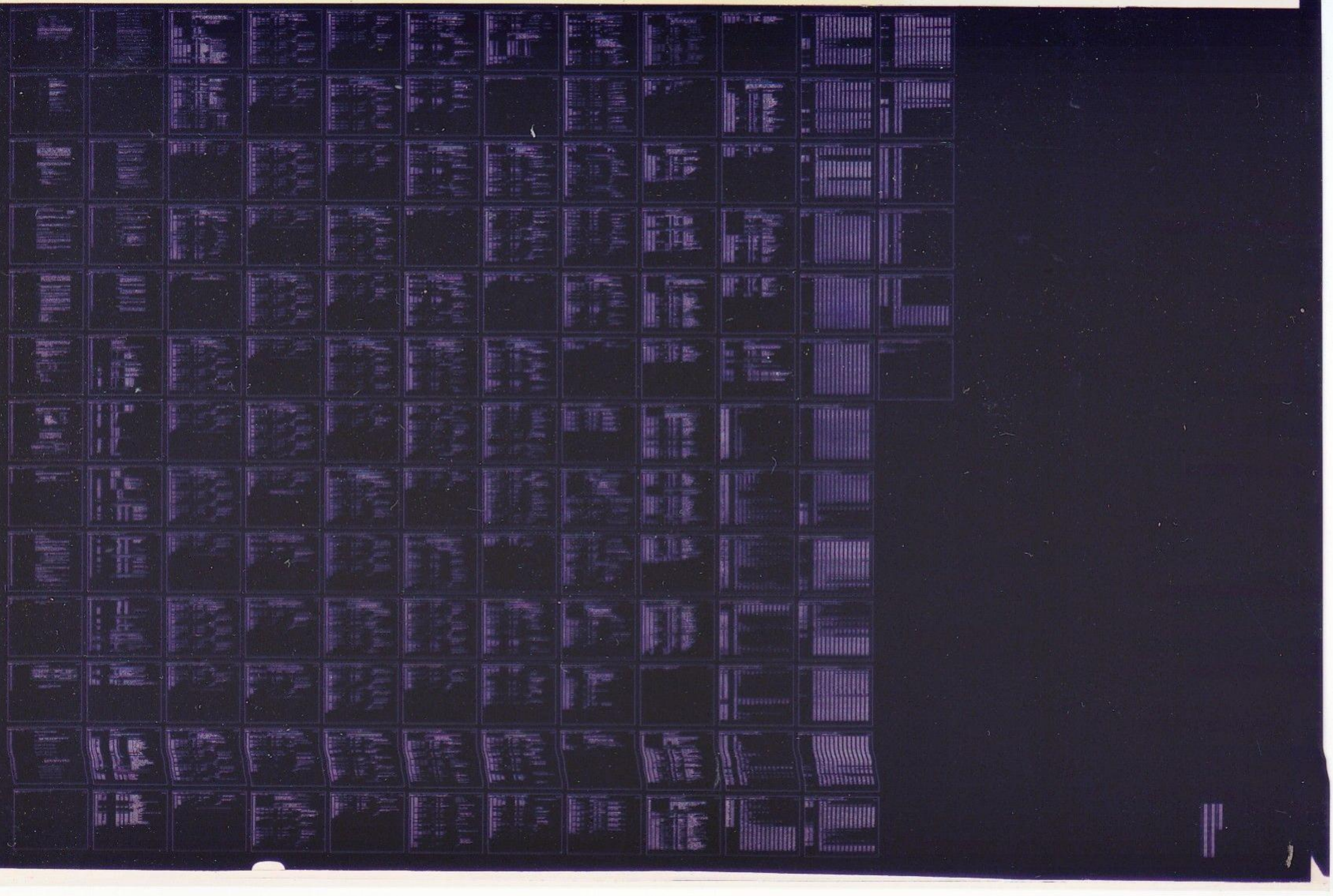
DLV11-E

OFFLINE TEST
CVDVAC0

AH B151C MC

FICHE 1 OF 1

JAN 1979
digital
MADE IN USA



```
                        .REM    @
```

IDENTIFICATION

PRODUCT CODE: AC-B150C-MC

PRODUCT TITLE: CVDVAC0 DLV11-E OFFLINE TEST

PRODUCT DATE: AUGUST, 1978

AUTHOR: ODES CHOATE

MAINTAINER: DIAGNOSTIC ENGINEERING GROUP

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
                                        H315 - CABLE TURN AROUND PLUG (OR EQUIVALENT)
                                        MODEM CABLE - BC01V-, 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)
```

```
                        1.3     RELATED DOCUMENTS AND STANDARDS.

