ; TRY TO SET XMITIE BIT
1440  002622                                                  BGNSUB
1441  002622  012767  002630  176260           MOV     #64$,$LPERR
1442  002630                                                      LET     @TCSR := @TCSR SET.BY #XMITIE
1443  002630  052777  000100  176426           BIS     #XMITIE,@TCSR
1444                                                      ; STUCK TO 0
1445  002636                                              IF      #XMITIE NOTSETIN @TCSR THEN
1446  002636  032777  000100  176420           BIT     #XMITIE,@TCSR
1447  002644  001001                           BNE     $15
1448                                                      ; XMIT DID NOT RESET IN TCSR
1449  002646                                              ERRHRD 13,,DIDNOT
1450  002646  104013                           ERROR   13
1451  002650                                              ENDIF
1452  002650                           $15:
1453  002650                                                  ENDSUB
1454
1455                                                    ; TRY TO CLEAR A SET BIT
1456  002650                                                  BGNSUB
1457  002650  012767  002656  176232           MOV     #64$,$LPERR
1458
1459  002656                                                      LET     @TCSR := @TCSR CLR.BY #XMITIE
1460  002656  042777  000100  176400           BIC     #XMITIE,@TCSR
1461                                                      ; SHOULD HAVE CLEARED
1462  002664                                              IF      #XMITIE SETIN @TCSR THEN
1463  002664  032777  000100  176372           BIT     #XMITIE,@TCSR
1464  002672  001401                           BEQ     $16
1465                                                      ;  XMIT DID NOT CLEAR IN TCSR
1466  002674                                              ERRHRD 14,,DIDNOT
1467  002674  104014                           ERROR   14
1468  002676                                              ENDIF
```

```
1469  002676                          $16:
1470  002676                                                          ENDSUB
1471
1472                                                          ; NOW SEE IF RESET CLEARS IT
1473  002676                                                          BGNSUB
1474  002676  012767  002704  176204          MOV     #64$,$LPERR
1475
1476  002704                                                              LET     @TCSR := @TCSR SET.BY #XMITIE
1477  002704  052777  000100  176352          BIS     #XMITIE,@TCSR
1478                                                                  ; ISSUE BUS RESET
1479  002712                                                          BRESET
1480  002712  000005                          RESET
1481  002714                                                          IF      #XMITIE SETIN @TCSR THEN
1482  002714  032777  000100  176342          BIT     #XMITIE,@TCSR
1483  002722  001401                          BEQ     $17
1484                                                                  ; XMIT DID NOT RESET IN TCSR
1485  002724                                                          ERRHRD 15,,DIDNOT
1486  002724  104015                          ERROR   15
1487  002726                                                          ENDIF
1488  002726                          $17:
1489  002726                                                      ENDSUB
1490  002726                                                  ENDTST
1491
1492
1493
1494                                  ;;*****************************************************************
```

```
1495                                    ;;*****************************************************************
1496                                    ;*TEST 5          DTR   RCSR1    SET, CLEAR
1497                                    ;*               NOTE:  RESET DOES NOT CLEAR THIS BIT
1498                                    ;*               WE CANNOT TEST FOR AN INITIAL CONDITION
1499                                    ;*               AS THIS BIT IS UNDEFINED UPON POWER UP AND
1500                                    ;*               INIT DOESN'T AFFECT IT.
1501                                    ;*              THE (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK.
1502                                    ;;*****************************************************************
1503  002726  000004                   TST5:   SCOPE
1504  002730  012767  000010  176222           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1505  002736  012767  000005  176234           MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1506  002744                                                                   IF #FRFD NOTSETIN $USWR THEN
1507  002744  032767  040000  176246           BIT     #FRFD,$USWR
1508  002752  001004                           BNE     $20
1509  002754                                                                       EXIT TST
1510  002754  012767  000001  176176           MOV     #1,$TIMES
1511  002762  000441                           BR      TST6            ;;;EXIT THIS TEST
1512  002764                                                                   ENDIF
1513  002764                           $20:
1514                                                           ; TRY TO CLEAR DTR BIT
1515  002764                                                                   BGNSUB
1516  002764  012767  002772  176116           MOV     #64$,$LPERR
1517  002772                                                                       LET     @RCSR := @RCSR CLR.BY #DTR
1518  002772  042777  000002  176260           BIC     #DTR,@RCSR
1519                                                                           ; STUCK TO 0
1520  003000                                                                   IF      #DTR SETIN @RCSR THEN
1521  003000  032777  000002  176252           BIT     #DTR,@RCSR
1522  003006  001401                           BEQ     $21
1523                                                                           ; DTR DID NOT CLEAR IN RCSR
1524  003010                                                                   ERRHRD 16,,DIDNOT
1525  003010  104016                           ERROR   16
1526  003012                                                                   ENDIF
1527  003012                           $21:
1528  003012                                                                   ENDSUB
1529
1530                                                           ; TRY TO SET  DTR
1531  003012                                                                   BGNSUB
1532  003012  012767  003020  176070           MOV     #64$,$LPERR
1533
1534  003020                                                                       LET     @RCSR := @RCSR SET.BY #DTR
1535  003020  052777  000002  176232           BIS     #DTR,@RCSR
1536  003026                                                                   IF      #DTR NOTSETIN @RCSR THEN
1537  003026  032777  000002  176224           BIT     #DTR,@RCSR
1538  003034  001001                           BNE     $22
1539                                                                           ; DTR DID NOT SET IN RCSR
1540  003036                                                                   ERRHRD 17,,DIDNOT
1541  003036  104017                           ERROR   17
1542  003040                                                                   ENDIF
1543  003040                           $22:
1544  003040                                                                   ENDSUB
1545
1546                                                           ; TRY TO CLEAR IT AGAIN
1547  003040                                                                   BGNSUB
1548  003040  012767  003046  176042           MOV     #64$,$LPERR
1549  003046                                                                       LET     @RCSR := @RCSR CLR.BY #DTR
1550  003046  042777  000002  176204           BIC     #DTR,@RCSR
```

```
1551                                                                    ; SHOULD HAVE CLEARED IT
1552  003054                                                          IF      #DTR SETIN @RCSR THEN
1553  003054  032777  000002  176176          BIT     #DTR,@RCSR
1554  003062  001401                          BEQ     $23
1555                                                                    ; DTR DID NOT CLEAR IN RCSR
1556  003064                                                            ERRHRD 20,,DIDNOT
1557  003064  104020                          ERROR   20
1558  003066                                                          ENDIF
1559  003066                          $23:
1560  003066                                                      ENDSUB
1561  003066                                                  ENDTST
1562
1563
1564
1565                            ;;*****************************************************************
```

```
1566                                    ;;*****************************************************************
1567                                    ;*TEST 6          REQSEND - RCSR2          SET, CLEAR, RESET
1568                                    ;*               THIS TEST ASSUMES THAT JUMPER -(FR) IS IN
1569                                    ;;*****************************************************************
1570  003066  000004                   TST6:   SCOPE
1571  003070  012767  000010  176062           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1572  003076  012767  000006  176074           MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1573  003104                                                           IF @#RFD NOTSETIN $USWR THEN
1574  003104  032767  040000  176106           BIT     @#RFD,$USWR
1575  003112  001004                           BNE     $24
1576  003114                                                               EXIT TST
1577  003114  012767  000001  176036           MOV     #1,$TIMES
1578  003122  000452                           BR      TST7            ;;;EXIT THIS TEST
1579  003124                                                           ENDIF
1580  003124                           $24:
1581
1582                                                   ; SEE IF IT IS CLEAR
1583  003124                                                           BGNSUB
1584  003124  012767  003132  175756           MOV     #64$,$LPERR
1585
1586  003132                                                               IF      @REQSEND SETIN @RCSR THEN
1587  003132  032777  000004  176120           BIT     @REQSEND,@RCSR
1588  003140  001401                           BEQ     $25
1589                                                                        ; REQSEND DID NOT RESET IN RCSR
1590  003142                                                                ERRHRD 21,,DIDNOT
1591  003142  104021                           ERROR   21
1592  003144                                                               ENDIF
1593  003144                           $25:
1594  003144                                                           ENDSUB
1595
1596                                                   ; TRY TO SET REQSEND BIT
1597  003144                                                           BGNSUB
1598  003144  012767  003152  175736           MOV     #64$,$LPERR
1599  003152                                                               LET     @RCSR := @RCSR SET.BY @REQSEND
1600  003152  052777  000004  176100           BIS     @REQSEND,@RCSR
1601                                                                     ; STUCK TO 0
1602  003160                                                               IF      @REQSEND NOTSETIN @RCSR THEN
1603  003160  032777  000004  176072           BIT     @REQSEND,@RCSR
1604  003166  001001                           BNE     $26
1605                                                                        ; REQSEND DID NOT SET IN RCSR
1606  003170                                                                ERRHRD 22,,DIDNOT
1607  003170  104022                           ERROR   22
1608  003172                                                               ENDIF
1609  003172                           $26:
1610  003172                                                           ENDSUB
1611
1612                                                   ; TRY TO CLEAR A SET BIT
1613  003172                                                           BGNSUB
1614  003172  012767  003200  175710           MOV     #64$,$LPERR
1615
1616  003200                                                               LET     @RCSR := @RCSR CLR.BY @REQSEND
1617  003200  042777  000004  176052           BIC     @REQSEND,@RCSR
1618                                                                     ; SHOULD HAVE CLEARED
1619  003206                                                               IF      @REQSEND SETIN @RCSR THEN
1620  003206  032777  000004  176044           BIT     @REQSEND,@RCSR
1621  003214  001401                           BEQ     $27
```

```
1622
1623  003216                                                        ; REQSEND DID NOT CLEAR IN RCSR
1624  003216  104023                        ERROR   23             ERRHRD 23,,DIDNOT
1625  003220                                                        ENDIF
1626  003220                          $27:
1627  003220                                                      ENDSUB
1628
1629                                                  ; NOW SEE IF RESET CLEARS IT
1630  003220                                                      BGNSUB
1631  003220  012767  003226  175662        MOV     #64$,$LPERR
1632
1633  003226                                                        LET     @RCSR := @RCSR SET.BY @REQSEND
1634  003226  052777  000004  176024        BIS     #REQSEND,@RCSR
1635                                                              ; ISSUE BUS RESET
1636  003234                                                        BRESET
1637  003234  000005                        RESET
1638  003236                                                        IF      #REQSEND SETIN @RCSR THEN
1639  003236  032777  000004  176014        BIT     #REQSEND,@RCSR
1640  003244  001401                        BEQ     $30
1641
1642  003246                                                        ; REQSEND DID NOT RESET IN RCSR
1643  003246  104024                        ERROR   24             ERRHRD 24,,DIDNOT
1644  003250                                                        ENDIF
1645  003250                          $30:
1646  003250                                                      ENDSUB
1647  003250                                                  ENDTST
1648
1649
1650
1651                            ;;*****************************************************************
```

```
1652                                    ;;*****************************************************************
1653                                    ;*TEST 7         SECXMIT - RCSR3         SET, CLEAR, RESET
1654                                    ;;*****************************************************************
1655  003250  000004                   TST7:   SCOPE
1656  003252  012767  000010  175700           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1657  003260  012767  000007  175712           MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1658                                                   ; SEE IF IT IS CLEAR
1659  003266                                                           BGNSUB
1660  003266  012767  003274  175614           MOV     #64$,$LPERR
1661
1662  003274                                                               IF      #SECXMIT SETIN @RCSR THEN
1663  003274  032777  000010  175756           BIT     #SECXMIT,@RCSR
1664  003302  001401                          BEQ     $31
1665                                                                       ; SECXMIT DID NOT RESET IN RCSR
1666  003304                                                               ERRHRD 25,,DIDNOT
1667  003304  104025                          ERROR   25
1668  003306                                                               ENDIF
1669  003306                                  $31:
1670  003306                                                           ENDSUB
1671
1672                                                   ; TRY TO SET SECXMIT BIT
1673  003306                                                           BGNSUB
1674  003306  012767  003314  175574           MOV     #64$,$LPERR
1675  003314                                                               LET     @RCSR := @RCSR SET.BY #SECXMIT
1676  003314  052777  000010  175736           BIS     #SECXMIT,@RCSR
1677                                                                       ; STUCK TO 0
1678  003322                                                               IF      #SECXMIT NOTSETIN @RCSR THEN
1679  003322  032777  000010  175730           BIT     #SECXMIT,@RCSR
1680  003330  001001                          BNE     $32
1681                                                                       ; SECXMIT DID NOT SET IN RCSR
1682  003332                                                               ERRHRD 26,,DIDNOT
1683  003332  104026                          ERROR   26
1684  003334                                                               ENDIF
1685  003334                                  $32:
1686  003334                                                           ENDSUB
1687
1688                                                   ; TRY TO CLEAR A SET BIT
1689  003334                                                           BGNSUB
1690  003334  012767  003342  175546           MOV     #64$,$LPERR
1691
1692  003342                                                               LET     @RCSR := @RCSR CLR.BY #SECXMIT
1693  003342  042777  000010  175710           BIC     #SECXMIT,@RCSR
1694                                                                       ; SHOULD HAVE CLEARED
1695  003350                                                               IF      #SECXMIT SETIN @RCSR THEN
1696  003350  032777  000010  175702           BIT     #SECXMIT,@RCSR
1697  003356  001401                          BEQ     $33
1698                                                                       ; SECXMIT DID NOT CLEAR IN RCSR
1699  003360                                                               ERRHRD 27,,DIDNOT
1700  003360  104027                          ERROR   27
1701  003362                                                               ENDIF
1702  003362                                  $33:
1703  003362                                                           ENDSUB
1704
1705  003362                                                           BGNSUB
1706  003362  012767  003370  175520           MOV     #64$,$LPERR
1707                                                   ; NOW SEE IF RESET CLEARS IT
```

```
1708
1709  003370                                                    LET     @RCSR := @RCSR SET.BY #SECXMIT
1710  003370  052777  000010  175662          BIS     #SECXMIT,@RCSR
1711                                                        ; ISSUE BUS RESET
1712  003376                                                BRESET
1713  003376  000005                          RESET
1714  003400                                                IF      #SECXMIT SETIN @RCSR THEN
1715  003400  032777  000010  175652          BIT     #SECXMIT,@RCSR
1716  003406  001401                          BEQ     $34
1717                                                        ; SECXMIT DID NOT RESET IN RCSR
1718  003410                                                ERRHRD 30,,DIDNOT
1719  003410  104030                          ERROR   30
1720  003412                                                ENDIF
1721  003412                          $34:
1722  003412                                              ENDSUB
1723  003412                                            ENDTST
1724
1725
1726
1727                             ;;*****************************************************************
```

```
1728                                    ;;*************************************************************
1729                                    ;*TEST 10        DATAIE - RCSR5 SET, CLEAR, RESET
1730                                    ;;*************************************************************
1731  003412  000004                    TST10:  SCOPE
1732  003414  012767  000010  175536            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1733  003422  012767  000010  175550            MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1734                                                            ; SEE IF IT IS CLEAR
1735  003430                                                            BGNSUB
1736  003430  012767  003436  175452            MOV     #64$,$LPERR
1737
1738  003436                                                                IF      #DATAIE SETIN @RCSR THEN
1739  003436  032777  000040  175614            BIT     #DATAIE,@RCSR
1740  003444  001401                            BEQ     $35
1741                                                                    ; DATAIE DID NOT RESET IN RCSR
1742  003446                                                            ERRHRD 31,,DIDNOT
1743  003446  104031                            ERROR   31
1744  003450                                                            ENDIF
1745  003450                                $35:
1746  003450                                                        ENDSUB
1747
1748                                                    ; TRY TO SET DATAIE BIT
1749  003450                                                            BGNSUB
1750  003450  012767  003456  175432            MOV     #64$,$LPERR
1751  003456                                                            LET     @RCSR := @RCSR SET.BY #DATAIE
1752  003456  052777  000040  175574            BIS     #DATAIE,@RCSR
1753                                                                ; STUCK TO 0
1754  003464                                                            IF      #DATAIE NOTSETIN @RCSR THEN
1755  003464  032777  000040  175566            BIT     #DATAIE,@RCSR
1756  003472  001001                            BNE     $36
1757                                                                ; DATAIE DID NOT SET IN RCSR
1758  003474                                                            ERRHRD 32,,DIDNOT
1759  003474  104032                            ERROR   32
1760  003476                                                            ENDIF
1761  003476                                $36:
1762  003476                                                        ENDSUB
1763
1764                                                    ; TRY TO CLEAR A SET BIT
1765  003476                                                            BGNSUB
1766  003476  012767  003504  175404            MOV     #64$,$LPERR
1767
1768  003504                                                            LET     @RCSR := @RCSR CLR.BY #DATAIE
1769  003504  042777  000040  175546            BIC     #DATAIE,@RCSR
1770                                                                ; SHOULD HAVE CLEARED
1771  003512                                                            IF      #DATAIE SETIN @RCSR THEN
1772  003512  032777  000040  175540            BIT     #DATAIE,@RCSR
1773  003520  001401                            BEQ     $37
1774                                                                ; DATAIE DID NOT CLEAR IN RCSR
1775  003522                                                            ERRHRD 33,,DIDNOT
1776  003522  104033                            ERROR   33
1777  003524                                                            ENDIF
1778  003524                                $37:
1779  003524                                                        ENDSUB
1780
1781                                                    ; NOW SEE IF RESET CLEARS IT
1782  003524                                                            BGNSUB
1783  003524  012767  003532  175356            MOV     #64$,$LPERR
```

```
1784
1785  003532                                                        LET     @RCSR := @RCSR SET.BY #DATAIF
1786  003532  052777  000040  175520          BIS     #DATAIE,@RCSR
1787                                                                ; ISSUE BUS RESET
1788  003540                                                        BRESET
1789  003540  000005                          RESET
1790  003542                                                        IF      #DATAIE SETIN @RCSR THEN
1791  003542  032777  000040  175510          BIT     #DATAIE,@RCSR
1792  003550  001401                          BEQ     $40
1793                                                                ; DATAIE DID NOT RESET IN RCSR
1794  003552                                                        ERRHRD 34,,DIDNOT
1795  003552  104034                          ERROR   34
1796  003554                                                        ENDIF
1797  003554                          $40:
1798  003554                                                    ENDSUB
1799  003554                                                ENDTST
1800
1801
1802
1803                            ;;*****************************************************************
```

```
1804                                    ;;*****************************************************************
1805                                    ;*TEST 11        RCVRIE - RCSR6 SET, CLEAR, RESET
1806                                    ;;*****************************************************************
1807  003554  000004                    TST11:  SCOPE
1808  003556  012767  000010  175374            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1809  003564  012767  000011  175406            MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1810                                                    ; SEE IF IT IS CLEAR
1811  003572                                                            BGNSUB
1812  003572  012767  003600  175310            MOV     #64$,$LPERR
1813
1814  003600                                                                IF      #RCVRIE SETIN @RCSR THEN
1815  003600  032777  000100  175452            BIT     #RCVRIE,@RCSR
1816  003606  001401                            BEQ     $41
1817                                                                        ; RCVRIE DID NOT RESET IN RCSR
1818  003610                                                                ERRHRD 35,,DIDNOT
1819  003610  104035                            ERROR   35
1820  003612                                                                ENDIF
1821  003612                                $41:
1822  003612                                                            ENDSUB
1823
1824                                                    ; TRY TO SET RCVRIE BIT
1825  003612                                                            BGNSUB
1826  003612  012767  003620  175270            MOV     #64$,$LPERR
1827  003620                                                                LET     @RCSR := @RCSR SET.BY #RCVRIE
1828  003620  052777  000100  175432            BIS     #RCVRIE,@RCSR
1829                                                                      ; STUCK TO 0
1830  003626                                                                IF      #RCVRIE NOTSETIN @RCSR THEN
1831  003626  032777  000100  175424            BIT     #RCVRIE,@RCSR
1832  003634  001001                            BNE     $42
1833                                                                        ; RCVRIE DID NOT SET IN RCSR
1834  003636                                                                ERRHRD 36,,DIDNOT
1835  003636  104036                            ERROR   36
1836  003640                                                                ENDIF
1837  003640                                $42:
1838  003640                                                            ENDSUB
1839
1840                                                    ; TRY TO CLEAR A SET BIT
1841  003640                                                            BGNSUB
1842  003640  012767  003646  175242            MOV     #64$,$LPERR
1843
1844  003646                                                                LET     @RCSR := @RCSR CLR.BY #RCVRIE
1845  003646  042777  000100  175404            BIC     #RCVRIE,@RCSR
1846                                                                      ; SHOULD HAVE CLEARED
1847  003654                                                                IF      #RCVRIE SETIN @RCSR THEN
1848  003654  032777  000100  175376            BIT     #RCVRIE,@RCSR
1849  003662  001401                            BEQ     $43
1850                                                                        ; RCVRIE DID NOT CLEAR IN RCSR
1851  003664                                                                ERRHRD 37,,DIDNOT
1852  003664  104037                            ERROR   37
1853  003666                                                                ENDIF
1854  003666                                $43:
1855  003666                                                            ENDSUB
1856
1857                                                    ; NOW SEE IF RESET CLEARS IT
1858  003666                                                            BGNSUB
1859  003666  012767  003674  175214            MOV     #64$,$LPERR
```

```
1860
1861  003674                                                    LET     @RCSR := @RCSR SET.BY #RCVRIE
1862  003674  052777  000100  175356          BIS     #RCVRIE,@RCSR
1863                                                            ; ISSUE BUS RESET
1864  003702                                                    BRESET
1865  003702  000005                          RESET
1866  003704                                                    IF      #RCVRIE SETIN @RCSR THEN
1867  003704  032777  000100  175346          BIT     #RCVRIE,@RCSR
1868  003712  001401                          BEQ     $44
1869                                                                ; RCVRIE DID NOT RESET IN RCSR
1870  003714                                                        ERRHRD 40,,DIDNOT
1871  003714  104040                          ERROR   40
1872  003716                                                    ENDIF
1873  003716                          $44:
1874  003716                                                    CKLOOP
1875  003716                                                ENDSUB
1876  003716                                            ENDTST
1877
1878
1879
1880
1881                            ;;**********************************************************************
1882                            ;* THE FOLLOWING 4 TESTS VERIFY
1883                            ;* THAT RESET (INIT) INITIALIZES READ ONLY BITS.
1884                            ;;**********************************************************************
1885
1886
```

```
1887                                    ;;*****************************************************************
1888                                    ;*TEST 12        TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
1889                                    ;;*****************************************************************
1890  003716  000004                    TST12:  SCOPE
1891  003720  012767  000010  175232            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1892  003726  012767  000012  175244            MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1893
1894
1895
1896
1897
1898  003734                                                                    BGNSUB
1899  003734  012767  003742  175146            MOV     #64$,$LPERR
1900  003742                                                                        IF      #RCVRDONE SETIN @RCSR THEN
1901  003742  032777  000200  175310            BIT     #RCVRDONE,@RCSR
1902  003750  001402                            BEQ     $45
1903
1904                                                                                  ;RCVRDONE SHOULD HAVE CLEARED BY INIT
1905                                                                                  ; RCVRDONE DID NOT CLEAR IN RCSR
1906  003752                                                                          ERRHRD 41,HRESET, DIDNOT
1907  003752  104041                            ERROR   41
1908                                                                                  ;REISSUE RESET
1909  003754                                                                          BRESET
1910  003754  000005                            RESET
1911  003756                                                                        ENDIF
1912  003756                                $45:
1913                                                                                  ;ALLOW LOOPING AFTER ERROR
1914  003756                                                                        CKLOOP
1915  003756                                                                    ENDSUB
1916  003756                                                                ENDTST
1917
1918
1919
```

```
1920                                    ;;**********************************************************************
1921                                    ;*TEST 13        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
1922                                    ;;**********************************************************************
1923  003756  000004                    TST13:  SCOPE
1924  003760  012767  000010  175172            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1925  003766  012767  000013  175204            MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1926
1927
1928
1929
1930  003774                                                                    BGNSUB
1931  003774  012767  004002  175106            MOV     #64$,$LPERR
1932
1933  004002                                                                        IF      #XMITRDY NOTSETIN @TCSR THEN
1934  004002  032777  000200  175254            BIT     #XMITRDY,@TCSR
1935  004010  001002                            BNE     $46
1936
1937                                                                              ;RESET SHOULD HAVE SET BIT.
1938                                                                              ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
1939  004012                                                                      ERRHRD 42,HRESET,DIDNOT
1940  004012  104042                            ERROR   42
1941                                                                              ;ISSUE ANOTHER RESET
1942  004014                                                                      BRESET
1943  004014  000005                            RESET
1944  004016                                                                    ENDIF
1945  004016                                $46:
1946                                                                              ;ALLOW LOOPING ON ERROR
1947  004016                                                                    CKLOOP
1948  004016                                                                ENDSUB
1949  004016                                                            ENDTST
1950
1951
1952
```

```
1953                                    ;;**********************************************************************
1954                                    ;*TEST 14        TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.
1955                                    ;;**********************************************************************
1956  004016  000004                   TST14:  SCOPE
1957  004020  012767  000010  175132           MOV     #10.,$TIMES      ;;DO 10 ITERATIONS
1958  004026  012767  000014  175144           MOV     #14,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1959
1960
1961
1962
1963  004034                                                        BGNSUB
1964  004034  012767  004042  175046           MOV     #64$,$LPERR
1965  004042                                                            IF      #DATAINT SETIN @RCSR THEN
1966  004042  032777  100000  175210           BIT     #DATAINT,@RCSR
1967  004050  001402                           BEQ     $47
1968
1969  004052                                                                ERRHRD 43, HRESET, DIDNOT
1970  004052  104043                           ERROR   43
1971
1972                                                                      ;TESTING EFFECT OF RESET ON BIT
1973
1974                                                                      ;DATAINT DID NOT CLEAR IN RCSR
1975                                                                      ;ALLOW A FRESH START
1976  004054                                                              BRESET
1977  004054  000005                           RESET
1978  004056                                                            ENDIF
1979  004056                                  $47:
1980  004056                                                            CKLOOP
1981  004056                                                        ENDSUB
1982  004056                                                    ENDTST
1983
1984
1985
1986                                    ;;**********************************************************************
```

```
1987                                    ;;**********************************************************************
1988                                    ;*TEST 15        TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT
1989                                    ;;**********************************************************************
1990  004056  000004                    TST15:  SCOPE
1991  004060  012767  000010  175072            MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
1992  004066  012767  000015  175104            MOV     #15.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1993
1994
1995  004074                                                            IF      @CABLE NOTSETIN $USWR THEN
1996  004074  032767  020000  175116            BIT     @CABLE,$USWR
1997  004102  001004                            BNE     $50
1998                                                                        ; CAN'T TEST WITHOUT BERG OR M315.
1999  004104                                                            EXIT TST
2000  004104  012767  000001  175046            MOV     #1,$TIMES
2001  004112  000411                            BR      TST16           ;;;EXIT THIS TEST
2002  004114                                                            ENDIF
2003  004114                            $50:
2004
2005
2006
2007  004114                                                            BGNSUB
2008  004114  012767  004122  174766            MOV     @64$,$LPERR
2009
2010  004122                                                                IF      @RCVRACT SETIN @RCSR THEN
2011  004122  032777  004000  175130            BIT     @RCVRACT,@RCSR
2012  004130  001402                            BEQ     $51
2013
2014                                                                        ;RESET SHOULD HAVE CLEARED RCVRACT
2015  004132                                                                ERRHRD 44, HRESET, DIDNOT
2016  004132  104044                            ERROR   44
2017
2018                                                                        ;TESTING EFFECT OF RESET ON BIT
2019
2020                                                                        ;RCVRACT DID NOT CLEAR IN RCSR
2021
2022                                                                        ;ALLOW ANOTHER TRY
2023  004134                                                                BRESET
2024  004134  000005                            RESET
2025  004136                                                            ENDIF
2026  004136                            $51:
2027                                                                        ;ALLOW LOOPING ON ERROR
2028  004136                                                            CKLOOP
2029  004136                                                        ENDSUB
2030  004136                                                    ENDTST
2031
```

```
2032                                ;;**********************************************************************
2033                                ;* THE FOLLOWING 4 TESTS VERIFY
2034                                ;* THAT THE EIA SIGNALS CAN BE TRANSMITTED
2035                                ;* AND RECEIVED THROUGH THE CABLE
2036                                ;;**********************************************************************
2037
2038
2039
2040                                ;;**********************************************************************
2041                                ;*TEST 16        TEST THAT CARDET SETS AND CLEARS
2042                                ;*              AS  DTR   SETS AND CLEARS
2043                                ;*              THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
2044                                ;;**********************************************************************
2045  004136  000004                TST16:  SCOPE
2046  004140  012767  000010  175012        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2047  004146  012767  000016  175024        MOV     #16.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2048                                                                        ; CAN WE USE THE WRAPAROUND??
2049  004154                                                            IF      #CABLE+FRFD NOTSETIN $USWR THEN
2050  004154  032767  060000  175036        BIT     #CABLE+FRFD,$USWR
2051  004162  001004                        BNE     $52
2052                                                                        ; CAN'T TEST WITHOUT BERG OR H315
2053                                                                        ; OR WITH (-FD) JUMPER OUT.
2054                                                                        ; OR WITH (-FR) JUMPER OUT.
2055  004164                                                            EXIT TST
2056  004164  012767  000001  174766        MOV     #1,$TIMES
2057  004172  000441                        BR      TST17           ;;;EXIT THIS TEST
2058  004174                                                            ENDIF
2059  004174                        $52:
2060
2061                                                                        ; DTR AND
2062                                                                        ; CARDET ARE CONNECTED
2063
2064                                                                    ; BY THE H315 OR EQUIV.
2065
2066                                                    ; CLEAR
2067  004174                                                    BGNSUB
2068  004174  012767  004202  174706        MOV     #64$,$LPERR
2069
2070                                                                        ; CLEAR  DTR
2071  004202                                                            LET     @RCSR := @RCSR CLR.BY #DTR
2072  004202  042777  000002  175050        BIC     #DTR,@RCSR
2073                                                                        ; CARDET SHOULD FOLLOW
2074  004210                                                            IF      #CARDET SETIN @RCSR THEN
2075  004210  032777  010000  175042        BIT     #CARDET,@RCSR
2076  004216  001401                        BEQ     $53
2077                                                                        ; CARDET DID NOT
2078  004220                                                                ERRHRD 45,,FORCE
2079  004220  104045                        ERROR   45
2080
2081                                                                        ;  CLEAR WITH DTR
2082  004222                                                            ENDIF
2083  004222                        $53:
2084  004222                                                        ENDSUB
2085
2086                                                    ; SET
2087  004222                                                        BGNSUB
```

```
2088  004222  012767  004230  174660          MOV     #64$,$LPERR
2089
2090
2091  004230                                                      ; SET DTR
2092  004230  052777  000002  175022          BIS     #DTR,@RCSR  LET   @RCSR := @RCSR SET.BY #DTR
2093                                                              ; CARDET SHOULD FOLLOW
2094  004236                                                      IF    #CARDET NOTSETIN @RCSR THEN
2095  004236  032777  010000  175014          BIT     #CARDET,@RCSR
2096  004244  001001                          BNE     $54
2097                                                                ; CARDET DID NOT SET
2098  004246                                                        ERRHRD 46,,FORCE
2099  004246  104046                          ERROR   46
2100
2101                                                              ;  WITH DTR
2102  004250                                                      ENDIF
2103  004250                          $54:
2104  004250                                                  ENDSUB
2105
2106                                                                  ; CLEAR
2107  004250                                                  BGNSUB
2108  004250  012767  004256  174632          MOV     #64$,$LPERR
2109
2110                                                              ; CLEAR  DTR
2111  004256                                                      LET   @RCSR := @RCSR CLR.BY #DTR
2112  004256  042777  000002  174774          BIC     #DTR,@RCSR
2113                                                              ; CARDET SHOULD FOLLOW
2114  004264                                                      IF    #CARDET SETIN @RCSR THEN
2115  004264  032777  010000  174766          BIT     #CARDET,@RCSR
2116  004272  001401                          BEQ     $55
2117                                                                ; CARDET DID NOT
2118  004274                                                        ERRHRD 47,,FORCE
2119  004274  104047                          ERROR   47
2120
2121                                                              ;  CLEAR WITH DTR
2122  004276                                                      ENDIF
2123  004276                          $55:
2124  004276                                                  ENDSUB
2125  004276                                              ENDTST
2126
2127
2128
2129
```

```
2130                                    ;;*****************************************************************
2131                                    ;;*****************************************************************
2132                                    ;*TEST 17        TEST THAT CLRSEND SETS AND CLEARS
2133                                    ;*              AS  DTR   SETS AND CLEARS
2134                                    ;*              (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK
2135                                    ;;*****************************************************************
2136  004276  000004                   TST17:  SCOPE
2137  004300  012767  000010  174652           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2138  004306  012767  000017  174664           MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2139                                                                            ; CAN WE USE THE WRAPAROUND??
2140  004314                                                                IF      @CABLE*FRFD NOTSETIN $USWR THEN
2141  004314  032767  060000  174676           BIT     @CABLE*FRFD,$USWR
2142  004322  001004                          BNE     $56
2143                                                                            ; CAN'T TEST WITHOUT BERG OR M315
2144  004324                                                                EXIT TST
2145  004324  012767  000001  174626           MOV     #1,$TIMES
2146  004332  000441                          BR      TST20           ;;;EXIT THIS TEST
2147  004334                                                                ENDIF
2148  004334                           $56:
2149
2150                                                                            ; DTR AND
2151                                                                            ; CLRSEND ARE CONNECTED
2152
2153                                                                    ; BY THE M315 OR EQUIV.
2154
2155                                                    ; CLEAR
2156  004334                                                                BGNSUB
2157  004334  012767  004342  174546           MOV     #64$,$LPERR
2158
2159                                                                            ; CLEAR  DTR
2160  004342                                                                LET     @RCSR :* @RCSR CLR.BY @DTR
2161  004342  042777  000002  174710           BIC     @DTR,@RCSR
2162                                                                            ; CLRSEND SHOULD FOLLOW
2163  004350                                                                IF      @CLRSEND SETIN @RCSR THEN
2164  004350  032777  020000  174702           BIT     @CLRSEND,@RCSR
2165  004356  001401                          BEQ     $57
2166                                                                            ; CLRSEND DID NOT
2167  004360                                                                ERRMRD 50.,FORCE
2168  004360  104050                          ERROR   50
2169
2170                                                                            ;  CLEAR WITH DTR
2171  004362                                                                ENDIF
2172  004362                           $57:
2173  004362                                                                ENDSUB
2174
2175                                                    ; SET
2176  004362                                                                BGNSUB
2177  004362  012767  004370  174520           MOV     #64$,$LPERR
2178
2179                                                                            ; SET DTR
2180  004370                                                                LET     @RCSR :* @RCSR SET.BY @DTR
2181  004370  052777  000002  174662           BIS     @DTR,@RCSR
2182                                                                            ; CLRSEND SHOULD FOLLOW
2183  004376                                                                IF      @CLRSEND NOTSETIN @RCSR THEN
2184  004376  032777  020000  174654           BIT     @CLRSEND,@RCSR
2185  004404  001001                          BNE     $60
```

```
2186                                                                    ; CLRSEND DID NOT SET
2187  004406                                                            ERRHRD 51,,FORCE
2188  004406  104051                       ERROR   51
2189
2190                                                                    ;  WITH DTR
2191  004410                                                            ENDIF
2192  004410                        $60:
2193  004410                                                          ENDSUB
2194
2195                                                  ; CLEAR
2196  004410                                                          BGNSUB
2197  004410  012767  004416  174472       MOV     #64$,$LPERR
2198
2199                                                                    ; CLEAR  DTR
2200  004416                                                            LET     @RCSR := @RCSR CLR.BY #DTR
2201  004416  042777  000002  174634       BIC     #DTR,@RCSR
2202                                                                    ; CLRSEND SHOULD FOLLOW
2203  004424                                                            IF      #CLRSEND SETIN @RCSR THEN
2204  004424  032777  020000  174626       BIT     #CLRSEND,@RCSR
2205  004432  001401                       BEQ     $61
2206                                                                    ; CLRSEND DID NOT
2207  004434                                                            ERRHRD 52,,FORCE
2208  004434  104052                       ERROR   52
2209
2210                                                                    ;  CLEAR WITH DTR
2211  004436                                                            ENDIF
2212  004436                        $61:
2213  004436                                                          ENDSUB
2214  004436                                                  ENDTST
2215
2216
2217
2218
```

```
2219                                    ;;*****************************************************************
2220                                    ;;*****************************************************************
2221                                    ;*TEST 20        TEST THAT RING SETS AND CLEARS
2222                                    ;*              AS  REQSEND    SETS AND CLEARS
2223                                    ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
2224                                    ;;*****************************************************************
2225  004436  000004                   TST20:  SCOPE
2226  004440  012767  000010  174512           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2227  004446  012767  000020  174524           MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2228                                                                           ; CAN WE USE THE WRAPAROUND??
2229  004454                                                                 IF      #CABLE+FRFD NOTSETIN $USWR THEN
2230  004454  032767  060000  174536           BIT     #CABLE+FRFD,$USWR
2231  004462  001004                          BNE     $62
2232                                                                           ; CAN'T TEST WITHOUT BERG OR H315
2233                                                                           ; OR WITH (-FR) JUMPER OUT.
2234  004464                                                                 EXIT TST
2235  004464  012767  000001  174466          MOV     #1,$TIMES
2236  004472  000441                          BR      TST21           ;;;EXIT THIS TEST
2237  004474                                                                 ENDIF
2238  004474                          $62:
2239
2240                                                                           ; REQSEND AND
2241                                                                           ; RING ARE CONNECTED
2242
2243                                                                    ; BY THE H315 OR EQUIV.
2244
2245                                                    ; CLEAR
2246  004474                                                           BGNSUB
2247  004474  012767  004502  174406          MOV     #64$,$LPERR
2248
2249                                                                        ; CLEAR  REQSEND
2250  004502                                                             LET     @RCSR := @RCSR CLR.BY #REQSEND
2251  004502  042777  000004  174550          BIC     #REQSEND,@RCSR
2252                                                                        ; RING SHOULD FOLLOW
2253  004510                                                             IF      #RING SETIN @RCSR THEN
2254  004510  032777  040000  174542          BIT     #RING,@RCSR
2255  004516  001401                          BEQ     $63
2256                                                                        ; RING DID NOT
2257  004520                                                               ERRHRD 53,,FORCE
2258  004520  104053                          ERROR   53
2259
2260                                                                        ;  CLEAR WITH REQSEND
2261  004522                                                             ENDIF
2262  004522                          $63:
2263  004522                                                           ENDSUB
2264
2265                                                            ; SET
2266  004522                                                           BGNSUB
2267  004522  012767  004530  174360          MOV     #64$,$LPERR
2268
2269                                                                        ; SET REQSEND
2270  004530                                                             LET     @RCSR := @RCSR SET.BY #REQSEND
2271  004530  052777  000004  174522          BIS     #REQSEND,@RCSR
2272                                                                        ; RING SHOULD FOLLOW
2273  004536                                                             IF      #RING NOTSETIN @RCSR THEN
2274  004536  032777  040000  174514          BIT     #RING,@RCSR
```

```
004544  001001                        BNE     $64
                                                                          ; RING DID NOT SET
004546                                                                    ERRHRD 54,,FORCE
004546  104054                        ERROR   54

                                                                       ;  WITH REQSEND
004550                                                                 ENDIF
004550                     $64:
004550                                                              ENDSUB

                                                  ; CLEAR
004550                                                              BGNSUB
004550  012767  004556  174332        MOV     #64$,$LPERR

                                                                       ; CLEAR  REQSEND
004556                                                                 LET     @RCSR := @RCSR CLR.BY #REQSEND
004556  042777  000004  174474        BIC     #REQSEND,@RCSR

                                                                       ; RING SHOULD FOLLOW
004564                                                                 IF      #RING SETIN @RCSR THEN
004564  032777  040000  174466        BIT     #RING,@RCSR
004572  001401                        BEQ     $65

                                                                          ; RING DID NOT
004574                                                                    ERRHRD 55,,FORCE
004574  104055                        ERROR   55

                                                                       ;  CLEAR WITH REQSEND
004576                                                                 ENDIF
004576                     $65:
004576                                                              ENDSUB
004576                                                           ENDTST
```

```
2309                                    ;;*****************************************************************
2310                                    ;;*****************************************************************
2311                                    ;*TEST 21        TEST THAT SECREC SETS AND CLEARS
2312                                    ;*              AS  SECXMIT    SETS AND CLEARS
2313                                    ;;*****************************************************************
2314  004576  000004                    TST21:  SCOPE
2315  004600  012767  000010  174352            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2316  004606  012767  000021  174364            MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2317                                                                            ; CAN WE USE THE WRAPAROUND??
2318  004614                                                                IF      @CABLE NOTSETIN $USWR THEN
2319  004614  032767  020000  174376            BIT     @CABLE,$USWR
2320  004622  001004                            BNE     $66
2321                                                                                ; CAN'T TEST WITHOUT BERG OR H315.
2322  004624                                                                    EXIT TST
2323  004624  012767  000001  174326            MOV     #1,$TIMES
2324  004632  000441                            BR      TST22           ;;;EXIT THIS TEST
2325  004634                                                                ENDIF
2326  004634                                $66:
2327
2328                                                                            ; SECXMIT AND
2329                                                                            ; SECREC ARE CONNECTED
2330
2331                                                                        ; BY THE H315 OR EQUIV.
2332
2333                                                        ; CLEAR
2334  004634                                                                BGNSUB
2335  004634  012767  004642  174246            MOV     #64$,$LPERR
2336
2337                                                                            ; CLEAR  SECXMIT
2338  004642                                                                    LET     @RCSR := @RCSR CLR.BY @SECXMIT
2339  004642  042777  000010  174410            BIC     @SECXMIT,@RCSR
2340                                                                            ; SECREC SHOULD FOLLOW
2341  004650                                                                    IF      @SECREC SETIN @RCSR THEN
2342  004650  032777  002000  174402            BIT     @SECREC,@RCSR
2343  004656  001401                            BEQ     $67
2344                                                                                ; SECREC DID NOT
2345  004660                                                                        ERRHRD 56,,FORCE
2346  004660  104056                            ERROR   56
2347                                                                            ;  CLEAR WITH SECXMIT
2348                                                                        ENDIF
2349  004662
2350  004662                                $67:
2351  004662                                                                ENDSUB
2352
2353                                                        ; SET
2354  004662                                                                BGNSUB
2355  004662  012767  004670  174220            MOV     #64$,$LPERR
2356
2357                                                                            ; SET SECXMIT
2358  004670                                                                    LET     @RCSR := @RCSR SET.BY @SECXMIT
2359  004670  052777  000010  174362            BIS     @SECXMIT,@RCSR
2360                                                                            ; SECREC SHOULD FOLLOW
2361  004676                                                                    IF      @SECREC NOTSETIN @RCSR THEN
2362  004676  032777  002000  174354            BIT     @SECREC,@RCSR
2363  004704  001001                            BNE     $70
2364                                                                                ; SECREC DID NOT SET
```

```
2365  004706                                                          ERRHRD 57,,FORCE
2366  004706  104057                      ERROR   57
2367
2368                                                                  ;  WITH SECXMIT
2369  004710                                                          ENDIF
2370  004710                      $70:
2371  004710                                                      ENDSUB
2372
2373                                                  ; CLEAR
2374  004710                                                      BGNSUB
2375  004710  012767  004716  174172      MOV     #64$,$LPERR
2376
2377                                                              ; CLEAR  SECXMIT
2378  004716                                                      LET     @RCSR := @RCSR CLR.BY #SECXMIT
2379  004716  042777  000010  174334      BIC     #SECXMIT,@RCSR
2380                                                              ; SECREC SHOULD FOLLOW
2381  004724                                                      IF      #SECREC SETIN @RCSR THEN
2382  004724  032777  002000  174326      BIT     #SECREC,@RCSR
2383  004732  001401                      BEQ     $71
2384
2385  004734                                                          ; SECREC DID NOT
2386  004734  104060                      ERROR   60                 ERRHRD 60,,FORCE
2387
2388                                                                  ;  CLEAR WITH SECXMIT
2389  004736                                                          ENDIF
2390  004736                      $71:
2391  004736                                                      ENDSUB
2392  004736                                                  ENDTST
2393
2394
2395
2396
```

```
2397                                    ;;*****************************************************************
2398                                    ;;*****************************************************************
2399                                    ;*TEST 22        TEST THAT DATAINT (RCSR-15) SETS
2400                                    ;*              WHEN DTR CHANGES STATE
2401                                    ;*              AND THAT DATAINT IS CLEARED AFTER READING RCSR
2402                                    ;*              NOTE DTR IS TIED TO BOTH CARDET AND CLRSEND BY THE H315
2403                                    ;*              THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
2404                                    ;;*****************************************************************
2405  004736  000004                    TST22:  SCOPE
2406  004740  012767  000010  174212            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2407  004746  012767  000022  174224            MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2408                                                                            ; CAN WE USE THE WRAPAROUND??
2409  004754                                                            IF      #CABLE+FRFD NOTSETIN $USWR THEN
2410  004754  032767  060000  174236            BIT     #CABLE+FRFD,$USWR
2411  004762  001004                            BNE     $72
2412                                                                            ; CAN'T TEST WITHOUT BERG OR H315
2413                                                                            ; OR WITH (-FD) JUMPER OUT.
2414  004764                                                            EXIT TST
2415  004764  012767  000001  174166            MOV     #1,$TIMES
2416  004772  000463                            BR      TST23           ;;;EXIT THIS TEST
2417  004774                                                            ENDIF
2418  004774                            $72:
2419
2420                                                                            ;MAKE SURE NOTHING UNEXPECTED HAPPENS
2421  004774  012746  000340                    MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
2422  005000  012746  005006                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
2423  005004  000002                            RTI                     ;;POP NEW PC AND PS
2424  005006                            64$:
2425
2426                                                    ;READ TWICE - CLEARS
2427  005006                                                            BGNSUB
2428  005006  012767  005014  174074            MOV     #65$,$LPERR
2429                                                                            ; CLEAR DTR
2430  005014                                                            LET     @RCSR := @RCSR CLR.BY #DTR
2431  005014  042777  000002  174236            BIC     #DTR,@RCSR
2432                                                                            ;WAIT 1 MILLI-SEC FOR CABLE
2433  005022                                                            WAITMS  1
2434  005022  010546                            MOV     R5,-(SP)
2435  005024  012745  000001                    MOV     #1,-(R5)
2436  005030  004767  004716                    JSR     PC,WAIT
2437  005034  012605                            MOV     (SP)+,R5
2438                                                                            ; READ RCSR - TO CLEAR DATAINT
2439  005036                                                            LET     R3 := @RCSR
2440  005036  017703  174216                    MOV     @RCSR,R3
2441                                                                            ; READ RCSR AGAIN
2442  005042                                                            IF      #DATAINT SETIN @RCSR THEN
2443  005042  032777  100000  174210            BIT     #DATAINT,@RCSR
2444  005050  001401                            BEQ     $73
2445                                                                            ; READING RCSR DID NOT CLEAR DATAINT
2446  005052                                                            ERRHRD 61,EDATAINT
2447  005052  104061                            ERROR   61
2448  005054                                                            ENDIF
2449  005054                            $73:
2450
2451  005054                                                    ENDSUB
2452
```

```
2453                                                    ; DTR SETTING SETS DATAINT
2454  005054                                                BGNSUB
2455  005054  012767  005062  174026          MOV     #64$,$LPERR
2456
2457                                                        ;SET DTR
2458  005062                                              LET     @RCSR  := @RCSR SET.BY #DTR
2459  005062  052777  000002  174170          BIS     #DTR,@RCSR
2460  005070                                              IF      #DATAINT NOTSETIN @RCSR THEN
2461  005070  032777  100000  174162          BIT     #DATAINT,@RCSR
2462  005076  001001                          BNE     $74
2463
2464  005100                                                  ;SETTING DTR DID NOT SET DATAINT
2465  005100  104062                          ERROR   62     ERRHRD 62,, E2DATA
2466  005102                                              ENDIF
2467  005102                          $74:
2468
2469  005102                                              IF      #DATAINT SETIN @RCSR THEN
2470  005102  032777  100000  174150          BIT     #DATAINT,@RCSR
2471  005110  001401                          BEQ     $75
2472
2473  005112                                                  ;READING RCSR DID NOT CLEAR DATAINT
2474  005112  104063                          ERROR   63     ERRHRD 63,EDATAINT
2475  005114                                              ENDIF
2476  005114                          $75:
2477  005114                                            ENDSUB
2478
2479                                                    ; DTR CLEARING SETS DATAINT
2480  005114                                                BGNSUB
2481  005114  012767  005122  173766          MOV     #64$,$LPERR
2482
2483                                                        ;CLEAR DTR
2484  005122                                              LET     @RCSR := @RCSR CLR.BY #DTR
2485  005122  042777  000002  174130          BIC     #DTR,@RCSR
2486  005130                                            IF        #DATAINT NOTSETIN @RCSR THEN
2487  005130  032777  100000  174122          BIT     #DATAINT,@RCSR
2488  005136  001001                          BNE     $76
2489
2490  005140                                                  ;CLEARING DTR DID NOT SET DATAINT
2491  005140  104064                          ERROR   64     ERRHRD 64,,E2DATA
2492  005142                                              ENDIF
2493  005142                          $76:
2494  005142                                          ENDSUB
2495  005142                                      ENDTST
2496
2497
2498
```

```
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;*TEST 23        TEST THAT DATAINT SETS WHEN RING SETS
                                        ;*              AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
                                        ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
                                        ;;*****************************************************************
005142  000004                TST23:  SCOPE
005144  012767  000010  174006        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005152  012767  000023  174020        MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; CAN WE USE THE WRAPAROUND??
005160                                                      IF      #CABLE+FRFD NOTSETIN $USWR THEN
005160  032767  060000  174032        BIT     #CABLE+FRFD,$USWR
005166  001004                        BNE     $77
                                                                ; CAN'T TEST WITHOUT BERG OR H315
                                                                ; OR WITH (-FR) JUMPER OUT.
005170                                                      EXIT TST
005170  012767  000001  173762        MOV     #1,$TIMES
005176  000473                        BR      TST24           ;;;EXIT THIS TEST
005200                                                      ENDIF
005200                          $77:
                                                                ;NO INTERRUPTS
005200  012746  000340                MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
005204  012746  005212                MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005210  000002                        RTI                     ;;POP NEW PC AND PS
005212                          64$:
                                                    ;START OFF WITH EVERYTHING CLEAR
005212                                                      BGNSUB
005212  012767  005220  173670        MOV     #65$,$LPERR
                                                                ;CLEAR RING
005220                                                  LET     @RCSR := @RCSR CLR.BY #REQSEND
005220  042777  000004  174032        BIC     #REQSEND,@RCSR
                                                                ;WAIT 1 MILLI-SEC FOR CABLE
005226                                                  WAITMS  1
005226  010546                        MOV     R5,-(SP)
005230  012745  000001                MOV     #1,-(R5)
005234  004767  004512                JSR     PC,WAIT
005240  012605                        MOV     (SP)+,R5
                                                                ;READ ONCE
005242                                                  LET     R3 := @RCSR
005242  017703  174012                MOV     @RCSR,R3
                                                                ;READ TWICE
005246                                                  IF      #DATAINT SETIN @RCSR THEN
005246  032777  100000  174004        BIT     #DATAINT,@RCSR
005254  001401                        BEQ     $100
                                                                ;READING RCSR DID NOT CLEAR DATAINT
005256                                                          ERRHRD 65, EDATAINT
005256  104065                        ERROR   65
005260                                                  ENDIF
005260                          $100:
005260                                                  ENDSUB
                                                ;       SET RING --> SET DATAINT
005260                                                  BGNSUB
```

```
2555  005260  012767  005266  173622          MOV     #64$,$LPERR
2556
2557                                                        ;WE ARE SETTING RING
2558                                                        ; SET RING
2559  005266                                              LET     @RCSR := @RCSR SET.BY #REQSEND
2560  005266  052777  000004  173764          BIS     #REQSEND,@RCSR
2561                                                        ;WAIT 1 MILLI-SEC FOR CABLE
2562  005274                                              WAITMS  1
2563  005274  010546                          MOV     R5,-(SP)
2564  005276  012745  000001                  MOV     #1,-(R5)
2565  005302  004767  004444                  JSR     PC,WAIT
2566  005306  012605                          MOV     (SP)+,R5
2567  005310                                              IF      #DATAINT NOTSETIN @RCSR THEN
2568  005310  032777  100000  173742          BIT     #DATAINT,@RCSR
2569  005316  001001                          BNE     $101
2570                                                        ;SETTING RING DID NOT SET DATAINT
2571  005320                                              ERRHRD 122,, E2DATA
2572  005320  104122                          ERROR   122
2573  005322                                              ENDIF
2574  005322                          $101:
2575  005322                                          ENDSUB
2576
2577                                                ;CLEAR RING CAUSES NO CHANGE IN DATAINT (CLEAR)
2578  005322                                          BGNSUB
2579  005322  012767  005330  173560          MOV     #64$,$LPERR
2580
2581                                                        ;CLEAR RING
2582  005330                                              LET     @RCSR := @RCSR CLR.BY #REQSEND
2583  005330  042777  000004  173722          BIC     #REQSEND,@RCSR
2584                                                        ;WAIT 1 MILLI-SEC FOR CABLE
2585  005336                                              WAITMS  1
2586  005336  010546                          MOV     R5,-(SP)
2587  005340  012745  000001                  MOV     #1,-(R5)
2588  005344  004767  004402                  JSR     PC,WAIT
2589  005350  012605                          MOV     (SP)+,R5
2590  005352                                              IF      #DATAINT SETIN @RCSR THEN
2591  005352  032777  100000  173700          BIT     #DATAINT,@RCSR
2592  005360  001401                          BEQ     $102
2593                                                        ;CLEARING RING SET DATAINT
2594  005362                                              ERRHRD 123, CLRRNG
2595  005362  104123                          ERROR   123
2596  005364                                              ENDIF
2597  005364                          $102:
2598  005364                                          ENDSUB
2599  005364                                          EXIT    ;SKIP AROUND MESSAGE
2600  005364  000400                          BR      TST24           ;;;EXIT THIS TEST
2601  005366                                      ENDTST
2602
2603
2604
```

```
2605                                    ;;**********************************************************************
2606                                    ;;**********************************************************************
2607                                    ;*TEST 24        TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE
2608                                    ;;**********************************************************************
2609  005366  000004                    TST24:  SCOPE
2610  005370  012767  000010  173562            MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
2611  005376  012767  000024  173574            MOV     #24.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2612                                                                            ; CAN WE USE THE WRAPAROUND??
2613  005404                                                            IF      @CABLE NOTSETIN $USWR THEN
2614  005404  032767  020000  173606            BIT     @CABLE,$USWR
2615  005412  001004                            BNE     $103
2616                                                                            ; CAN'T TEST WITHOUT BERG OR M315.
2617  005414                                                            EXIT TST
2618  005414  012767  000001  173536            MOV     #1,$TIMES
2619  005422  000454                            BR      TST25           ;;;EXIT THIS TEST
2620  005424                                                            ENDIF
2621  005424                            $103:
2622
2623                                                                            ;NO INTERRUPTS
2624  005424  012746  000340                    MOV     @PR7,-(SP)      ;;PUT NEW PS ON STACK
2625  005430  012746  005436                    MOV     @64$,-(SP)      ;;PUT NEW PC ON  STACK
2626  005434  000002                            RTI                     ;;POP NEW PC AND PS
2627  005436                            64$:
2628
2629
2630                                                                    ;START FRESH
2631                                                                            ;CLEAR SECREC
2632  005436                                                            LET     @RCSR := @RCSR CLR.BY @SECXMIT
2633  005436  042777  000010  173614            BIC     @SECXMIT,@RCSR
2634  005444                                                            LET     R3 := @RCSR
2635  005444  017703  173610                    MOV     @RCSR,R3
2636
2637                                                    ;SET SECREC --> DATAINT SET
2638  005450                                                            BGNSUB
2639  005450  012767  005456  173432            MOV     @65$,$LPERR
2640
2641                                                                            ;SET SECREC
2642  005456                                                            LET     @RCSR := @RCSR SET.BY @SECXMIT
2643  005456  052777  000010  173574            BIS     @SECXMIT,@RCSR
2644                                                                            ;WAIT 1 MILLI-SEC FOR CABLE
2645  005464                                                            WAITMS  1
2646  005464  010546                            MOV     R5,-(SP)
2647  005466  012745  000001                    MOV     #1,-(R5)
2648  005472  004767  004254                    JSR     PC,WAIT
2649  005476  012605                            MOV     (SP)+,R5
2650  005500                                                            IF      @DATAINT NOTSETIN @RCSR THEN
2651  005500  032777  100000  173552            BIT     @DATAINT,@RCSR
2652  005506  001001                            BNE     $104
2653                                                                            ;SETTING SECREC DID NOT SET DATAINT
2654  005510                                                                    ERRHRD 124,, E2DATA
2655  005510  104124                            ERROR   124
2656  005512                                                            ENDIF
2657  005512                            $104:
2658  005512                                                            ENDSUB
2659
2660                                                    ;CLEAR SECREC --> DATAINT SET
```

```
2661  005512                                                        BGNSUB
2662  005512  012767  005520  173370          MOV     #64$,$LPERR
2663                                                                    ;CLEAR SECREC
2664  005520                                                        LET     @RCSR := @RCSR CLR.BY #SECXMIT
2665  005520  042777  000010  173532          BIC     #SECXMIT,@RCSR
2666                                                                    ;WAIT 1 MILLI-SEC FOR CABLE
2667  005526                                                        WAITMS  1
2668  005526  010546                          MOV     R5,-(SP)
2669  005530  012745  000001                  MOV     #1,-(R5)
2670  005534  004767  004212                  JSR     PC,WAIT
2671  005540  012605                          MOV     (SP)+,R5
2672  005542                                                        IF      #DATAINT NOTSETIN @RCSR THEN
2673  005542  032777  100000  173510          BIT     #DATAINT,@RCSR
2674  005550  001001                          BNE     $105
2675                                                                    ;CLEARING SECREC DID NOT SET DATAINT
2676  005552                                                            ERRHRD 125,, E2DATA
2677  005552  104125                          ERROR   125
2678  005554                                                        ENDIF
2679  005554                          $105:
2680  005554                                                    ENDSUB
2681  005554                                                ENDTST
2682
2683
2684
```

```
2685                                    ;;**********************************************************************
2686                                    ;;**********************************************************************
2687                                    ;*TEST 25        TEST THAT XMIT RDY - TCSR 7 - CLEARS
2688                                    ;*              WHEN TBUF IS LOADED WITH A CHARACTER
2689                                    ;*              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
2690                                    ;;**********************************************************************
2691  005554  000004                  TST25:  SCOPE
2692  005556  012767  000001  173374          MOV     #1,$TIMES       ;;DO 1 ITERATION
2693  005564  012767  000025  173406          MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2694
2695                                                  ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
2696  005572                                          IF @APTENV SETIN $ENV THEN
2697  005572  032767  000001  173414          BIT     @APTENV,$ENV
2698  005600  001404                          BEQ     $106
2699  005602                                                  EXIT TEST
2700  005602  012767  000001  173350          MOV     #1,$TIMES
2701  005610  000454                          BR      TST26           ;;;EXIT THIS TEST
2702  005612                                          ENDIF
2703  005612                                  $106:
2704  005612                                                  BGNSUB
2705  005612  012767  005620  173270          MOV     #64$,$LPERR
2706                                                  ; LOAD TBUF WITH ONE CHARACTER
2707                                                  ; WAIT FOR READY TO SET
2708                                                  ; (SHOULD BE VERY SHORT WAIT
2709                                                  ; SINCE UART DOUBLE BUFFERS ITS INPUT)
2710
2711                                                          ;SEND A CHARACTER
2712  005620                                                LET @TBUF :B= #0
2713  005620  105077  173444                  CLRB    @TBUF
2714                                                          ;WAIT A MAXIMUM
2715                                                          ;OF 50 MSEC FOR
2716                                                          ;XMIT RDY TO SET IN TCSR
2717  005624                                                CALL TIMER IN <#1000,#XMITRDY,TCSR,#SET>
2718  005624  010546                          MOV     R5,-(SP)
2719  005626  012745  177777                  MOV     #SET,-(R5)
2720  005632  016745  173426                  MOV     TCSR,-(R5)
2721  005636  012745  000200                  MOV     #XMITRDY,-(R5)
2722  005642  012745  001000                  MOV     #1000,-(R5)
2723  005646  004767  003622                  JSR     PC,TIMER
2724  005652  012605                          MOV     (SP)+,R5
2725                                                          ;TIMER RETURNS AN ERROR IF BIT DID
2726                                                          ;NOT MEET CONDITION WITHIN TIME LIMIT
2727  005654                                                IF.ERROR THEN
2728  005654  103001                          BCC     $107
2729                                                          ;XMIT RDY DID NOT SET IN TCSR
2730  005656                                                  ERRHRD 66,,DIDNOT
2731  005656  104066                          ERROR   66
2732  005660                                                ENDIF
2733  005660                                  $107:
2734  005660                                                ENDSUB
2735
2736  005660                                                BGNSUB
2737  005660  012767  0056/%>  173222          MOV     #64$,$LPERR
2738                                                  ; LOAD TBUF WITH A SECOND CHARACTER
2739                                                  ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
2740                                                  ; AND THEN WAIT FOR IT TO SET
```

```
2741
2742                                                                    ;SEND SECOND CHARACTER
2743  005666                                                          LET @TBUF :B= #0
2744  005666  105077  173376          CLRB    @TBUF
2745  005672  000240                  NOP                          ; GIVE IT TIME TO CLEAR
2746                                                                    ; XMITRDY SHOULD HAVE CLEARED UPON
2747                                                                    ; RECEIPT OF A CHARACTER
2748  005674                                                          IF #XMITRDY SETIN @TCSR THEN
2749  005674  032777  000200  173362  BIT     #XMITRDY,@TCSR
2750  005702  001401                  BEQ     $110
2751                                                                    ; XMITRDY DID NOT CLEAR IN TCSR
2752  005704                                                          ERRHRD 67,,DIDNOT
2753  005704  104067                  ERROR   67
2754  005706                                                          ENDIF
2755  005706                  $110:
2756
2757                                                                    ;WAIT A MAXIMUM
2758                                                                    ;OF 50 MSEC FOR
2759                                                                    ;XMIT RDY TO SET IN TCSR
2760  005706                                                          CALL TIMER IN <#1000,#YMITRDY,TCSR,#SET>
2761  005706  010546                  MOV     R5,-(SP)
2762  005710  012745  177777          MOV     #SET,-(R5)
2763  005714  016745  173344          MOV     TCSR,-(R5)
2764  005720  012745  000200          MOV     #XMITRDY,-(R5)
2765  005724  012745  001000          MOV     #1000,-(R5)
2766  005730  004767  003540          JSR     PC,TIMER
2767  005734  012605                  MOV     (SP)+,R5
2768  005736                                                          IF.ERROR THEN
2769  005736  103001                  BCC     $111
2770                                                                    ;XMIT RDY DID NOT SET IN TCSR
2771  005740                                                          ERRHRD 70,,DIDNOT
2772  005740  104070                  ERROR   70
2773  005742                                                          ENDIF
2774  005742                  $111:
2775  005742                                                      ENDSUB
2776  005742                                                  ENDTST
```

```
MAINDEC-ZZ CVDVA-D      MACY11 30A(1052)  12 JUL 84  09:52  PAGE 70
CVDVAD.P11    12-JUL-84 05:04           T25      TEST THAT XMIT RDY - TCSR 7 - CLEARS

 2777                                   ;;*****************************************************************
 2778                                   ;;*****************************************************************
 2779                                   ;*TEST 26        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
 2780                                   ;*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
 2781                                   ;*              AND THAT RESET CLEARS THE BIT.
 2782                                   ;;*****************************************************************
 2783  005742  000004                  TST26:  SCOPE
 2784  005744  012767  000010  173206          MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
 2785  005752  012767  000026  173220          MOV     #26.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2786
 2787                                                                          ; SET THE MAINTENANCE BIT
 2788  005760                                                                LET @TCSR :* @TCSR SET.BY #MAINT
 2789  005760  052777  000004  173276          BIS     #MAINT,@TCSR
 2790
 2791  005766                                                                BGNSUB
 2792  005766  012767  005774  173114          MOV     #64$,$LPERR
 2793                                              ; SEND A CHARACTER AND LET IT WRAP AROUND
 2794
 2795  005774                                                                    LET @TBUF :B= #0
 2796  005774  105077  173270                  CLRB    @TBUF
 2797
 2798                                                                          ; WAIT A MAXIMUM OF 50 MSEC
 2799                                                                          ; FOR RCVR DONE TO SET IN
 2800                                                                          ; RCSR
 2801  006000                                                                  CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
 2802  006000  010546                          MOV     R5,-(SP)
 2803  006002  012745  177777                  MOV     #SET,-(R5)
 2804  006006  016745  173246                  MOV     RCSR,-(R5)
 2805  006012  012745  000200                  MOV     #RCVRDONE,-(R5)
 2806  006016  012745  001000                  MOV     #1000,-(R5)
 2807  006022  004767  003446                  JSR     PC,TIMER
 2808  006026  012605                          MOV     (SP)+,R5
 2809
 2810                                                                      ;DIDN'T SET IN TIME
 2811  006030                                                              IF.ERROR THEN
 2812  006030  103001                          BCC     $112
 2813                                                                          ; RCVRDONE DID NOT SET IN RCSR
 2814  006032                                                                  ERRHRD 71,,DIDNOT
 2815  006032  104071                          ERROR   71
 2816  006034                                                              ENDIF
 2817  006034                          $112:
 2818
 2819  006034                                                            ENDSUB
 2820
 2821  006034                                                            BGNSUB
 2822  006034  012767  006042  173046          MOV     #64$,$LPERR
 2823                                              ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
 2824                                              ; THIS ALSO WILL CLEAR THE MAINT. BIT
 2825  006042                                                                BRESET
 2826  006042  000005                          RESET
 2827
 2828  006044                                                                IF #RCVRDONE SETIN @RCSR THEN
```

```
2829  006044  032777  000200  173206          BIT     @RCVRDONE,@RCSR
2830  006052  001401                          BEQ     $113
2831                                                                    ; RCVRDONE DID NOT RESET IN RCSR.
2832  006054                                                            ERRHRD 72,,DIDNOT
2833  006054  104072                          ERROR   72
2834  006056                                                          ENDIF
2835  006056                          $113:
2836  006056                                                      ENDSUB
2837  006056                                                  ENDTST
```

```
2838                                    ;;*****************************************************************
2839                                    ;;*****************************************************************
2840                                    ;*TEST 27        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
2841                                    ;;*****************************************************************
2842  006056  000004                   TST27:  SCOPE
2843  006060  012767  000010  173072           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2844  006066  012767  000027  173104           MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2845
2846                                                                    ; SET MAINT. BIT
2847  006074                                                          LET @TCSR := @TCSR SET.BY #MAINT
2848  006074  052777  000004  173162           BIS     #MAINT,@TCSR
2849  006102                                                          BGNSUB
2850  006102  012767  006110  173000           MOV     #64$,$LPERR
2851                                            ; OUTPUT A CHARACTER WITH MAINTENANCE
2852                                            ; SET, AND WAIT FOR XMITRDY TO SET.
2853
2854                                                                        ; OUTPUT A CHARACTER
2855  006110                                                              LET @TBUF :B= #0
2856  006110  105077  173154                   CLRB    @TBUF
2857                                                                        ; WAIT MAXIMUM OF 1000 MSEC
2858                                                                        ; FOR RCVRDONE TO SET IN
2859                                                                        ; RCSR
2860  006114                                                                CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
2861  006114  010546                           MOV     R5,-(SP)
2862  006116  012745  177777                   MOV     #SET,-(R5)
2863  006122  016745  173132                   MOV     RCSR,-(R5)
2864  006126  012745  000200                   MOV     #RCVRDONE,-(R5)
2865  006132  012745  001000                   MOV     #1000,-(R5)
2866  006136  004767  003332                   JSR     PC,TIMER
2867  006142  012605                           MOV     (SP)+,R5
2868                                                                        ; DID IT BECAME READY?
2869  006144                                                                IF.ERROR  THEN
2870  006144  103001                           BCC     $114
2871                                                                            ;RCVRDONE DID NOT SET IN RCSR
2872  006146                                                                    ERRHRD 73,, DIDNOT
2873  006146  104073                           ERROR   73
2874  006150                                                                ENDIF
2875  006150                           $114:
2876  006150                                                              ENDSUB
2877
2878                                            ; NOW THAT IT IS SET LETS SEE IF READING THE
2879                                            ; BUFFER CLEARS RCVRDONE.
2880
2881                                                                        ;READ BUFFER
2882  006150                                                                LET R0 :B= @RBUF
2883  006150  117700  173106                   MOVB    @RBUF,R0
2884
2885  006154                                                                IF #RCVRDONE SETIN @RCSR THEN
2886  006154  032777  000200  173076           BIT     #RCVRDONE,@RCSR
2887  006162  001401                           BEQ     $115
2888                                                                            ;RCVRDONE DID NOT CLEAR IN RCSR
2889  006164                                                                    ERRHRD 74,DIDNOT
2890  006164  104074                           ERROR   74
2891  006166                                                                ENDIF
2892  006166                           $115:
2893  006166                                                  ENDTST
```

```
2894                                    ;;*****************************************************************
2895                                    ;;*****************************************************************
2896                                    ;*TEST 30        TEST THAT RCVRACT - RCSR 11 - SETS
2897                                    ;*              WHEN A START BIT IS RECEIVED AND
2898                                    ;*              CLEARS WHEN RCVRDONE - RCSR 7 - SETS
2899                                    ;;*****************************************************************
2900  006166  000004                   TST30:  SCOPE
2901  006170  012767  000010  172762           MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
2902  006176  012767  000030  172774           MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2903
2904                                                   ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
2905  006204                                           IF @APTENV SETIN $ENV THEN
2906  006204  032767  000001  173002           BIT     @APTENV,$ENV
2907  006212  001404                           BEQ     $116
2908  006214                                                   EXIT TEST
2909  006214  012767  000001  172736           MOV     #1,$TIMES
2910  006222  000500                           BR      TST31           ;;;EXIT THIS TEST
2911  006224                                           ENDIF
2912  006224                                   $116:
2913  006224                                                   LET @TCSR := @TCSR SET.BY @MAINT
2914  006224  052777  000004  173032           BIS     @MAINT,@TCSR
2915  006232                                                       LET R0 := @CLR
2916  006232  012700  000000                   MOV     @CLR,R0
2917  006236                                                       LET R1 := @0
2918  006236  005001                           CLR     R1
2919                                                   ;LOAD A CHARACTER INTO TBUF
2920                                                   ;WAIT FOR RCVRACT TO SET
2921
2922                                                                   ;SEND A CHARACTER
2923  006240                                                       LET @TBUF :B= @0
2924  006240  105077  173024                   CLRB    @TBUF
2925  006244                                                   REPEAT
2926  006244                                   $117:
2927  006244                                                       IF @RCVRACT SETIN @RCSR THEN
2928  006244  032777  004000  173006           BIT     @RCVRACT,@RCSR
2929  006252  001403                           BEQ     $120
2930  006254
2931  006254  012700  177777                   MOV     @SET,R0                 LET R0 := @SET
2932  006260                                                       ELSE
2933  006260  000401                           BR      $121
2934  006262                                   $120:
2935  006262                                                       LET R1 := R1 + @1
2936  006262  005201                           INC     R1
2937  006264                                                       ENDIF
2938  006264                                   $121:
2939  006264                                                   UNTIL R0 EQ @SET OR R1 HI MAX
2940  006264  020027  177777                   CMP     R0,@SET
2941  006270  001403                           BEQ     $122
2942  006272  020167  000124                   CMP     R1,MAX
2943  006276  101762                           BLOS    $117
2944  006300                                   $122:
2945  006300                                                   IF R1 HI MAX THEN
2946  006300  020167  000116                   CMP     R1,MAX
2947  006304  101407                           BLOS    $123
2948                                                               ;IT NEVER SET
2949                                                               ;RCVRACT DID NOT SET IN RCSR.
```

```
2950  006306                                                          ERRHRD 75,, DIDNOT
2951  006306  104075                      ERROR   75
2952  006310                                                          LET R0 := @RBUF ; CLEAR BUFFER
2953  006310  017700  172746              MOV     @RBUF,R0
2954  006314                                                          EXIT TEST
2955  006314  012767  000001  172636      MOV     #1,$TIMES
2956  006322  000440                      BR      TST31           ;;;EXIT THIS TEST
2957  006324                                                      ENDIF
2958  006324                      $123:
2959
2960
2961                                                  ;CHECK FOR TIMING OF RCVRACT. CLEARING
2962                                                  ;VS RCVRDONE SETTING
2963
2964
2965  006324                                                      WHILE @RCVRACT SETIN @RCSR DO
2966  006324                      $124:
2967  006324  032777  004000  172726      BIT     @RCVRACT,@RCSR
2968  006332  001416                      BEQ     $125
2969
2970  006334                                                          IF @RCVRDONE SETIN @RCSR THEN
2971  006334  032777  000200  172716      BIT     @RCVRDONE,@RCSR
2972  006342  001411                      BEQ     $126
2973  006344                                                             IF @RCVRACT SETIN @RCSR THEN
2974  006344  032777  004000  172706      BIT     @RCVRACT,@RCSR
2975  006352  001405                      BEQ     $127
2976                                                                  ;RCVRDONE AND RCVRACT
2977                                                                      ;BOTH SET
2978  006354                                                              ERRHRD 76, DONEACT
2979  006354  104076                      ERROR   76
2980                                                                  ;NO USE CONTINUING
2981  006356                                                              EXIT TST
2982  006356  012767  000001  172574      MOV     #1,$TIMES
2983  006364  000417                      BR      TST31           ;;;EXIT THIS TEST
2984  006366                                                          ENDIF
2985  006366                      $127:
2986  006366                                                      ENDIF
2987  006366                      $126:
2988  006366                                                  ENDDO
2989  006366  000756                      BR      $124
2990  006370                      $125:
2991
2992                                                          ;RCVRACT = 0 NOW.
2993  006370                                                  IF @RCVRDONE NOTSETIN @RCSR THEN
2994  006370  032777  000200  172662      BIT     @RCVRDONE,@RCSR
2995  006376  001001                      BNE     $130
2996                                                              ;RCVRDONE DID NOT SET IN RCSR
2997  006400                                                      ERRHRD 77,,DIDNOT
2998  006400  104077                      ERROR   77
2999                                                              ;SET IT BACK.
3000  006402                                                  ENDIF
3001  006402                      $130:
3002                                          ;TEST THAT READING THE RECEIVER
3003                                          ;BUFFER CLEARS RCVRDONE
3004
3005
```

```
3006                                                                    ;READ CHAR.
3007  006402                                                          LET R0 := @RBUF
3008  006402  017700  172654          MOV     @RBUF,R0
3009
3010  006406                                                          IF #RCVRDONE SETIN @RCSR THEN
3011  006406  032777  000200  172644  BIT     #RCVRDONE,@RCSR
3012  006414  001401                  BEQ     $131
3013                                                                      ;RCVRDONE DID NOT CLEAR IN RCSR
3014  006416                                                              ERRHRD 100,,DIDNOT
3015  006416  104100                  ERROR   100
3016  006420                                                          ENDIF
3017  006420                  $131:
3018
3019  006420                                                                  EXIT
3020  006420  000401                  BR      TST31           ;;;EXIT THIS TEST
3021  006422  070000                  MAX:70000
3022
3023  006424                                                              ENDTST
3024
```