                                DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS        175-003-009-02
                                APT                                                     MD-11-DZZMA
                                ACT                                                     AUTOCAT-11-QZAUB
                                SYSMAC                                                  MD-11-DZQAC


                        1.4     DIAGNOSTIC HIERARCHY PREREQUISITES.
                                NO SPECIAL DIAGNOSTICS ARE REQUIRED TO RUN BEFORE THIS, BUT
                                THE PROCESSOR, MEMORY, AND BUS ARE ASSUMMED TO BE FULLY
                                OPERATIONAL.


                        1.5     ASSUMPTIONS.


                                THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED
                                LOCATION '$USWR' AND '$DEVM' TO THE PROPER VALUES.
                                THE (H) JUMPER MUST BE REMOVED FROM ALL DLV11-E'S UNDER TEST.


                        2.0     OPERATING INSTRUCTIONS.
                                --------- ------------


                        2.1     LOADING AND STARTING PROCEDURES.

                                USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED
                                MEDIA.

                                THIS DIAGNOSTIC HAS ONLY ONE (1) STARTING ADDRESS.  200 FOR
                                START AND RESTART.

                                THE USER CAN SELECT A SPECIFIC TEST TO BE EXECUTED BY SETTING
                                SWITCH 8 IN THE SWITCH REGISTER AND THE TEST NUMBER (IN OCTAL)
                                IN THE LOWER BYTE.  (NOTE:  ALL TESTS PREVIOUS TO THE SELECTED
                                ONE ARE EXECUTED WITHOUT ITERATIONS.)


                        2.2     SPECIAL ENVIRONMENTS.

                                THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING
                                UDER APT,ACT,XXDP MONITORS, AS DESCRIBED IN THEIR RESPECTIVE
                                PROCEDURES MANUAL AND SYSMAC PACKAGE.
```

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)

```
                                2.4     PROGRAM OPTIONS.

                                        THIS PROGRAM WILL SUPPORT TESTING OF MULTIPLE  DLV11-E'S.   IT
                                        REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT '$BASE') AND
                                        ITS INTERRUPT VECTOR (STORED AT '$VECT1');  AND WILL  BE  ABLE
                                        TO  ADDRESS ANY DLV11-E STARTING AT THE SPECIFIED BASE ADDRESS
                                        UP TO 16 CONSECUTIVE DEVICES.
                                        EXAMPLES:       $BASE:  175610
                                                        $VECT1:    300

                                        THE PROGRAM WILL BE  ABLE  TO  TEST  ANY  DLV11-E  WITHIN  THE
                                        ADDRESS RANGE 175610 --> 176000

                                        $BASE AND $VECT1 DEFAULT TO 175610 AND 300 RESPECTIVELY.
                                        THE PROGRAM ASSOCIATES UNIT NUMBERS AS FOLLOWS:   (NUMBERS  IN
                                        PARENTHESIS ARE OCTAL)
                                        UNIT#0 -- BASE ADDRESS STORED AT '$BASE'
                                                  ASSOCIATED BASE VECTOR STORED AT '$VECT1'
                                        UNIT#1 -- BASE ADDRESS + (10)
                                                  BASE VECTOR  + (10)
                                                  !
                                                  !     UP TO
                                                  !
                                        UNIT#15 -- BASE ADDRESS + (170)
                                                   BASE VECTOR  + (170)

                                        LOCATION '$DEVM' IS USED AS A BIT MAP TO INDICATE  WHICH  UNIT
                                        NUMBERS ARE PRESENT AND WILL BE TESTED.

                                        BIT 15                      BIT 1    BIT 0
                                        ------------------------------------------
                                        !UNIT!                     !UNIT  !   UNIT!
                                        ! 15 !                     ! #1   !    #0 !
                                        ------------------------------------------

                                        A BIT MAP CAN BE ENTERED AT  '$DEVM'  PRIOR  TO  STARTING  THE
                                        PROGRAM.

                                        EXAMPLE:
                                                $BASE:  175610
                                                $VECTOR:   300
                                                $DEVM:      13

                                        THE PROGRAM WILL TEST-

                                                UNIT#0  175610  300
                                                UNIT#1  175620  310
                                                UNIT#3  175640  330
```

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.

NOTE

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

6.0 SUMMARY OF TESTS AND SPECIAL SUBROUTINES.

TEST 1 ADDRESSABILITY

THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS TO THAT ADDRESS SPACE.

THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS

TEST 2 BREAK - TCSR0 SET, CLEAR, RESET

TEST 3 MAINT - TCSR2 SET, CLEAR, RESET

TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET

TEST 5 DTR - RCSR1 SET, CLEAR

NOTE

RESET DOES NOT CLEAR THIS BIT. WE CANNOT TEST FOR AN INITIAL CONDITION AS THIS BIT IS UNDEFINED UPON POWER UP AND INIT DOESN'T AFFECT IT.

TEST 6 REQSEND - RCSR2 SET, CLEAR, RESET

THIS TEST ASSUMES THAT JUMPER FR IS IN.

TEST 7 SECXMIT - RCSR3 SET, CLEAR, RESET

TEST 10 DATAIE - RCSR5 SET, CLEAR, RESET

TEST 11 RCVRIE - RCSR6 SET, CLEAR, RESET

482

```
                        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.