```
3025
3026                                    ;;**********************************************************************
3027                                    ;;**********************************************************************
3028                                    ;*TEST 31        TEST THE OVERRUN BIT - RBUF 14
3029                                    ;;**********************************************************************
3030  006424  000004                    TST31:  SCOPE
3031  006426  012767  000010  172524            MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
3032  006434  012767  000031  172536            MOV     #31.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
3033
3034  006442                                                            BGNSUB
3035  006442  012767  006450  172440            MOV     #64$,$LPERR
3036                                                ;OUTPUT 2 CHARACTERS WITH
3037                                                ;AMPLE DELAYS BETWEEN FOR RECEPTION.
3038                                                ;THIS SHOULD AN CAUSE OVERRUN ERROR.
3039
3040                                                                        ;OUTPUT 1 CHARACTER
3041  006450                                                            LET @TBUF :B= #0
3042  006450  105077  172614                    CLRB    @TBUF
3043                                                                        ;GO AWAY FOR 1024. M SEC
3044  006454                                                            WAITMS 1024.
3045  006454  010546                            MOV     R5,-(SP)
3046  006456  012745  002000                    MOV     #1024.,-(R5)
3047  006462  004767  003264                    JSR     PC,WAIT
3048  006466  012605                            MOV     (SP)+,R5
3049
3050                                                                        ;OUTPUT 2ND CHARACTER
3051  006470                                                            LET @TBUF :B= #0
3052  006470  105077  172574                    CLRB    @TBUF
3053                                                                        ;LET OVERRUN HAPPEN
3054  006474                                                            WAITMS 1024.
3055  006474  010546                            MOV     R5,-(SP)
3056  006476  012745  002000                    MOV     #1024.,-(R5)
3057  006502  004767  003244                    JSR     PC,WAIT
3058  006506  012605                            MOV     (SP)+,R5
3059
3060                                                                        ;READ BUFFER AND ERROR BITS
3061  006510                                                            LET R4 := @RBUF
3062  006510  017704  172546                    MOV     @RBUF,R4
3063
3064                                                                        ;IT DIDN'T SET
3065  006514                                                            IF #ORERR NOTSETIN R4 THEN
3066  006514  032704  040000                    BIT     #ORERR,R4
3067  006520  001005                            BNE     $132
3068                                                                        ;ORERR DID NOT SET IN RBUF
3069  006522                                                                ERRHRD 101.,DIDNOT
3070  006522  104101                            ERROR   101
3071
3072                                                                        ;NO USE COMPOUNDING ERRORS
3073  006524                                                                EXIT TST
3074  006524  012767  000001  172426            MOV     #1,$TIMES
3075  006532  000456                            BR      TST32           ;;;EXIT THIS TEST
3076  006534                                                            ENDIF
3077  006534                                $132:
3078  006534                                                          ENDSUB
3079
3080                                            ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
```

```
3081  006534                                                          BGNSUB
3082  006534  012767  006542  172346          MOV     #64$,$LPERR
3083  006542                                                              IF #ERROR NOTSETIN R4 THEN
3084  006542  032704  100000                  BIT     #ERROR,R4
3085  006546  001005                          BNE     $133
3086
3087                                                                        ;ERROR DID NOT SET IN RBUF
3088  006550                                                                ERRHRD 102.,DIDNOT
3089  006550  104102                          ERROR   102
3090
3091                                                                        ;-:HEN ORERR SET.
3092                                                                        ;GET OUT NOW.
3093  006552                                                                EXIT TST
3094  006552  012767  000001  172400          MOV     #1,$TIMES
3095  006560  000443                          BR      TST32           ;;;EXIT THIS TEST
3096  006562                                                              ENDIF
3097  006562                          $133:
3098  006562                                                          ENDSUB
3099
3100  006562                                                          BGNSUB
3101  006562  012767  006570  172320          MOV     #64$,$LPERR
3102                                              ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
3103
3104  006570                                                              IF #ORERR NOTSETIN @RBUF THEN
3105  006570  032777  040000  172464          BIT     #ORERR,@RBUF
3106  006576  001002                          BNE     $134
3107
3108                                                                        ;READING RBUF CLEARED ORERR.
3109  006600                                                                ERRHRD 103,ITCLRED
3110  006600  104103                          ERROR   103
3111                                                                        ;SKIP REST OF TEST
3112  006602                                                                EXIT
3113  006602  000432                          BR      TST32           ;;;EXIT THIS TEST
3114  006604                                                              ENDIF
3115  006604                          $134:
3116  006604                                                          ENDSUB
3117
3118  006604                                                          BGNSUB
3119  006604  012767  006612  172276          MOV     #64$,$LPERR
3120                                              ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
3121
3122                                                                        ;SEND A CHARACTER AROUND.
3123  006612                                                            LET @TBUF :B= #0
3124  006612  105077  172452                  CLRB    @TBUF
3125                                                                        ;LET IT CIRCULATE
3126  006616                                                            WAITMS 1024.
3127  006616  010546                          MOV     R5,-(SP)
3128  006620  012745  002000                  MOV     #1024.,-(R5)
3129  006624  004767  003122                  JSR     PC,WAIT
3130  006630  012605                          MOV     (SP)+,R5
3131
3132  006632                                                              IF #ORERR SETIN @RBUF THEN
3133  006632  032777  040000  172422          BIT     #ORERR,@RBUF
3134  006640  001405                          BEQ     $135
3135                                                                        ;ORERR DID NOT CLEAR IN P9UF
3136  006642                                                                ERRHRD 104.,DIDNOT
```

```
006642  104104                          ERROR   104


                                                                                ;-AFTER RECEIVING ANOTHER CHAR
                                                                                ;SKIP AROUND REST
006644                                                                          EXIT TST
006644  012767  000001  172306          MOV     #1,$TIMES
006652  000406                          BR      TST32           ;;;EXIT THIS TEST
006654                                                                  ENDIF
006654                          $135:

006654                                                                  IF #ERROR SETIN @RBUF THEN
006654  032777  100000  172400          BIT     #ERROR,@RBUF
006662  001401                          BEQ     $136
                                                                                ;ERROR DID NOT CLEAR IN RBUF
006664                                                                          ERRHRD 105,,DIDNOT
006664  104105                          ERROR   105

006666                                                                  ENDIF
006666                          $136:
006666                                                          ENDSUB
006666                                                          EXIT
006666  000400                          BR      TST32           ;;;EXIT THIS TEST
                                                                .EVEN
006670                                                      ENDTST

```

```
3162
3163                                    ;;**********************************************************************
3164                                    ;;**********************************************************************
3165                                    ;*TEST 32        PROGRAMMABLE BAUD RATE TEST
3166                                    ;*              TEST AT ALL SPEEDS AVAILABLE
3167                                    ;*              A COMPARISON WILL BE MADE TO SEE
3168                                    ;*              IF NEW TIME IS LESS THAN PREVIOUS.
3169                                    ;;**********************************************************************
3170  006670  000004                   TST32:  SCOPE
3171  006672  012767  000010  172260           MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
3172  006700  012767  000032  172272           MOV     #32.,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
3173  006706                                                           IF @#PBR NOTSETIN $USWR THEN
3174  006706  032767  000200  172304           BIT     @#PBR,$USWR
3175  006714  001004                           BNE     $137
3176  006716                                                               EXIT TST
3177  006716  012767  000001  172234           MOV     #1,$TIMES
3178  006724  000552                           BR      TST33           ;;;EXIT THIS TEST
3179  006726                                                           ENDIF
3180  006726                           $137:
3181                                                   ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
3182  006726                                                           IF @#APTENV SETIN $ENV THEN
3183  006726  032767  000001  172260           BIT     @#APTENV,$ENV
3184  006734  001404                           BEQ     $140
3185  006736                                                           EXIT TEST
3186  006736  012767  000001  172214           MOV     #1,$TIMES
3187  006744  000542                           BR      TST33           ;;;EXIT THIS TEST
3188  006746                                                           ENDIF
3189  006746                           $140:
3190
3191  006746                                                           LET OLD := #-1
3192  006746  012767  177777  000272           MOV     #-1,OLD
3193  006754                                                           LET OLD+2 := #-1
3194  006754  012767  177777  000266           MOV     #-1,OLD+2
3195  006762                                                           LET @TCSR := @TCSR SET.BY @#MAINT
3196  006762  052777  000004  172274           BIS     @#MAINT,@TCSR
3197                                                               ;EACH BAUD RATE
3198  006770                                                       INCR R3 FROM #0 TO #15. BY #1
3199  006770  005003                           CLR     R3
3200  006772  000401                           BR      $141
3201  006774                           $142:
3202  006774  005203                           INC     R3
3203  006776                           $141:
3204  006776  020327  000017                   CMP     R3,#15.
3205  007002  003060                           BGT     $143
3206  007004                                                           LET R0 := @RBUF
3207  007004  017700  172252                   MOV     @RBUF,R0
3208                                                                 ;CHANGE BAUDE RATE
3209  007010                                                       LET @TCSRHI :B= RATES(R3)
3210  007010  116377  007170  172250           MOVB    RATES(R3),@TCSRHI
3211                                                                 ;FLAG
3212  007016                                                       LET BIT := #0
3213  007016  005002                           CLR     BIT
3214                                                                 ;OUTPUT THE CHARACTER
3215  007020                                                       LET @TBUF := #0
3216  007020  005077  172244                   CLR     @TBUF
3217                                                                 ;INITIALIZE COUNTER
```

```
3218  007024
3219  007024  005067  000212                CLP     NEW
3220  007030
3221  007030  005067  000210                CLR     NEW+2
3222  007034
3223  007034                          $144:
3224  007034  005702                        TST     BIT
3225  007036  001014                        BNE     $145
3226  007040
3227  007040  032777  000200  172212        BIT     @RCVRDONE,@RCSR
3228  007046  001403                        BEQ     $146
3229
3230  007050
3231  007050  012702  000001                MOV     #1,BIT
3232  007054
3233  007054  000404                        BR      $147
3234  007056                          $146:
3235
3236  007056
3237  007056  005267  000160                INC     NEW
3238  007062
3239  007062  005567  000156                ADC     NEW+2
3240  007066
3241  007066                          $147:
3242
3243  007066
3244  007066  000762                        BR      $144
3245  007070                          $145:
3246
3247  007070
3248  007070  026767  000150  000152        CMP     NEW+2,OLD+2
3249  007076  103001                        BHIS    $150
3250
3251  007100
3252  007100  000412                        BR      $151
3253  007102                          $150:
3254
3255  007102
3256  007102  026767  000136  000140        CMP     NEW+2,OLD+2
3257  007110  001005                        BNE     $152
3258  007112  026767  000124  000126        CMP     NEW,OLD
3259  007120  103001                        BHIS    $152
3260
3261  007122
3262  007122  000401                        BR      $153
3263  007124                          $152:
3264
3265
3266
3267
3268  007124
3269  007124  104126                        ERROR   126
3270  007126
3271  007126                          $153:
3272  007126
3273  007126                          $151:
```

SEQ 78

```
LET NEW := #0

LET NEW+2 := #0

WHILE BIT EQ #0 DO


    IF #RCVRDONE SETIN @RCSR THEN


    ;DONE - ITS READY
        LET BIT := #1

    ELSE


    ;OTHERWISE-INCREMENT TIME
        LET NEW := NEW + #1

        LET NEW+2 := NEW+2 + CARRY

    ENDIF

    ;SIGNALS DONE
ENDDO


IF NEW+2 LO OLD+2 THEN


    ; OK
ELSE


    ; NEW+2 >= OLD+2
    IF NEW+2 EQ OLD+2 AND NEW LO OLD THEN



        ;OK
    ELSE


        ;NEW+2 > OLD+2 OR
        ;(NEW+2 = OLD+2 AND
        ; NEW >= OLD)
        ;BAUD RATE DIDN'T CHANGE
        ERRHRD 126, BAUDRATE

    ENDIF

ENDIF
```

```
3274                                                                    ;UPDATE OLD TIME
3275  007126                                                          LET OLD :* NEW
3276  007126  016767  000110  000112          MOV     NEW,OLD
3277  007134                                                          LET OLD*2 :* NEW*2
3278  007134  016767  000104  000106          MOV     NEW*2,OLD*2
3279
3280  007142                                                        ENDINC  ;BAUD RATE
3281  007142  000714                          BR      $142
3282  007144                          $143:
3283  007144                                                  LET R3 :B* $USWR*1 AND #17      ; PUT BAUD BACK
3284  007144  116703  172051                 MOVB    $USWR*1,R3
3285  007150  110346                          MOVB    R3,-(SP)
3286  007152  142716  000017                 BICB    #17,(SP)
3287  007156  142603                          BICB    (SP)*,R3
3288  007160                                                  LET BTCSRHI :B* RATES(R3)       ; LIKE HE WANTED IT
3289  007160  116377  007170  172100          MOVB    RATES(R3),BTCSRHI
3290
3291  007166                                                          EXIT    ;SKIP TABLE
3292  007166  000431                          BR      TST33           ;;;EXIT THIS TEST
3293
3294  007170                          RATES:  ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
3295                                  ;UPPER BYTE OF XCSR FOR EACH BAUD RATE
3296                                  ;** NOTE::      THE VALUE INDICATED IN THE COLUMN 'OFFSET
3297                                  ;**             INTO TABLE' CAN BE PLACED INTO BITS<11:8>
3298                                  ;**             OF LOCATION '$USWR' TO CAUSE THE CORROSPONDING
3299                                  ;**             BAUD TO BE SELECTED IN THE DLV11-E UPON
3300                                  ;**             COMPLETION OF THIS TEST.
3301
3302                                                          ;       BAUD    OFFSET INTO TABLE
3303  007170     010                  R0050:  .BYTE   010     ;       50      0
3304  007171     030                  R0070:  .BYTE   030     ;       70      1
3305  007172     050                  R0110:  .BYTE   050     ;       110     2
3306  007173     070                  R0135:  .BYTE   070     ;       135     3
3307  007174     110                  R0150:  .BYTE   110     ;       150     4
3308  007175     130                  R0300:  .BYTE   130     ;       300     5
3309  007176     150                  R0600:  .BYTE   150     ;       600     6
3310  007177     170                  R0200:  .BYTE   170     ;       1200    7
3311  007200     210                  R1800:  .BYTE   210     ;       1800    10
3312  007201     230                  R2000:  .BYTE   230     ;       2000    11
3313  007202     250                  R2400:  .BYTE   250     ;       2400    12
3314  007203     270                  R3600:  .BYTE   270     ;       3600    13
3315  007204     310                  R4800:  .BYTE   310     ;       4800    14
3316  007205     330                  R7200:  .BYTE   330     ;       7200    15
3317  007206     350                  R9600:  .BYTE   350     ;       9600    16
3318  007207     370                  R10000: .BYTE   370     ;       19200   17
3319
```

```
3320  007210  040502  042125  051040  BAUDRATE:       .ASCIZ  /BAUD RATE DIDN'T CHANGE./
3321  007216  052101  020105  044504
3322  007224  047104  052047  041440
3323  007232  040510  043516  027105
3324  007240     000
3325          007242                  .EVEN
3326  007242  000000  000000          NEW:    0,0
3327  007246  000000  000000          OLD:    0,0
3328  007252                                                          ENDTST
3329
3330
3331
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 33       TRANSMITTER INTERRUPT LOGIC TEST
                                ;*             LOGICALLLY THIS IS 4 SEPARATE TESTS
                                ;*             A) DOES TRANSMITTER INTERRUPT LOGIC WORK
                                ;*             B) AT PRIORITY OF 0
                                ;*             C) AND ONLY ONCE
                                ;*             D) BUT NOT WITH INTERRUPT ENABLE CLEAR
                                ;;*****************************************************************
007252  000004                  TST33:  SCOPE
007254  012767  000010  171676          MOV     #10.,$TIMES     ;;DO 10 ITERATIONS
007262  012767  000033  171710          MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR 'INTERRUPT OCCURED  FLAG
007270                                                                  LET INTFLAG := #0
007270  005067  002544                  CLR     INTFLAG

                                                                        ;GET VECTOR ADDRESS
007274                                                                  LET R3 := DLVEC
007274  016703  171756                  MOV     DLVEC,R3
                                                                        ;FOR THE TRANSMITTER
007300                                                                  LET R3 := R3 + #4
007300  062703  000004                  ADD     #4,R3
                                                                        ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
007304                                                                  SETVEC R3, #INTSRV, #PR7
007304  010146                  MOV     R1,-(SP)
007306  010301                  MOV     R3,R1
007310  012721  012032          MOV     #INTSRV,(R1)+
007314  012711  000340          MOV     #PR7,(R1)
007320  012601                  MOV     (SP)+,R1
007322                                                                  BGNSUB
007322  012767  007330  171560          MOV     #64$,$LPERR
                                                                        ;CLEAR INTERRUPT ENABLE
007330                                                                  LET @TCSR := @TCSR CLR.BY #XMITIE
007330  042777  000100  171726          BIC     #XMITIE,@TCSR

                                                                        ;SET IT TO 0
007336  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007342  012746  007350          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
007346  000002                  RTI                     ;;POP NEW PC AND PS
007350                          65$:

                                                                        ;NOW SET I.E. BIT
007350                                                                  LET @TCSR := @TCSR SET.BY #XMITIE
007350  052777  000100  171706          BIS     #XMITIE,@TCSR

                                                                        ;LET INTERRUPT HAVE TIME TO OCCUR
007356                                                                  WAITMS 1024.
007356  010546                  MOV     R5,-(SP)
007360  012745  002000          MOV     #1024.,-(R5)
007364  004767  002362          JSR     PC,WAIT
007370  012605                  MOV     (SP)+,R5

                                                                        ;DID EXACTLY 1 INTERRUPT OCCUR
007372                                                                  IF INTFLAG NE #1 THEN
007372  026727  002442  000001          CMP     INTFLAG,#1
007400  001406                  BEQ     $154
```

```
3388                                                                    ;NO - WAS IT 0 OR MORE THAN ONCE
3389  007402                                                            IF INTFLAG EQ #0 THEN
3390  007402  005767  002432          TST     INTFLAG
3391  007406  001002                  BNE     $155
3392                                                                    ;TRANSMITTER DID NOT INTERRUPT IN TIME
3393  007410                                                                ERRHRD 106,,DIDNOT
3394  007410  104106                  ERROR   106
3395  007412                                                            ELSE
3396  007412  000401                  BR      $156
3397  007414                  $155:
3398                                                                    ;TWICE
3399                                                                    ;TRANSMITTER INTERRUPTED TWICE
3400  007414                                                                ERRHRD 107,,TWICE
3401  007414  104107                  ERROR   107
3402  007416                                                            ENDIF
3403  007416                  $156:
3404  007416                                                        ENDIF
3405  007416                  $154:
3406  007416                                                    ENDSUB
3407                                                    ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
3408  007416                                                    BGNSUB
3409  007416  012767  007424  171464  MOV     #64$,@LPERR
3410                                                                    ;CLEAR 'INTERRUPT OCCURED' FLAG
3411  007424                                                        LET INTFLAG := #0
3412  007424  005067  002410          CLR     INTFLAG
3413                                                                    ;CLEAR INTERRUPT ENABLE
3414  007430                                                        LET @TCSR := @TCSR CLR.BY #XMITIE
3415  007430  042777  000100  171626  BIC     #XMITIE,@TCSR
3416                                                                    ;NO INTERRUPTS SHOULD OCCUR.
3417  007436  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3418  007442  012746  007450          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
3419  007446  000002                  RTI                     ;;POP NEW PC AND PS
3420  007450                  65$:
3421                                                                    ;DARE IT TO HAPPEN
3422  007450                                                        WAITMS 2
3423  007450  010546                  MOV     R5,-(SP)
3424  007452  012745  000002          MOV     #2,-(R5)
3425  007456  004767  002270          JSR     PC,WAIT
3426  007462  012605                  MOV     (SP)+,R5
3427  007464                                                        IF INTFLAG NE #0 THEN
3428  007464  005767  002350          TST     INTFLAG
3429  007470  001401                  BEQ     $157
3430                                                                    ;INTERRUPT OCCURED WITH I E CLEARED
3431  007472                                                                ERRHRD 110,NOTENAB
3432  007472  104110                  ERROR   110
3433  007474                                                        ENDIF
3434  007474                  $157:
3435  007474                                                        BRESET
3436  007474  000005                  RESET
3437  007476                                                    ENDSUB
3438                                                                    ;RESTORE VECTOR AREA
3439  007476                                                        LET R4 := @DLVEC
3440  007476  017704  171554          MOV     @DLVEC,R4
3441  007502                                                        CLRVEC R4
3442  007502  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3443  007504  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
```

```
3444  007506  012701  000004          MOV     #R4,R1
3445  007512  010102                  MOV     R1,R2
3446  007514  062702  000002          ADD     #2,R2
3447  007520  010221                  MOV     R2,(R1)+
3448  007522  005011                  CLR     (R1)
3449  007524  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
3450  007526  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
3451  007530                                                      ENDTST
3452
3453
3454
3455
3456
3457
```

```
3458                                    ;;**********************************************************************
3459                                    ;;**********************************************************************
3460                                    ;*TEST 34        RECEIVER INTERRUPT LOGIC TEST
3461                                    ;*              THIS TEST COVERS ALL OF THE RECEIVER
3462                                    ;*              SIDE OF THE INTERRUPT LOGIC, BOTH DATASET
3463                                    ;*              AND CHARACTER MODES.
3464                                    ;;**********************************************************************
3465  007530  000004                    TST34:  SCOPE
3466  007532  012767  000010  171420            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
3467  007540  012767  000034  171432            MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3468                                                                            ;CLEAR INTERRUPT OCCURED FLAG
3469                                                                            ;SET UP RECEIVER INTER.VECTOR
3470  007546                                                            SETVEC DLVEC,#INTSRV,#PR7
3471  007546  010146                            MOV     R1,-(SP)
3472  007550  016701  171502                    MOV     DLVEC,R1
3473  007554  012721  012032                    MOV     #INTSRV,(R1)+
3474  007560  012711  000340                    MOV     #PR7,(R1)
3475  007564  012601                            MOV     (SP)+,R1
3476                                            ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
3477  007566                                                            BGNSUB
3478  007566  012767  007574  171314            MOV     #64$,$LPERR
3479  007574                                                            LET INTFLAG := #0
3480  007574  005067  002240                    CLR     INTFLAG
3481                                                                          ;SET MAINT. BIT
3482  007600                                                            LET @TCSR := @TCSR SET.BY #MAINT
3483  007600  052777  000004  171456            BIS     #MAINT,@TCSR
3484                                                                          ;CLEAR INTERRUPTS
3485  007606                                                          LET @RCSR := @RCSR CLR.BY #RCVRIE
3486  007606  042777  000100  171444            BIC     #RCVRIE,@RCSR
3487                                                                          ;CHANGE PRIORITY
3488                                                                          ;..TO 0
3489  007614  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3490  007620  012746  007626                    MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
3491  007624  000002                            RTI                     ;;POP NEW PC AND PS
3492  007626                            65$:
3493
3494                                                                          ;SEND A CHARACTER
3495  007626                                                          LET @TBUF :B= #0
3496  007626  105077  171436                    CLRB    @TBUF
3497                                                                          ;WAIT A MAXIMUM
3498                                                                  ;OF 1000 MSEC FOR
3499                                                                  ;RCVR RDY TO SET IN RCSR
3500  007632                                                          CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
3501  007632  010546                            MOV     R5,-(SP)
3502  007634  012745  177777                    MOV     #SET,-(R5)
3503  007640  016745  171414                    MOV     RCSR,-(R5)
3504  007644  012745  000200                    MOV     #RCVRDONE,-(R5)
3505  007650  012745  001000                    MOV     #1000,-(R5)
3506  007654  004767  001614                    JSR     PC,TIMER
3507  007660  012605                            MOV     (SP)+,R5
3508                                                                          ;SET INTERRUPT ENABLE
3509  007662                                                          LET @RCSR := @RCSR SET.BY #RCVRIE
3510  007662  052777  000100  171370            BIS     #RCVRIE,@RCSR
3511                                                                          ;LET IT COME IN.
3512  007670                                                          WAITMS 1
3513  007670  010546                            MOV     R5,-(SP)
```

```
3514  007672  012745  000001          MOV     #1,-(R5)
3515  007676  004767  002050          JSR     PC,WAIT
3516  007702  012605                  MOV     (SP)+,R5
3517  007704                                                  LET R0 := @RBUF ; CLEAR RCVRDONE
3518  007704  017700  171352          MOV     @RBUF,R0
3519
3520                                                            ;DID HE DO IT RIGHT?
3521  007710                                                  IF INTFLAG NE #1 THEN
3522  007710  026727  002124  000001  CMP     INTFLAG,#1
3523  007716  001406                  BEQ     $160
3524                                                              ;NONE OCCURED
3525  007720                                                      IF INTFLAG EQ #0 THEN
3526  007720  005767  002114          TST     INTFLAG
3527  007724  001002                  BNE     $161
3528                                                                ;RECEIVER DID NOT INTERRUPT IN TIME
3529  007726                                                            ERRHRD 111,,DIDNOT
3530  007726  104111                  ERROR   111
3531                                                              ;TWICE OR MORE
3532  007730                                                      ELSE
3533  007730  000401                  BR      $162
3534  007732                  $161:
3535                                                                ;RECEIVER INTERRUPTED TWICE
3536  007732                                                            ERRHRD 112,,TWICE
3537  007732  104112                  ERROR   112
3538  007734                                                      ENDIF
3539  007734                  $162:
3540  007734                                                  ENDIF
3541  007734                  $160:
3542                                                            ;RESET MAINT. BIT.
3543  007734                                                  LET @TCSR := @TCSR CLR.BY @MAINT
3544  007734  042777  000004  171322  BIC     @MAINT,@TCSR
3545                                                            ; CLEAR INTERRUPT ENABLE
3546  007742                                                  LET @RCSR := @RCSR CLR.BY @RCVRIE
3547  007742  042777  000100  171310  BIC     @RCVRIE,@RCSR
3548  007750                                                  ENDSUB
3549
3550
3551
3552
3553
3554
3555
3556                                                    ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3557  007750                                                  BGNSUB
3558  007750  012767  007756  171132  MOV     #64$,$LPERR
3559  007756                                                  IF @CABLE NOTSETIN $USWR THEN
3560  007756  032767  020000  171234  BIT     @CABLE,$USWR
3561  007764  001004                  BNE     $163
3562                                                            ;CAN'T TEST WITHOUT A CABLE
3563  007766                                                  EXIT TST
3564  007766  012767  000001  171164  MOV     #1,$TIMES
3565  007774  000466                  BR      TST35           ;;;EXIT THIS TEST
3566  007776                                                  ENDIF
3567  007776                  $163:
3568                                                            ; CLEAR 'INTFLAG'
3569  007776                                                  LET INTFLAG := #0
```

```
3570  007776  005067  002036          CLR     INTFLAG
3571                                                        ;CLEAR INTERRUPTS
3572  010002                                              LET @RCSR := @RCSR CLR.BY #DATAIE
3573  010002  042777  000040  171250  BIC     #DATAIE,@RCSR
3574                                                        ;CHANGE PRIORITY
3575                                                        ;..TO 0
3576  010010  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3577  010014  012746  010022          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3578  010020  000002                  RTI                     ;;POP NEW PC AND PS
3579  010022                  64$:
3580  010022                                              LET @RCSR := @RCSR CLR.BY #REQSEND
3581  010022  042777  000004  171230  BIC     #REQSEND,@RCSR
3582                                                        ;SET INTERRUPT ENABLE
3583  010030                                              LET @RCSR := @RCSR SET.BY #DATAIE
3584  010030  052777  000040  171222  BIS     #DATAIE,@RCSR
3585  010036                                              LET    @RCSR := @RCSR SET.BY #REQSEND
3586  010036  052777  000004  171214  BIS     #REQSEND,@RCSR
3587                                                        ;LET IT COME IN.
3588  010044                                              WAITMS 1
3589  010044  010546                  MOV     R5,-(SP)
3590  010046  012745  000001          MOV     #1,-(R5)
3591  010052  004767  001674          JSR     PC,WAIT
3592  010056  012605                  MOV     (SP)+,R5
3593
3594                                                        ; DID IT DO IT RIGHT?
3595  010060                                              IF INTFLAG NE #1 THEN
3596  010060  026727  001754  000001  CMP     INTFLAG,#1
3597  010066  001406                  BEQ     $164
3598
3599  010070                                                  ;NONE OCCURED
3600  010070  005767  001744          TST     INTFLAG         IF INTFLAG EQ #0 THEN
3601  010074  001002                  BNE     $165
3602                                                          ;DATAINT  DID NOT INTERRUPT IN TIME
3603  010076                                                      ERRHRD 113,,DIDNOT
3604  010076  104113                  ERROR   113
3605                                                          ;TWICE OR MORE
3606  010100                                                  ELSE
3607  010100  000401                  BR      $166
3608  010102                  $165:
3609                                                          ; DATAINT INTERRUPTED TWICE
3610  010102                                                      ERRHRD 114,,TWICE
3611  010102  104114                  ERROR   114
3612  010104                                                  ENDIF
3613  010104                  $166:
3614  010104                                              ENDIF
3615  010104                  $164:
3616  010104                                              LET @RCSR := @RCSR CLR.BY #DATAIE
3617  010104  042777  000040  171146  BIC     #DATAIE,@RCSR
3618  010112                                              LET @RCSR := @RCSR CLR.BY #REQSEND
3619  010112  042777  000004  171140  BIC     #REQSEND,@RCSR
3620  010120                                          ENDSUB
3621
3622  010120                                          LET R4 := @DLVEC
3623  010120  017704  171132          MOV     @DLVEC,R4
3624  010124                                          CLRVEC R4
3625  010124  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
 3626  010126  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 3627  010130  012701  000004              MOV     #R4,R1
 3628  010134  010102                      MOV     R1,R2
 3629  010136  062702  000002              ADD     #2,R2
 3630  010142  010221                      MOV     R2,(R1)+
 3631  010144  005011                      CLR     (R1)
 3632  010146  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
 3633  010150  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
 3634  010152                                              ENDTST
```

```
3635                                    ;;*****************************************************************
3636                                    ;;*****************************************************************
3637                                    ;*TEST 35        TEST ACTUAL DATA TRANSFERED
3638                                    ;*               NON-INTERRUPT MAINTENANCE BIT SET
3639                                    ;;*****************************************************************
3640  010152  000004                   TST35:  SCOPE
3641  010154  012767  000001  170776           MOV     #1,$TIMES       ;;DO 1 ITERATION
3642  010162  012767  000035  171010           MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3643                                                                           ;SET MAINT. BIT
3644  010170                                                                LET @TCSR := @TCSR SET.BY #MAINT
3645  010170  052777  000004  171066           BIS     #MAINT,@TCSR
3646
3647                                                                           ;CHANGE PRIORITY
3648                                                                           ;..TO 0
3649  010176  012746  000000                   MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3650  010202  012746  010210                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3651  010206  000002                           RTI                     ;;POP NEW PC AND PS
3652  010210                           64$:
3653                                                                           ;GET DATA MASK.
3654  010210                                                                CALL DATLNG OUT <R1>
3655  010210  162705  000002                   SUB     #1*2,R5
3656  010214  004767  001432                   JSR     PC,DATLNG
3657  010220  012501                           MOV     (R5)+,R1
3658                                                                           ; START CLEAN
3659  010222                                                                LET R0 := @RBUF
3660  010222  017700  171034                   MOV     @RBUF,R0
3661
3662                                                                           ;ALL BINARY CHAR.
3663  010226                                                                INCR R2 FROM #0 TO #377 BY #1
3664  010226  005002                           CLR     R2
3665  010230  000401                           BR      $167
3666  010232                           $170:
3667  010232  005202                           INC     R2
3668  010234                           $167:
3669  010234  020227  000377                   CMP     R2,#377
3670  010240  003047                           BGT     $171
3671
3672
3673                                                                      ;TRANSMIT CHAR IN R2
3674
3675  010242                                                              CALL TIMER IN <#1000,#XMITRDY,TCSR,#SET>
3676  010242  010546                           MOV     R5,-(SP)
3677  010244  012745  177777                   MOV     #SET,-(R5)
3678  010250  016745  171010                   MOV     TCSR,-(R5)
3679  010254  012745  000200                   MOV     #XMITRDY,-(R5)
3680  010260  012745  001000                   MOV     #1000,-(R5)
3681  010264  004767  001204                   JSR     PC,TIMER
3682  010270  012605                           MOV     (SP)+,R5
3683
3684                                                                           ;TRANSMIT IT
3685  010272                                                                LET @TBUF :B= R2
3686  010272  110277  170772                   MOVB    R2,@TBUF
3687
3688  010276                                                              CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
3689  010276  010546                           MOV     R5,-(SP)
3690  010300  012745  177777                   MOV     #SET,-(R5)
```

```
MAINDEC-ZZ-CVDVA D      MACY11 30A(1052)  12-JUL-84  09:52  PAGE 91
CVDVAD.P11     12 JUL-84 05:04           T35     TEST ACTUAL DATA TRANSFERED

 3691  010304  016745  170750                   MOV     RCSR,-(R5)
 3692  010310  012745  000200                   MOV     #RCVRDONE,-(R5)
 3693  010314  012745  001000                   MOV     #1000,-(R5)
 3694  010320  004767  001150                   JSR     PC,TIMER
 3695  010324  012605                           MOV     (SP)+,R5
 3696                                                                    ;AND SAVE IT
 3697  010326                                                          LET R3 := @RBUF
 3698  010326  017703  170730                   MOV     @RBUF,R3
 3699
 3700                                                                    ;COMPARE TO SEE IF WE RECEIVED IT ALL
 3701
 3702                                                                        ;CLEAN OFF NON-DATA BITS
 3703                                                                        ;ON BOTH TRANSMITTED AND
 3704                                                                    LET R4 := R2 CLR.BY R1
 3705  010332
 3706  010332  010204                           MOV     R2,R4
 3707  010334  040104                           BIC     R1,R4
 3708  010336                                                          LET R3 := R3 CLR.BY R1
 3709  010336  040103                           BIC     R1,R3
 3710
 3711                                                                    ;RECEIVED DATA
 3712  010340                                                          IF R4 NE R3 THEN
 3713  010340  020403                           CMP     R4,R3
 3714  010342  001405                           BEQ     $172
 3715
 3716  010344                                                              ;DATA COMPARE ERROR
 3717  010344  104116                           ERROR   116             ERRHRD 116,COMP,SBWAS
 3718  010346                                                              EXIT TST  ; ON ERROR
 3719  010346  012767  000001  170604           MOV     #1,$TIMES
 3720  010354  000404                           BR      TST36       ;;;EXIT THIS TEST
 3721  010356                                                          ENDIF
 3722  010356                           $172:
 3723  010356                                                      ENDINC  ; R2
 3724  010356  000725                           BR      $170
 3725  010360                           $171:
 3726
 3727                                                                    ;RESET MAINT. BIT.
 3728  010360                                                          LET @TCSR := @TCSR CLR.BY #MAINT
 3729  010360  042777  000004  170676           BIC     #MAINT,@TCSR
 3730  010366                                                      ENDTST
 3731
 3732
 3733
```

```
                                        ;;**********************************************************************
                                        ;;**********************************************************************
                                        ;*TEST 36        TEST DATA THROUGH CABLE
                                        ;;**********************************************************************
010366  000004                  TST36:  SCOPE
010370  012767  000001  170562          MOV     #1,$TIMES       ;;DO 1 ITERATION
010376  012767  000036  170574          MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
010404                                                              IF @CABLE NOTSETIN $USWR THEN
010404  032767  020000  170606          BIT     @CABLE,$USWR
010412  001004                          BNE     $173
                                                                        ;CAN'T TEST WITHOUT A CABLE
010414                                                              EXIT TST
010414  012767  000001  170536          MOV     #1,$TIMES
010422  000474                          BR      TST37           ;;;EXIT THIS TEST
010424                                                              ENDIF
010424                          $173:
                                                                        ;DON'T USE MAINT.
010424                                                              LET @TCSR := @TCSR CLR.BY @MAINT
010424  042777  000004  170632          BIC     @MAINT,@TCSR
                                                                        ;CHANGE PRIORITY
                                                                        ;..TO 0
010432  012746  000000                  MOV     @PR0,-(SP)      ;;PUT NEW PS ON STACK
010436  012746  010444                  MOV     @64$,-(SP)      ;;PUT NEW PC ON  STACK
010442  000002                          RTI                     ;;POP NEW PC AND PS
010444                          64$:
                                                                        ;GET DATA MASK
010444                                                              CALL DATLNG OUT <R1>
010444  162705  000002                  SUB     #1*2,R5
010450  004767  001176                  JSR     PC,DATLNG
010454  012501                          MOV     (R5)+,R1
010456                                                              LET R0 := @RBUF ; START CLEAN
010456  017700  170600                  MOV     @RBUF,R0
                                                                        ;BINARY COUNT PATTERN
010462                                                              INCR R2 FROM #0 TO #377 BY #1
010462  005002                          CLR     R2
010464  000401                          BR      $174
010466                          $175:
010466  005202                          INC     R2
010470                          $174:
010470  020227  000377                  CMP     R2,#377
010474  003047                          BGT     $176


                                                                    ;TRANSMIT THE CHAR. IN R2.

010476                                                              CALL TIMER IN <#1000,@XMITRDY,TCSR,@SET>
010476  010546                          MOV     R5,-(SP)
010500  012745  177777                  MOV     @SET,-(R5)
010504  016745  170554                  MOV     TCSR,-(R5)
010510  012745  000200                  MOV     @XMITRDY,-(R5)
010514  012745  001000                  MOV     #1000,-(R5)
010520  004767  000750                  JSR     PC,TIMER
010524  012605                          MOV     (SP)+,R5


                                                                    ;START IT ON ITS WAY
010526                                                              LET @TBUF :B= R2
```

```
3790  010526  110277  170536            MOVB    R2,@TBUF                    CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
3791  010532
3792  010532  010546                    MOV     R5,-(SP)
3793  010534  012745  177777            MOV     #SET,-(R5)
3794  010540  016745  170514            MOV     RCSR,-(R5)
3795  010544  012745  000200            MOV     #RCVRDONE,-(R5)
3796  010550  012745  001000            MOV     #1000,-(R5)
3797  010554  004767  000714            JSR     PC,TIMER
3798  010560  012605                    MOV     (SP)+,R5
3799
3800                                                                            ;RETRIEVE
3801  010562                                                                    LET R3 := @RBUF
3802  010562  017703  170474            MOV     @RBUF,R3
3803
3804                                                                            ;STRIP OFF JUNK ON BOTH
3805  010566                                                                    LET R4 := R2 CLR.BY R1
3806  010566  010204                    MOV     R2,R4
3807  010570  040104                    BIC     R1,R4
3808  010572                                                                    LET R3 := R3 CLR.BY R1
3809  010572  040103                    BIC     R1,R3
3810
3811                                                                            ;WE HAVE TROUBLE
3812  010574                                                                    IF R4 NE R3 THEN
3813  010574  020403                    CMP     R4,R3
3814  010576  001405                    BEQ     $177
3815
3816  010600                                                                        ;DATA COMPARE ERROR
3817  010600  104117                    ERROR   117                               ERRHRD 117,COMP,SBWAS
3818  010602
3819  010602  012767  000001  170350    MOV     #1,$TIMES                         EXIT TST  ; ON ERROR
3820  010610  000401                    BR      TST37           ;;;EXIT THIS TEST
3821  010612                                                                    ENDIF
3822  010612                          $177:
3823
3824  010612                                                                ENDINC  ; R2
3825  010612  000725                    BR      $175
3826  010614                          $176:
3827
3828
3829
3830  010614                                                            ENDTST
3831
3832
3833
3834
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 37        FULL DATA TRANSFER WITH INTERRUPTS
                                ;*              AND MAINTENANCE MODE.
                                ;;**********************************************************************
010614  000004                  TST37:  SCOPE
010616  012767  000001  170334          MOV     #1,$TIMES       ;;DO 1 ITERATION
010624  012767  000037  170346          MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                
                                                ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
010632                                          IF @APTENV SETIN $ENV THEN
010632  032767  000001  170354          BIT     @APTENV,$ENV
010640  001404                        BEQ     $200
010642                                                  EXIT TEST
010642  012767  000001  170310          MOV     #1,$TIMES
010650  000550                        BR      TST40           ;;;EXIT THIS TEST
010652                                          ENDIF
010652                          $200:
                                                        ;GET DATA MASK
010652                                          CALL DATLNG OUT <R3>
010652  162705  000002                  SUB     #1*2,R5
010656  004767  000770                  JSR     PC,DATLNG
010662  012503                        MOV     (R5)+,R3
                                
                                ; THIS TEST WILL RUN BOTH TRANSMITTER AND
                                ; RECIEVER AT FULL SPEED TESTING
                                ; THE ABILITY OF THE MODULE
                                ; TO HANDLE INTERRUPTS FROM BOTH SIDES
                                ; AT ONCE. ALSO, THE DOUBLE BUFFERING LOGIC
                                ; OF THE UART WILL BE FULLY TESTED.
                                ;  THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
                                ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
                                ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS.
                                
                                                        ;CHANGE PRIORITY
                                                        ;..TO 0
010664  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
010670  012746  010676                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
010674  000002                        RTI                     ;;POP NEW PC AND PS
010676                          64$:
                                                        ;GET VECTOR ADDRESS
010676                                          LET R1 := DLVEC
010676  016701  170354                  MOV     DLVEC,R1
                                                        ;RCVR VECTOR
010702                                          LET (R1)+ := @REC
010702  012721  011076                  MOV     @REC,(R1)+
010706                                          LET (R1)+ := @PR7
010706  012721  000340                  MOV     @PR7,(R1)+
                                                        ;POINT TO TRANSMITTER VECTOR
                                                        ;AND SET IT UP ALSO
010712                                          LET (R1)+ := @TRAN
010712  012721  011034                  MOV     @TRAN,(R1)+
010716                                          LET (R1)  := @PR7
```

```
3891  010716  012711  000340                  MOV     #PR7,(R1)
3892
3893                                                            ; INITIALIZE COUNTERS
3894  010722                                                LET R1 := #-1
3895  010722  012701  177777                  MOV     # 1,R1
3896                                                            ;RECEIVER STORAGE
3897  010726                                                LET R2 := #0
3898  010726  005002                          CLR     R2
3899                                                            ;# OF RECEIVED CHAR. COUNT.
3900  010730                                                LET R4 := #-1
3901  010730  012704  177777                  MOV     #-1,R4
3902
3903                                                            ; CLEAR ERROR COUNT.
3904  010734                                                LET ERRCNT := #0
3905  010734  005067  000066                  CLR     ERRCNT
3906
3907  010740                                                BRESET  ;SET UP ALL REGISTERS
3908  010740  000005                          RESET
3909                                                            ;SET UP MAINTENANCE
3910  010742                                                LET @TCSR := @TCSR SET.BY #MAINT
3911  010742  052777  000004  170314          BIS     #MAINT,@TCSR
3912
3913                                                            ;SET I.E. IN TRANSMITTER
3914  010750                                                LET @TCSR := @TCSR SET.BY #XMITIE
3915  010750  052777  000100  170306          BIS     #XMITIE,@TCSR
3916                                                            ;AND RECEIVER
3917  010756                                                LET @RCSR := @RCSR SET.BY #RCVRIE
3918  010756  052777  000100  170274          BIS     #RCVRIE,@RCSR
3919
3920
3921                                                        ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3922  010764                                            REPEAT
3923  010764                          $201:
3924  010764                                            UNTIL R4 EQ NUMBER OR ERRCNT GT #0
3925  010764  020467  000040                  CMP     R4,NUMBER
3926  010770  001403                          BEQ     $202
3927  010772  005767  000030                  TST     ERRCNT
3928  010776  003772                          BLE     $201
3929  011000                          $202:
3930
3931                                                            ;DATA COMPARE ERRORS.
3932  011000                                            IF ERRCNT NE #0 THEN
3933  011000  005767  000022                  TST     ERRCNT
3934  011004  001401                          BEQ     $203
3935                                                            ;DATA COMPARE ERROR
3936  011006                                              ERRHRD 120,COMP,FIRST
3937  011006  104120                          ERROR   120
3938  011010                                            ENDIF
3939  011010                          $203:
3940
3941  011010                                            LET @TCSR := @TCSR CLR.BY #XMITIE
3942  011010  042777  000100  170246          BIC     #XMITIE,@TCSR
3943  011016                                            LET @RCSR := @RCSR CLR.BY #XMITIE
3944  011016  042777  000100  170234          BIC     #XMITIE,@RCSR
3945
3946  011024                                            EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
```

```
                T37     FULL DATA TRANSFER WITH INTERRUPTS

 3947 011024 000462             BR      TST40           ;;;EXIT THIS TEST
 3948
 3949 011026 000000                                     ERRCNT: 0
 3950 011030 000400                                     NUMBER: 400
 3951 011032    000                                     SB:     .BYTE 0
 3952 011033    000                                     WAS:    .BYTE 0
 3953
 3954
 3955
 3956                    ;*************************************************************
 3957                            ;TRANSMIT INTERRUPT HANDLER
 3958
 3959 011034                                            BGNSRV  TRAN
 3960 011034                TRAN:
 3961                    ;*************************************************************
 3962
 3963                                                        ;INCREMENT CHAR COUNT
 3964 011034                                            LET R1 := R1 + #1
 3965 011034 005201             INC     R1
 3966                                                        ;SET UP FOR TRANSFER
 3967 011036                                            LET HOLD := R1      CLR.BY R3
 3968 011036 010167 000030      MOV     R1,HOLD
 3969 011042 040367 000024      BIC     R3,HOLD
 3970                                                        ;AND SEND.
 3971 011046                                            LET @TBUF := HOLD
 3972 011046 016777 000020 170214       MOV     HOLD,@TBUF
 3973                                                        ;ALL DONE
 3974 011054                                            IF R1 EQ NUMBER THEN
 3975 011054 020167 177750      CMP     R1,NUMBER
 3976 011060 001003             BNE     $204
 3977                                                        ;STOP   INTERRUPT PROCESSING
 3978 011062                                                LET @TCSR := @TCSR CLR.BY #XMITIE
 3979 011062 042777 000100 170174       BIC     #XMITIE,@TCSR
 3980 011070                                            ENDIF
 3981 011070                $204:
 3982
 3983 011070 000401             BR      ZZZ                     ; EXIT SRV
 3984
 3985 011072 000000                                     HOLD:0
 3986
 3987 011074                ZZZ:                        ENDSRV
 3988 011074 000002             RTI
 3989
 3990
 3991                    ;*************************************************************
 3992
 3993                            ;RECEIVER INTERRUPT HANDLER
 3994 011076                                            BGNSRV  REC
 3995 011076                REC:
 3996                    ;*************************************************************
 3997
 3998                                                        ;COUNT THIS CHAR.
 3999 011076                                            LET R4 := R4 + #1
 4000 011076 005204             INC     R4
 4001                                                        ;GET CHAR IN + MASK IT
 4002 011100                                            LET R2 := @RBUF CLR.BY R3
```

```
4003  011100  017702  170156          MOV     @RBUF,R2
4004  011104  040302                  BIC     R3,R2
4005                                                        ;RHLD WILL CONTAIN EXPECTED INPUT
4006  011106                                              LET RHLD := R4 CLR BY R3
4007  011106  010467  000054          MOV     R4,RHLD
4008  011112  040367  000050          BIC     R3,RHLD
4009
4010                                                        ;DO THEY COMPARE
4011  011116                                              IF R2 NE RHLD THEN
4012  011116  020267  000044          CMP     R2,RHLD
4013  011122  001412                  BEQ     $205
4014
4015  011124                                                  ;FIRST ERROR
4016  011124  005767  177676          TST     ERRCNT         IF ERRCNT EQ #0 THEN
4017  011130  001005                  BNE     $206
4018                                                        ;SAVE RECORD OF FIRST MISS
4019  011132                                                      LET SB :B= RHLD
4020  011132  116767  000030  177672  MOVB    RHLD,SB
4021  011140                                                      LET WAS :B= R2
4022  011140  110267  177667          MOVB    R2,WAS
4023  011144                                                  ENDIF
4024  011144                          $206:
4025                                                        ;COUNT IT.
4026  011144                                                  LET ERRCNT := ERRCNT + #1
4027  011144  005267  177656          INC     ERRCNT
4028  011150                                              ENDIF
4029  011150                          $205:
4030
4031                                                        ;ALL DONE?
4032  011150                                              IF R4 EQ NUMBER THEN
4033  011150  020467  177654          CMP     R4,NUMBER
4034  011154  001003                  BNE     $207
4035                                                        ;STOP RECEIVER INTERRUPTS
4036  011156                                                  LET @RCSR := @RCSR CLR.BY #RCVRIE
4037  011156  042777  000100  170074  BIC     #RCVRIE,@RCSR
4038                                                        ;MAIN REPEAT LOOP IS CHECKING
4039  011164                                              ENDIF
4040  011164                          $207:
4041
4042                                                        ;FOR 'R4 = NUMBER' ALSO
4043  011164  000401                  BR      ZZZZ       ;  EXIT SRV
4044
4045  011166  000000                                          RHLD:0
4046  011170                          ZZZZ:
4047  011170                                              ENDSRV
4048  011170  000002                  RTI
4049
4050  011172                                          ENDTST
4051
4052
4053
```

```
                                        ;;*****************************************************************
                                        ;;*****************************************************************
                                        ;*TEST 40        TEST BREAK GENERATION LOGIC
                                        ;*              TRANSMIT KNOWN CHAR WITH BREAK SET
                                        ;*              AND COMPARE RECEIVED WITH 0.
                                        ;;*****************************************************************
011172  000004                  TST40:  SCOPE
011174  012767  000010  167756          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
011202  012767  000040  167770          MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
011210                                                                  IF @BRK NOTSETIN $USWR THEN
011210  032767  010000  170002          BIT     @BRK,$USWR
011216  001004                          BNE     $210
011220                                                                      EXIT TST
011220  012767  000001  167732          MOV     #1,$TIMES
011226  000452                          BR      TST41           ;;;EXIT THIS TEST
011230                                                                  ENDIF
011230                          $210:
011230                                                                  ;SET MAINTENANCE BIT
011230                                                                  LET @TCSR := @TCSR SET.BY @MAINT
011230  052777  000004  170026          BIS     @MAINT,@TCSR
                                                                        ; CLEAR RCVRDONE JUST IN CASE
011236                                                                  LET R0 := @RBUF
011236  017700  170020                  MOV     @RBUF,R0
                                                                        ;SET BREAK BIT
011242                                                                  LET @TCSR := @TCSR SET.BY @BREAK
011242  052777  000001  170014          BIS     @BREAK,@TCSR
                                                                        ;NON-ZERO CHAR. '*'
011250                                                                  LET @TBUF := #252
011250  012777  000252  170012          MOV     #252,@TBUF
011256                                                                  CALL TIMER IN <#1000,#RCVRDONE,RCSR,#SET>
011256  010546                          MOV     R5,-(SP)
011260  012745  177777                  MOV     #SET,-(R5)
011264  016745  167770                  MOV     RCSR,-(R5)
011270  012745  000200                  MOV     #RCVRDONE,-(R5)
011274  012745  001000                  MOV     #1000,-(R5)
011300  004767  000170                  JSR     PC,TIMER
011304  012605                          MOV     (SP)+,R5
011306                                                                  IF.ERROR THEN
011306  103001                          BCC     $211
                                                                        ; RECIEVER DONE DID NOT SET
011310                                                                      ERRHRD 115
011310  104115                          ERROR   115
011312                                                                  ENDIF
011312                          $211:
011312                                                                  IFB @RBUF NE #0 THEN
011312  105777  167744                  TSTB    @RBUF
011316  001401                          BEQ     $212
                                                                        ; BREAK DID NOT EQUAL 0
011320                                                                      ERRHRD 121 ,BADBRK
011320  104121                          ERROR   121
011322                                                                  ENDIF
011322                          $212:
```

```
4110  011322                                                    BRESET  ;CLEAN UP
4111  011322  000005                          RESET
4112  011324                                                    EXIT
4113  011324  000413                          BR      TST41    ;;;EXIT THIS TEST
4114  011326  051102  040505  020113                           BADBRK: .ASCIZ  /BREAK DID NOT EQUAL 0/
4115  011334  044504  020104  047516
4116  011342  020124  050505  040525
4117  011350  020114  000060
4118
4119  011354                                                ENDTST
```

```
4120
4121                                    ;;**********************************************************************
4122                                    ;*TEST 41        NOT A TEST - SEND BACK TO LOOP
4123                                    ;;**********************************************************************
4124  011354  000004                   TST41:  SCOPE
4125  011356  012767  000001  167574           MOV     #1,$TIMES       ;;DO 1 ITERATION
4126  011364  104401  011372                   TYPE    ,65$            ;;TYPE ASCIZ STRING
4127  011370  000404                           BR      64$             ;;GET OVER THE ASCIZ
4128                                    ;;65$:  .ASCIZ  <CRLF>*CSR: *
4129  011402                           64$:
4130  011402  016746  167646                   MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
4131  011406  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4132  011410  104401  011416                   TYPE    ,67$            ;;TYPE ASCIZ STRING
4133  011414  000405                           BR      66$             ;;GET OVER THE ASCIZ
4134                                    ;;67$:  .ASCIZ  *,VECTOR: *
4135  011430                           66$:
4136  011430  016746  167622                   MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
4137  011434  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4138  011436  104401  011444                   TYPE    ,69$            ;;TYPE ASCIZ STRING
4139  011442  000405                           BR      68$             ;;GET OVER THE ASCIZ
4140                                    ;;69$:  .ASCIZ  *,ERRORS: *
4141  011456                           68$:
4142  011456  016746  167430                   MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
4143  011462  104405                           TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
4144  011464  005067  167422                   CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
4145  011470  000167  170250                   JMP     LOOP            ; BACK UP TO THE BEGINNING
```

```
                                        ;;BGNMOD        SUBS
                                ;;**************************************************************
                                                        ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
011474                          TIMER:
011474                          ;* ROUTINE:TIMER
                                ;*      THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
                                ;*      IN ANY REGISTER.
                                ;*      INPUTS:
                                ;*            HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
                                ;*                       THIS ROUTINE.
                                ;*            WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
                                ;*                         TO BE CHECKED.
                                ;*            REG      A POINTER TO THE REGISTER TO BE CHECKED
                                ;*            SETCLR   THE DESIRED RESULTS
                                ;*                     EITHER #SET OR #CLEAR
                                ;*      OUTPUT:
                                ;*            THE 'C' BIT IS SET TO INDICATE AN ERROR
                                ;*            BUT IT IS TESTED BY THE IF.ERROR STATEMENT
                                ;*
                                ;*
                                ;*            NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
                                ;*                    MECHANISM BETWEEN THE CALLER AND THE CALLED
                                ;;**************************************************************

000001                                                          TRUE=   1
000000                                                          FALSE=  0

011474                                                          LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
011474  016567  000004  000136          MOV     REG(R5),REGSAV
011502                                                          LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
011502  016567  000000  000132          MOV     HOWLONG(R5),TIMSAV
011510                                                          LET     FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA
011510  112767  000000  000126          MOVB    #FALSE,FLAG


                                                                ;       START OF AN INFINITE LOOP

011516                                                          LOOP
011516                          $215:
                                                                    ; TEST TO SEE IF WHICHBIT IS SET
011516                                                          IF      WHICHBIT(R5)  NOTSETIN &REGSAV THEN
011516  036577  000002  000114          BIT     WHICHBIT(R5),@REGSAV
011524  001004                          BNE     $217
011526                                                              LET     HOLDSC :B= #CLR
011526  112767  000000  000111          MOVB    #CLR,HOLDSC
011534                                                          ELSE
011534  000403                          BR      $220
011536                          $217:
011536                                                              LET     HOLDSC :B= #SET ; REMEMBER THIS
011536  112767  177777  000101          MOVB    #SET,HOLDSC
011544                                                          ENDIF
011544                          $220:

                                                                ; NOW SEE IF THAT WAS WHAT WE WANTED
011544                                                          IFB     HOLDSC EQ SETCLR(R5) THEN
```

```
4202  011544  126765  000075  000006          CMPB    HOLDSC,SETCLR(R5)
4203  011552  001003                          BNE     $221
4204                                                    ; JUST THE THING WE NEEDED
4205  011554                                                LET     FLAG :B= #TRUE
4206  011554  112767  000001  000062          MOVB    #TRUE,FLAG
4207  011562                                            ENDIF
4208  011562                          $221:
4209
4210  011562                                          EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
4211  011562  126727  000056  000001          CMPB    FLAG,#TRUE
4212  011570  001414                          BEQ     $216
4213  011572  005767  000044                  TST     TIMSAV
4214  011576  003411                          BLE     $216
4215                                                    ; ONE WAY OR THE OTHER, WE ARE DONE
4216                                                ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
4217
4218  011600                                            WAITMS  1       ;WAIT FOR 10 MILLI-SECONDS
4219  011600  010546                          MOV     R5,-(SP)
4220  011602  012745  000001                  MOV     #1,-(R5)
4221  011606  004767  000140                  JSR     PC,WAIT
4222  011612  012605                          MOV     (SP)+,R5
4223  011614
4224  011614  005367  000022                  DEC     TIMSAV                  LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4225  011620                                          ENDLOOP                 ; CONTINUED AT THE TOP
4226  011620  000736                          BR      $215
4227  011622                          $216:
4228
4229                                    ; ONLY 2 WAYS TO GET HERE
4230                                    ;       1). WE RAN OUT OF TIME---ERROR ! !
4231                                    ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
4232
4233  011622                                          IFB     FLAG EQ #TRUE THEN
4234  011622  126727  000016  000001          CMPB    FLAG,#TRUE
4235  011630  001001                          BNE     $222
4236  011632                                            RETURN  NO.ERROR    ; GOOD
4237  011632  000405                          BR      $213
4238  011634                                          ENDIF
4239  011634                          $222:
4240  011634                                          RETURN  ERROR           ; BAD
4241  011634  000261                          SEC
4242  011636  000404                          BR      $214
4243
4244  011640  000000                                  REGSAV: .WORD 0
4245  011642  000000                                  TIMSAV: .WORD 0
4246  011644     000                                  FLAG:   .BYTE 0
4247  011645     000                                  HOLDSC: .BYTE 0
4248                                                ; WE ARE DONE  GO BACK HOME
4249  011646                                                  ENDRTN
4250  011646                          $213:
4251  011646  000241                          CLC
4252  011650                          $214:
4253  011650  000207                          RTS     PC
```

```
4254
4255                                    ;;**********************************************************************
4256  011652                            ROUTINE DATLNG <MASK>
4257  011652                            DATLNG:
4258                                    ;* ROUTINE:DATLNG
4259                                    ;*       THIS ROUTINE SETS UP A MASK FOR DATA, WITH
4260                                    ;*  INPUT - NOTHING IS PASSED TO THIS ROUTINE
4261                                    ;*      BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:
4262                                    ;*      $USWR-- THE WORD FOR SOFTWARE PARAMETERS
4263                                    ;*      DATA-- A MASK FOR THE LOCATION OF THE OCTAL
4264                                    ;*              NUMBER OF DATA BITS
4265                                    ;*  OUTPUT----
4266                                    ;*      MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED
4267                                    ;*              THE NUMBER OF WHICH IS DEFINED IN $USWR WORD.
4268
4269                                    ;;**********************************************************************
4270
4271  011652                                                                    LET MASK(R5) := #0              ; START
4272  011652  005065  000000                   CLR     MASK(R5)
4273  011656                                                                    LET NUMBR := $USWR AND #DATA
4274  011656  016767  167336  000062           MOV     $USWR,NUMBR
4275  011664  016746  000056                   MOV     NUMBR,-(SP)
4276  011670  042716  000017                   BIC     #DATA,(SP)
4277  011674  042667  000046                   BIC     (SP)+,NUMBR
4278
4279  011700                                                                    INCR I FROM #1 TO NUMBR BY #1
4280  011700  012767  000001  167364           MOV     #1,I
4281  011706  000402                           BR      $225
4282  011710                            $226:
4283  011710  005267  167356                   INC     I
4284  011714                            $225:
4285  011714  026767  167352  000024           CMP     I,NUMBR
4286  011722  003006                           BGT     $227
4287  011724                                                                        LET MASK(R5) := MASK(R5) SHIFT #1
4288  011724  006365  000000                   ASL     MASK(R5)
4289  011730                                                                        LET MASK(R5) := MASK(R5) SET.BY #1
4290  011730  052765  000001  000000           BIS     #1,MASK(R5)
4291  011736                                                                    ENDINC
4292  011736  000764                           BR      $226
4293  011740                            $227:
4294  011740                                                                    LET MASK(R5) := COMP MASK(R5)
4295  011740  005165  000000                   COM     MASK(R5)
4296  011744                                                                RETURN
4297  011744  000401                           BR      $223
4298  011746  000000                           NUMBR:0
4299  011750                                                            ENDRTN
4300  011750                            $223:
4301  011750                            $224:
4302  011750  000207                           RTS     PC
```

```
4303
4304                                    ;;*****************************************************************
4305  011752                            ROUTINE WAIT    <TIME>
4306  011752                            WAIT:
4307                                    ;* ROUTINE:WAIT
4308                                    ;*      THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
4309                                    ;*      MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
4310                                    ;*      THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
4311                                    ;*      REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
4312                                    ;*      TO APPROXIMATE 1 MILLI SEC.
4313                                    ;;*****************************************************************
4314  011752  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
4315  011754  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
4316  011756  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
4317  011760                                    LET R1 := TIME(R5)
4318  011760  016501  000000                    MOV     TIME(R5),R1
4319  011764                                    INCRU R2 FROM #1 TO R1 BY #1
4320  011764  012702  000001                    MOV     #1,R2
4321  011770  000402                            BR      $232
4322  011772                            $233:
4323  011772  062702  00000'                    ADD     #01,R2
4324  011776                            $232:
4325  011776  020201                            CMP     R2,R1
4326  012000  101010                            BHI     $234
4327  012002                                    INCR R3 FROM #0 TO #100 BY #1
4328  012002  005003                            CLR     R3
4329  012004  000401                            BR      $235
4330  012006                            $236:
4331  012006  005203                            INC     R3
4332  012010                            $235:
4333  012010  020327  000100                    CMP     R3,#100
4334  012014  003001                            BGT     $237
4335  012016                                    ENDINC
4336  012016  000773                            BR      $236
4337  012020                            $237:
4338  012020                                    ENDINC
4339  012020  000764                            BR      $233
4340  012022                            $234:
4341  012022  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
4342  012024  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
4343  012026  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
4344  012030                            ENDRTN
4345  012030                            $230:
4346  012030                            $231:
4347  012030  000207                            RTS     PC
```

```
4348
4349                                    .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
4350                                    ;;*****************************************************************
4351  012032                            INTSRV:
4352                                    ;* SERVICE ROUTINE: INTSRV
4353                                    ;*       THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
4354                                    ;*       'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
4355                                    ;*       THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
4356                                    ;*       TO LOOK FOR.
4357                                    ;;*****************************************************************
4358
4359                                                    ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
4360  012032                                                            LET INTFLAG := INTFLAG + #1
4361  012032  005267  000002                    INC     INTFLAG
4362  012036                                    ENDSRV                          ;THAT'S ALL
4363  012036  000002                            RTI
4364  012040  000000                    INTFLAG:  0
```

```
 4365
 4366  012042                          ROUTINE MYTYPE
 4367  012042                          MYTYPE:
 4368                                  ;;*****************************************************************
 4369  012042  104401  012050                TYPE    .65$            ;;TYPE ASCIZ STRING
 4370  012046  000405                        BR      64$             ;;GET OVER THE ASCIZ
 4371                                  ;;65$:  .ASCIZ  <CRLF>*TEST # *
 4372  012062                          64$:
 4373  012062  016746  167112                MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
 4374  012066  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4375  012070  104401  012076                TYPE    .67$            ;;TYPE ASCIZ STRING
 4376  012074  000405                        BR      66$             ;;GET OVER THE ASCIZ
 4377                                  ;;67$:  .ASCIZ  *,ERROR # *
 4378  012110                          66$:
 4379  012110  116767  167000  167060        MOVB    $ITEMB,$FATAL   ; APT FATAL ERROR NUMBER
 4380  012116  016746  167054                MOV     $FATAL,-(SP)    ;;SAVE $FATAL FOR TYPEOUT
 4381  012122  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4382  012124  104401  012132                TYPE    .69$            ;;TYPE ASCIZ STRING
 4383  012130  000404                        BR      68$             ;;GET OVER THE ASCIZ
 4384                                  ;;69$:  .ASCIZ  *,PC = *
 4385  012142                          68$:
 4386  012142  016746  166750                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 4387  012146  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4388  012150  104401  012156                TYPE    .71$            ;;TYPE ASCIZ STRING
 4389  012154  000404                        BR      70$             ;;GET OVER THE ASCIZ
 4390                                  ;;71$:  .ASCIZ  *,CSR: *
 4391  012166                          70$:
 4392  012166  016746  167062                MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
 4393  012172  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4394  012174  104401  012202                TYPE    .73$            ;;TYPE ASCIZ STRING
 4395  012200  000405                        BR      72$             ;;GET OVER THE ASCIZ
 4396                                  ;;73$:  .ASCIZ  *,VECTOR: *
 4397  012214                          72$:
 4398  012214  016746  167036                MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
 4399  012220  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4400  012222                          ENDRTN
 4401  012222                          $240:
 4402  012222                          $241:
 4403  012222  000207                        RTS     PC
```

```
                                ROUTINE CYCLE
                                CYCLE:
                                ;;*****************************************************************
                                ;* ROUTINE:     CYCLE
                                ;*       THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
                                ;*       ADDRESS  OF  DLV11-E  UNDER  TEST,  ADRS +2 TO
                                ;*       POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
                                ;*      IT  KEEPS  TRACK  OF THE CURRENT DEVICE AND BIT
                                ;*      MASKS.
                                ;;*****************************************************************
                                                                        REPEAT
                                $244:
                                                                            IF BITMASK EQ #0 THEN
012224  005767  000122                TST     BITMASK
012230  001027                        BNE     $245
                                                                                IF INITFLAG EQ #1 THEN
012232  026727  000116  000001        CMP     INITFLAG,#1
012240  001003                        BNE     $246
                                                                                    LET INITFLAG := #0
012242  005067  000106                CLR     INITFLAG
                                                                                ELSE
012246  000403                        BR      $247
                                $246:
                                                                                    CALL $EOP   ; AS A SUBROUTINE
012250  004767  000110                JSR     PC,$EOP
                                SPECIALADDRESS:                         ; BECAUSE $EOP RETURNS AS A JUMP
                                                                                    LET R0 := POP
012254  012600                        MOV     (SP)+,R0
                                                                                ENDIF
                                $247:
                                                                                LET BITMASK := #1
012256  012767  000001  000066        MOV     #1,BITMASK
                                                                                LET $DEVCT := #1
012264  012767  000001  166712        MOV     #1,$DEVCT
                                                                                LET ADDRESS := $BASE
012272  016767  166752  000056        MOV     $BASE,ADDRESS
                                                                                LET VECTOR := $VECT1
012300  016767  166740  000052        MOV     $VECT1,VECTOR
                                                                            ELSE
012306  000410                        BR      $250
                                $245:
                                                                                LET R4 := #10
012310  012704  000010                MOV     #10,R4
                                                                                LET BITMASK := BITMASK ROTATE 1
012314  006167  000032                ROL     BITMASK
                                                                                LET ADDRESS  := ADDRESS + R4
012320  060467  000032                ADD     R4,ADDRESS
                                                                                LET VECTOR := VECTOR + R4
012324  060467  000030                ADD     R4,VECTOR
                                                                            ENDIF
                                $250:
                                                                        UNTIL   BITMASK SETIN $DEVM
012330  036767  000016  166714        BIT     BITMASK,$DEVM
012336  001732                        BEQ     $244
```

```
012340                                                  LET ADRS :- #ADDRESS
012340  012701  012356                MOV     #ADDRESS,ADRS
012344                                                  LET $DEVCT :- $DEVCT + #1
012344  005267  166634                INC     $DEVCT
012350                                                  RETURN
012350  000404                        BR      $242
012352  000000              BITMASK:          0
012354  000001              INITFLAG:         1
012356  000000              ADDRESS:          0
012360  000000              VECTOR:           0

012362                                                  ENDRTN
012362                      $242:
012362                      $243:
012362  000207                        RTS     PC
```

```
                                .SBTTL  END OF PASS ROUTINE

                                ;;**********************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO SPECIALADDRESS

012364                          $EOP:
012364  000004                          SCOPE
012366  005067  166510                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
012372  005067  166562                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
012376  005267  166600                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
012402  042767  100000  166572        BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
012410  005327                        DEC     (PC)+           ;;LOOP?
012412  000001                  $EOPCT: .WORD   1
012414  003022                          BGT     $DOAGN          ;;YES
012416  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
012420  000001                  $ENDCT: .WORD   1
012422  012412                          $EOPCT
012424  104401  012471                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
012430  016746  166546                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
012434  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
012436  104401  012466                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
012442  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
012446  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
012450  000005                          RESET                   ;;CLEAR THE WORLD
012452  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
012454  000240                          NOP                     ;;SAVE ROOM
012456  000240                          NOP                     ;;FOR
012460  000240                          NOP                     ;;ACT11
012462                          $DOAGN:
012462  000137                          JMP     @(PC)+          ;;RETURN
012464  012254                  $RTNAD: .WORD   SPECIALADDRESS
012466     377     377     000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
012471     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
012476  050040  051501  020123
012504  000043
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;**********************************************************************
                                        ;POWER DOWN ROUTINE
012506  012737  012652  000024  $PWRDN: MOV     @#$ILLUP,@#PWRVEC ;;SET FOR FAST UP
012514  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
012522  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
012524  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
012526  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
012530  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
012532  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
012534  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
012536  017746  166376                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
012542  010667  000110                  MOV     SP,$SAVR6       ;;SAVE SP
012546  012737  012560  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
012554  000000                          HALT
012556  000776                          BR      .-2             ;;HANG UP

                                        ;;**********************************************************************
                                        ;POWER UP ROUTINE
012560  012737  012652  000024  $PWRUP: MOV     @#$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
012566  016706  000064                  MOV     $SAVR6,SP       ;;GET SP
012572  005067  000060                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
012576  005267  000054          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
012602  001375                          BNE     1$              ;;OF  WORD
012604  012677  166330                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
012610  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
012612  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
012614  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
012616  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
012620  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
012622  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
012624  012737  012506  000024          MOV     @#PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
012632  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
012640  104401                          TYPE                    ;REPORT THE POWER FAILURE
012642  012660                  $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
012644  012716                          MOV     (PC)+,(SP)      ;;RESTART AT START
012646  001336                  $PWRAD: .WORD   START           ;;RESTART ADDRESS
012650  000002                          RTI
012652  000000                  $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
012654  000775                          BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
012656  000000                  $SAVR6: 0                       ;;PUT THE SP HERE
012660  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
012666  000122
                                        .EVEN
```

```
                                        .SBTTL  TYPE ROUTINE

                                        ;;**********************************************************************
                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                        ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                        ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                        ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*

012670  105767  166263          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
012674  100002                          BPL     1$              ;;BR IF YES
012676  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
012700  000430                          BR      3$              ;;LEAVE
012702  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
012704  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
012710  122767  000001  166276          CMPB    @APTENV,$ENV    ;;RUNNING IN APT MODE
012716  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
012720  132767  000100  166267          BITB    @APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
012726  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
012730  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
012734  004767  000774                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
012740  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
012742  132767  000040  166245  62$:    BITB    @APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
012750  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
012752  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
012754  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
012756  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
012760  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
012762  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
012766  000002                          RTI                     ;;RETURN
012770  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
012774  001430                          BEQ     8$
012776  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
013002  001006                          BNE     5$
013004  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
013006  104401                          TYPE                    ;;TYPE A CR AND LF
013010  001171                          $CRLF
013012  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
013016  000755                          BR      2$              ;;GET NEXT CHARACTER
013020  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
013024  126726  166126          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
013030  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
013032  016746  166116                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
013036  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
013042  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
013044  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
013050  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
 4617  013054  000770                        BR      7$              ;;LOOP
 4618
 4619                                ;HORIZONTAL TAB PROCESSOR
 4620
 4621  013056  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 4622  013062  004767  000014          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
 4623  013066  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 4624  013074  001372                        BNE     9$              ;;TAB STOP
 4625  013076  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 4626  013100  000724                        BR      2$              ;;GET NEXT CHARACTER
 4627  013102  105777  166042          $TYPEC: TSTB    @#TPS           ;;WAIT UNTIL PRINTER IS READY
 4628  013106  100375                        BPL     $TYPEC
 4629  013110  116677  000002  166034        MOVB    2(SP),@#TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 4630  013116  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 4631  013124  001003                        BNE     1$              ;;BRANCH IF NO
 4632  013126  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 4633  013132  000406                        BR      $TYPEX          ;;EXIT
 4634  013134  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 4635  013142  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 4636  013144  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 4637  013146  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 4638  013150  000207          $TYPEX: RTS     PC
 4639
```

```
                                        .SBTTL  TTY INPUT ROUTINE
                                        ;;*****************************************************************
                                        .ENABL  LSB
                                        ;;*****************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
013152  022767  000176  165760  $CKSWR: CMP     @SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
013160  001074                          BNE     15$             ;;BRANCH IF NO
013162  105777  165756                  TSTB    @$TKS           ;;CHAR THERE?
013166  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
013170  117746  165752                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
013174  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
013200  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
013204  001062                          BNE     15$             ;;NO, RETURN TO USER
013206  126727  165722  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
013214  001456                          BEQ     15$             ;;BRANCH IF YES
                                        
013216  104401  013677                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
013222  104401  013704          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
013226  016746  164744                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
013232  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013234  104401  013715                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
013240  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
013242  005046                          CLR     -(SP)           ;;THE NEW SWR
013244  105777  165674          7$:     TSTB    @$TKS           ;;CHAR THERE?
013250  100375                          BPL     7$              ;;IF NOT TRY AGAIN
                                        
013252  117746  165670                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
013256  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        
                                        
                                        
013262  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
013266  001005                          BNE     10$             ;;BRANCH IF NOT
013270  104401  013672                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
013274  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
013300  000757                          BR      19$             ;;LET'S TRY IT AGAIN
                                        
                                        
013302  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
013306  001022                          BNE     16$             ;;BRANCH IF NO
013310  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
013314  001403                          BEQ     11$             ;;BRANCH IF YES
013316  016677  000002  165614          MOV     2(SP),@SWR      ;;SAVE NEW SWR
013324  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
013330  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
013334  126727  165575  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
013342  001003                          BNE     15$             ;;BRANCH IF NOT
013344  012777  000100  165572          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
013352  000002                  15$:    RTI                     ;;RETURN
013354  004767  177522          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
013360  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
```

```
 4696  013364  002420                        BLT     18$             ;;BRANCH IF YES
 4697  013366  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
 4698  013372  003015                        BGT     18$             ;;BRANCH IF YES
 4699  013374  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 4700  013400  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
 4701  013404  001403                        BEQ     17$             ;;BRANCH IF YES
 4702  013406  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
 4703  013410  006316                        ASL     (SP)            ;;    CHAR OVER TO MAKE
 4704  013412  006316                        ASL     (SP)            ;;    ROOM FOR NEW ONE.
 4705  013414  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 4706  013420  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 4707  013424  000707                        BR      7$              ;;GET THE NEXT ONE
 4708  013426  104401  001170        18$:    TYPE    .$QUES          ;;TYPE ?<CR><LF>
 4709  013432  000720                        BR      20$             ;;SIMULATE CONTROL-U
 4710                                .DSABL  LSB
 4711
 4712
 4713                                ;;*****************************************************************
 4714                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 4715                                ;*CALL:
 4716                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 4717                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 4718                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
 4719                                ;
 4720
 4721  013434  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 4722  013436  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
 4723  013444  105777  165474        1$:     TSTB    @#TKS           ;;WAIT FOR
 4724  013450  100375                        BPL     1$              ;;A CHARACTER
 4725  013452  117766  165470  000004        MOVB    @#TKB,4(SP)     ;;READ THE TTY
 4726  013460  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 4727  013466  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 4728  013474  001013                        BNE     3$              ;;BRANCH IF NO
 4729  013476  105777  165442        2$:     TSTB    @#TKS           ;;WAIT FOR A CHARACTER
 4730  013502  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
 4731  013504  117746  165436                MOVB    @#TKB,-(SP)     ;;GET CHARACTER
 4732  013510  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 4733  013514  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 4734  013520  001366                        BNE     2$              ;;IF NOT DISCARD IT
 4735  013522  000750                        BR      1$              ;;YES, RESUME
 4736  013524  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
 4737  013532  002407                        BLT     4$              ;;BRANCH IF YES
 4738  013534  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 4739  013542  003003                        BGT     4$              ;;BRANCH IF YES
 4740  013544  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 4741  013552  000002                4$:     RTI                     ;;GO BACK TO USER
 4742                                ;;*****************************************************************
 4743                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
 4744                                ;*CALL:
 4745                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
 4746                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
 4747                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 4748
 4749  013554  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
 4750  013556  012703  013662        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 4751  013562  022703  013672        2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
```

```
 4752  013566  101405                          BLOS    4$              ;;BR IF YES
 4753  013570  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4754  013572  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4755  013574  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4756  013600  001003                          BNE     3$              ;;SKIP IF NOT
 4757  013602  104401  001170          4$:     TYPE    .$QUES          ;;TYPE A '?'
 4758  013606  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4759  013610  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 4760  013614  104401  013660                  TYPE    .9$
 4761  013620  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 4762  013624  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 4763  013626  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4764  013632  104401  001172                  TYPE    ,$LF            ;;;YPE A LINE FEED
 4765  013636  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 4766  013640  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4767  013642  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 4768  013650  012766  013662  000004          MOV     #$TTYIN,4(SP)
 4769  013656  000002                          RTI                     ;;RETURN
 4770  013660     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
 4771  013661     000                  .BYTE   0                       ;;TERMINATOR
 4772  013662  000010          $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
 4773  013672  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
 4774  013677     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
 4775  013704  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
 4776  013712  020075     000
 4777  013715     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
 4778  013722  036440  000040
```