                        TEST 23 DATAINT SETS WHEN RING SETS AND  THAT  DATAINT
                        ---- --
                            DOES NOT SET WHEN RING CLEARS


                        TEST 24 DATAINT SETS WHEN SECREC CHANGES STATE
                        ---- --
```

```
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
```

```
593                                     TEST 36 TEST DATA THROUGH CABLE
594                                     ---- --
595
596
597                                     TEST 37 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE
598                                     ---- --
599                                             MODE.
600
601
602                                     TEST 40 TEST BREAK GENERATION LOGIC  TRANSMIT  KNOWN  CHAR
603                                     ---- --
604                                             WITH BREAK SET AND COMPARE RECEIVED WITH 0.
605
606
607                                     TEST 41 NOT A TEST - SEND BACK TO LOOP
608                                     ---- --
609
610                                                             NOTE
611
612                                                 FOR ALL OF THE FOLLOWING ROUTINES THE  USE
613                                                 OF  (R5)  IS PART OF THE LINKAGE MECHANISM
614                                                 BETWEEN THE CALLER AND THE CALLED.
615
616
617                                     ROUTINE:TIMER
618                                     -------------
619
620                                             THIS ROUTINE IS USED TO TEST THE STATUS OF
621                                             ANY BIT IN ANY REGISTER.
622                                     INPUTS:
623                                             HOWLONG     THE  MAXIMUM  AMOUNT  OF  TIME  TO
624                                                         SPEND IN THIS ROUTINE.
625                                             WHICHBIT    A MASK WITH THE  BIT(S)  SET  THAT
626                                                         ARE TO BE CHECKED
627                                             REG         A POINTER TO THE  REGISTER  TO  BE
628                                                         CHECKED
629                                             SETCLR      THE DESIRED RESULTS -- EITHER  SET
630                                                         OR  CLEAR
631                                     OUTPUT:
632                                             THE 'C' BIT IS SET TO INDICATE AN ERROR BUT IT
633                                             IS TESTED BY THE IF.ERROR STATEMENT.
634
635
```

```
                        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-B
                        ;*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
```

```
                                                                H 2
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7

                        ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  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)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  BIT00=  1
                        .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=   R1
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=         BIT04           ;   \   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  012446                          $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)
```

```
                                .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 BYTE   --  (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
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


001254                          $ERRTB:
                                ;;              GLOBAL DATA
001254  175610                      DLADD:  DLADDR
001256  000300                      DLVEC:  300
001260  175610                      RCSR:   DLADDR  +  0
001262  175612                      RBUF:   DLADDR  +  2
001264  175614                      TCSR:   DLADDR  +  4
001266  175615                      TCSRHI: DLADDR  +  5
001270  175616                      TBUF:   DLADDR  +  6
001272  000000                      I:      0
001274  000020                      .BLKW 20    ;FOR R5 STACK
001334  000000                      R5STACK:    .WORD 0
001336                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001336  012706  001100              MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001342  005026                      CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001344  022706  001140              CMP     #SWR,R6 ;;DONE?
001350  001374                      BNE     .-6             ;;LOOP BACK IF NO
001352  012706  001100              MOV     #STACK,SP       ;;SETUP THE STACK POINTER

                                ;;INITIALIZE A FEW VECTORS
001356  012737  014370  000020      MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001364  012737  000340  000022      MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001372  012737  014170  000030      MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001400  012737  000340  000032      MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001406  012737  015322  000034      MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001414  012737  000340  000036      MOV     #340,@#TRAPVEC+2;LEVEL 7
001422  012737  012502  000024      MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001430  012737  000340  000026      MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001436  016767  010752  010742      MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
001444  005067  177510              CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001450  005067  177506              CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
001454  112767  000001  177433      MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
001462  012767  001462  177416      MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
001470  012767  001470  177412      MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS

                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001476  013746  000004              MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001502  012737  001536  000004      MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001510  012767  177570  177422      MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001516  012767  177570  177416      MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
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     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1131  001606                        67$:
1132                                .SBTTL  TYPE PROGRAM NAME
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  012446  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$:  .ASCIZ  <CRLF>*ZZ-CVDVA-C  DLV11-E OFFLINE TEST*<CRLF>
1152  001736                        68$:    .
1153  001736                                LET INITFLAG := #1
1154  001736  012767  000001  010404        MOV     #1,INITFLAG
1155  001744                        LOOP:
1156  001744                                CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
1157  001744  004767  010250                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
```

```
                                ;;*****************************************************************
                                ;*TEST 1         ADDRESSABILITY
                                ;*              THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
                                ;*              THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS
                                ;*              TO THAT ADDRESS SPACE
                                ;;*****************************************************************
002060  000004          TST1:   SCOPE
002062  012767  000002  177070          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
002070  012767  000001  177102          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002076                                                                  LET     ADRS := DLADD
002076  016701  177152                  MOV     DLADD,ADRS
                                                                        ; SET UP INTERRUPT
002102                                                                  SETVEC  #ILLMEM,#INTSRV,#PR7
002102  010146                  MOV     R1,-(SP)
002104  012701  000004          MOV     #ILLMEM,R1
002110  012721  012026          MOV     #INTSRV,(R1)+
002114  012711  000340          MOV     #PR7,(R1)
002120  012601                  MOV     (SP)+,R1
002122                                                                  LET     I := #0
002122  005067  177144          CLR     I
002126                                                                  REPEAT
002126                  $1:
002126                                                                      BGNSUB
002126  012767  002134  176754          MOV     #64$,$LPERR
002134                                                                          ;CLEAR FLAG
                                                                                LET INTFLAG := #0
002134  005067  007674          CLR     INTFLAG
                                                                        ;READ FLAG
002140  005711                  TST @ADRS
                                                                        IF INTFLAG NE #0 THEN
002142  005767  007666          TST     INTFLAG
002146  001401                  BEQ     $2
                                                                        ; FATAL ERROR
002150                                                                      ERRDF 1,,NODL
002150  104001                  ERROR   1
                                                                        ENDIF
002152                  $2:
                                                                    ENDSUB
                                                                    LET     I := I + #2
002152  062767  000002  177112          ADD     #2,I
                                                                    LET     ADRS := DLADD + I
002160  016701  177070          MOV     DLADD,ADRS
002164  066701  177102          ADD     I,ADRS
                                                                UNTIL I EQ #8.
002170  026727  177076  000010          CMP     I,#8.
002176  001353                  BNE     $1
                                                                CLRVEC ILLMEM
002200  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
002202  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
002204  012701  000004          MOV     #ILLMEM,R1
002210  010102                  MOV     R1,R2
002212  062702  000002          ADD     #2,R2
002216  010221                  MOV     R2,(R1)+
002220  005011                  CLR     (R1)
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
```

```
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
002426                                                          BGNSUB
002426  012767  002434  176454          MOV     #64$,$LPERR

002434                                                              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
002446                                                              ENDIF
002446                  $10:
002446                                                          ENDSUB

                                                ; TRY TO SET MAINT BIT
002446                                                          BGNSUB
002446  012767  002454  176434          MOV     #64$,$LPERR
002454                                                              LET     @TCSR := @TCSR SET.BY #MAINT
002454  052777  000004  176602          BIS     #MAINT,@TCSR

                                                                    ; STUCK TO 0
002462                                                              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
002474                                                              ENDIF
002474                  $11:
002474                                                          ENDSUB

                                                ; TRY TO CLEAR A SET BIT
002474                                                          BGNSUB
002474  012767  002502  176406          MOV     #64$,$LPERR

002502                                                              LET     @TCSR := @TCSR CLR.BY #MAINT
002502  042777  000004  176554          BIC     #MAINT,@TCSR

                                                                    ; SHOULD HAVE CLEARED
002510                                                              IF      #MAINT SETIN @TCSR THEN
002510  032777  000004  176546          BIT     #MAINT,@TCSR
002516  001401                          BEQ     $12

                                                                    ; MAINT DID NOT CLEAR INTCSR
                                                                    ERRHRD 10,,DIDNOT
002520  104010                          ERROR   10
002522                                                              ENDIF
002522                  $12:
002522                                                          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 SETIN @TCSR THEN
1400  002540  032777  000004  176516          BIT     #MAINT,@TCSR
1401  002546  001401                          BEQ     $13
1402                                                                  ; MAINT DID NOT RESET IN TCSR
1403  002550                                                          ERRHRD 11,,DIDNOT
1404  002550  104011                          ERROR   11
1405  002552                                                      ENDIF
1406  002552                          $13:
1407  002552                                                  ENDSUB
1408  002552                                              ENDTST
1409
1410
1411
1412                            ;;*****************************************************************
```