```
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;**************************************************************
013726  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
013734  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
013742  000403                         BR      $ATYC
013744  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
013752                          $ATYC:
013752  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013754  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013756  105767  000206                 TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
013762  001450                         BEQ     5$              ;;IF NOT:  BR
013764  122767  000001  165222         CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
013772  001031                         BNE     3$              ;;IF NOT:  BR
013774  132767  000100  165213         BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
014002  001425                         BEQ     3$              ;;IF NOT:  BR
014004  017600  000004                 MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
014010  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
014016  005767  165152          1$:    TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
014022  001375                         BNE     1$              ;;IF NOT:  WAIT
014024  010067  165160                 MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
014030  105720                  2$:    TSTB    (R0)+           ;;FIND END OF MESSAGE
014032  001376                         BNE     2$
014034  166700  165150                 SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
014040  006200                         ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
014042  010067  165144                 MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
014046  012767  000004  165120         MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
014054  000413                         BR      5$
014056  017667  000004  000016  3$:    MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
014064  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
014072  016746  163700                 MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
014076  004767  176566                 JSR     PC,$TYPE        ;;CALL TYPE MACRO
014102  000000                  4$:    .WORD   0
014104                          5$:
014104  105767  000062         10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
014110  001416                         BEQ     12$             ;;IF NOT:  BR
014112  005767  165076                 TST     $ENV            ;;RUNNING UNDER APT?
014116  001413                         BEQ     12$             ;;IF NOT:  BR
014120  005767  165050         11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
014124  001375                         BNE     11$             ;;IF NOT:  WAIT
014126  017667  000004  165042         MOV     @4(SP),$FATAL   ;;GET ERROR #
014134  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
014142  005267  165026                 INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
014146  105067  000020         12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
014152  105067  000013                 CLRB    $LFLG           ;;CLEAR LOG FLAG
014156  105067  000006                 CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
014162  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
014164  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
```

```
 4827  014166  000207                      RTS     PC              ;;RETURN
 4828  014170     000              $MFLG:  .BYTE   0               ;;MESSG. FLAG
 4829  014171     000              $LFLG:  .BYTE   0               ;;LOG FLAG
 4830  014172     000              $FFLG:  .BYTE   0               ;;FATAL FLAG
 4831          014174                      .EVEN
 4832          000200              APTSIZE=200
 4833          000001              APTENV=001
 4834          000100              APTSPOOL=100
 4835          000040              APTCSUP=040
```

```
                                        .SBTTL  ERROR HANDLER ROUTINE

                                        ;;**************************************************************
                                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                        ;*AND GO TO MYTYPE ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1         HALT ON ERROR
                                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1         BELL ON ERROR
                                        ;*SW09=1         LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

014174                                  $ERROR:
014174  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
014176  105267  164701          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
014202  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
014204  016777  164672  164730          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
014212  032777  002000  164720          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
014220  001402                          BEQ     1$              ;;NO - SKIP
014222  104401  001164                  TYPE    .$BELL          ;;RING BELL
014226  005267  164660          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
014232  011667  164660                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
014236  162767  000002  164652          SUB     #2,$ERRPC
014244  117767  164646  164642          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
014252  032777  020000  164660          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
014260  001004                          BNE     20$             ;;SKIP TYPEOUTS
014262  004767  175554                  JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
014266  104401  001171                  TYPE    .$CRLF
014272                                  20$:
014272  122767  000001  164714          CMPB    @APTENV,$ENV    ;;RUNNING IN APT MODE
014300  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
014302  116767  164606  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
014310  004767  177430                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
014314     000                  21$:    .BYTE   0
014315     000                          .BYTE   0
014316  000777                  22$:    BR      22$             ;;APT ERROR LOOP
014320  005777  164614          2$:     TST     @SWR            ;;HALT ON ERROR
014324  100002                          BPL     3$              ;;SKIP IF CONTINUE
014326  000000                          HALT                    ;;HALT ON ERROR!
014330  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
014332  032777  001000  164600  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
014340  001402                          BEQ     4$              ;;BR IF NO
014342  016716  164542                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
014346  005767  164610          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
014352  001402                          BEQ     5$              ;;BR IF NONE
014354  016716  164602                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
014360                                  5$:
014360  022737  012452  000042          CMP     @#ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
014366  001001                          BNE     6$              ;;BRANCH IF NO
014370  000000                          HALT                    ;;YES
014372                                  6$:
014372  000002                          RTI                     ;;RETURN
```

```
                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                ;*SW14=1         LOOP ON TEST
                                ;*SW11=1         INHIBIT ITERATIONS
                                ;*SW09=1         LOOP ON ERROR
                                ;*SW08=1         LOOP ON TEST IN SWR<7:0>
                                ;*CALL
                                ;*      SCOPE           ;;SCOPE=IOT

014374                          $SCOPE:
014374  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
014376  032777  040000  164534  1$:     BIT     @#BIT14,@SWR    ;;LOOP ON PRESENT TEST?
014404  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;@@@@@START OF CODE FOR THE XOR TESTER@@@@@
014406  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
014410  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
014414  012737  014434  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
014422  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
014426  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
014432  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
014434  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
014436  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
014442  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
014444                          6$:;@@@@@END OF CODE FOR THE XOR TESTER@@@@@
014444  032777  000400  164466          BIT     @#BIT08,@SWR    ;;LOOP ON SPEC. TEST?
014452  001404                          BEQ     2$              ;;BR IF NO
014454  127767  164460  164420          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
014462  001465                          BEQ     $OVER           ;;BR IF YES
014464  105767  164413          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
014470  001421                          BEQ     3$              ;;BR IF NO
014472  126767  164417  164403          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
014500  101015                          BHI     3$              ;;BR IF NO
014502  032777  001000  164430          BIT     @#BIT09,@SWR    ;;LOOP ON ERROR?
014510  001404                          BEQ     4$              ;;BR IF NO
014512  016767  164372  164366  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
014520  000446                          BR      $OVER
014522  105067  164355          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
014526  005067  164426                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
014532  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
014534  032777  004000  164376  3$:     BIT     @#BIT11,@SWR    ;;INHIBIT ITERATIONS?
014542  001011                          BNE     1$              ;;BR IF YES
014544  005767  164432                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
014550  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
014552  005267  164326                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
014556  026767  164376  164320          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
014564  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
014566  012767  000001  164310  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
014574  016767  000052  164356          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
014602  105267  164274          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
014606  116767  164270  164364          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 4946  014614  011667  164266                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 4947  014620  011667  164264                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 4948  014624  005067  164332                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 4949  014630  112767  000001  164257         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 4950  014636  016777  164240  164276 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 4951  014644  016716  164236                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 4952  014650  000002                         RTI                     ;;FIXES PS
 4953  014652  003720                 $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
```

```
                                        .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

                                        ;;**********************************************************************
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE

014654                                  $TYPDS:
014654  010046                                 MOV     R0,-(SP)        ;;PUSH R0 ON STACK
014656  010146                                 MOV     R1,-(SP)        ;;PUSH R1 ON STACK
014660  010246                                 MOV     R2,-(SP)        ;;PUSH R2 ON STACK
014662  010346                                 MOV     R3,-(SP)        ;;PUSH R3 ON STACK
014664  010546                                 MOV     R5,-(SP)        ;;PUSH R5 ON STACK
014666  012746  020200                        MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
014672  016605  000020                        MOV     20(SP),R5       ;;GET THE INPUT NUMBER
014676  100004                                BPL     1$              ;;BR IF INPUT IS POS.
014700  005405                                NEG     R5              ;;MAKE THE BINARY NUMBER POS.
014702  112766  000055  000001                MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
014710  005000                          1$:   CLR     R0              ;;ZERO THE CONSTANTS INDEX
014712  012703  015070                        MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
014716  112723  000040                        MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
014722  005002                          2$:   CLR     R2              ;;CLEAR THE BCD NUMBER
014724  016001  015060                        MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
014730  160105                          3$:   SUB     R1,R5           ;;FORM THIS BCD DIGIT
014732  002402                                BLT     4$              ;;BR IF DONE
014734  005202                                INC     R2              ;;INCREASE THE BCD DIGIT BY 1
014736  000774                                BR      3$
014740  060105                          4$:   ADD     R1,R5           ;;ADD BACK THE CONSTANT
014742  005702                                TST     R2              ;;CHECK IF BCD DIGIT=0
014744  001002                                BNE     5$              ;;FALL THROUGH IF 0
014746  105716                                TSTB    (SP)            ;;STILL DOING LEADING 0'S?
014750  100407                                BMI     7$              ;;BR IF YES
014752  106316                          5$:   ASLB    (SP)            ;;MSD?
014754  103003                                BCC     6$              ;;BR IF NO
014756  116663  000001  177777                MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
014764  052702  000060                  6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
014770  052702  000040                  7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
014774  110223                                MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
014776  005720                                TST     (R0)+           ;;JUST INCREMENTING
015000  020027  000010                        CMP     R0,#10          ;;CHECK THE TABLE INDEX
015004  002746                                BLT     2$              ;;GO DO THE NEXT DIGIT
015006  003002                                BGT     8$              ;;GO TO EXIT
015010  010502                                MOV     R5,R2           ;;GET THE LSD
015012  000764                                BR      6$              ;;GO CHANGE TO ASCII
015014  105726                          8$:   TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
015016  100003                                BPL     9$              ;;BR IF NO
015020  116663  177777  177776                MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
015026  105013                          9$:   CLRB    (R3)            ;;SET THE TERMINATOR
015030  012605                                MOV     (SP)+,R5        ;;POP STACK INTO R5
015032  012603                                MOV     (SP)+,R3        ;;POP STACK INTO R3
015034  012602                                MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
5010  015036  012601                        MOV     (SP)+,P1        ;;POP STACK INTO R1
5011  015040  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
5012  015042  104401  015070                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
5013  015046  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
5014  015054  012616                        MOV     (SP)+,(SP)
5015  015056  000002                        RTI                     ;;RETURN TO USER
5016  015060  023420                $DTBL:  10000.
5017  015062  001750                        1000.
5018  015064  000144                        100.
5019  015066  000012                        10.
5020  015070  000004                $DBLK:  .BLKW   4
```

```
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;**********************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

015100  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
015104  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
015112  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
015116  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
015122  000406                          BR      $TYPON
015124  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
015132  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
015140  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
015146  010346                          MOV     R3,-(SP)        ;;SAVE R3
015150  010446                          MOV     R4,-(SP)        ;;SAVE R4
015152  010546                          MOV     R5,-(SP)        ;;SAVE R5
015154  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
015160  005404                          NEG     R4
015162  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
015166  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
015172  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
015176  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
015202  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
015204  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
015206  000404                          BR      3$              ;;GO DO MSB
015210  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
015212  006105                          ROL     R5
015214  006105                          ROL     R5
015216  010503                          MOV     R5,R3
015220  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
015222  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
015226  100016                          BPL     7$              ;;BR IF NO
015230  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
015234  001002                          BNE     4$              ;;TEST FOR 0
015236  005704                          TST     R4              ;;SUPPRESS THIS 0?
015240  001403                          BEQ     5$              ;;BR IF YES
```

```
5077  015242  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5078  015244  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5079  015250  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5080  015254  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
5081  015260  104401  015320                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
5082  015264  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
5083  015270  003347                  BGT     2$              ;;BR IF MORE TO DO
5084  015272  002402                  BLT     6$              ;;BR IF DONE
5085  015274  005204                  INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5086  015276  000744                  BR      2$              ;;GO DO THE LAST DIGIT
5087  015300  012605          6$:     MOV     (SP)+,R5        ;;RESTORE R5
5088  015302  012604                  MOV     (SP)+,R4        ;;RESTORE R4
5089  015304  012603                  MOV     (SP)+,R3        ;;RESTORE R3
5090  015306  016666  000002  000004  MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5091  015314  012616                  MOV     (SP)+,(SP)
5092  015316  000002                  RTI                     ;;RETURN
5093  015320     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5094  015321     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5095  015322     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
5096  015323     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
5097  015324  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
MAINDEC-ZZ CVDVA D      MACY11 30A(1052)  12-JUL-84  09:52  PAGE 125
CVDVAD.P11    12-JUL 84 05:04           TRAP DECODER

                                        .SBTTL  TRAP DECODER

                                        ;;**********************************************************************
                                        ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                        ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                        ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                        ;*GO TO THAT ROUTINE.

015326  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
015330  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
015334  005740                          TST     -(R0)           ;;BACKUP BY 2
015336  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
015340  006300                          ASL     R0              ;;POSITION FOR INDEXING
015342  016000  015362                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
015346  000200                          RTS     R0              ;;GO TO ROUTINE

                                        ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

015350  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
015352  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
015360  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

                                        ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                        ;*BY THE "TRAP" INSTRUCTION.

                                        ;       ROUTINE
                                        ;       -------
015362  015350                  $TRPAD: .WORD   $TRAP2
015364  012670                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
015366  015124                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
015370  015100                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
015372  015140                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
015374  014654                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

015376  013222                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

015400  013152                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
015402  013434                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
015404  013554                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
        000001                  .END
```

```
ABASE = 175610            1#   1017    1058
ACDW1 = 000000          1017
ACDW2 = 000000          1017
ACPUOP= 000000          1017    1032
ADDRES  012356          4440*   4451*   4461    4468#
ADDW0 = 000000          1017
ADDW1 = 000000          1017
ADDW10= 000000          1017
ADDW11= 000000          1017
ADDW12= 000000          1017
ADDW13= 000000          1017
ADDW14= 000000          1017
ADDW15= 000000          1017
ADDW2 = 000000          1017
ADDW3 = 000000          1017
ADDW4 = 000000          1017
ADDW5 = 000000          1017
ADDW6 = 000000          1017
ADDW7 = 000000          1017
ADDW8 = 000000          1017
ADDW9 = 000000          1017
ADEVCT= 000000          1017    1023
ADEVM = 000001            1#    1017    1059
AENV  = 000000          1017    1028
AENVM = 000000          1017    1029
AFATAL= 000000          1017    1020
AMADR1= 000000          1017    1045
AMADR2= 000000          1017    1049
AMADR3= 000000          1017    1052
AMADR4= 000000          1017    1055
AMAMS1= 000000          1017    1039
AMAMS2= 000000          1017    1047
AMAMS3= 000000          1017    1050
AMAMS4= 000000          1017    1053
AMSGAD= 000000          1017    1025
AMSGLG= 000000          1017    1026
AMSGTY= 000000          1017    1019
AMTYP1= 000000          1017    1040
AMTYP2= 000000          1017    1048
AMTYP3= 000000          1017    1051
AMTYP4= 000000          1017    1054
APASS = 000000          1017    1022
APRIOR= 000000          1017
APTCSU= 000040          4591    4835#
APTENV= 000001          2697    2906    3183    3847    4584    4791    4833#   4867
APTSIZ= 000200          1128    4832#
APTSPO= 000100          4586    4793    4834#
ASWREG= 000000          1017    1030
ATESTN= 000000          1017    1021
AUNIT = 000000          1017    1024
AUSWR = 071110            1#    1017    1031
AVECT1= 000300            1#    1017    1056
AVECT2= 000000          1017    1057
BADBRK  011326          4114#
BAUD  = 007400           921#
BAUDRA  007210          3320#
```

```
BITMAS  012352          4417    4436*   4449*   4457    4466#
BIT0    = 000001        796#
BIT00   = 000001        786#    796     865     886     907
BIT01   = 000002        785#    795     844     864     906
BIT02   = 000004        784#    794     843     863     884     905
BIT03   = 000010        783#    793     842     862     904
BIT04   = 000020        782#    792     861     903
BIT05   = 000040        781#    791     840     860     902
BIT06   = 000100        780#    790     839     859     880     901
BIT07   = 000200        779#    789     838     858     879     900
BIT08   = 000400        778#    788     4920
BIT09   = 001000        777#    787     4878    4928
BIT1    = 000002        795#
BIT10   = 002000        776#    835     4855
BIT11   = 004000        775#    834     874     4935
BIT12   = 010000        774#    833     853     873
BIT13   = 020000        773#    832     852     872     4862
BIT14   = 040000        772#    831     851     871     4906
BIT15   = 100000        771#    830     850     870
BIT2    = 000004        794#
BIT3    = 000010        793#
BIT4    = 000020        792#
BIT5    = 000040        791#
BIT6    = 000100        790#
BIT7    = 000200        789#
BIT8    = 000400        788#
BIT9    = 001000        787#
BPTVEC  = 000014        803#
BREAK   = 000001        886#    1272    1285    1288    1302    1305    1319    1324    4082
BRK     = 010000        922#    1260    4064
CABLE   = 020000        923#    1996    2050    2141    2230    2319    2410    2510    2614    3560    3742
CARDET  = 010000        833#    2075    2095    2115
CKSWR   = 104407        4851    4877    4905    5137#
CLR     = 000000        824#    2916    4191
CLRSEN  = 020000        832#    2164    2184    2204
COMSPD  = 000100        916#
CR      = 000015        711#    4630    4640
CRLF    = 000200        712#    1152    4129    4372    4601    4640
CYCLE   012224          1157    4405#
DATA    = 000017        913#    4275    4276
DATAIE  = 000040        840#    1739    1752    1755    1769    1772    1786    1791    3573    3584    3617
DATAIN  = 100000        830#    1966    2443    2461    2470    2487    2544    2568    2591    2651    2673
DATLNG  011652          3656    3762    3857    4257#
DDISP   = 177570        718#     998    1116
DISPLA  001142          998#    1116*   1124*   4854*   4950*
DISPRE  000174          933#    1124
DLADD   001254          1078#   1161*   1166    1169    1171    1174    1177    1180    1196    1228    4130    4392
DLADDR  = 175610        820#    1078    1080    1081    1082    1083    1084
DLVEC   001256          1079#   1164*   3350    3440    3472    3623    3880    4136    4398
DSWR    = 177570        717#     997    1115
DTR     = 000002        844#    1518    1521    1535    1537    1550    1553    2072    2092    2112    2161    2181    2201
                        2431    2459    2485
EMTVEC  = 000030        806#    1099*   1100*
ERRCNT  011026          3905*   3927    3933    3949#   4016    4027*
ERROR   = 100000        850#    3084    3148
ERRVEC  = 000004        799#    1113    1114*   1125*   1911    4912*   4914*   4917*
```

```
EVENOD= 000040          915#
FALSE = 000000          4172#   4179
FLAG    011644          4179*   4206*   4211    4234    4246#
FRERR = 020000          852#
FRFD  = 040000          924#    1507    1574    2050    2141    2230    2410    2510
GNS   = ****** U        932     1151    4128    4134    4140    4371    4377    4384    4390    4396    5129    5130    5131
                        5132    5133    5135    5137    5138    5139
GTSWR = 104406          1146    5135#
HOLD    011072          3968*   3969*   3972    3985#
HOLDSC  011645          4191*   4196*   4202    4247#
HOWLON= 000000          4151#   4177
MT    = 000011          709#    4599    4640
I       001272          1085#   1205*   1226*   1229    1231    4280*   4283*   4285
ILLMEM= 000004          811#    1200    1236
INITFL  012354          1154*   4420    4423*   4467#
INTFLA  012040          1212*   1217    3346*   3386    3390    3412*   3428    3480*   3522    3526    3570*   3596    3600
                        4361*   4364#
INTSRV  012032          1201    3358    3473    4351#
IOTVEC= 000020          804#    1097*   1098*
LF    = 000012          710#    4634    4640
LOOP    001744          1155#   4145
MAINT = 000004          884#    1348    1361    1364    1378    1381    1395    1400    2789    2848    2914    3196    3483
                        3544    3645    3729    3752    3911    4073
MASK  = 000000          4258#   4272*   4288*   4290*   4295*
MAX     006422          2942    2946    3021#
MYTYPE  012042          4367#   4864
NEW     007242          3219*   3221*   3237*   3239*   3248    3256    3258    3276    3278    3326#
NUMBER  011030          3925    3950#   3975    4033
NUMBR   011746          4274*   4275    4277*   4285    4298#
OLD     007246          3192*   3194*   3248    3256    3258    3276*   3278*   3327#
ORERR = 040000          851#    3066    3105    3133
PARITY= 000020          914#
PBAUDS= 004000          874#
PBAUDO= 010000          873#
PBAUD1= 020000          872#
PBAUD2= 040000          871#
PBAUD3= 100000          870#
PBR   = 000200          917#    3174
PERR  = 010000          853#
PIRQ  = 177772          716#
PIRQVE= 000240          810#
PR0   = 000000          733#    3368    3417    3489    3576    3649    3755    3874
PR1   = 000040          734#
PR2   = 000100          735#
PR3   = 000140          736#
PR4   = 000200          737#
PR5   = 000240          738#
PR6   = 000300          739#
PR7   = 00034C          740#    1202    1420    2421    2521    2624    3359    3474    3885    3891
PS    = 177776          713#    714
PSW   = 177776          714#
PWRVEC= 000024          805#    1103*   1104*   4520*   4521*   4530*   4536*   4548*   4549*
RATES   007170          3210    3289    3294#
RBUF    001262          1081#   1171*   1172*   2883    2953    3008    3062    3105    3133    3148    3207    3518    3660
                        3698    3765    3802    4003    4077    4103
RCSR    001260          1080#   1169*   1518*   1521    1535*   1537    1550*   1553    1587    1600*   1603    1617*   1620
```

```
                    1634*  1639   1663   1676*  1679   1693*  1696   1710*  1715   1739   1752*  1755   1769*
                    1772   1786*  1791   1815   1828*  1831   1845*  1848   1862*  1867   1901   1966   2011
                    2072*  2075   2092*  2095   2112*  2115   2161*  2164   2181*  2184   2201*  2204   2251*
                    2254   2271*  2274   2291*  2294   2339*  2342   2359*  2362   2379*  2382   2431*  2440
                    2443   2459*  2461   2470   2485*  2487   2532*  2541   2544   2560*  2568   2583*  2591
                    2633*  2635   2643*  2651   2665*  2673   2804   2829   2863   2886   2928   2967   2971
                    2974   2994   3011   3227   3486*  3503   3510*  3547*  3573*  3581*  3584*  3586*  3617*
                    3619*  3691   3794   3918*  3944*  4037*  4089
RCVRAC= 004000       834#  2011   2928   2967   2974
RCVRDO= 000200       838#  1901   2805   2829   2864   2886   2971   2994   3011   3227   3504   3692   3795
                    4090
RCVRIE= 000100       839#  1815   1828   1831   1845   1848   1862   1867   3486   3510   3547   3918   4037
RDATA0= 000001       865#
RDATA1= 000002       864#
RDATA2= 000004       863#
RDATA3= 000010       862#
RDATA4= 000020       861#
RDATA5= 000040       860#
RDATA6= 000100       859#
RDATA7= 000200       858#
RDCHR = 104410      4753   5138#
RDLIN = 104411      5139#
REC     011076      3883   3995#
REG   = 000004     4151#   4175
REGSAV  011640     4175*   4188   4244#
REQSEN= 000004       843#  1587   1600   1603   1617   1620   1634   1639   2251   2271   2291   2532   2560
                    2583   3581   3586   3619
RESVEC= 000010       800#
RHLD    011166     4007*   4008*  4012   4020   4045#
RING  = 040000       831#  2254   2274   2294
R0050   007170     3303#
R0070   007171     3304#
R0110   007172     3305#
R0135   007173     3306#
R0150   007174     3307#
R0200   007177     3310#
R0300   007175     3308#
R0600   007176     3309#
R10000  007207     3318#
R1800   007200     3311#
R2000   007201     3312#
R2400   007202     3313#
R3600   007203     3314#
R4800   007204     3315#
R5STAC  001334     1087#  1183
R7200   007205     3316#
R9600   007206     3317#
SB      011032     3951#  4020*
SECREC= 002000       835#  2342   2362   2382
SECXHI= 000010       842#  1663   1676   1679   1693   1696   1710   1715   2339   2359   2379   2633   2643
                    2665
SET   = 177777       823#  2719   2762   2803   2862   2931   2940   3502   3677   3690   3781   3793   4088
                    4196
SETCLR= 000006     4151#   4202
SPECIA  012254     4430#   4511
STACK = 001100       704#  1095
```

```
START   001336            936    1088#  4553
STKLMT= 177774            715#
SWR     001140            997#   1093   1115*  1117   1123*  1130*  1144   4528   4541*  4650   4687*  4855   4862
                         4874    4878   4906   4920   4922   4928   4935
SWREG   000176            934#   1123   1144   4650   4663
SW0   = 000001            768#
SW00  = 000001            758#    768
SW01  = 000002            757#    767
SW02  = 000004            756#    766
SW03  = 000010            755#    765
SW04  = 000020            754#    764
SW05  = 000040            753#    763
SW06  = 000100            752#    762
SW07  = 000200            751#    761
SW08  = 000400            750#    760
SW09  = 001000            749#    759
SW1   = 000002            767#
SW10  = 002000            748#
SW11  = 004000            747#
SW12  = 010000            746#
SW13  = 020000            745#
SW14  = 040000            744#
SW15  = 100000            743#
SW2   = 000004            766#
SW3   = 000010            765#
SW4   = 000020            764#
SW5   = 000040            763#
SW6   = 000100            762#
SW7   = 000200            761#
SW8   = 000400            760#
SW9   = 001000            759#
TBITVE= 000014            810#
TBUF    001270           1084#   1180*  1181*  2713*  2744*  2796*  2856*  2924*  3042*  3052*  3124*  3216*  3496*
                         3686*   3790*  3972*  4085*
TCSR    001264           1082#   1174*  1175*  1272   1285*  1288   1302*  1305   1319*  1324   1348   1361*  1364
                         1378*   1381   1395*  1400   1430   1443*  1446   1460*  1463   1477*  1482   1934   2720
                         2749    2763   2789*  2848*  2914*  3196*  3365*  3375*  3415*  3483*  3544*  3645*  3678
                         3729*   3752*  3782   3911*  3915*  3942*  3979*  4073*  4082*
TCSRHI  001266           1083#   1177*  1178*  3210*  3289*
TDATA0= 000001            907#
TDATA1= 000002            906#
TDATA2= 000004            905#
TDATA3= 000010            904#
TDATA4= 000020            903#
TDATA5= 000040            902#
TDATA6= 000100            901#
TDATA7= 000200            900#
TIME  = 000000           4307#   4318
TIMER   011474           2723    2766   2807   2866   3506   3681   3694   3785   3797   4092   4150#
TIMSAV  011642           4177*   4213   4224*  4245#
TKVEC = 000060            808#
TPVEC = 000064            809#
TRAN    011034           3889    3960#
TRAPVE= 000034            807#   1101*  1102*
TRTVEC= 000014            802#
TRUE  = 000001           4171#   4206   4211   4234
```

```
TST1    002060          1192#
TST10   003412          1731#
TST11   003554          1807#
TST12   003716          1890#
TST13   003756          1923#
TST14   004016          1956#
TST15   004056          1990#
TST16   004136          2001    2045#
TST17   004276          2057    2136#
TST2    002226          1255#
TST20   004436          2146    2225#
TST21   004576          2236    2314#
TST22   004736          2324    2405#
TST23   005142          2416    2505#
TST24   005366          2516    2600    2609#
TST25   005554          2619    2691#
TST26   005742          2701    2783#
TST27   006056          2842#
TST3    002410          1264    1339#
TST30   006166          2900#
TST31   006424          2910    2956    2983    3020    3030#
TST32   006670          3075    3095    3113    3143    3158    3170#
TST33   007252          3178    3187    3292    3341#
TST34   007530          3465#
TST35   010152          3565    3640#
TST36   010366          3720    3738#
TST37   010614          3747    3820    3840#
TST4    002552          1416#
TST40   011172          3851    3947    4060#
TST41   011354          4068    4113    4124#
TST5    002726          1503#
TST6    003066          1511    1570#
TST7    003250          1578    1655#
TYPDS   104405          4143    4500    5133#
TYPE    104401          1138    4126    4132    4138    4369    4375    4382    4388    4394    4498    4501    4550    4604
                        4661    4662    4665    4678    4689    4708    4757    4760    4764    4857    4865    5012    5081
                        5129#
TYPDC   104402          4131    4137    4374    4381    4387    4393    4399    4664    5130#
TYPDN   104404          5132#
TYPDS   104403          5131#
VECTOR  012360          4442*   4453*   4469#
WAIT    011752          2436    2537    2565    2588    2648    2670    3047    3057    3129    3381    3425    3515    3591
                        4221    4306#
WAS     011033          3952#   4022*
WHICHB  000002          4151#   4188
XMITIE  000100           880#   1430    1443    1446    1460    1463    1477    1482    3365    3375    3415    3915    3942
                        3944    3979
XMITRD  000200           879#   1934    2721    2749    2764    3679    3783
ZZZ     011074          3983    3987#
ZZZZ    011170          4043    4046#
$APTMD  001000           957     963#
$ASTAT  ****** U        4813    4828
$ATYC   013752          4784    4786#
$ATY1   013726          4782#
$ATY3   013734          4589    4783#
$ATY4   013744          4785#   4870
```

```
$AUTOB  001134          994#  1148*  4658  4779
$BASE   001250         1058#  4440
$BDADR  001122          989#
$BDDAT  001126          991#
$BELL   001164         1009#  4857  4890
$BGNLF= 177777            1#
$CHARC  013146         4606*  4616*  4623  4632*  4637#
$CKSWR  013152         4650#  5137
$CMTAG  001100          977#  1090  1091  1099  1105  1106  1107
$CM3  = 000000         1007#
$CNTLG  013677         4661   4774#
$CNTLU  013672         4678   4773#
$CPUOP  001222         1032#
$CRLF   001171         1011#  4605  4640  4689  4773  4865  4890
$DBLK   015070         4978   5012  5020#
$DEVCT  001204         1023#  4438*  4463*
$DEVM   001252         1059#  4457
$DOAGN  012462         4494   4503  4509#
$DTBL   015060         4981   5016#
$ENDAD  012452          943   1136  4505#  4885
$ENDCT  012420         1105   4496#
$ENDMG  012471         4498   4513#
$ENULL  012466         4501   4512#
$ENV    001214         1028#  1142  2697  2906  3183  3847  4584  4791  4815  4867
$ENVM   001215         1029#  1128  4586  4591  4793
$EOP    012364         4428   4486#
$EOPCT  012412         1105*  4493#  4497
$ERFLG  001103          980#  4852*  4890  4895  4924  4926  4932*  4954
$ERMAX  001115          986#  1108*  4926  4949*  4954
$ERRFL= 000000        1154#  1155  1157#  1161#  1162  1164#  1165  1166#  1167  1169#  1170  1171#  1173
                      1174#  1176  1177#  1179  1180#  1182  1183#  1184  1196#  1197  1199#  1200#  1201#
                      1202#  1203#  1204  1205#  1206  1209#  1210  1212#  1213  1226#  1227  1228#  1230
                      1236#  1237#  1239#  1240#  1241  1263#  1264  1269#  1270  1283#  1284  1285#  1286
                      1299#  1300  1302#  1303  1316#  1317  1319#  1320  1345#  1346  1359#  1360  1361#
                      1362   1375#  1376  1378#  1379  1392#  1393  1395#  1396  1427#  1428  1441#  1442
                      1443#  1444  1457#  1458  1460#  1461  1474#  1475  1477#  1478  1510#  1511  1516#
                      1517   1518#  1519  1532#  1533  1535#  1536  1548#  1549  1550#  1551  1577#  1578
                      1584#  1585  1598#  1599  1600#  1601  1614#  1615  1617#  1618  1631#  1632  1634#
                      1635   1660#  1661  1674#  1675  1676#  1677  1690#  1691  1693#  1694  1706#  1707
                      1710#  1711  1736#  1737  1750#  1751  1752#  1753  1766#  1767  1769#  1770  1783#
                      1784   1786#  1787  1812#  1813  1826#  1827  1828#  1829  1842#  1843  1845#  1846
                      1859#  1860  1862#  1863  1899#  1900  1931#  1932  1964#  1965  2000#  2001  2008#
                      2009   2056#  2057  2068#  2069  2072#  2073  2088#  2089  2092#  2093  2108#  2109
                      2112#  2113  2145#  2146  2157#  2158  2161#  2162  2177#  2178  2181#  2182  2197#
                      2198   2201#  2202  2235#  2236  2247#  2248  2251#  2252  2267#  2268  2271#  2272
                      2287#  2288  2291#  2292  2323#  2324  2335#  2336  2339#  2340  2355#  2356  2359#
                      2360   2375#  2376  2379#  2380  2415#  2416  2428#  2429  2431#  2432  2434#  2440#
                      2441   2455#  2456  2459#  2460  2481#  2482  2485#  2486  2515#  2516  2528#  2529
                      2532#  2533  2535#  2541#  2542  2555#  2556  2560#  2561  2563#  2579#  2580  2583#
                      2584   2586#  2618#  2619  2633#  2634  2635#  2636  2639#  2640  2643#  2644  2646#
                      2662#  2663  2665#  2666  2668#  2700#  2701  2705#  2706  2713#  2714  2718#  2737#
                      2738   2744#  2745  2761#  2789#  2790  2792#  2793  2796#  2797  2802#  2822#  2823
                      2848#  2849  2850#  2851  2856#  2857  2861#  2883#  2884  2909#  2910  2914#  2915
                      2916#  2917  2918#  2919  2924#  2925  2931#  2932  2936#  2937  2953#  2954  2955#
                      2956   2982#  2983  3008#  3009  3035#  3036  3042#  3043  3045#  3052#  3053  3055#
                      3062#  3063  3074#  3075  3082#  3083  3094#  3095  3101#  3102  3119#  3120  3124#
```