```
                        ;;*****************************************************************
                        ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
                        ;;*****************************************************************
002552  000004          TST4:   SCOPE
002554  012767  000010  176376          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002562  012767  000004  176410          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; USE PRIORITY OF 7
002570  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
002574  012746  002602                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
002600  000002                          RTI                     ;;POP NEW PC AND PS
002602                  64$:
                                                ; SEE IF IT IS CLEAR
002602                                                          BGNSUB
002602  012767  002610  176300          MOV     #65$,$LPERR
                                                                IF      #XMITIE SETIN @TCSR THEN
002610
002610  032777  000100  176446          BIT     #XMITIE,@TCSR
002616  001401                          BEQ     $14
                                                                ; XMITIE DID NOT RESET IN TCSR
002620                                                          ERRHRD 12,,DIDNOT
002620  104012                          ERROR   12
002622                                                          ENDIF
002622                  $14:
002622                                                          ENDSUB
                                                ; TRY TO SET XMITIE BIT
002622                                                          BGNSUB
002622  012767  002630  176260          MOV     #64$,$LPERR
                                                                LET     @TCSR := @TCSR SET.BY #XMITIE
002630
002630  052777  000100  176426          BIS     #XMITIE,@TCSR
                                                                ; STUCK TO 0
002636                                                          IF      #XMITIE NOTSETIN @TCSR THEN
002636  032777  000100  176420          BIT     #XMITIE,@TCSR
002644  001001                          BNE     $15
                                                                ; XMIT DID NOT RESET IN TCSR
002646                                                          ERRHRD 13,,DIDNOT
002646  104013                          ERROR   13
002650                                                          ENDIF
002650                  $15:
002650                                                          ENDSUB
                                                ; TRY TO CLEAR A SET BIT
002650                                                          BGNSUB
002650  012767  002656  176232          MOV     #64$,$LPERR
                                                                LET     @TCSR := @TCSR CLR.BY #XMITIE
002656
002656  042777  000100  176400          BIC     #XMITIE,@TCSR
                                                                ; SHOULD HAVE CLEARED
002664                                                          IF      #XMITIE SETIN @TCSR THEN
002664  032777  000100  176372          BIT     #XMITIE,@TCSR
002672  001401                          BEQ     $16
                                                                ;  XMIT DID NOT CLEAR IN TCSR
002674                                                          ERRHRD 14,,DIDNOT
002674  104014                          ERROR   14
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
1485  002724                                                              ; XMIT DID NOT RESET IN TCSR
1486  002724  104015                                                     ERRHRD 15,,DIDNOT
1487  002726                          ERROR   15
1488  002726                                                          ENDIF
1489  002726                          $17:
1490  002726                                                      ENDSUB
1491                                                          ENDTST
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
1552  003054                                                        ; SHOULD HAVE CLEARED IT
1553  003054  032777  000002  176176          BIT     #DTR,@RCSR    IF    #DTR SETIN @RCSR THEN
1554  003062  001401                          BEQ     $23
1555
1556  003064                                                        ; DTR DID NOT CLEAR IN RCSR
1557  003064  104020                          ERROR   20            ERRHRD 20,,DIDNOT
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 #FRFD NOTSETIN $USWR THEN
1574  003104  032767  040000  176106            BIT     #FRFD,$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
1590  003142                                                                            ; REQSEND DID NOT RESET IN RCSR
1591  003142  104021                            ERROR   21                             ERRHRD 21,,DIDNOT
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
1666  003304                                                                    ; SECXMIT DID NOT RESET IN RCSR
1667  003304  104025                            ERROR   25                      ERRHRD 25,,DIDNOT
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
1682  003332                                                                    ; SECXMIT DID NOT SET IN RCSR
1683  003332  104026                            ERROR   26                      ERRHRD 26,,DIDNOT
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
1699  003360                                                                    ; SECXMIT DID NOT CLEAR IN RCSR
1700  003360  104027                            ERROR   27                      ERRHRD 27,,DIDNOT
1701  003362                                                                ENDIF
1702  003362                                $33:
1703  003362                                                            ENDSUB
1704
1705  003362                                                            BGNSUB
1706  003362  012767  003370  175520            MOV     #0'$,$LPERR
1707                                                          ; NOW SEE IF RESET CLEARS IT
```

```
1708
1709  003370
1710  003370  052777  000010  175662          BIS     #SECXMIT,@RCSR                  LET     @RCSR := @RCSR SET.BY #SECXMIT
1711                                                                                  ; ISSUE BUS RESET
1712  003376                                                                          BRESET
1713  003376  000005                          RESET                                  IF      #SECXMIT SETIN @RCSR THEN
1714  003400
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                          ;;*****************************************************************
```

```
                                ;;*****************************************************************
                                ;*TEST 10        DATAIE - RCSR5 SET, CLEAR, RESET
                                ;;*****************************************************************
003412  000004                  TST10:  SCOPE
003414  012767  000010  175536          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003422  012767  000010  175550          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                        ; SEE IF IT IS CLEAR
                                                                BGNSUB
003430  012767  003436  175452          MOV     #64$,$LPERR
                                                                IF      #DATAIE SETIN @RCSR THEN
003436  032777  000040  175614          BIT     #DATAIE,@RCSR
003444  001401                          BEQ     $35
                                                                    ; DATAIE DID NOT RESET IN RCSR
                                                                    ERRHRD 31,,DIDNOT
003446  104031                          ERROR   31
                                                                ENDIF
003450                          $35:
                                                            ENDSUB
                                        ; TRY TO SET DATAIE BIT
                                                                BGNSUB
003450  012767  003456  175432          MOV     #64$,$LPERR
                                                                LET     @RCSR := @RCSR SET.BY #DATAIE
003456  052777  000040  175574          BIS     #DATAIE,@RCSR
                                                                ; STUCK TO 0
                                                                IF      #DATAIE NOTSETIN @RCSR THEN
003464  032777  000040  175566          BIT     #DATAIE,@RCSR
003472  001001                          BNE     $36
                                                                    ; DATAIE DID NOT SET IN RCSR
                                                                    ERRHRD 32,,DIDNOT
003474  104032                          ERROR   32
                                                                ENDIF
003476                          $36:
                                                            ENDSUB
                                        ; TRY TO CLEAR A SET BIT
                                                                BGNSUB
003476  012767  003504  175404          MOV     #64$,$LPERR
                                                                LET     @RCSR := @RCSR CLR.BY #DATAIE
003504  042777  000040  175546          BIC     #DATAIE,@RCSR
                                                                ; SHOULD HAVE CLEARED
                                                                IF      #DATAIE SETIN @RCSR THEN
003512  032777  000040  175540          BIT     #DATAIE,@RCSR
003520  001401                          BEQ     $37
                                                                    ; DATAIE DID NOT CLEAR IN RCSR
                                                                    ERRHRD 33,,DIDNOT
003522  104033                          ERROR   33
                                                                ENDIF
003524                          $37:
                                                            ENDSUB
                                        ; NOW SEE IF RESET CLEARS IT
                                                                BGNSUB
003524  012767  003532  175356          MOV     #64$,$LPERR
```

```
1784
1785  003532
1786  003532  052777  000040  175520          BIS     #DATAIE,@RCSR                   LET     @RCSR := @RCSR SET.BY #DATAIE
1787
1788  003540                                                                          ; ISSUE BUS RESET
1789  003540  000005                          RESET                                  BRESET
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,$TIMES      ;;DO 10 ITERATIONS
1809  003564  012767  00001   175406            MOV     #11,$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
1862  003674  052777  000100  175356          BIS     #RCVRIE,@RCSR                   LET     @RCSR := @RCSR SET.BY #RCVRIE
1863                                                                                      ; ISSUE BUS RESET
1864  003702                                                                            BRESET
1865  003702  000005                          RESET                                    IF      #RCVRIE SETIN @RCSR THEN
1866  003704
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
```

```
                                ;;*****************************************************************
                                ;*TEST 13        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
                                ;;*****************************************************************
003756  000004                  TST13:  SCOPE
003760  012767  000010  175172          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003766  012767  000013  175204          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


003774                                                          BGNSUB
003774  012767  004002  175106          MOV     #64$,$LPERR

004002                                                              IF      #XMITRDY NOTSETIN @TCSR THEN
004002  032777  000200  175254          BIT     #XMITRDY,@TCSR
004010  001002                        BNE     $46

                                                                    ;RESET SHOULD HAVE SET BIT.
                                                                    ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
004012                                                              ERRHRD 42,HRESET,DIDNOT
004012  104042                        ERROR   42
                                                                    ;ISSUE ANOTHER RESET
004014                                                              BRESET
004014  000005                        RESET
004016                                                          ENDIF
004016                          $46:
                                                                    ;ALLOW LOOPING ON ERROR
004016                                                          CKLOOP
004016                                                      ENDSUB
004016                                                  ENDTST
```

```
                        ;;*****************************************************************
                        ;*TEST 14        TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.
                        ;;*****************************************************************
004016  000004          TST14:  SCOPE
004020  012767  000010  175132          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004026  012767  000014  175144          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


004034                                                  BGNSUB
004034  012767  004042  175046          MOV     #64$,$LPERR
004042                                          IF      #DATAINT SETIN @RCSR THEN
004042  032777  100000  175210          BIT     #DATAINT,@RCSR
004050  001402                          BEQ     $47

004052                                              ERRHRD 43, HRESET, DIDNOT
004052  104043                          ERROR   43

                                                    ;TESTING EFFECT OF RESET ON BIT

                                                    ;DATAINT DID NOT CLEAR IN RCSR
                                                    ;ALLOW A FRESH START
004054                                              BRESET
004054  000005                          RESET
004056                                          ENDIF
004056                  $47:
004056                                          CKLOOP
004056                                      ENDSUB
004056                                  ENDTST