```
                        3125    3127#   3142#   3143    3177#   3178    3186#   3187    3192#   3193    3194#   3195    3196#
                        3197    3199#   3200    3202    3206    3207#   3208    3210#   3211    3213#   3214    3216#   3217
                        3219#   3220    3221#   3222    3231#   3232    3237#   3238    3239#   3240    3276#   3277    3278#
                        3279    3284#   3288    3289#   3290    3346#   3347    3350#   3351    3353#   3354    3356#   3357#
                        3358#   3359#   3360#   3361    3362#   3363    3365#   3366    3375#   3376    3379#   3409#   3410
                        3412#   3413    3415#   3416    3423#   3442#   3441    3444#   3445#   3447#   3448#   3449    3471#
                        3472#   3473#   3474#   3475#   3476    3478#   3479    3480#   3481    3483#   3484    3486#   3487
                        3496#   3497    3501#   3510#   3511    3513#   3518#   3519    3544#   3545    3547#   3548    3558#
                        3559    3564#   3565    3570#   3571    3573#   3574    3581#   3582    3584#   3585    3586#   3587
                        3589#   3617#   3618    3619#   3620    3623#   3624    3627#   3628#   3630#   3631#   3632    3645#
                        3646    3655#   3660#   3661    3664#   3665    3667    3671    3676#   3686#   3687    3689#   3698#
                        3699    3706#   3708    3709#   3710    3719#   3720    3729#   3730    3746#   3747    3752#   3753
                        3761#   3765#   3766    3768#   3769    3771    3775    3780#   3790#   3791    3792#   3802#   3803
                        3806#   3808    3809#   3810    3819#   3820    3850#   3851    3856#   3880#   3881    3883#   3884
                        3885#   3886    3889#   3890    3891#   3892    3895#   3896    3898#   3899    3901#   3902    3905#
                        3906    3911#   3912    3915#   3916    3918#   3919    3942#   3943    3944#   3945    3965#   3966
                        3968#   3970    3972#   3973    3979#   3980    4000#   4001    4003#   4005    4007#   4009    4020#
                        4021    4022#   4023    4027#   4028    4037#   4038    4067#   4068    4073#   4074    4077#   4078
                        4082#   4083    4085#   4086    4087#   4175#   4176    4177#   4178    4179#   4180    4191#   4192
                        4196#   4197    4206#   4207    4219#   4224#   4225    4237#   4238    4241#   4243    4272#   4273
                        4274#   4278    4280#   4281    4283    4287    4288#   4289    4290#   4291    4295#   4296    4318#
                        4319    4320#   4321    4323    4327    4328#   4329    4331    4335    4361#   4362    4423#   4424
                        4428#   4432#   4433    4436#   4437    4438#   4439    4440#   4441    4442#   4443    4447#   4448
                        4449#   4450    4451#   4452    4453#   4454    4461#   4462    4463#   4464
$ERROR    014174        1099    4850#
$ERRPC    001116         987#   4386    4859*   4860*   4861    4890
$ERRTB    001254        1076#
$ERTTL    001112         984#   4142    4144*   4858*   4890
$ESCAP    001162        1008#   1107*   4881    4883    4890    4948*
$ETABL    001214        1027#
$ETEND    001254         969    1060#
$FATAL    001176        1020#   4379*   4380    4819*
$FFLG     014172        4782*   4785*   4813    4822*   4830#
$FILLC    001156        1005#   4609    4640
$FILLS    001155        1004#   4640
$F$AND=   000310          1#    1219    1262    1274    1290    1307    1326    1350    1366    1383    1402    1432    1448
                        1465    1484    1509    1523    1539    1555    1576    1589    1605    1622    1641    1665    1681
                        1698    1717    1741    1757    1774    1793    1817    1833    1850    1869    1903    1936    1968
                        1998    2013    2052    2077    2097    2117    2143    2166    2186    2206    2232    2256    2276
                        2296    2321    2344    2364    2384    2412    2445    2463    2472    2489    2512    2546    2570
                        2593    2616    2653    2675    2699    2751    2831    2888    2908    2930    2948    2969    2973
                        2976    2996    3013    3068    3086    3107    3135    3150    3176    3185    3226    3229    3250
                        3258    3388    3392    3430    3524    3528    3562    3598    3602    3715    3744    3815    3849
                        3935    3977    4014    4018    4035    4066    4105    4190    4204    4236    4419    4422
$F$BAD=   000401          1#    1219    1262    1274    1290    1307    1326    1350    1366    1383    1402    1432    1448
                        1465    1484    1509    1523    1539    1555    1576    1589    1605    1622    1641    1665    1681
                        1698    1717    1741    1757    1774    1793    1817    1833    1850    1869    1903    1936    1968
                        1998    2013    2052    2077    2097    2117    2143    2166    2186    2206    2232    2256    2276
                        2296    2321    2344    2364    2384    2412    2445    2463    2472    2489    2512    2546    2570
                        2593    2616    2653    2675    2699    2751    2831    2888    2908    2930    2948    2969    2973
                        2976    2996    3013    3068    3086    3107    3135    3150    3176    3185    3226    3229    3250
                        3260    3388    3392    3430    3524    3528    3562    3598    3602    3715    3744    3815    3849
                        3935    3977    4014    4018    4035    4066    4105    4190    4204    4236    4419    4422
$F$BLA=   000170          1#
$F$CAS=   000150          1#
$F$DEC=   000220          1#
```

FGOO= 000400 1# 1217 1260 1272 1288 1305 1324 1348 1364 1381 1400 1430 1446
1463 1482 1507 1521 1537 1553 1574 1587 1603 1620 1639 1663 1679
1696 1715 1739 1755 1772 1791 1815 1831 1848 1867 1901 1934 1966
1996 2011 2050 2075 2095 2115 2141 2164 2184 2204 2230 2254 2274
2294 2319 2342 2362 2382 2410 2443 2461 2470 2487 2510 2544 2568
2591 2614 2651 2673 2697 2728 2749 2769 2812 2829 2870 2886 2906
2928 2946 2966 2967 2971 2974 2994 3011 3046 3084 3105 3133 3148
3174 3183 3223 3224 3227 3248 3256 3386 3390 3428 3522 3526 3560
3596 3600 3713 3742 3813 3847 3933 3975 4012 4016 4033 4064 4095
4103 4188 4202 4234 4417 4420

FIF = 000110 1# 1217 1223 1260 1266 1272 1278 1288 1294 1305 1311 1324 1330
1348 1354 1364 1370 1381 1387 1400 1406 1430 1436 1446 1452 1463
1469 1482 1488 1507 1513 1521 1527 1537 1543 1553 1559 1574 1580
1587 1593 1603 1609 1620 1626 1639 1645 1663 1669 1679 1685 1696
1702 1715 1721 1739 1745 1755 1761 1772 1778 1791 1797 1815 1821
1831 1837 1848 1854 1867 1873 1901 1912 1934 1945 1966 1979 1996
2003 2011 2026 2050 2059 2075 2083 2095 2103 2115 2123 2141 2148
2164 2172 2184 2192 2204 2212 2230 2238 2254 2262 2274 2282 2294
2302 2319 2326 2342 2350 2362 2370 2382 2390 2410 2418 2443 2449
2461 2467 2470 2476 2487 2493 2510 2518 2544 2550 2568 2574 2591
2597 2614 2621 2651 2657 2673 2679 2697 2703 2728 2733 2749 2755
2769 2774 2812 2817 2829 2835 2870 2875 2886 2892 2906 2912 2928
2933 2938 2946 2958 2971 2974 2985 2987 2994 3001 3011 3017 3066
3077 3084 3097 3105 3115 3133 3145 3148 3155 3174 3180 3183 3189
3227 3233 3241 3248 3252 3256 3258 3262 3271 3273 3386 3390 3396
3403 3405 3428 3434 3522 3526 3533 3539 3541 3560 3567 3596 3600
3607 3613 3615 3713 3722 3742 3749 3813 3822 3847 3853 3933 3939
3975 3981 4012 4016 4024 4029 4033 4040 4064 4070 4095 4100 4103
4109 4188 4193 4198 4202 4208 4234 4239 4417 4420 4425 4434 4444
4455

FINC= 000210 1# 3199 3281 3664 3724 3768 3825 4280 4292 4320 4328 4336 4339

FLOO= 000200 1# 4185 4211 4226

FNAM= 000160 1#

FNO = 000403 1# 1217 1260 1272 1288 1305 1324 1348 1364 1381 1400 1430 1446
1463 1482 1507 1521 1537 1553 1574 1587 1603 1620 1639 1663 1679
1696 1715 1739 1755 1772 1791 1815 1831 1848 1867 1901 1934 1966
1996 2011 2050 2075 2095 2115 2141 2164 2184 2204 2230 2254 2274
2294 2319 2342 2362 2382 2410 2443 2461 2470 2487 2510 2544 2568
2591 2614 2651 2673 2697 2749 2829 2886 2906 2928 2946 2967 2971
2974 2994 3011 3066 3084 3105 3133 3148 3174 3183 3224 3227 3248
3256 3258 3386 3390 3428 3522 3526 3560 3596 3600 3713 3742 3813
3847 3933 3975 4012 4016 4033 4064 4188 4417 4420

FOR = 000320 1# 1219 1262 1274 1290 1307 1326 1350 1366 1383 1402 1432 1448
1465 1484 1509 1523 1539 1555 1576 1589 1605 1622 1641 1665 1681
1698 1717 1741 1757 1774 1793 1817 1833 1850 1869 1903 1936 1968
1998 2013 2052 2077 2097 2117 2143 2166 2186 2206 2232 2256 2276
2296 2321 2344 2364 2384 2412 2445 2463 2472 2489 2512 2546 2570
2593 2616 2653 2675 2699 2751 2831 2888 2908 2930 2948 2969 2973
2976 2996 3013 3068 3086 3107 3135 3150 3176 3185 3226 3229 3250
3258 3260 3388 3392 3430 3524 3528 3562 3598 3602 3715 3744 3815
3849 3935 3977 4014 4018 4035 4066 4105 4190 4204 4236 4419 4422

FRTN= 000300 1# 4151 4250 4258 4300 4307 4345 4368 4401 4406 4472

FSEL= 000140 1#

FUNT= 000130 1# 1207 1231 2926 2940 3923 3925 4415 4457

FWMI= 000120 1# 2966 2967 2989 3223 3224 3244 3258

FYES= 000402 1# 1217 1260 1272 1288 1305 1324 1348 1364 1381 1400 1430 1446

```
$ITEMB  001114          985#  4379    4861*  4869    4890
$LF     001172         1012#  4640    4764   4773    4890
$LFLG   014171         4823*  4829#
$LOCTA* 177777            1#  1207    1208   1218    1219    1223    1224    1232    1233    1261    1262    1266    1267
                       1273   1274    1278   1279    1289    1290    1294    1295    1306    1307    1311    1312    1325
                       1326   1330    1331   1349    1350    1354    1355    1365    1366    1370    1371    1382    1383
                       1387   1388    1401   1402    1406    1407    1431    1432    1436    1437    1447    1448    1452
                       1453   1464    1465   1469    1470    1483    1484    1488    1489    1508    1509    1513    1514
                       1522   1523    1527   1528    1538    1539    1543    1544    1554    1555    1559    1560    1575
                       1576   1580    1581   1588    1589    1593    1594    1604    1605    1609    1610    1621    1622
                       1626   1627    1640   1641    1645    1646    1664    1665    1669    1670    1680    1681    1685
                       1686   1697    1698   1702    1703    1716    1717    1721    1722    1740    1741    1745    1746
                       1756   1757    1761   1762    1773    1774    1778    1779    1792    1793    1797    1798    1816
                       1817   1821    1822   1832    1833    1837    1838    1849    1850    1854    1855    1868    1869
                       1873   1874    1902   1903    1912    1913    1935    1936    1945    1946    1967    1968    1979
                       1980   1997    1998   2003    2004    2012    2013    2026    2027    2051    2052    2059    2060
                       2076   2077    2083   2084    2096    2097    2103    2104    2116    2117    2123    2124    2142
                       2143   2148    2149   2165    2166    2172    2173    2185    2186    2192    2193    2205    2206
                       2212   2213    2231   2232    2238    2239    2255    2256    2262    2263    2275    2276    2282
                       2283   2295    2296   2302    2303    2320    2321    2326    2327    2343    2344    2350    2351
                       2363   2364    2370   2371    2383    2384    2390    2391    2411    2412    2418    2419    2444
                       2445   2449    2450   2462    2463    2467    2468    2471    2472    2476    2477    2488    2489
                       2493   2494    2511   2512    2518    2519    2545    2546    2550    2551    2569    2570    2574
                       2575   2592    2593   2597    2598    2615    2616    2621    2622    2652    2653    2657    2658
                       2674   2675    2679   2680    2698    2699    2703    2704    2728    2729    2733    2734    2750
                       2751   2755    2756   2769    2770    2774    2775    2812    2813    2817    2818    2830    2831
                       2835   2836    2870   2871    2875    2876    2887    2888    2892    2893    2907    2908    2912
                       2913   2926    2927   2929    2930    2933    2934    2935    2938    2939    2941    2942    2943
                       2944   2945    2947   2948    2958    2959    2966    2967    2968    2969    2972    2973    2975
                       2976   2985    2986   2987    2988    2989    2990    2991    2995    2996    3001    3002    3012
                       3013   3017    3018   3067    3068    3077    3078    3085    3086    3097    3098    3106    3107
                       3115   3116    3134   3135    3145    3146    3149    3150    3155    3156    3175    3176    3180
                       3181   3184    3185   3189    3190    3200    3201    3202    3203    3204    3205    3206    3223
                       3224   3225    3226   3228    3229    3233    3234    3235    3241    3242    3244    3245    3246
                       3249   3250    3252   3253    3254    3257    3258    3259    3260    3262    3263    3264    3271
                       3272   3273    3274   3281    3282    3283    3387    3388    3391    3392    3396    3397    3398
                       3403   3404    3405   3406    3429    3430    3434    3435    3523    3524    3527    3528    3533
                       3534   3535    3539   3540    3541    3542    3561    3562    3567    3568    3597    3598    3601
                       3602   3607    3608   3609    3613    3614    3615    3616    3665    3666    3667    3668    3669
                       3670   3671    3714   3715    3722    3723    3724    3725    3726    3743    3744    3749    3750
                       3769   3770    3771   3772    3773    3774    3775    3814    3815    3822    3823    3825    3826
                       3827   3848    3849   3853    3854    3923    3924    3926    3927    3928    3929    3930    3934
                       3935   3939    3940   3976    3977    3981    3982    4013    4014    4017    4018    4024    4025
                       4029   4030    4034   4035    4040    4041    4065    4066    4070    4071    4095    4096    4100
                       4101   4104    4105   4109    4110    4150    4185    4186    4189    4190    4193    4194    4195
                       4198   4199    4203   4204    4208    4209    4212    4213    4214    4215    4226    4227    4228
                       4235   4236    4237   4238    4239    4240    4242    4243    4250    4251    4252    4253    4257
                       4281   4282    4283   4284    4285    4286    4287    4292    4293    4294    4297    4298    4300
                       4301   4302    4306   4321    4322    4323    4324    4325    4326    4327    4329    4330    4331
                       4332   4333    4334   4335    4336    4337    4338    4339    4340    4341    4345    4346    4347
                       4367   4401    4402   4403    4405    4415    4416    4418    4419    4421    4422    4425    4426
                       4427   4434    4435   4444    4445    4446    4455    4456    4458    4459    4465    4466    4472
                       4473   4474
$LPADR  001106          982#  1109*  4930*  4946*  4951    4953
$LPERR  001110          983#  1110*  1209*  1269*  1283*  1299*  1316*  1345*  1359*  1375*  1392*  1427*  1441*
                       1457*  1474*  1516*  1532*  1548*  1584*  1598*  1614*  1631*  1660*  1674*  1690*  1706*
```

```
                1736* 1750* 1766* 1783* 1812* 1826* 1842* 1859* 1899* 1931* 1964* 2008* 2068*
                2088* 2108* 2157* 2177* 2197* 2247* 2267* 2287* 2335* 2355* 2375* 2428* 2455*
                2481* 2528* 2555* 2579* 2639* 2662* 2705* 2737* 2792* 2822* 2850* 3035* 3082*
                3101* 3119* 3362* 3409* 3478* 3558* 4880  4930  4947* 4953
$LSTCN= 177777    10  1207  1208  1217  1219  1223  1231  1260  1262  1266  1272  1274  1278
                1288  1290  1294  1305  1307  1311  1324  1326  1330  1348  1350  1354  1364
                1366  1370  1381  1383  1387  1400  1402  1406  1430  1432  1436  1446  1448
                1452  1463  1465  1469  1482  1484  1488  1507  1509  1513  1521  1523  1527
                1537  1539  1543  1553  1555  1559  1574  1576  1580  1587  1589  1593  1603
                1605  1609  1620  1622  1626  1639  1641  1645  1663  1665  1669  1679  1681
                1685  1696  1698  1702  1715  1717  1721  1739  1741  1745  1755  1757  1761
                1772  1774  1778  1791  1793  1797  1815  1817  1821  1831  1833  1837  1848
                1850  1854  1867  1869  1873  1901  1903  1912  1934  1936  1945  1966  1968
                1979  1996  1998  2003  2011  2013  2026  2050  2052  2059  2075  2077  2083
                2095  2097  2103  2115  2117  2123  2141  2143  2148  2164  2166  2172  2184
                2186  2192  2204  2206  2212  2230  2232  2238  2254  2256  2262  2274  2276
                2282  2294  2296  2302  2319  2321  2326  2342  2344  2350  2362  2364  2370
                2382  2384  2390  2410  2412  2418  2443  2445  2449  2461  2463  2467  2470
                2472  2476  2487  2489  2493  2510  2512  2518  2544  2546  2550  2568  2570
                2574  2591  2593  2597  2614  2616  2621  2651  2653  2657  2673  2675  2679
                2697  2699  2703  2728  2729  2733  2749  2751  2755  2769  2770  2774  2812
                2813  2817  2829  2831  2835  2870  2871  2875  2886  2888  2892  2906  2908
                2912  2926  2927  2928  2930  2934  2935  2938  2940  2945  2946  2948  2958
                2966  2967  2969  2971  2973  2974  2976  2985  2987  2989  2990  2994  2996
                3001  3011  3013  3017  3066  3068  3077  3084  3086  3097  3105  3107  3115
                3133  3135  3145  3148  3150  3155  3174  3176  3180  3183  3185  3189  3199
                3201  3202  3203  3206  3223  3224  3226  3227  3229  3234  3235  3241  3244
                3245  3248  3250  3253  3254  3256  3260  3263  3264  3271  3273  3281  3282
                3386  3388  3390  3392  3397  3398  3403  3405  3428  3430  3434  3522  3524
                3526  3528  3534  3535  3539  3541  3560  3562  3567  3596  3598  3600  3602
                3608  3609  3613  3615  3664  3666  3667  3668  3671  3713  3715  3722  3724
                3725  3742  3744  3749  3768  3770  3771  3772  3775  3813  3815  3822  3825
                3826  3847  3849  3853  3923  3924  3925  3930  3933  3935  3939  3975  3977
                3981  4012  4014  4016  4018  4024  4029  4033  4035  4040  4064  4066  4070
                4095  4096  4100  4103  4105  4109  4151  4185  4186  4188  4190  4194  4195
                4198  4202  4204  4208  4226  4227  4234  4236  4239  4250  4258  4280  4282
                4283  4284  4287  4292  4293  4300  4307  4320  4322  4323  4324  4327  4328
                4330  4331  4332  4335  4336  4337  4339  4340  4345  4368  4401  4406  4415
                4416  4417  4419  4420  4422  4426  4427  4434  4445  4446  4455  4457  4472
$LSTIN= 000000    10  1154  1155  1157  1158  1161  1162  1164  1165  1166  1167  1169  1170
                1171  1172  1173  1174  1175  1176  1177  1178  1179  1180  1181  1182  1183
                1184  1196  1197  1199  1200  1201  1202  1203  1204  1205  1206  1209  1210
                1212  1213  1217  1218  1219  1226  1227  1228  1229  1230  1231  1232  1233
                1236  1237  1238  1239  1240  1241  1260  1261  1262  1263  1264  1269  1270
                1272  1273  1274  1283  1284  1285  1286  1288  1289  1290  1299  1300  1302
                1303  1305  1306  1307  1316  1317  1319  1320  1324  1325  1326  1345  1346
                1348  1349  1350  1359  1360  1361  1362  1364  1365  1366  1375  1376  1378
                1379  1381  1382  1383  1392  1393  1395  1396  1400  1401  1402  1427  1428
                1430  1431  1432  1441  1442  1443  1444  1446  1447  1448  1457  1458  1460
                1461  1463  1464  1465  1474  1475  1477  1478  1482  1483  1484  1507  1508
                1509  1510  1511  1516  1517  1518  1519  1521  1522  1523  1532  1533  1535
                1536  1537  1538  1539  1548  1549  1550  1551  1553  1554  1555  1574  1575
                1576  1577  1578  1584  1585  1587  1588  1589  1598  1599  1600  1601  1603
                1604  1605  1614  1615  1617  1618  1620  1621  1622  1631  1632  1634  1635
                1639  1640  1641  1660  1661  1663  1664  1665  1674  1675  1676  1677  1679
                1680  1681  1690  1691  1693  1694  1696  1697  1698  1706  1707  1710  1711
```

1715	1716	1717	1736	1737	1739	1740	1741	1750	1751	1752	1753	1755
1756	1757	1766	1767	1769	1770	1772	1773	1774	1783	1784	1786	1787
1791	1792	1793	1812	1813	1815	1816	1817	1826	1827	1828	1829	1831
1832	1833	1842	1843	1845	1846	1848	1849	1850	1859	1860	1862	1863
1867	1868	1869	1899	1900	1901	1902	1903	1931	1932	1934	1935	1936
1964	1965	1966	1967	1968	1996	1997	1998	2000	2001	2008	2009	2011
2012	2013	2050	2051	2052	2056	2057	2068	2069	2072	2073	2075	2076
2077	2088	2089	2092	2093	2095	2096	2097	2108	2109	2112	2113	2115
2116	2117	2141	2142	2143	2145	2146	2157	2158	2161	2162	2164	2165
2166	2177	2178	2181	2182	2184	2185	2186	2197	2198	2201	2202	2204
2205	2206	2230	2231	2232	2235	2236	2247	2248	2251	2252	2254	2255
2256	2267	2268	2271	2272	2274	2275	2276	2287	2288	2291	2292	2294
2295	2296	2319	2320	2321	2323	2324	2335	2336	2339	2340	2342	2343
2344	2355	2356	2359	2360	2362	2363	2364	2375	2376	2379	2380	2382
2383	2384	2410	2411	2412	2415	2416	2428	2429	2431	2432	2434	2435
2436	2437	2438	2440	2441	2443	2444	2445	2455	2456	2459	2460	2461
2462	2463	2470	2471	2472	2481	2482	2485	2486	2487	2488	2489	2510
2511	2512	2515	2516	2528	2529	2532	2533	2535	2536	2537	2538	2539
2541	2542	2544	2545	2546	2555	2556	2560	2561	2563	2564	2565	2566
2567	2568	2569	2570	2579	2580	2583	2584	2586	2587	2588	2589	2590
2591	2592	2593	2614	2615	2616	2618	2619	2633	2634	2635	2636	2639
2640	2643	2644	2646	2647	2648	2649	2650	2651	2652	2653	2662	2663
2665	2666	2668	2669	2670	2671	2672	2673	2674	2675	2697	2698	2699
2700	2701	2705	2706	2713	2714	2718	2719	2720	2721	2722	2723	2724
2725	2728	2729	2737	2738	2744	2745	2749	2750	2751	2761	2762	2763
2764	2765	2766	2767	2768	2769	2770	2789	2790	2792	2793	2796	2797
2802	2803	2804	2805	2806	2807	2808	2809	2812	2813	2822	2823	2829
2830	2831	2848	2849	2850	2851	2856	2857	2861	2862	2863	2864	2865
2866	2867	2868	2870	2871	2883	2884	2886	2887	2888	2906	2907	2908
2909	2910	2914	2915	2916	2917	2918	2919	2924	2925	2928	2929	2930
2931	2932	2933	2934	2936	2937	2940	2941	2942	2943	2944	2946	2947
2948	2953	2954	2955	2956	2967	2968	2969	2971	2972	2973	2974	2975
2976	2982	2983	2989	2990	2994	2995	2996	3008	3009	3011	3012	3013
3035	3036	3042	3043	3045	3046	3047	3048	3049	3052	3053	3055	3056
3057	3058	3059	3062	3063	3066	3067	3068	3074	3075	3082	3083	3084
3085	3086	3094	3095	3101	3102	3105	3106	3107	3119	3120	3124	3125
3127	3128	3129	3130	3131	3133	3134	3135	3142	3143	3148	3149	3150
3174	3175	3176	3177	3178	3183	3184	3185	3186	3187	3192	3193	3194
3195	3196	3197	3199	3200	3201	3202	3203	3204	3205	3206	3207	3208
3210	3211	3213	3214	3216	3217	3219	3220	3221	3222	3224	3225	3226
3227	3228	3229	3231	3232	3233	3234	3237	3238	3239	3240	3244	3245
3248	3249	3250	3252	3253	3256	3257	3258	3259	3260	3262	3263	3276
3277	3278	3279	3281	3282	3284	3285	3286	3287	3288	3289	3290	3346
3347	3350	3351	3353	3354	3356	3357	3358	3359	3360	3361	3362	3363
3365	3366	3375	3376	3379	3380	3381	3382	3383	3386	3387	3388	3390
3391	3392	3396	3397	3409	3410	3412	3413	3415	3416	3423	3424	3425
3426	3427	3428	3429	3430	3440	3441	3444	3445	3446	3447	3448	3449
3471	3472	3473	3474	3475	3476	3478	3479	3480	3481	3483	3484	3486
3487	3496	3497	3501	3502	3503	3504	3505	3506	3507	3508	3510	3511
3513	3514	3515	3516	3517	3518	3519	3522	3523	3524	3526	3527	3528
3533	3534	3544	3545	3547	3548	3558	3559	3560	3561	3562	3564	3565
3570	3571	3573	3574	3581	3582	3584	3585	3586	3587	3589	3590	3591
3592	3593	3596	3597	3598	3600	3601	3602	3607	3608	3617	3618	3619
3620	3623	3624	3627	3628	3629	3630	3631	3632	3645	3646	3655	3656
3657	3658	3660	3661	3664	3665	3666	3667	3668	3669	3670	3671	3676
3677	3678	3679	3680	3681	3682	3683	3686	3687	3689	3690	3691	3692

	3693	3694	3695	3696	3698	3699	3706	3707	3708	3709	3710	3713	3714
	3715	3719	3720	3724	3725	3729	3730	3742	3743	3744	3746	3747	3752
	3753	3761	3762	3763	3764	3765	3766	3768	3769	3770	3771	3772	3773
	3774	3775	3780	3781	3782	3783	3784	3785	3786	3787	3790	3791	3792
	3793	3794	3795	3796	3797	3798	3799	3802	3803	3806	3807	3808	3309
	3810	3813	3814	3815	3819	3820	3825	3826	3847	3848	3849	3850	3851
	3856	3857	3858	3859	3880	3881	3883	3884	3885	3886	3889	3890	3891
	3892	3895	3896	3898	3899	3901	3902	3905	3906	3911	3912	3915	3916
	3918	3919	3925	3926	3927	3928	3929	3933	3934	3935	3942	3943	3944
	3945	3965	3966	3968	3969	3970	3972	3973	3975	3976	3977	3979	3980
	4000	4001	4003	4004	4005	4007	4008	4009	4012	4013	4014	4016	4017
	4018	4020	4021	4022	4023	4027	4028	4033	4034	4035	4037	4038	4064
	4065	4066	4067	4068	4073	4074	4077	4078	4082	4083	4085	4086	4087
	4088	4089	4090	4091	4092	4093	4094	4095	4096	4103	4104	4105	4175
	4176	4177	4178	4179	4180	4188	4189	4190	4191	4192	4193	4194	4196
	4197	4202	4203	4204	4206	4207	4211	4212	4213	4214	4215	4219	4220
	4221	4222	4223	4224	4225	4226	4227	4234	4235	4236	4237	4238	4241
	4242	4243	4251	4252	4253	4254	4272	4273	4274	4275	4276	4277	4278
	4280	4281	4282	4283	4284	4285	4286	4287	4288	4289	4290	4291	4292
	4293	4295	4296	4297	4298	4302	4303	4318	4319	4320	4321	4322	4323
	4324	4325	4326	4327	4328	4329	4330	4331	4332	4333	4334	4335	4336
	4337	4339	4340	4347	4348	4361	4362	4403	4404	4417	4418	4419	4420
	4421	4422	4423	4424	4425	4426	4428	4429	4432	4435	4436	4437	4438
	4439	4440	4441	4442	4443	4444	4445	4447	4448	4449	4450	4451	4452
	4453	4454	4457	4458	4459	4461	4462	4463	4464	4465	4466	4474	4475
$LSTST= 177777	10	1207	1208	1217	1219	1223	1231	1260	1262	1266	1272	1274	1278
	1288	1290	1294	1305	1307	1311	1324	1326	1330	1348	1350	1354	1364
	1366	1370	1381	1383	1387	1400	1402	1406	1430	1432	1436	1446	1448
	1452	1463	1465	1469	1482	1484	1488	1507	1509	1513	1521	1523	1527
	1537	1539	1543	1553	1555	1559	1574	1576	1580	1587	1589	1593	1603
	1605	1609	1620	1622	1626	1639	1641	1645	1663	1665	1669	1679	1681
	1685	1696	1698	1702	1715	1717	1721	1739	1741	1745	1755	1757	1761
	1772	1774	1778	1791	1793	1797	1815	1817	1821	1831	1833	1837	1848
	1850	1854	1867	1869	1873	1901	1903	1912	1934	1936	1945	1966	1968
	1979	1996	1998	2003	2011	2013	2026	2050	2052	2059	2075	2077	2083
	2095	2097	2103	2115	2117	2123	2141	2143	2148	2164	2166	2172	2184
	2186	2192	2204	2206	2212	2230	2232	2238	2254	2256	2262	2274	2276
	2282	2294	2296	2302	2319	2321	2326	2342	2344	2350	2362	2364	2370
	2382	2384	2390	2410	2412	2418	2443	2445	2449	2461	2463	2467	2470
	2472	2476	2487	2489	2493	2510	2512	2518	2544	2546	2550	2568	2570
	2574	2591	2593	2597	2614	2616	2621	2651	2653	2657	2673	2675	2679
	2697	2699	2703	2728	2729	2733	2749	2751	2755	2769	2770	2774	2812
	2813	2817	2829	2831	2835	2870	2871	2875	2886	2888	2892	2906	2908
	2912	2926	2927	2928	2930	2933	2934	2935	2938	2940	2946	2948	2958
	2966	2967	2969	2971	2973	2974	2976	2985	2987	2989	2990	2994	2996
	3001	3011	3013	3017	3066	3068	3077	3084	3086	3097	3105	3107	3115
	3133	3135	3145	3148	3150	3155	3174	3176	3180	3183	3185	3189	3199
	3201	3202	3203	3206	3223	3224	3226	3227	3229	3233	3234	3235	3241
	3244	3245	3248	3250	3252	3253	3254	3256	3260	3262	3263	3264	3271
	3273	3281	3282	3386	3388	3390	3392	3396	3397	3398	3403	3405	3428
	3430	3434	3522	3524	3526	3528	3533	3534	3535	3539	3541	3560	3562
	3567	3596	3598	3600	3602	3607	3608	3609	3613	3615	3664	3666	3667
	3668	3671	3713	3715	3722	3724	3725	3742	3744	3749	3768	3770	3771
	3772	3775	3813	3815	3822	3825	3826	3847	3849	3853	3923	3924	3925
	3933	3935	3939	3975	3977	3981	4012	4014	4016	4018	4024	4029	4033
	4035	4040	4064	4066	4070	4095	4096	4100	4103	4105	4109	4151	4185

		4186 4188 4190 4193 4194 4195 4198 4202 4204 4208 4211 4226 4227
		4234 4236 4239 4250 4258 4280 4282 4283 4284 4287 4292 4293 4300
		4307 4320 4322 4323 4324 4327 4328 4330 4331 4332 4335 4336 4337
		4339 4340 4345 4368 4401 4406 4415 4416 4417 4419 4420 4422 4425
		4426 4427 4434 4444 4445 4446 4455 4457 4472
$LSTTA*	000000	1# 1207 1208 1223 1224 1266 1267 1278 1279 1294 1295 1311 1312
		1330 1331 1354 1355 1370 1371 1387 1388 1406 1407 1436 1437 1452
		1453 1469 1470 1488 1489 1513 1514 1527 1528 1543 1544 1559 1560
		1580 1581 1593 1594 1609 1610 1626 1627 1645 1646 1669 1670 1685
		1686 1702 1703 1721 1722 1745 1746 1761 1762 1778 1779 1797 1798
		1821 1822 1837 1838 1854 1855 1873 1874 1912 1913 1945 1946 1979
		1980 2003 2004 2026 2027 2059 2060 2083 2084 2103 2104 2123 2124
		2148 2149 2172 2173 2192 2193 2212 2213 2238 2239 2262 2263 2282
		2283 2302 2303 2326 2327 2350 2351 2370 2371 2390 2391 2418 2419
		2449 2450 2467 2468 2476 2477 2493 2494 2518 2519 2550 2551 2574
		2575 2597 2598 2621 2622 2657 2658 2679 2680 2703 2704 2733 2734
		2755 2756 2774 2775 2817 2818 2835 2836 2875 2876 2892 2893 2912
		2913 2926 2927 2934 2935 2938 2939 2944 2945 2958 2959 2966 2967
		2985 2986 2987 2988 2990 2991 3001 3002 3017 3018 3077 3078 3097
		3098 3115 3116 3145 3146 3155 3156 3180 3181 3189 3190 3201 3202
		3203 3204 3223 3224 3234 3235 3241 3242 3245 3246 3253 3254 3263
		3264 3271 3272 3273 3274 3282 3283 3397 3398 3403 3404 3405 3406
		3434 3435 3534 3535 3539 3540 3541 3542 3567 3568 3608 3609 3613
		3614 3615 3616 3666 3667 3668 3669 3722 3723 3725 3726 3749 3750
		3770 3771 3772 3773 3822 3823 3826 3827 3853 3854 3923 3924 3929
		3930 3939 3940 3981 3982 4024 4025 4029 4030 4040 4041 4070 4071
		4100 4101 4109 4110 4150 4151 4185 4186 4194 4195 4198 4199 4208
		4209 4227 4228 4239 4240 4250 4251 4252 4253 4257 4258 4282 4283
		4284 4285 4293 4294 4300 4301 4302 4306 4307 4322 4323 4324 4325
		4330 4331 4332 4333 4337 4338 4340 4341 4345 4346 4347 4367 4368
		4401 4402 4403 4405 4406 4415 4416 4426 4427 4434 4435 4445 4446
		4455 4456 4472 4473 4474
$MADR1	001226	1045#
$MADR2	001232	1049#
$MADR3	001236	1052#
$MADR4	001242	1055#
$MAIL	001174	965 969 1018# 1127 1142 1194 1257 1341 1418 1505 1572 1657 1733
		1809 1892 1925 1958 1992 2047 2138 2227 2316 2407 2507 2611 2693
		2785 2844 2902 3032 3172 3343 3467 3642 3740 3842 4062 4584 4867
		4945
$MAMS1	001224	1039#
$MAMS2	001230	1047#
$MAMS3	001234	1050#
$MAMS4	001240	1053#
$MBADR	001002	965#
$MCALL*	000000	1#
$MFLG	014170	4783* 4789 4824* 4828#
$MNEW	013715	4665 4777#
$MSGAD	001210	1025# 4799* 4802
$MSGLG	001212	1026# 4804*
$MSGTY	001174	1019# 4797 4805* 4817 4821*
$MSWR	013704	4662 4775#
$MTYP1	001225	1040#
$MTYP2	001231	1048#
$MTYP3	001235	1051#
$MTYP4	001241	1054#

MAINDEC-ZZ-CVDVA-D MACY11 30A(1052) 12-JUL-84 09:52 PAGE 142
CVDVAD.P11 12-JUL-84 05:04 CROSS REFERENCE TABLE -- USER SYMBOLS

```
$MXCNT  014652          4943    4953#
$NESTL= 177777            1#    1207#  1217#  1223#  1231#  1260#  1266#  1272#  1278#  1288#  1294#  1305#  1311#
                        1324#  1330#  1348#  1354#  1364#  1370#  1381#  1387#  1400#  1406#  1430#  1436#  1446#
                        1452#  1463#  1469#  1482#  1488#  1507#  1513#  1521#  1527#  1537#  1543#  1553#  1559#
                        1574#  1580#  1587#  1593#  1603#  1609#  1620#  1626#  1639#  1645#  1663#  1669#  1679#
                        1685#  1696#  1702#  1715#  1721#  1739#  1745#  1755#  1761#  1772#  1778#  1791#  1797#
                        1815#  1821#  1831#  1837#  1848#  1854#  1867#  1873#  1901#  1912#  1934#  1945#  1966#
                        1979#  1996#  2003#  2011#  2026#  2050#  2059#  2075#  2083#  2095#  2103#  2115#  2123#
                        2141#  2148#  2164#  2172#  2184#  2192#  2204#  2212#  2230#  2238#  2254#  2262#  2274#
                        2282#  2294#  2302#  2319#  2326#  2342#  2350#  2362#  2370#  2382#  2390#  2410#  2418#
                        2443#  2449#  2461#  2467#  2470#  2476#  2487#  2493#  2510#  2518#  2544#  2550#  2568#
                        2574#  2591#  2597#  2614#  2621#  2651#  2657#  2673#  2679#  2697#  2703#  2728#  2733#
                        2749#  2755#  2769#  2774#  2812#  2817#  2829#  2835#  2870#  2875#  2886#  2892#  2906#
                        2912#  2926#  2938#  2933    2938#  2940#  2946#  2958#  2966#  2971#  2974#  2985#  2987#
                        2989#  2994#  3001#  3011#  3017#  3066#  3077#  3084#  3097#  3105#  3115#  3133#  3145#
                        3148#  3155#  3174#  3180#  3183#  3189#  3199#  3223#  3227#  3233    3241#  3244#  3248#
                        3252    3256#  3262    3271#  3273#  3281#  3386#  3390#  3396    3403#  3405#  3428#  3434#
                        3522#  3526#  3533    3539#  3541#  3560#  3567#  3596#  3615    3664#  3724    3742#  3749
                        3768#  3825    3847#  3853    3923#  3925    3933#  3939    3975#  3981    4012#  4029    4033#
                        4040    4064#  4070    4095#  4100    4103#  4109    4151#  4250    4258#  4300    4307#  4345
                        4368#  4401    4406#  4415#  4417#  4420#  4425    4434#  4444    4455#  4457#  4472#
$NSK0 = 000300          1207#  1231    1260#  1266    1272#  1278    1288#  1294    1305#  1311    1324#  1330    1348#
                        1354    1364#  1370    1381#  1387    1400#  1406    1430#  1436    1446#  1452    1463#  1469
                        1482#  1488    1507#  1513    1521#  1527    1537#  1543    1553#  1559    1574#  1580    1587#
                        1593    1603#  1609    1620#  1626    1639#  1645    1663#  1669    1679#  1685    1696#  1702
                        1715#  1721    1739#  1745    1755#  1761    1772#  1778    1791#  1797    1815#  1821    1831#
                        1837    1848#  1854    1867#  1873    1901#  1912    1934#  1945    1966#  1979    1996#  2003
                        2011#  2026    2050#  2059    2075#  2083    2095#  2103    2115#  2123    2141#  2148    2164#
                        2172    2184#  2192    2204#  2212    2230#  2238    2254#  2262    2274#  2282    2294#  2302
                        2319#  2326    2342#  2350    2362#  2370    2382#  2390    2410#  2418    2443#  2449    2461#
                        2467    2470#  2476    2487#  2493    2510#  2518    2544#  2550    2568#  2574    2591#  2597
                        2614#  2621    2651#  2657    2673#  2679    2697#  2703    2728#  2733    2749#  2755    2769#
                        2774    2812#  2817    2829#  2835    2870#  2875    2886#  2892    2906#  2912    2926#  2940
                        2946#  2958    2966#  2989    2994#  3001    3011#  3017    3066#  3077    3084#  3097    3105#
                        3115    3133#  3145    3148#  3155    3174#  3180    3183#  3189    3199#  3281    3386#  3405
                        3428#  3434    3522#  3541    3560#  3567    3596#  3615    3664#  3724    3742#  3749    3768#
                        3825    3847#  3853    3923#  3925    3933#  3939    3975#  3981    4012#  4029    4033#  4040
                        4064#  4070    4095#  4100    4103#  4109    4151#  4250    4258#  4300    4307#  4345    4368#
                        4401    4406#  4472
$NSK1 = 000130          1217#  1223    2928#  2933    2938    2971#  2987    3223#  3244    3248#  3252    3273    3390#
                        3396    3403    3526#  3533    3539    3600#  3607    3613    3713#  3722    3813#  3822    4016#
                        4024    4185#  4211    4226    4234#  4239    4280#  4292    4320#  4339    4415#  4457
$NSK2 = 000110          2974#  2985    3227#  3233    3241    3256#  3262    3271    4188#  4193    4198    4202#  4208
                        4328#  4336    4417#  4444    4455
$NSK3 = 000110          4420#  4425    4434
$NULL   001154          1003#  4611    4640
$NWTST= 000001          1186#  1188    1250#  1252    1336#  1413#  1495#  1497    1566#  1568    1652#  1728#  1804#
                        1887#  1920#  1953#  1987#  2040#  2042    2131#  2133    2220#  2222    2310#  2312    2398#
                        2400    2500#  2502    2606#  2686#  2688    2778#  2780    2839#  2895#  2897    3027#  3164#
                        3166    3333#  3335    3459#  3461    3636#  3638    3735#  3836#  3838    4055#  4057    4121#
$OCNT   015322          5053*  5082*  5095#
$OMODE  015324          5048*  5052*  5057    5060*  5071*  5097#
$OVER   014636          4907    4923    4931    4941    4950#
$PASS   001202          1022#  1127*  4490*  4491*  4499    4512    4937    4954
```

```
$PASTM  001006          9670
$POWER  012660        4551    4558#
$PWRAD  012646        4553#
$PWRDN  012506        1103    4520#   4548
$PWRMG  012642        4551#
$PWRUP  012560        4530    4536#
$QUES   001170        1010#   4640    4708    4757    4773    4890
$RDCHR  013434        4721#   5138
$RDDEC- ****** U      5140
$RDLIN  013554        4749#   5139
$RDOCT- ****** U      5140
$RDSZ - 000010        4742#
$RTNAD  012464        4511#
$R2A  - ****** U      5140
$SAVLE- 177777           1#   2989#   2990#   3202#   3206#   3244#   3245#   3667#   3671#   3771#   3775#   4226#   4227#
                      4283#   4287#   4323#   4327#   4331#   4335#
$SAVRE- ****** U      5140
$SAVR6  012656        4529#   4537    4538#   4539#   4557#
$SCOPE  014374        1097    4904#
$SETUP- 000137        1088#   1096    1097    1099    1101    1103    1105    1106    1107    1109    1136    1139    4488
                      4645    4779    4851    4877    4885    4905
$SSKO - 000236        2989#   2990    3202#   3206    3244#   3245    3667#   3671    3771#   3775    4226#   4227    4283#
                      4287    4323#   4327    4331#   4335
$STUP - 177777        1088#
$SVLAD  014602        4915    4944#
$SVPC - 000204         941#    946
$SWR  - 167400           1#    689     693     694     695     696     697     698     699    1007    1008    1009    1106
                      1107    1109    1110    1193    1256    1340    1417    1504    1571    1656    1732    1808    1891
                      1924    1957    1991    2046    2137    2226    2315    2406    2506    2610    2692    2784    2843
                      2901    3031    3171    3342    3466    3641    3739    3841    4061    4125    4483    4489    4504
                      4510    4512    4554    4842    4843    4844    4845    4846    4855    4862    4874    4878    4890
                      4896    4897    4898    4899    4900    4906    4918    4920    4921    4924    4925    4926    4933
                      4934    4935    4947    4950    4953
$SWREG  001216        1030#   1130
$SWRMK- 000000         699     700    4900    4901    4922
$TAGLE- 177777           1#   1208#   1219#   1223#   1231#   1262#   1266#   1274#   1278#   1290#   1294#   1307#   1311#
                      1326#   1330#   1350#   1354#   1366#   1370#   1383#   1387#   1402#   1406#   1432#   1436#   1448#
                      1452#   1465#   1469#   1484#   1488#   1509#   1513#   1523#   1527#   1539#   1543#   1555#   1559#
                      1576#   1580#   1589#   1593#   1605#   1609#   1622#   1626#   1641#   1645#   1665#   1669#   1681#
                      1685#   1698#   1702#   1717#   1721#   1741#   1745#   1757#   1761#   1774#   1778#   1793#   1797#
                      1817#   1821#   1833#   1837#   1850#   1854#   1869#   1873#   1903#   1912#   1936#   1945#   1968#
                      1979#   1998#   2003#   2013#   2026#   2052#   2059#   2077#   2083#   2097#   2103#   2117#   2123#
                      2143#   2148#   2166#   2172#   2186#   2192#   2206#   2212#   2232#   2238#   2256#   2262#   2276#
                      2282#   2296#   2302#   2321#   2326#   2344#   2350#   2364#   2370#   2384#   2390#   2412#   2418#
                      2445#   2449#   2463#   2467#   2472#   2476#   2489#   2493#   2512#   2518#   2546#   2550#   2570#
                      2574#   2593#   2597#   2616#   2621#   2653#   2657#   2675#   2679#   2699#   2703#   2729#   2733#
                      2751#   2755#   2770#   2774#   2813#   2817#   2831#   2835#   2871#   2875#   2888#   2892#   2908#
                      2912#   2927#   2930#   2934#   2935#   2938#   2940#   2948#   2958#   2967#   2969#   2973#   2976#
                      2985#   2987#   2989#   2996#   3001#   3013#   3017#   3068#   3077#   3086#   3097#   3107#   3115#
                      3135#   3145#   3150#   3155#   3176#   3180#   3185#   3189#   3201#   3203#   3206#   3224#   3226#
                      3229#   3234#   3235#   3241#   3244#   3250#   3253#   3254#   3260#   3263#   3264#   3271#   3273#
                      3281#   3282#   3288#   3292#   3297#   3298#   3403#   3405#   3430#   3434#   3524#   3528#   3534#
                      3535#   3539#   3541#   3562#   3567#   3598#   3602#   3608#   3609#   3613#   3615#   3666#   3668#
                      3671#   3715#   3722#   3724#   3725#   3744#   3749#   3770#   3772#   3775#   3815#   3822#   3825#
                      3826#   3849#   3853#   3924#   3925#   3935#   3939#   3977#   3981#   4014#   4018#   4024#   4029#
                      4035#   4040#   4066#   4070#   4096#   4100#   4105#   4109#   4186#   4190#   4194#   4195#   4198#
```

	4204#	4208#	4211	4226#	4236#	4239#	4282#	4284#	4287#	4292#	4293#	4322#	4324#
	4327#	4330#	4332#	4335#	4336#	4337#	4339#	4340#	4416#	4419#	4422#	4426#	4427#
	4434#	4445#	4446#	4455#	4457#								
$TAGNU= 000251	1#	1207	1208#	1218	1219#	1261	1262#	1273	1274#	1289	1290#	1306	1307#
	1325	1326#	1349	1350#	1365	1366#	1382	1383#	1401	1402#	1431	1432#	1447
	1448#	1464	1465#	1483	1484#	1508	1509#	1522	1523#	1538	1539#	1554	1555#
	1575	1576#	1588	1589#	1604	1605#	1621	1622#	1640	1641#	1664	1665#	1680
	1681#	1697	1698#	1716	1717#	1740	1741#	1756	1757#	1773	1774#	1792	1793#
	1816	1817#	1832	1833#	1849	1850#	1868	1869#	1902	1903#	1935	1936#	1967
	1968#	1997	1998#	2012	2013#	2051	2052#	2076	2077#	2096	2097#	2116	2117#
	2142	2143#	2165	2166#	2185	2186#	2205	2206#	2231	2232#	2255	2256#	2275
	2276#	2295	2296#	2320	2321#	2343	2344#	2363	2364#	2383	2384#	2411	2412#
	2444	2445#	2462	2463#	2471	2472#	2488	2489#	2511	2512#	2545	2546#	2569
	2570#	2592	2593#	2615	2616#	2652	2653#	2674	2675#	2698	2699#	2728	2729#
	2750	2751#	2769	2770#	2812	2813#	2830	2831#	2870	2871#	2887	2888#	2907
	2908#	2926	2927#	2929	2930#	2933	2935#	2941	2944	2945#	2947	2948#	2966
	2967#	2968	2969#	2972	2973#	2975	2976#	2995	2996#	3012	3013#	3067	3066#
	3085	3086#	3106	3107#	3134	3135#	3149	3150#	3175	3176#	3184	3185#	3200
	3201#	3202#	3205	3206#	3223	3224#	3225	3226#	3228	3229#	3233	3235#	3249
	3250#	3252	3254#	3257	3259	3260#	3262	3264#	3387	3388#	3391	3392#	3396
	3398#	3429	3430#	3523	3524#	3527	3528#	3533	3535#	3561	3562#	3597	3598#
	3601	3602#	3607	3609#	3665	3666#	3667#	3670	3671#	3714	3715#	3743	3744#
	3769	3770#	3771#	3774	3775#	3814	3815#	3848	3849#	3923	3924#	3926	3929
	3930#	3934	3935#	3976	3977#	4013	4014#	4017	4018#	4034	4035#	4065	4066#
	4095	4096#	4104	4105#	4151#	4185	4186#	4189	4190#	4193	4195#	4203	4204#
	4235	4236#	4258#	4281	4282#	4283#	4286	4287#	4307#	4321	4322#	4323#	4326
	4327#	4329	4330#	4331#	4334	4335#	4368#	4406#	4415	4416#	4418	4419#	4421
	4422#	4425	4427#	4444	4446#								
$TEMP = 000300	1154#	1155#	1161#	1162#	1164#	1165#	1166#	1167#	1169#	1170#	1171#	1173#	1174#
	1176#	1177#	1179#	1180#	1182#	1183#	1184#	1196#	1197#	1199#	1200#	1201#	1202#
	1203#	1204#	1205#	1206#	1209#	1210#	1212#	1213#	1223#	1226#	1227#	1228#	1230#
	1231#	1232	1236#	1237#	1239#	1240#	1241#	1263#	1264#	1266#	1269#	1270#	1278#
	1283#	1284#	1285#	1286#	1294#	1299#	1300#	1302#	1303#	1311#	1316#	1317#	1319#
	1320#	1330#	1345#	1346#	1354#	1359#	1360#	1361#	1362#	1370#	1375#	1376#	1378#
	1379#	1387#	1392#	1393#	1395#	1396#	1406#	1427#	1428#	1436#	1441#	1442#	1443#
	1444#	1452#	1457#	1458#	1460#	1461#	1469#	1474#	1475#	1477#	1478#	1488#	1510#
	1511#	1513#	1516#	1517#	1518#	1519#	1527#	1532#	1533#	1535#	1536#	1543#	1548#
	1549#	1550#	1551#	1559#	1577#	1578#	1580#	1584#	1585#	1593#	1598#	1599#	1600#
	1601#	1609#	1614#	1615#	1617#	1618#	1626#	1631#	1632#	1634#	1635#	1645#	1660#
	1661#	1669#	1674#	1675#	1676#	1677#	1685#	1690#	1691#	1693#	1694#	1702#	1706#
	1707#	1710#	1711#	1721#	1736#	1737#	1745#	1750#	1751#	1752#	1753#	1761#	1766#
	1767#	1769#	1770#	1778#	1783#	1784#	1786#	1787#	1797#	1812#	1813#	1821#	1826#
	1827#	1828#	1829#	1837#	1842#	1843#	1845#	1846#	1854#	1859#	1860#	1862#	1863#
	1873#	1899#	1900#	1912#	1931#	1932#	1945#	1964#	1965#	1979#	2000#	2001#	2003#
	2008#	2009#	2026#	2056#	2057#	2059#	2068#	2069#	2072#	2073#	2083#	2088#	2089#
	2092#	2093#	2103#	2108#	2109#	2112#	2113#	2123#	2145#	2146#	2148#	2157#	2158#
	2161#	2162#	2172#	2177#	2178#	2181#	2182#	2192#	2197#	2198#	2201#	2202#	2212#
	2235#	2236#	2238#	2247#	2248#	2251#	2252#	2262#	2267#	2268#	2271#	2272#	2282#
	2287#	2288#	2291#	2292#	2302#	2323#	2324#	2326#	2335#	2336#	2339#	2340#	2350#
	2355#	2356#	2359#	2360#	2370#	2375#	2376#	2379#	2380#	2390#	2415#	2416#	2418#
	2428#	2429#	2431#	2432#	2440#	2441#	2449#	2455#	2456#	2459#	2460#	2467#	2476#
	2481#	2482#	2485#	2486#	2493#	2515#	2516#	2518#	2528#	2529#	2532#	2533#	2541#
	2542#	2550#	2555#	2556#	2560#	2561#	2574#	2579#	2580#	2583#	2584#	2597#	2618#
	2619#	2621#	2633#	2634#	2635#	2636#	2639#	2640#	2643#	2644#	2657#	2662#	2663#
	2665#	2666#	2679#	2700#	2701#	2703#	2705#	2706#	2713#	2714#	2733#	2737#	2738#
	2744#	2745#	2755#	2774#	2789#	2790#	2792#	2793#	2796#	2797#	2817#	2822#	2823#

		2835# 2848# 2849# 2850# 2851# 2856# 2857# 2875# 2883# 2884# 2892# 2909# 2910#
		2912# 2914# 2915# 2916# 2917# 2918# 2919# 2924# 2925# 2931# 2932# 2933# 2934#
		2936# 2937# 2938# 2940# 2943 2953# 2954# 2955# 2956# 2958# 2982# 2983# 2985#
		2987# 2989# 2990# 3001# 3008# 3009# 3017# 3035# 3036# 3042# 3043# 3052# 3053#
		3062# 3063# 3074# 3075# 3077# 3082# 3083# 3094# 3095# 3097# 3101# 3102# 3115#
		3119# 3120# 3124# 3125# 3142# 3143# 3145# 3155# 3177# 3178# 3180# 3186# 3187#
		3189# 3192# 3193# 3194# 3195# 3196# 3197# 3203# 3206# 3207# 3208# 3210# 3211#
		3213# 3214# 3216# 3217# 3219# 3220# 3221# 3222# 3231# 3232# 3233# 3234# 3237#
		3238# 3239# 3240# 3241# 3244# 3245# 3252# 3253# 3262# 3263# 3271# 3273# 3276#
		3277# 3278# 3279# 3281# 3282# 3284# 3288# 3289# 3290# 3346# 3347# 3350# 3351#
		3353# 3354# 3356# 3357# 3358# 3359# 3360# 3361# 3362# 3363# 3365# 3366# 3375#
		3376# 3396# 3397# 3403# 3405# 3409# 3410# 3412# 3413# 3415# 3416# 3434# 3440#
		3441# 3444# 3445# 3447# 3448# 3449# 3471# 3472# 3473# 3474# 3475# 3476# 3478#
		3479# 3480# 3481# 3483# 3484# 3486# 3487# 3496# 3497# 3510# 3511# 3518# 3519#
		3533# 3534# 3539# 3541# 3544# 3545# 3547# 3548# 3558# 3559# 3564# 3565# 3567#
		3570# 3571# 3573# 3574# 3581# 3582# 3584# 3585# 3586# 3587# 3607# 3608# 3613#
		3615# 3617# 3618# 3619# 3620# 3623# 3624# 3627# 3628# 3630# 3631# 3632# 3645#
		3646# 3660# 3661# 3668# 3671# 3686# 3687# 3698# 3699# 3706# 3708# 3709# 3710#
		3719# 3720# 3722# 3724# 3725# 3729# 3730# 3746# 3747# 3749# 3752# 3753# 3765#
		3766# 3772# 3775# 3790# 3791# 3802# 3803# 3806# 3808# 3809# 3810# 3819# 3820#
		3822# 3825# 3826# 3850# 3851# 3853# 3880# 3881# 3883# 3884# 3885# 3886# 3889#
		3890# 3891# 3892# 3895# 3896# 3898# 3899# 3901# 3902# 3905# 3906# 3911# 3912#
		3915# 3916# 3918# 3919# 3925# 3928 3939# 3942# 3943# 3944# 3945# 3965# 3966#
		3968# 3970# 3972# 3973# 3979# 3980# 3981# 4000# 4001# 4003# 4005# 4007# 4009#
		4020# 4021# 4022# 4023# 4024# 4027# 4028# 4029# 4037# 4038# 4040# 4067# 4068#
		4070# 4073# 4074# 4077# 4078# 4082# 4083# 4085# 4086# 4100# 4109# 4175# 4176#
		4177# 4178# 4179# 4180# 4191# 4192# 4193# 4194# 4196# 4197# 4198# 4206# 4207#
		4208# 4211# 4212 4214 4224# 4225# 4226# 4227# 4239# 4250# 4272# 4273# 4274#
		4278# 4284# 4287# 4288# 4289# 4290# 4291# 4292# 4293# 4295# 4296# 4300# 4318#
		4319# 4324# 4327# 4332# 4335# 4336# 4337# 4339# 4340# 4345# 4361# 4362# 4401#
		4423# 4424# 4425# 4426# 4432# 4433# 4434# 4436# 4437# 4438# 4439# 4440# 4441#
		4442# 4443# 4444# 4445# 4447# 4448# 4449# 4450# 4451# 4452# 4453# 4454# 4455#
		4457# 4458 4461# 4462# 4463# 4464# 4472#
$TESTN	001200	1021# 1194* 1257* 1341* 1418* 1505* 1572* 1657* 1733* 1809* 1892* 1925* 1958*
		1992* 2047* 2138* 2227* 2316* 2407* 2507* 2611* 2693* 2785* 2844* 2902* 3032*
		3172* 3343* 3467* 3642* 3740* 3842* 4062* 4373 4945*
$TIMES	001160	1007# 1106* 1193* 1256* 1263* 1340* 1417* 1504* 1510* 1571* 1577* 1656* 1732*
		1808* 1891* 1924* 1957* 1991* 2000* 2046* 2056* 2137* 2145* 2226* 2235* 2315*
		2323* 2406* 2415* 2506* 2515* 2610* 2618* 2692* 2700* 2784* 2843* 2901* 2909*
		2955* 2982* 3031* 3074* 3094* 3142* 3171* 3177* 3186* 3342* 3466* 3564* 3641*
		3719* 3739* 3746* 3819* 3841* 3850* 4061* 4067* 4125* 4489* 4933* 4940 4943*
		4953
$TKB	001146	1000# 4643 4654 4671 4725 4731
$TKS	001144	999# 4643 4652 4668 4692* 4723 4729
$TN	000042	1# 689 1186 1193# 1194 1250 1256# 1257 1264 1336 1340# 1341 1413
		1417# 1418 1495 1504# 1505 1511 1566 1571# 1572 1578 1652 1656# 1657
		1728 1732# 1733 1804 1808# 1809 1887 1891# 1892 1920 1924# 1925 1953
		1957# 1958 1987 1991# 1992 2001 2040 2046# 2047 2057 2131 2137# 2138
		2146 2220 2226# 2227 2236 2310 2315# 2316 2324 2398 2406# 2407 2416
		2500 2506# 2507 2516 2600 2606 2610# 2611 2619 2686 2692# 2693 2701
		2778 2784# 2785 2839 2843# 2844 2895 2901# 2902 2910 2956 2983 3020
		3027 3031# 3032 3075 3095 3113 3143 3158 3164 3171# 3172 3178 3187
		3292 3333 3342# 3343 3459 3466# 3467 3565 3636 3641# 3642 3720 3735
		3739# 3740 3747 3820 3836 3841# 3842 3851 3947 4055 4061# 4062 4068
		4113 4121 4125#
$TPB	001152	1002# 4629* 4640

```
$IPFLG  001157          10060#  4578    4640
$IPS    001150          10010#  4627    4640
$TRAP   015326          1101    5106#
$TRAP2  015350          5117#   5128
$TRP  - 000012          5121#   5130#   5131#   5132#   5133#   5134#   5135    5136#   5137    5138#   5139#   5140#
$TRPAD  015362          5111    5128#
$TSK0 - 000244          1208#   1231    1262#   1266    1274#   1278    1290#   1294    1307#   1311    1326#   1330    1350#
                        1354    1366#   1370    1383#   1387    1402#   1406    1432#   1436    1448#   1452    1465#   1469
                        1484#   1488    1509#   1513    1523#   1527    1539#   1543    1555#   1559    1576#   1580    1589#
                        1593    1605#   1609    1622#   1626    1641#   1645    1665#   1669    1681#   1685    1698#   1702
                        1717#   1721    1741#   1745    1757#   1761    1774#   1778    1793#   1797    1817#   1821    1833#
                        1837    1850#   1854    1869#   1873    1903#   1912    1936#   1945    1968#   1979    1998#   2003
                        2013#   2026    2052#   2059    2077#   2083    2097#   2103    2117#   2123    2143#   2148    2166#
                        2172    2186#   2192    2206#   2212    2232#   2238    2256#   2262    2276#   2282    2296#   2302
                        2321#   2326    2344#   2350    2364#   2370    2384#   2390    2412#   2418    2445#   2449    2463#
                        2467    2472#   2476    2489#   2493    2512#   2518    2546#   2550    2570#   2574    2593#   2597
                        2616#   2621    2653#   2657    2675#   2679    2699#   2703    2729#   2733    2751#   2755    2770#
                        2774    2813#   2817    2831#   2835    2871#   2875    2888#   2892    2908#   2912    2927#   2940
                        2948#   2958    2967#   2989    2996#   3001    3013#   3017    3068#   3077    3086#   3097    3107#
                        3115    3135#   3145    3150#   3155    3176#   3180    3185#   3189    3201#   3203    3206#   3284
                        3388#   3405    3430#   3434    3524#   3541    3562#   3567    3598#   3615    3666#   3668    3671#
                        3725    3744#   3749    3770#   3772    3775#   3826    3849#   3853    3924#   3925    3935#   3939
                        3977#   3981    4014#   4029    4035#   4040    4066#   4070    4096#   4100    4105#   4109    4186#
                        4226    4236#   4239    4282#   4284    4287#   4293    4322#   4324    4327#   4340    4416#   4457
$TSK1 - 000250          1219#   1223    2930#   2934    2935#   2938    2969#   2989    3206#   3281    3392#   3397    3398#
                        3403    3528#   3534    3535#   3539    3602#   3608    3609#   3613    3671#   3724    3775#   3825
                        4018#   4024    4186#   4211    4226    4287#   4292    4327#   4339    4419#   4445    4446#   4455
$TSK2 - 000247          2973#   2987    3224#   3244    3250#   3253    3254#   3273    3715#   3722    3815#   3822    4190#
                        4194    4195#   4198    4204#   4208    4330#   4332    4335#   4337    4422#   4426    4427#   4434
$TSK3 - 000236          2976#   2985    3226#   3244    3260#   3263    3264#   3271    4335#   4336
$TSK4 - 000147          3229#   3234    3235#   3241
$TSTM   001004           966#
$TSTNM  001102           979#   4488*   4854    4890    4895    4922    4944*   4945    4950    4954
$TTYIN  013662          4750    4751    4768    4772#
$TYPBN- ****** U        5134
$TYPDS  014654          4966#   5133
$TYPE   012670          4578#   4810    5121    5139
$TYPEC  013102          4608    4615    4622    4627#   4628    4694
$TYPEX  013150          4633    4635    4638#
$TYPOC  015124          5051#   5130
$TYPON  015140          5050    5053#   5132
$TYPOS  015100          5046#   5131
$UNIT   001206          1024#
$UNITM  001010           968#
$USWR   001220          1031#   1260    1507    1574    1996    2050    2141    2230    2319    2410    2510    2614    3174
                        3284    3560    3742    4064    4274
$VECT1  001244          1056#   4442
$VECT2  001246          1057#
$XTSTR  014406          4909#
$YESNO- 000001          1172#   1173#   1175#   1176#   1178#   1179#   1181#   1182#   1226#   1227#   1229#   1230#   1238#
                        1239#   1285#   1286#   1302#   1303#   1319#   1320#   1361#   1362#   1378#   1379#   1395#   1396#
                        1443#   1444#   1460#   1461#   1477#   1478#   1518#   1519#   1535#   1536#   1550#   1551#   1600#
                        1601#   1617#   1618#   1634#   1635#   1676#   1677#   1693#   1694#   1710#   1711#   1752#   1753#
                        1769#   1770#   1786#   1787#   1828#   1829#   1845#   1846#   1862#   1863#   2072#   2073#   2092#
                        2093#   2112#   2113#   2161#   2162#   2181#   2182#   2201#   2202#   2251#   2252#   2271#   2272#
                        2291#   2292#   2339#   2340#   2359#   2360#   2379#   2380#   2431#   2432#   2459#   2460#   2485#
```

```
                24860  25320  25330  25600  25610  25830  25840  26330  26340  26430  26440  26650  26660
                27890  27900  28480  28490  29140  29150  29360  29370  31960  31970  32020  32030  32370
                32380  32390  32400  32850  32880  33530  33540  33650  33660  33750  33760  34150  34160
                34460  34470  34850  34840  34860  34870  35100  35110  35440  35450  35470  35480  35730
                35740  35810  35820  35840  35850  35860  35870  36170  36180  36190  36200  36290  36300
                36450  36460  36670  36680  37070  37080  37090  37100  37290  37300  37520  37530  37710
                37720  38070  38080  38090  38100  39110  39120  39150  39160  39180  39190  39420  39430
                39440  39450  39650  39660  39690  39700  39790  39800  40000  40010  40040  40050  40080
                40090  40270  40280  40370  40380  40730  40740  40820  40830  42240  42250  42750  42780
                42830  42840  42880  42890  42900  42910  42950  42960  43230  43240  43310  43320  43610
                43620  44490  44500  44510  44520  44530  44540  44630  44640
??ARGC= 000000  41510  42580  43070  43680  44060
??BYTE= 000403  12170  12600  12720  12880  13050  13240  13480  13640  13810  14000  14300  14460  14630
                14820  15070  15210  15370  15530  15740  15870  16030  16200  16390  16630  16790  16960
                17150  17390  17550  17720  17910  18150  18310  18480  18670  19010  19340  19660  19960
                20110  20500  20750  20950  21150  21410  21640  21840  22040  22300  22540  22740  22940
                23190  23420  23620  23820  24100  24430  24610  24700  24870  25100  25440  25680  25910
                26140  26510  26730  26970  27490  28290  28860  29060  29280  29460  29670  29710  29740
                29940  30110  30660  30840  31050  31330  31480  31740  31830  32240  32270  32480  32560
                32580  33860  33900  34280  35220  35260  35600  35960  36000  37130  37420  38130  38470
                39330  39750  40120  40160  40330  40640  41030  41880  42020  42340  44170  44200
??DST = 000067  32850  42750
??FLAG= 000001  12170  1219   12230  12600  1262   12660  12720  1274   12780  12880  1290   12940  13050
                1307   13110  13240  1326   13300  13480  1350   13540  13640  1366   13700  13810  1383
                13870  14000  1402   14060  14300  1432   14360  14460  1448   14520  14630  1465   14690
                14820  1484   14880  15070  1509   15130  15210  1523   15270  15370  1539   15430  15530
                1555   15590  15740  1576   15800  15870  1589   15930  16030  1605   16090  16200  1622
                16260  16390  1641   16450  16630  1665   16690  16790  1681   16850  16960  1698   17020
                17150  1717   17210  17390  1741   17450  17550  1757   17610  17720  1774   17780  17910
                1793   17970  18150  1817   18210  18310  1833   18370  18480  1850   18540  18670  1869
                18730  19010  1903   19120  19340  1936   19450  19660  1968   19790  19960  1998   20030
                20110  2013   20260  20500  2052   20590  20750  2077   20830  20950  2097   21030  21150
                2117   21230  21410  2143   21480  21640  2166   21720  21840  2186   21920  22040  2206
                22120  22300  2232   22380  22540  2256   22620  22740  2276   22820  22940  2296   23020
                23190  2321   23260  23420  2344   23500  23620  2364   23700  23820  2384   23900  24100
                2412   24184  24430  2445   24490  24610  2463   24670  24700  2472   24760  24870  2489
                24930  25100  2512   25180  25440  2546   25500  25680  2570   25740  25910  2593   25970
                26140  2616   26210  26510  2653   26570  26730  2675   26790  26970  2699   27030  27280
                27330  27490  2751   27550  27690  27740  28120  28170  28290  2831   28350  28700  28750
                28860  2888   28920  29060  2908   29120  29280  293C   29380  29460  2948   29580  29660
                29670  2969   29710  2973   29740  2976   29850  29870  29940  2996   30010  30110  3013
                30170  30660  3068   30770  30840  3086   30970  31050  3107   31150  31330  3135   31450
                31480  3150   31550  31740  3176   31800  31830  3185   31890  32230  32240  3226   32270
                3229   32410  32480  3250   32560  32580  3260   32710  32730  33860  3388   33900  3392
                34030  34050  34280  3430   34340  35220  3524   35260  3528   35390  35410  35600  3562
                35670  35960  3598   36000  3602   36130  36150  37130  3715   37220  37420  3744   37490
                38130  3815   38220  38470  3849   38530  39330  3935   39390  39750  3977   39810  40120
                4014   40160  4018   40240  40290  40330  4035   40400  40640  4066   40700  40950  41000
                41030  4105   41090  41880  4190   41980  42020  4204   42080  42340  4236   42390  44170
                4419   44200  4422   44340  44550
??FROM= 000000  11570  24340  25350  25630  25860  26460  26680  27180  27610  28020  28610  30450  30550
                31270  33790  34230  35010  35130  35890  36550  36760  36890  37610  37800  37920  38560
                40870  42190  44280
??GET4= 000000  45040
??LOC = 012336  12180  1219   12320  1233   12610  1262   12730  1274   12890  1290   13060  1307   13250
                1326   13490  1350   13650  1366   13820  1383   14010  1402   14310  1432   14470  1448
```

```
                    14640  1465   14830  1484   15080  1509   15220  1525   15380  1539   15540  1555   15750
                    1576   15880  1589   16040  1605   16210  1622   16400  1641   16640  1665   16800  1681
                    16970  1698   17160  1717   17400  1741   17560  1757   17730  1774   17920  1793   18160
                    1817   18320  1833   18490  1850   18680  1869   19020  1903   19350  1936   19670  1968
                    19970  1998   20120  2013   20510  2052   20760  2077   20960  2097   21160  2117   21420
                    2143   21650  2166   21850  2186   22050  2206   22310  2232   22550  2256   22750  2276
                    22950  2296   23200  2321   23430  2344   23630  2364   23830  2384   24110  2412   24440
                    2445   24620  2463   24710  2472   24880  2489   25110  2512   25450  2546   25690  2570
                    25920  2593   26150  2616   26520  2653   26740  2675   26980  2699   27280  2729   27500
                    2751   27690  2770   28120  2813   28300  2831   28700  2871   28870  2888   29070  2908
                    29290  2930   29430  2944   29470  2948   29680  2969   29720  2973   29750  2976   29950
                    2996   30120  3013   30670  3068   30850  3086   31060  3107   31340  3135   31490  3150
                    31750  3176   31840  3185   32250  3226   32280  3229   32490  3250   32570  3258   32590
                    3260   33870  3388   33910  3392   34290  3430   35230  3524   35270  3528   35610  3562
                    35970  3598   36010  3602   37140  3715   37430  3744   38140  3815   38480  3849   39280
                    3929   39340  3935   39760  3977   40130  4014   40170  4018   40340  4035   40650  4066
                    40950  4096   41040  4105   41890  4190   42030  4204   42350  4236   44180  4419   44210
                    4422   44580  4459
$$LOCN= 000000      41510  4253   42580  4302   43070  4347   43680  4403   44060  4474
$$METU= 000000      41510  42370  42410  4251   42580  4301   43070  4346   43680  4402   44060  4473
$$RTN1= 000242      41510  4237   4250   42580  4297   4300   43070  4345   43680  4401   44060  4465   4472
$$RTN2= 000243      41510  4242   4252   42580  4301   43070  4346   43680  4402   44060  4473
$$SAC = 000027      32850  42750
$$TO  = 000000      11570  1158   24340  2437   25350  2538   25630  2566   25860  2589   26460  2649   26680
                    2671   27180  2724   27610  2767   28020  2808   28610  2867   30450  3048   30550  3058
                    31270  3130   33790  3382   34230  3426   35010  3507   35130  3516   35890  3592   36550
                    3656   3657   36760  3682   36890  3695   37610  3762   3763   37800  3786   37920  3798
                    38560  3857   3858   40870  4093   42190  4222   44280  4429
$OFILL  015323      5047*  5051*  5061   50960
$1      002126      12070  1232
$10     002446      1349   13540
$100    005260      2545   25500
$101    005322      2569   25740
$102    005364      2592   25970
$103    005424      2615   26210
$104    005512      2652   26570
$105    005554      2674   26790
$106    005612      2698   27030
$107    005660      2728   27330
$11     002474      1365   13700
$110    005706      2750   27550
$111    005742      2769   27740
$112    006034      2812   28170
$113    006056      2830   28350
$114    006150      2870   28750
$115    006166      2887   28920
$116    006224      2907   29120
$117    006244      29260  2943
$12     002522      1382   13870
$120    006262      2929   29340
$121    006264      2933   29380
$122    006300      2941   29440
$123    006324      2947   29580
$124    006324      29660  2989
$125    006370      2968   29900
$126    006366      2972   29870
```

$12"	006366	2975	2985#	
$13	002552	1401	1406#	
$130	006402	2995	3001#	
$131	006420	3012	3017#	
$132	006534	3067	3077#	
$133	006562	3085	3097#	
$134	006604	3106	3115#	
$135	006654	3134	3145#	
$136	006666	3149	3155#	
$137	006726	3175	3180#	
$14	002622	1431	1436#	
$140	006746	3184	3189#	
$141	006776	3200	3203#	
$142	006774	3201#	3281	
$143	007144	3205	3282#	
$144	007034	3223#	3244	
$145	007070	3225	3245#	
$146	007056	3228	3234#	
$147	007066	3233	3241#	
$15	002650	1447	1452#	
$150	007102	3249	3253#	
$151	007126	3252	3273#	
$152	007124	3257	3259	3263#
$153	007126	3262	3271#	
$154	007416	3387	3405#	
$155	007414	3391	3397#	
$156	007416	3396	3403#	
$157	007474	3429	3434#	
$16	002676	1464	1469#	
$160	007734	3523	3541#	
$161	007732	3527	3534#	
$162	007734	3533	3539#	
$163	007776	3561	3567#	
$164	010104	3597	3615#	
$165	010102	3601	3608#	
$166	010104	3607	3613#	
$167	010234	3665	3668#	
$17	002726	1483	1488#	
$170	010232	3666#	3724	
$171	010360	3670	3725#	
$172	010356	3714	3722#	
$173	010424	3743	3749#	
$174	010470	3769	3772#	
$175	010466	3770#	3825	
$176	010614	3774	3826#	
$177	010612	3814	3822#	
$2	002152	1218	1223#	
$20	002764	1508	1513#	
$200	010652	3848	3853#	
$201	010764	3923#	3928	
$202	011000	3926	3929#	
$203	011010	3934	3939#	
$204	011070	3976	3981#	
$205	011150	4013	4029#	
$206	011144	4017	4024#	
$207	011164	4034	4040#	

$21	003012	1522	1527#	
$210	011230	4065	4070#	
$211	011312	4095	4100#	
$212	011322	4104	4109#	
$213	011646	4237	4250#	
$214	011650	4242	4252#	
$215	011516	4185#	4226	
$216	011622	4212	4214	4227#
$217	011536	4189	4194#	
$22	003040	1538	1543#	
$220	011544	4193	4198#	
$221	011562	4203	4208#	
$222	011634	4235	4239#	
$223	011750	4297	4300#	
$224	011750	4301#		
$225	011714	4281	4284#	
$226	011710	4282#	4292	
$227	011740	4286	4293#	
$23	003066	1554	1559#	
$230	012030	4345#		
$231	012030	4346#		
$232	011776	4321	4324#	
$233	011772	4322#	4339	
$234	012022	4326	4340#	
$235	012010	4329	4332#	
$236	012006	4330#	4336	
$237	012020	4334	4337#	
$24	003124	1575	1580#	
$240	012222	4401#		
$241	012222	4402#		
$242	012362	4465	4472#	
$243	012362	4473#		
$244	012224	4415#	4458	
$245	012310	4418	4445#	
$246	012250	4421	4426#	
$247	012256	4425	4434#	
$25	003144	1588	1593#	
$250	012330	4444	4455#	
$26	003172	1604	1609#	
$27	003220	1621	1626#	
$3	002264	1261	1266#	
$30	003250	1640	1645#	
$31	003306	1664	1669#	
$32	003334	1680	1685#	
$33	003362	1697	1702#	
$34	003412	1716	1721#	
$35	003450	1740	1745#	
$36	003476	1756	1761#	
$37	003524	1773	1778#	
$4	002304	1273	1278#	
$40	003554	1792	1797#	
$40CAT=	****** U	4864	4906	
$41	003612	1816	1821#	
$42	003640	1832	1837#	
$43	003666	1849	1854#	
$44	003716	1868	1873#	

```
$45     003756          1902    1912#
$46     004016          1935    1945#
$47     004056          1967    1979#
$5      002332          1289    1294#
$50     004114          1997    2003#
$51     004136          2012    2026#
$52     004174          2051    2059#
$53     004222          2076    2083#
$54     004250          2096    2103#
$55     004276          2116    2123#
$56     004334          2142    2148#
$57     004362          2165    2172#
$6      002360          1306    1311#
$60     004410          2185    2192#
$61     004436          2205    2212#
$62     004474          2231    2238#
$63     004522          2255    2262#
$64     004550          2275    2282#
$65     004576          2295    2302#
$66     004634          2320    2326#
$67     004662          2343    2350#
$7      002410          1325    1330#
$70     004710          2363    2370#
$71     004736          2383    2390#
$72     004774          2411    2418#
$73     005054          2444    2449#
$74     005102          2462    2467#
$75     005114          2471    2476#
$76     005142          2488    2493#
$77     005200          2511    2518#
.       = 015406         928#    932#    941     942#    944#    946#    947#    953     954#    956#    958#    976#    1013
                        1086#   1094    1109    1110    1152#   1218    1219    1232    1233    1261    1262    1273    1274
                        1289    1290    1306    1307    1325    1326    1349    1350    1365    1366    1382    1383    1401
                        1402    1431    1432    1447    1448    1464    1465    1483    1484    1508    1509    1522    1523
                        1538    1539    1554    1555    1575    1576    1588    1589    1604    1605    1621    1622    1640
                        1641    1664    1665    1680    1681    1697    1698    1716    1717    1740    1741    1756    1757
                        1773    1774    1792    1793    1816    1817    1832    1833    1849    1850    1868    1869    1902
                        1903    1935    1936    1967    1968    1997    1998    2012    2013    2051    2052    2076    2077
                        2096    2097    2116    2117    2142    2143    2165    2166    2185    2186    2205    2206    2231
                        2232    2255    2256    2275    2276    2295    2296    2320    2321    2343    2344    2363    2364
                        2383    2384    2411    2412    2444    2445    2462    2463    2471    2472    2488    2489    2511
                        2512    2545    2546    2569    2570    2592    2593    2615    2616    2652    2653    2674    2675
                        2698    2699    2728    2729    2750    2751    2769    2770    2812    2813    2830    2831    2870
                        2871    2887    2888    2907    2908    2929    2930    2943    2944    2947    2948    2968    2969
                        2972    2973    2975    2976    2995    2996    3012    3013    3067    3068    3085    3086    3106
                        3107    3134    3135    3149    3150    3175    3176    3184    3185    3225    3226    3228    3229
                        3249    3250    3257    3258    3259    3260    3325#   3387    3388    3391    3392    3429    3430
                        3523    3524    3527    3528    3561    3562    3597    3598    3601    3602    3714    3715    3743
                        3744    3814    3815    3848    3849    3928    3929    3934    3935    3976    3977    4013    4014
                        4017    4018    4034    4035    4065    4066    4095    4096    4104    4105    4129#   4189    4190
                        4203    4204    4235    4236    4372#   4385#   4391#   4418    4419    4421    4422    4458    4459
                        4512    4516    4532    4556    4640    4643    4772#   4773    4779    4831#   4890    4953    4954
                        5020#
.$ASTA= ****** U        4783    4786
.$X   = 001000           953#    958
```