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

```
                        ;;*****************************************************************
                        ;*TEST 15        TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT
                        ;;*****************************************************************
004056  000004          TST15:  SCOPE
004060  012767  000010  175072          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004066  012767  000015  175104          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

004074                                                          IF      #CABLE NOTSETIN $USWR THEN
004074  032767  020000  175116          BIT     #CABLE,$USWR
004102  001004                        BNE     $50
                                                                    ; CAN'T TEST WITHOUT BERG OR H315.
                                                                EXIT TST
004104  012767  000001  175046          MOV     #1,$TIMES
004112  000411                        BR      TST16           ;;;EXIT THIS TEST
004114                                                          ENDIF
004114                  $50:

                                                                BGNSUB
004114  012767  004122  174766          MOV     #64$,$LPERR

004122                                                              IF      #RCVRACT SETIN @RCSR THEN
004122  032777  004000  175130          BIT     #RCVRACT,@RCSR
004130  001402                        BEQ     $51

                                                                    ;RESET SHOULD HAVE CLEARED RCVRACT
                                                                    ERRHRD 44, HRESET, DIDNOT
004132  104044                        ERROR   44

                                                                    ;TESTING EFFECT OF RESET ON BIT

                                                                    ;RCVRACT DID NOT CLEAR IN RCSR

                                                                    ;ALLOW ANOTHER TRY
004134                                                              BRESET
004134  000005                        RESET
004136                                                          ENDIF
004136                  $51:

                                                                    ;ALLOW LOOPING ON ERROR
004136                                                          CKLOOP
004136                                                      ENDSUB
004136                                                  ENDTST
```

```
                        ;;*****************************************************************
                        ;* THE FOLLOWING 4 TESTS VERIFY
                        ;* THAT THE EIA SIGNALS CAN BE TRANSMITTED
                        ;* AND RECEIVED THROUGH THE CABLE
                        ;;*****************************************************************



                        ;;*****************************************************************
                        ;*TEST 16        TEST THAT CARDET SETS AND CLEARS
                        ;*              AS  DTR   SETS AND CLEARS
                        ;*              THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
                        ;;*****************************************************************
004136  000004          TST16:  SCOPE
004140  012767  000010  175012          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004146  012767  000016  175024          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; CAN WE USE THE WRAPAROUND??
                                                        IF      #CABLE+FRFD NOTSETIN $USWR THEN
004154
004154  032767  060000  175036          BIT     #CABLE+FRFD,$USWR
004162  001004                          BNE     $52
                                                                ; CAN'T TEST WITHOUT BERG OR H315
                                                                ; OR WITH (-FD) JUMPER OUT.
                                                                ; OR WITH (-FR) JUMPER OUT.
                                                        EXIT TST
004164
004164  012767  000001  174766          MOV     #1,$TIMES
004172  000441                          BR      TST17           ;;;EXIT THIS TEST
                                                        ENDIF
004174
004174                  $52:
                                                                ; DTR AND
                                                                ; CARDET ARE CONNECTED

                                                ; BY THE H315 OR EQUIV.


                                                ; CLEAR
                                                        BGNSUB
004174
004174  012767  004202  174706          MOV     #64$,$LPERR

                                                                ; CLEAR  DTR
                                                    LET     @RCSR := @RCSR CLR.BY #DTR
004202
004202  042777  000002  175050          BIC     #DTR,@RCSR
                                                                ; CARDET SHOULD FOLLOW
004210                                                  IF      #CARDET SETIN @RCSR THEN
004210  032777  010000  175042          BIT     #CARDET,@RCSR
004216  001401                          BEQ     $53

                                                                ; CARDET DID NOT
                                                                ERRHRD 45,,FORCE
004220
004220  104045                          ERROR   45

                                                                ;  CLEAR WITH DTR
                                                    ENDIF
004222
004222                  $53:
004222                                                  ENDSUB


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

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*TEST 17        TEST THAT CLRSEND SETS AND CLEARS
                        ;*              AS  DTR   SETS AND CLEARS
                        ;*              (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK
                        ;;*****************************************************************
004276  000004          TST17:  SCOPE
004300  012767  000010  174652          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004306  012767  000017  174664          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
004314                                                          IF      #CABLE+FRFD NOTSETIN $USWR THEN
004314  032767  060000  174676          BIT     #CABLE+FRFD,$USWR
004322  001004                          BNE     $56
                                                                        ; CAN'T TEST WITHOUT BERG OR H315
004324                                                              EXIT TST
004324  012767  000001  174626          MOV     #1,$TIMES
004332  000441                          BR      TST20           ;;;EXIT THIS TEST
004334                                                          ENDIF
004334                          $56:
                                                                        ; DTR AND