```
BEGIN       1#
BGNHRD      1#
BGNHW       1#
BGNINI      1#
BGNMOD      1#  4147
BGNMSG      1#
BGNSFT      1#
BGNSRV      1#  3959  3994  4351
BGNSUB      1#  1208  1268  1282  1298  1315  1344  1358  1374  1391  1426  1440  1456  1473  1515
            1531  1547  1583  1597  1613  1630  1659  1673  1689  1705  1735  1749  1765  1782  1811
            1825  1841  1858  1898  1930  1963  2007  2067  2087  2107  2156  2176  2196  2246  2266
            2286  2334  2354  2374  2427  2454  2480  2527  2554  2578  2638  2661  2704  2736  2791
            2821  2849  3034  3081  3100  3118  3361  3408  3477  3557
BGNSW       1#
BRESET      1#  1184  1321  1397  1479  1636  1712  1788  1864  1909  1942  1976  2023  2825  3435
            3907  4110
CALL        1#  1156  2434  2535  2563  2586  2646  2668  2717  2760  2801  2860  3045  3055  3127
            3379  3423  3500  3513  3589  3654  3675  3688  3760  3779  3791  3855  4086  4219  4427
CASE        1#
CKLOOP      1#  1674  1914  1947  1980  2028
CLRVEC      1#  1233  3441  3624
COMMEN      1#  8110
DECR        1#
DECRU       1#
DEFAUL      1#
DEVREG      1#
DEVTYP      1#
DISPAT      1#
ELSE        1#  2932  3232  3251  3261  3395  3532  3606  4192  4424  4443
END         1#
ENDCLN      1#
ENDCOM      1#  8110
ENDDEC      1#
ENDDO       1#  2988  3243
ENDHRD      1#
ENDHW       1#
ENDIF       1#  1222  1265  1277  1293  1310  1329  1353  1369  1386  1405  1435  1451  1468  1487
            1512  1526  1542  1558  1579  1592  1608  1625  1644  1668  1684  1701  1720  1744  1760
            1777  1796  1820  1836  1853  1872  1911  1944  1978  2002  2025  2058  2082  2102  2122
            2147  2171  2191  2211  2237  2261  2281  2301  2325  2349  2369  2389  2417  2448  2466
            2475  2492  2513  2549  2573  2596  2620  2656  2678  2702  2732  2754  2773  2816  2834
            2874  2891  2911  2937  2957  2984  2986  3000  3016  3076  3096  3114  3144  3154  3179
            3188  3240  3270  3272  3402  3404  3433  3538  3540  3566  3612  3614  3721  3748  3821
            3852  3938  3980  4023  4028  4039  4069  4099  4108  4197  4207  4238  4433  4454
ENDINC      1#  3280  3723  3824  4291  4335  4338
ENDINI      1#
ENDLOO      1#  4225
ENDMOD      1#
ENDMSG      1#
ENDRTN      1#  4249  4299  4344  4400  4471
ENDSEL      1#
ENDSFT      1#
ENDSRV      1#  3987  4047  4362
ENDSUB      1#  1224  1279  1295  1312  1331  1355  1371  1388  1407  1437  1453  1470  1489  1528
            1544  1560  1594  1610  1627  1646  1670  1686  1703  1722  1746  1762  1779  1798  1822
            1838  1855  1875  1915  1948  1981  2029  2084  2104  2124  2173  2193  2213  2263  2283
```

	2068	2071	2088	2091	2108	2111	2145	2157	2160	2177	2180	2197	2200	2235	2247
	2250	2267	2270	2287	2290	2323	2335	2338	2355	2358	2375	2378	2415	2428	2430
	2439	2455	2458	2481	2484	2515	2528	2531	2540	2555	2559	2579	2582	2618	2632
	2634	2639	2642	2662	2664	2700	2705	2712	2737	2743	2788	2792	2795	2822	2847
	2850	2855	2882	2909	2913	2915	2917	2923	2930	2935	2952	2955	2982	3007	3035
	3041	3051	3061	3074	3082	3094	3101	3119	3123	3142	3177	3186	3191	3193	3195
	3206	3209	3212	3215	3218	3220	3230	3236	3238	3275	3277	3283	3288	3345	3349
	3352	3356	3357	3358	3359	3360	3362	3364	3374	3409	3411	3414	3439	3444	3445
	3447	3448	3471	3472	3473	3474	3475	3478	3479	3482	3485	3495	3509	3517	3543
	3546	3558	3564	3569	3572	3580	3583	3585	3616	3618	3622	3627	3628	3630	3631
	3644	3659	3685	3697	3705	3708	3719	3728	3746	3751	3764	3789	3801	3805	3808
	3819	3850	3879	3882	3884	3888	3890	3894	3897	3900	3904	3910	3914	3917	3941
	3943	3964	3967	3971	3978	3999	4002	4006	4019	4021	4026	4036	4067	4072	4076
	4081	4084	4174	4176	4178	4190	4195	4205	4223	4271	4273	4287	4289	4294	4317
	4360	4422	4431	4435	4437	4439	4441	4446	4448	4450	4452	4460	4462		
LOCAL	1#														
LOOP	1#	4184													
MSG	1186#	1188	1250#	1252	1495#	1497	1566#	1568	2040#	2042	2131#	2133	2220#	2222	2310#
	2312	2398#	2400	2500#	2502	2686#	2688	2778#	2780	2895#	2897	3164#	3166	3333#	3335
	3459#	3461	3636#	3638	3836#	3838	4055#	4057							
MULT	1#	811#													
NEWTST	1#	811#	1186	1250	1336	1413	1495	1566	1652	1728	1804	1887	1920	1953	1987
	2040	2131	2220	2310	2398	2500	2606	2686	2778	2839	2895	3027	3164	3333	3459
	3636	3735	3836	4055	4121										
NOLOCA	1#														
POINTE	1#														
POP	1#	811#	1241	3449	3632	4341	4541	4542	4825	4826	5007				
PRINTB	1#														
PUSH	1#	811#	1234	3442	3625	4314	4522	4528	4786	4788	4809	4966			
REPEAT	1#	1206	2925	3922	4414										
REPORT	1#	811#													
RETURN	1#	4236	4240	4296	4464										
ROUTIN	1#	4149	4256	4305	4366	4404									
SAVR14	1#														
SCOPE	706#	1192	1255	1339	1416	1503	1570	1655	1731	1807	1890	1923	1956	1990	2045
	2136	2225	2314	2405	2505	2609	2691	2783	2842	2900	3030	3170	3341	3465	3640
	3738	3840	4060	4124	4487										
SELECT	1#														
SETPRI	1#	811#	1420	2421	2521	2624	3368	3417	3489	3576	3649	3755	3874		
SETTRA	5121#	5130	5131	5132	5133	5135	5137	5138	5139						
SETUP	1#	811#	1088												
SETVEC	1#	1198	3355	3470											
SKIP	1#	811#	1264	1511	1578	2001	2057	2146	2236	2324	2416	2516	2600	2619	2701
	2910	2956	2983	3020	3075	3095	3113	3143	3158	3178	3187	3292	3565	3720	3747
	3820	3851	3947	4068	4113										
SLASH	1#	811#													
SPACE	811#														
STARS	1#	811#	827	829	847	849	867	869	889	891	910	925	939	950	952
	959	972	1013	1016	1186	1191	1245	1247	1250	1254	1335	1336	1338	1412	1413
	1415	1494	1495	1502	1565	1566	1569	1651	1652	1654	1727	1728	1730	1803	1804
	1806	1881	1884	1887	1889	1920	1922	1953	1955	1986	1987	1989	2032	2036	2040
	2044	2130	2131	2135	2219	2220	2224	2309	2310	2313	2397	2398	2404	2499	2500
	2504	2605	2606	2608	2685	2686	2690	2777	2778	2782	2838	2839	2841	2894	2895
	2899	3026	3027	3029	3163	3164	3169	3332	3333	3340	3458	3459	3464	3635	3636
	3639	3734	3735	3737	3835	3836	3839	4054	4055	4059	4121	4123	4148	4169	4255
	4269	4304	4313	4350	4357	4368	4406	4413	4479	4518	4534	4563	4642	4645	4713

```
        4742    4781    4838    4892    4956    5023    5100

STRUCT    1#
SWRSU     1#    8111#   11111#
TRMTRP  5121#
TYPBIN    1#    8111#
TYPDEC    1#    8111#   4142    4499
TYPNAM    1#    8111#   1132
TYPNUM    1#    8111#
TYPOCS    1#    8111#
TYPOCT    1#    8111#   4130    4136    4373    4380    4386    4392    4398    4663
TYPTXT    1#    8111#   4126    4132    4138    4369    4375    4382    4388    4394
UNTIL     1#    1230    2939    3924    4456
UNTILB    1#
WAITMS    1#    2433    2534    2562    2585    2645    2667    3044    3054    3126    3378    3422    3512    3588    4218
WHILE     1#    2965    3222
WHILEB    1#
$ADDON    1#    1207    1208    1217    1219    1260    1262    1272    1274    1288    1290    1305    1307    1324    1326
        1348    1350    1364    1366    1381    1383    1400    1402    1430    1432    1446    1448    1463    1465    1482
        1484    1507    1509    1521    1523    1537    1539    1553    1555    1574    1576    1587    1589    1603    1605
        1620    1622    1639    1641    1663    1665    1679    1681    1696    1698    1715    1717    1739    1741    1755
        1757    1772    1774    1791    1793    1815    1817    1831    1833    1848    1850    1867    1869    1901    1903
        1934    1936    1966    1968    1996    1998    2011    2013    2050    2052    2075    2077    2095    2097    2115
        2117    2141    2143    2164    2166    2184    2186    2204    2206    2230    2232    2254    2256    2274    2276
        2294    2296    2319    2321    2342    2344    2362    2364    2382    2384    2410    2412    2443    2445    2461
        2463    2470    2472    2487    2489    2510    2512    2544    2546    2568    2570    2591    2593    2614    2616
        2651    2653    2673    2675    2697    2699    2728    2729    2749    2751    2769    2770    2812    2813    2829
        2831    2870    2871    2886    2888    2906    2908    2926    2927    2928    2930    2935    2945    2946    2948
        2966    2967    2969    2971    2973    2974    2976    2989    2994    2996    3011    3013    3066    3068    3084
        3086    3105    3107    3133    3135    3148    3150    3174    3176    3183    3185    3199    3201    3202    3206
        3223    3224    3226    3227    3229    3235    3244    3248    3250    3254    3256    3260    3264    3386    3388
        3390    3392    3398    3428    3430    3522    3524    3526    3528    3535    3560    3562    3596    3598    3600
        3602    3609    3664    3666    3667    3671    3713    3715    3742    3744    3768    3770    3771    3775    3813
        3815    3847    3849    3923    3924    3930    3933    3935    3975    3977    4012    4014    4016    4018    4033
        4035    4064    4066    4095    4096    4103    4105    4151    4185    4186    4188    4190    4195    4202    4204
        4226    4234    4236    4258    4280    4282    4283    4287    4307    4320    4322    4323    4327    4328    4330
        4331    4335    4368    4406    4415    4416    4417    4419    4420    4422    4427    4446
$AND      1#    3256
$BRANC    1#    1218    1232    1261    1273    1289    1306    1325    1349    1365    1382    1401    1431    1447    1464
        1483    1508    1522    1538    1554    1575    1588    1604    1621    1640    1664    1680    1697    1716    1740
        1756    1773    1792    1816    1832    1849    1868    1902    1935    1967    1997    2012    2051    2076    2096
        2116    2142    2165    2185    2205    2231    2255    2275    2295    2320    2343    2363    2383    2411    2444
        2462    2471    2488    2511    2545    2569    2592    2615    2652    2674    2698    2728    2750    2769    2812
        2830    2870    2887    2907    2929    2933    2941    2943    2947    2968    2972    2975    2989    2995    3012
        3067    3085    3106    3134    3149    3175    3184    3200    3205    3225    3228    3233    3244    3249    3252
        3257    3259    3262    3281    3387    3391    3396    3429    3523    3527    3533    3561    3597    3601    3607
        3665    3670    3714    3724    3743    3769    3774    3814    3825    3848    3926    3928    3934    3976    4013
        4017    4034    4065    4095    4104    4189    4193    4203    4212    4214    4226    4235    4237    4242    4281
        4286    4292    4297    4321    4326    4329    4334    4336    4339    4418    4421    4425    4444    4458    4465
$BRCOD    1#    2940    3204    3669    3773    3925    4211    4213    4285    4325    4333
$CALL     1#    1157    2434    2535    2563    2586    2646    2668    2718    2761    2802    2861    3045    3055    3127
        3379    3423    3501    3513    3589    3655    3676    3689    3761    3780    3792    3856    4087    4219    4428
$CHECK    1#    1217    1260    1272    1288    1305    1324    1348    1364    1381    1400    1430    1446    1463    1482
        1507    1521    1537    1553    1574    1587    1603    1620    1639    1663    1679    1696    1715    1739    1755
        1772    1791    1815    1831    1848    1867    1901    1934    1966    1996    2011    2050    2075    2095    2115
        2141    2164    2184    2204    2230    2254    2274    2294    2319    2342    2362    2382    2410    2443    2461
        2470    2487    2510    2544    2568    2591    2614    2651    2673    2697    2749    2829    2886    2906    2928
```

	2116	2142	2165	2185	2205	2231	2255	2275	2295	2320	2343	2363	2383	2411	2444
	2462	2471	2488	2511	2545	2569	2592	2615	2652	2674	2698	2728	2750	2769	2812
	2830	2870	2887	2907	2929	2933	2941	2943	2947	2968	2972	2975	2989	2995	3012
	3067	3085	3106	3134	3149	3175	3184	3200	3205	3225	3228	3233	3244	3249	3252
	3257	3259	3262	3281	3387	3391	3596	3429	3523	3527	3533	3561	3597	3601	3607
	3665	3670	3714	3724	3743	3769	3774	3814	3825	3848	3926	3928	3934	3976	4013
	4017	4034	4065	4095	4104	4189	4193	4203	4212	4214	4226	4235	4237	4242	4281
	4286	4292	4297	4321	4326	4329	4334	4336	4339	4418	4421	4425	4444	4458	4465
$GENTA	10	1207	1223	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2926	2934	2938	2944	2958	2966	2985	2987	2990	3001	3017
	3077	3097	3115	3145	3155	3180	3189	3201	3203	3223	3234	3241	3245	3253	3263
	3271	3273	3282	3397	3403	3405	3434	3534	3539	3541	3567	3608	3613	3615	3666
	3668	3722	3725	3749	3770	3772	3822	3826	3853	3923	3929	3939	3981	4024	4029
	4040	4070	4100	4109	4185	4194	4198	4208	4227	4239	4250	4252	4282	4284	4293
	4300	4301	4322	4324	4330	4332	4337	4340	4345	4346	4401	4402	4415	4426	4434
	4445	4455	4472	4473											
$IF	10	1217	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463	1482
	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739	1755
	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095	2115
	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443	2461
	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906	2928
	2946	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183	3227	3248	3256
	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742	3813	3847	3933	3975	4012
	4016	4033	4064	4103	4188	4202	4234	4417	4420						
$IFCOD	10	1217	1231	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463
	1482	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739
	1755	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095
	2115	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443
	2461	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906
	2928	2942	2946	2967	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183
	3224	3227	3248	3256	3258	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742
	3813	3847	3927	3933	3975	4012	4016	4033	4064	4103	4188	4202	4234	4417	4420
	4457														
$IFCON	10	2728	2769	2812	2870	4095									
$IFOPR	10	1218	1232	1261	1273	1289	1306	1325	1349	1365	1382	1401	1431	1447	1464
	1483	1508	1522	1538	1554	1575	1588	1604	1621	1640	1664	1680	1697	1716	1740
	1756	1773	1792	1816	1832	1849	1868	1902	1935	1967	1997	2012	2051	2076	2096
	2116	2142	2165	2185	2205	2231	2255	2275	2295	2320	2343	2363	2383	2411	2444
	2462	2471	2488	2511	2545	2569	2592	2615	2652	2674	2698	2728	2750	2769	2812
	2830	2870	2887	2907	2929	2943	2947	2968	2972	2975	2995	3012	3067	3085	3106
	3134	3149	3175	3184	3225	3228	3249	3257	3259	3387	3391	3429	3523	3527	3561
	3597	3601	3714	3743	3814	3848	3928	3934	3976	4013	4017	4034	4065	4095	4104
	4189	4203	4235	4418	4421	4458									
$LET	10	1154	1161	1164	1166	1169	1171	1174	1177	1180	1183	1196	1199	1200	1201
	1202	1203	1205	1209	1212	1226	1228	1236	1237	1239	1240	1263	1269	1283	1285
	1299	1302	1316	1319	1345	1359	1361	1375	1378	1392	1395	1427	1441	1443	1457
	1460	1474	1477	1510	1516	1518	1532	1535	1548	1550	1577	1584	1598	1600	1614
	1617	1631	1634	1660	1674	1676	1690	1693	1706	1710	1736	1750	1752	1766	1769
	1783	1786	1812	1826	1828	1842	1845	1859	1862	1899	1931	1964	2000	2008	2056
	2068	2072	2088	2092	2108	2112	2145	2157	2161	2177	2181	2197	2201	2235	2247
	2251	2267	2271	2287	2291	2323	2335	2339	2355	2359	2375	2379	2415	2428	2431
	2440	2455	2459	2481	2485	2515	2528	2532	2541	2555	2560	2579	2583	2618	2633

	2635	2639	2643	2662	2665	2700	2705	2713	2737	2744	2789	2792	2796	2822	2848
	2850	2856	2883	2909	2914	2916	2918	2924	2931	2936	2953	2955	2982	3008	3035
	3042	3052	3062	3074	3082	3094	3101	3119	3124	3142	3177	3186	3192	3194	3196
	3207	3210	3213	3216	3219	3221	3231	3237	3239	3276	3278	3284	3289	3346	3350
	3353	3356	3357	3358	3359	3360	3362	3365	3375	3409	3412	3415	3440	3444	3445
	3447	3448	3471	3472	3473	3474	3475	3478	3480	3483	3486	3496	3510	3518	3544
	3547	3558	3564	3570	3573	3581	3584	3586	3617	3619	3623	3627	3628	3630	3631
	3645	3660	3686	3698	3706	3709	3719	3729	3746	3752	3765	3790	3802	3806	3809
	3819	3850	3880	3883	3885	3889	3891	3895	3898	3901	3905	3911	3915	3918	3942
	3944	3965	3968	3972	3979	4000	4003	4007	4020	4022	4027	4037	4067	4073	4077
	4082	4085	4175	4177	4179	4191	4196	4206	4224	4272	4274	4288	4290	4295	4318
	4361	4423	4432	4436	4438	4440	4442	4447	4449	4451	4453	4461	4463		
$LPCNT	10	3199	3664	3768	4280	4320	4328								
$OPADD	10	1172	1175	1178	1181	1226	1229	1238	2936	3202	3237	3239	3353	3446	3629
	3667	3771	3965	4000	4027	4283	4323	4331	4361	4451	4453	4463			
$OPAND	10	3285	4275												
$OPCD1	10	1172	1175	1178	1181	1226	1229	1238	1285	1302	1319	1361	1378	1395	1443
	1460	1477	1518	1535	1550	1600	1617	1634	1676	1693	1710	1752	1769	1786	1828
	1845	1862	2072	2092	2112	2161	2181	2201	2251	2271	2291	2339	2359	2379	2431
	2459	2485	2532	2560	2583	2633	2643	2665	2789	2848	2914	2936	3196	3202	3237
	3239	3285	3353	3365	3375	3415	3446	3483	3486	3510	3544	3547	3573	3581	3584
	3586	3617	3619	3629	3645	3667	3707	3709	3729	3752	3771	3807	3809	3911	3915
	3918	3942	3944	3965	3969	3979	4000	4004	4008	4027	4037	4073	4082	4224	4275
	4283	4288	4290	4295	4323	4331	4361	4449	4451	4453	4463				
$OPCD2	10	4288	4295	4449											
$OPCOD	10	1172	1175	1178	1181	1226	1229	1238	1285	1302	1319	1361	1378	1395	1443
	1460	1477	1518	1535	1550	1600	1617	1634	1676	1693	1710	1752	1769	1786	1828
	1845	1862	2072	2092	2112	2161	2181	2201	2251	2271	2291	2339	2359	2379	2431
	2459	2485	2532	2560	2583	2633	2643	2665	2789	2848	2914	2936	3196	3202	3237
	3239	3285	3353	3365	3375	3415	3446	3483	3486	3510	3544	3547	3573	3581	3584
	3586	3617	3619	3629	3645	3667	3707	3709	3729	3752	3771	3807	3809	3911	3915
	3918	3942	3944	3965	3969	3979	4000	4004	4008	4027	4037	4073	4082	4224	4275
	4283	4288	4290	4295	4323	4331	4361	4449	4451	4453	4463				
$OPCOM	10	4295													
$OPDEF	10	1154	1157	1161	1164	1166	1169	1171	1172	1174	1175	1177	1178	1180	1181
	1183	1196	1199	1200	1201	1202	1203	1205	1209	1212	1217	1218	1226	1228	1229
	1231	1232	1236	1237	1238	1239	1240	1260	1261	1263	1269	1272	1273	1283	1285
	1288	1289	1299	1302	1305	1306	1316	1319	1324	1325	1345	1348	1349	1359	1361
	1364	1365	1375	1378	1381	1382	1392	1395	1400	1401	1427	1430	1431	1441	1443
	1446	1447	1457	1460	1463	1464	1474	1477	1482	1483	1507	1508	1510	1516	1518
	1521	1522	1532	1535	1537	1538	1548	1550	1553	1554	1574	1575	1577	1584	1587
	1588	1598	1600	1603	1604	1614	1617	1620	1621	1631	1634	1639	1640	1660	1663
	1664	1674	1676	1679	1680	1690	1693	1696	1697	1706	1710	1715	1716	1736	1739
	1740	1750	1752	1755	1756	1766	1769	1772	1773	1783	1786	1791	1792	1812	1815
	1816	1826	1828	1831	1832	1842	1845	1848	1849	1859	1862	1867	1868	1899	1901
	1902	1931	1934	1935	1964	1966	1967	1996	1997	2000	2008	2011	2012	2050	2051
	2056	2068	2072	2075	2076	2088	2092	2095	2096	2108	2112	2115	2116	2141	2142
	2145	2157	2161	2164	2165	2177	2181	2184	2185	2197	2201	2204	2205	2230	2231
	2235	2247	2251	2254	2255	2267	2271	2274	2275	2287	2291	2294	2295	2319	2320
	2323	2335	2339	2342	2343	2355	2359	2362	2363	2375	2379	2382	2383	2410	2411
	2415	2428	2431	2434	2435	2436	2437	2440	2443	2444	2455	2459	2461	2462	2470
	2471	2481	2485	2487	2488	2510	2511	2515	2528	2532	2535	2536	2537	2538	2541
	2544	2545	2555	2560	2563	2564	2565	2566	2568	2569	2579	2583	2586	2587	2588
	2589	2591	2592	2614	2615	2618	2633	2635	2639	2643	2646	2647	2648	2649	2651
	2652	2662	2665	2668	2669	2670	2671	2673	2674	2697	2698	2700	2705	2713	2718
	2719	2720	2721	2722	2723	2724	2728	2737	2744	2749	2750	2761	2762	2763	2764

```
        2765    2766    2767    2769    2789    2792    2796    2802    2803    2804    2805    2806    2807    2808    2812
        2822    2829    2830    2848    2850    2856    2861    2862    2863    2864    2865    2866    2867    2870    2883
        2886    2887    2906    2907    2909    2914    2916    2918    2924    2928    2929    2931    2933    2936    2940
        2941    2942    2943    2946    2947    2953    2955    2967    2968    2971    2972    2974    2975    2982    2989
        2994    2995    3008    3011    3012    3035    3042    3045    3046    3047    3048    3052    3055    3056    3057
        3058    3062    3066    3067    3074    3082    3084    3085    3094    3101    3105    3106    3119    3124    3127
        3128    3129    3130    3133    3134    3142    3148    3149    3174    3175    3177    3183    3184    3186    3192
        3194    3196    3199    3200    3202    3204    3205    3207    3210    3213    3216    3219    3221    3224    3225
        3227    3228    3231    3233    3237    3239    3244    3248    3249    3252    3256    3257    3258    3259    3262
        3276    3278    3281    3284    3285    3286    3287    3289    3346    3350    3353    3356    3357    3358    3359
        3360    3362    3365    3375    3379    3380    3381    3382    3386    3387    3390    3391    3396    3409    3412
        3415    3423    3424    3425    3426    3428    3429    3440    3444    3445    3446    3447    3448    3471    3472
        3473    3474    3475    3478    3480    3483    3486    3496    3501    3502    3503    3504    3505    3506    3507
        3510    3513    3514    3515    3516    3518    3522    3523    3526    3527    3533    3544    3547    3558    3560
        3561    3564    3570    3573    3581    3584    3586    3589    3590    3591    3592    3596    3597    3600    3601
        3607    3617    3619    3623    3627    3628    3629    3630    3631    3645    3655    3656    3657    3660    3664
        3665    3667    3669    3670    3676    3677    3678    3679    3680    3681    3682    3686    3689    3690    3691
        3692    3693    3694    3695    3698    3706    3707    3709    3713    3714    3719    3724    3729    3742    3743
        3746    3752    3761    3762    3763    3765    3768    3769    3771    3773    3774    3780    3781    3782    3783
        3784    3785    3786    3790    3792    3793    3794    3795    3796    3797    3798    3802    3806    3807    3809
        3813    3814    3819    3825    3847    3848    3850    3856    3857    3858    3880    3883    3885    3889    3891
        3895    3898    3901    3905    3911    3915    3918    3925    3926    3927    3928    3933    3934    3942    3944
        3965    3968    3969    3972    3975    3976    3979    4000    4003    4004    4007    4008    4012    4013    4016
        4017    4020    4022    4027    4033    4034    4037    4064    4065    4067    4073    4077    4082    4085    4087
        4088    4089    4090    4091    4092    4093    4095    4103    4104    4175    4177    4179    4188    4189    4191
        4193    4196    4202    4203    4206    4211    4212    4213    4214    4219    4220    4221    4222    4224    4226
        4234    4235    4237    4241    4242    4251    4253    4272    4274    4275    4276    4277    4280    4281    4283
        4285    4286    4288    4290    4292    4295    4297    4302    4318    4320    4321    4323    4325    4326    4328
        4329    4331    4333    4334    4336    4339    4347    4361    4403    4417    4418    4420    4421    4423    4425
        4428    4432    4436    4438    4440    4442    4444    4447    4449    4451    4453    4457    4458    4461    4463
        4465    4474
$OPEQU     1#
$OPNAN     1#
$OPNEG     1#
$OPNOR     1#
$OPNOT     1#   1302    1378    1460    1518    1550    1617    1693    1769    1845    2072    2112    2161    2201    2251
        2291    2339    2379    2431    2485    2532    2583    2633    2665    3365    3415    3486    3544    3547    3573
        3581    3617    3619    3707    3709    3729    3752    3807    3809    3942    3944    3969    3979    4004    4008
        4037
$OPOR      1#   1285    1319    1361    1395    1443    1477    1535    1600    1634    1676    1710    1752    1786    1828
        1862    2092    2181    2271    2359    2459    2560    2643    2789    2848    2914    3196    3375    3483    3510
        3584    3586    3645    3911    3915    3918    4073    4082    4290
$OPROT     1#   4449
$OPRO      1#   1154    1161    1164    1166    1169    1183    1196    1199    1200    1201    1202    1203    1205    1209
        1212    1236    1239    1240    1263    1269    1283    1299    1316    1345    1359    1375    1392    1427    1441
        1457    1474    1510    1516    1532    1548    1577    1584    1598    1614    1631    1660    1674    1690    1706
        1736    1750    1766    1783    1812    1826    1842    1859    1899    1931    1964    2000    2008    2056    2068
        2088    2108    2145    2157    2177    2197    2235    2247    2267    2287    2323    2335    2355    2375    2415
        2428    2440    2455    2481    2515    2528    2541    2555    2579    2618    2635    2639    2662    2700    2705
        2713    2737    2744    2792    2796    2822    2850    2856    2883    2909    2916    2918    2924    2931    2953
        2955    2982    3008    3035    3042    3052    3062    3074    3082    3094    3101    3119    3124    3142    3177
        3186    3192    3194    3199    3207    3210    3213    3216    3219    3221    3231    3276    3278    3289    3346
        3350    3356    3357    3358    3359    3360    3362    3409    3412    3440    3444    3447    3448    3471    3472
        3473    3474    3475    3478    3480    3496    3518    3558    3564    3570    3623    3627    3630    3631    3660
        3664    3686    3698    3719    3746    3765    3768    3790    3802    3819    3850    3880    3883    3885    3889
        3891    3895    3898    3901    3905    3972    4020    4022    4067    4077    4085    4175    4177    4179    4191
```

	4196	4206	4272	4280	4318	4320	4328	4423	4432	4436	4438	4440	4442	4447	4461
$OPR1	10	3202	3667	3771	4283	4295	4323	4331							
$OPR2	10	1171	1174	1177	1180	1226	1228	1237	1285	1302	1319	1361	1378	1395	1443
	1460	1477	1518	1535	1550	1600	1617	1634	1676	1693	1710	1752	1769	1786	1828
	1845	1862	2072	2092	2112	2161	2181	2201	2251	2271	2291	2339	2359	2379	2431
	2459	2485	2532	2560	2583	2633	2643	2665	2789	2848	2914	2936	3196	3237	3239
	3284	3353	3365	3375	3415	3445	3483	3486	3510	3544	3547	3573	3581	3584	3586
	3617	3619	3628	3645	3706	3709	3729	3752	3806	3809	3911	3915	3918	3942	3944
	3965	3968	3979	4000	4003	4007	4027	4037	4073	4082	4224	4274	4288	4290	4361
	4449	4451	4453	4463											
$OPSHF	10	4288													
$OPSUB	10	4224													
$OPSWB	10														
$OPXOR	10														
$OR	10														
$PUT	10	2435	2536	2564	2587	2647	2669	2719	2762	2803	2862	3046	3056	3128	3380
	3424	3502	3514	3590	3677	3690	3781	3793	4088	4220					
$STRUC	10														
$SUBON	10	1223	1231	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2934	2938	2940	2958	2985	2987	2989	2990	3001	3017	3077
	3097	3115	3145	3155	3180	3189	3203	3206	3234	3241	3244	3245	3253	3263	3271
	3273	3281	3282	3397	3403	3405	3434	3534	3539	3541	3567	3608	3613	3615	3668
	3671	3722	3724	3725	3749	3772	3775	3822	3825	3826	3853	3925	3939	3981	4024
	4029	4040	4070	4100	4109	4194	4198	4208	4226	4227	4239	4250	4284	4287	4292
	4293	4300	4324	4327	4332	4335	4336	4337	4339	4340	4345	4401	4426	4434	4445
	4455	4457	4472												
$THEN	10	1217	1260	1272	1288	1305	1324	1348	1364	1381	1400	1430	1446	1463	1482
	1507	1521	1537	1553	1574	1587	1603	1620	1639	1663	1679	1696	1715	1739	1755
	1772	1791	1815	1831	1848	1867	1901	1934	1966	1996	2011	2050	2075	2095	2115
	2141	2164	2184	2204	2230	2254	2274	2294	2319	2342	2362	2382	2410	2443	2461
	2470	2487	2510	2544	2568	2591	2614	2651	2673	2697	2749	2829	2886	2906	2928
	2946	2971	2974	2994	3011	3066	3084	3105	3133	3148	3174	3183	3227	3248	3258
	3386	3390	3428	3522	3526	3560	3596	3600	3713	3742	3813	3847	3933	3975	4012
	4016	4033	4064	4103	4188	4202	4234	4417	4420						
$TILA	10														
$TILO	10														
$UNTL2	10	2940	3925												
$UNTL3	10														
$WHILE	10	2966	3223												
$$CMRE	9700														
$$CMTM	9700														
$$DEFA	10														
$$ENDS	10														
$$ERRO	10														
$$ESCA	10	8110													
$$GEN	10	1207	1223	1266	1278	1294	1311	1330	1354	1370	1387	1406	1436	1452	1469
	1488	1513	1527	1543	1559	1580	1593	1609	1626	1645	1669	1685	1702	1721	1745
	1761	1778	1797	1821	1837	1854	1873	1912	1945	1979	2003	2026	2059	2083	2103
	2123	2148	2172	2192	2212	2238	2262	2282	2302	2326	2350	2370	2390	2418	2449
	2467	2476	2493	2518	2550	2574	2597	2621	2657	2679	2703	2733	2755	2774	2817
	2835	2875	2892	2912	2926	2934	2938	2944	2958	2966	2985	2987	2990	3001	3017
	3077	3097	3115	3145	3155	3180	3189	3201	3203	3223	3234	3241	3245	3253	3263

```
        3271  3273  3282  3397  3403  3405  3434  3534  3539  3541  3567  3608  3613  3615  3666
        3668  3722  3725  3749  3770  3772  3822  3826  3853  3923  3929  3939  3981  4024  4029
        4040  4070  4100  4109  4150  4185  4194  4198  4208  4227  4239  4250  4252  4257  4282
        4284  4293  4300  4301  4306  4322  4324  4330  4332  4337  4340  4345  4346  4367  4401
        4402  4405  4415  4426  4434  4445  4455  4472  4473
$$GETS    1#  1223  1231  1266  1278  1294  1311  1330  1354  1370  1387  1406  1436  1452  1469
        1488  1513  1527  1543  1559  1580  1593  1609  1626  1645  1669  1685  1702  1721  1745
        1761  1778  1797  1821  1837  1854  1873  1912  1945  1979  2003  2026  2059  2083  2103
        2123  2148  2172  2192  2212  2238  2262  2282  2302  2326  2350  2370  2390  2418  2449
        2467  2476  2493  2518  2550  2574  2597  2621  2657  2679  2703  2733  2755  2774  2817
        2835  2875  2892  2912  2933  2934  2938  2940  2958  2985  2987  2989  2990  3001  3017
        3077  3097  3115  3145  3155  3180  3189  3203  3206  3233  3234  3241  3244  3245  3252
        3253  3262  3263  3271  3273  3281  3282  3396  3397  3403  3405  3434  3533  3534  3539
        3541  3567  3607  3608  3613  3615  3668  3671  3722  3724  3725  3749  3772  3775  3822
        3825  3826  3853  3925  3939  3981  4024  4029  4040  4070  4100  4109  4193  4194  4198
        4208  4211  4226  4227  4239  4250  4284  4287  4292  4293  4300  4324  4327  4332  4335
        4336  4337  4339  4340  4345  4401  4425  4426  4434  4444  4445  4455  4457  4472
$$GETT    1#  2933  3233  3252  3262  3396  3533  3607  4193  4211  4425  4444
$$LPCN    1#  3202  3667  3771  4283  4323  4331
$$NEWT    1#  8110  1186  1250  1336  1413  1495  1566  1652  1728  1804  1887  1920  1953  1987
        2040  2131  2220  2310  2398  2500  2606  2686  2778  2839  2895  3027  3164  3333  3459
        3636  3735  3836  4055  4121
$$POP     1#  1223  1231  1266  1278  1294  1311  1330  1354  1370  1387  1406  1436  1452  1469
        1488  1513  1527  1543  1559  1580  1593  1609  1626  1645  1669  1685  1702  1721  1745
        1761  1778  1797  1821  1837  1854  1873  1912  1945  1979  2003  2026  2059  2083  2103
        2123  2148  2172  2192  2212  2238  2262  2282  2302  2326  2350  2370  2390  2418  2449
        2467  2476  2493  2518  2550  2574  2597  2621  2657  2679  2703  2733  2755  2774  2817
        2835  2875  2892  2912  2934  2938  2940  2958  2985  2987  2989  2990  3001  3017  3077
        3097  3115  3145  3155  3180  3189  3203  3206  3234  3241  3244  3245  3253  3263  3271
        3273  3281  3282  3397  3403  3405  3434  3534  3539  3541  3567  3608  3613  3615  3668
        3671  3722  3724  3725  3749  3772  3775  3822  3825  3826  3853  3925  3939  3981  4024
        4029  4040  4070  4100  4109  4194  4198  4208  4226  4227  4239  4250  4284  4287  4292
        4293  4300  4324  4327  4332  4335  4336  4337  4339  4340  4345  4401  4426  4434  4445
        4455  4457  4472
$$PUSH    1#  1207  1208  1217  1219  1260  1262  1272  1274  1288  1290  1305  1307  1324  1326
        1348  1350  1364  1366  1381  1383  1400  1402  1430  1432  1446  1448  1463  1465  1482
        1484  1507  1509  1521  1523  1537  1539  1553  1555  1574  1576  1587  1589  1603  1605
        1620  1622  1639  1641  1663  1665  1679  1681  1696  1698  1715  1717  1739  1741  1755
        1757  1772  1774  1791  1793  1815  1817  1831  1833  1848  1850  1867  1869  1901  1903
        1934  1936  1966  1968  1996  1998  2011  2013  2050  2052  2075  2077  2095  2097  2115
        2117  2141  2143  2164  2166  2184  2186  2204  2206  2230  2232  2254  2256  2274  2276
        2294  2296  2319  2321  2342  2344  2362  2364  2382  2384  2410  2412  2443  2445  2461
        2463  2470  2472  2487  2489  2510  2512  2544  2546  2568  2570  2591  2593  2614  2616
        2651  2653  2673  2675  2697  2699  2728  2729  2749  2751  2769  2770  2812  2813  2829
        2831  2870  2871  2886  2888  2906  2908  2926  2927  2928  2930  2935  2946  2948  2966
        2967  2969  2971  2973  2974  2976  2989  2994  2996  3011  3013  3066  3068  3084  3086
        3105  3107  3133  3135  3148  3150  3174  3176  3183  3185  3199  3201  3202  3206  3223
        3224  3226  3227  3229  3235  3244  3248  3250  3254  3256  3260  3264  3386  3388  3390
        3392  3398  3428  3430  3522  3524  3526  3528  3535  3560  3562  3596  3598  3600  3602
        3609  3664  3666  3667  3671  3713  3715  3742  3744  3768  3770  3771  3775  3813  3815
        3847  3849  3923  3924  3933  3935  3975  3977  4012  4014  4016  4018  4033  4035  4064
        4066  4095  4096  4103  4105  4151  4185  4186  4188  4190  4195  4202  4204  4226  4234
        4236  4258  4280  4282  4283  4287  4307  4320  4322  4323  4327  4328  4330  4331  4335
        4368  4406  4415  4416  4417  4419  4420  4422  4427  4446
$$SELE    1#
$$SET    5121#  5130  5131  5132  5133  5135  5137  5138  5139
```

```
.$SUPR      1#
.$TRAP      1#    5098
.$TYPB      1#
.$TYPD      1#    4954
.$TYPE      1#    4561
.$TYPO      1#    5021
.$40CA      1#
.1170       1#


. ABS.  015406     000


 ERRORS DETECTED:  0

 CVDVAD,CVDVAD/SOL/CRF=CVDVAD.SML,CVDVAD.MAC,CVDVAD.P11
 RUN-TIME: 84 90 6 SECONDS
 RUN-TIME RATIO: 287/181=1.5
 CORE USED:  43K  (85 PAGES)
```