                                                                        ; CLRSEND ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

                                                ; CLEAR
004334                                                          BGNSUB
004334  012767  004342  174546          MOV     #64$,$LPERR

                                                                    ; CLEAR  DTR
004342                                                          LET     @RCSR := @RCSR CLR.BY #DTR
004342  042777  000002  174710          BIC     #DTR,@RCSR

                                                                    ; CLRSEND SHOULD FOLLOW
004350                                                          IF      #CLRSEND SETIN @RCSR THEN
004350  032777  020000  174702          BIT     #CLRSEND,@RCSR
004356  001401                          BEQ     $57

                                                                    ; CLRSEND DID NOT
004360                                                          ERRHRD 50,,FORCE
004360  104050                          ERROR   50

                                                                    ;  CLEAR WITH DTR
004362                                                          ENDIF
004362                          $57:
004362                                                  ENDSUB

                                                ; SET
004362                                                  BGNSUB
004362  012767  004370  174520          MOV     #64$,$LPERR

                                                                    ; SET DTR
004370                                                          LET     @RCSR := @RCSR SET.BY #DTR
004370  052777  000002  174662          BIS     #DTR,@RCSR

                                                                    ; CLRSEND SHOULD FOLLOW
004376                                                          IF      #CLRSEND NOTSETIN @RCSR THEN
004376  032777  020000  174654          BIT     #CLRSEND,@RCSR
004404  001001                          BNE     $60
```

```
                                                                ; CLRSEND DID NOT SET
                                                                ERRHRD 51,,FORCE
004406
004406  104051                        ERROR   51
                                                                ;  WITH DTR
                                                            ENDIF
004410
004410                          $60:                        ENDSUB
004410
                                                                ; CLEAR
                                                            BGNSUB
004410
004410  012767  004416  174472        MOV     #64$,$LPERR
                                                                ; CLEAR  DTR
004416                                                      LET     @RCSR := @RCSR CLR.BY #DTR
004416  042777  000002  174634        BIC     #DTR,@RCSR
                                                                ; CLRSEND SHOULD FOLLOW
004424                                                      IF      #CLRSEND SETIN @RCSR THEN
004424  032777  020000  174626        BIT     #CLRSEND,@RCSR
004432  001401                        BEQ     $61
                                                                ; CLRSEND DID NOT
004434                                                          ERRHRD 52,,FORCE
004434  104052                        ERROR   52
                                                                ;  CLEAR WITH DTR
004436                                                      ENDIF
004436                          $61:
004436                                                  ENDSUB
004436                                              ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 20        TEST THAT RING SETS AND CLEARS
                                ;*              AS  REQSEND    SETS AND CLEARS
                                ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
                                ;;*****************************************************************
004436  000004                  TST20:  SCOPE
004440  012767  000010  174512          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004446  012767  000020  174524          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
004454                                                          IF      #CABLE+FRFD NOTSETIN $USWR THEN
004454  032767  060000  174536          BIT     #CABLE+FRFD,$USWR
004462  001004                          BNE     $62
                                                                        ; CAN'T TEST WITHOUT BERG OR H315
                                                                        ; OR WITH (-FR) JUMPER OUT.
004464                                                              EXIT TST
004464  012767  000001  174466          MOV     #1,$TIMES
004472  000441                          BR      TST21           ;;;EXIT THIS TEST
004474                                                          ENDIF
004474                          $62:
                                                                        ; REQSEND AND
                                                                        ; RING ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

                                        ; CLEAR
004474                                                          BGNSUB
004474  012767  004502  174406          MOV     #64$,$LPERR

                                                                    ; CLEAR  REQSEND
004502                                                          LET     @RCSR := @RCSR CLR.BY #REQSEND
004502  042777  000004  174550          BIC     #REQSEND,@RCSR

                                                                    ; RING SHOULD FOLLOW
004510                                                          IF      #RING SETIN @RCSR THEN
004510  032777  040000  174542          BIT     #RING,@RCSR
004516  001401                          BEQ     $63

                                                                    ; RING DID NOT
004520                                                          ERRHRD 53,,FORCE
004520  104053                          ERROR   53

                                                                    ;  CLEAR WITH REQSEND
004522                                                          ENDIF
004522                          $63:
004522                                                  ENDSUB

                                                : SET
004522                                                  BGNSUB
004522  012767  004530  174360          MOV     #64$,$LPERR

                                                                    ; SET REQSEND
004530                                                          LET     @RCSR := @RCSR SET.BY #REQSEND
004530  052777  000004  174522          BIS     #REQSEND,@RCSR

                                                                    ; RING SHOULD FOLLOW
004536                                                          IF      #RING NOTSETIN @RCSR THEN
004536  032777  040000  174514          BIT     #RING,@RCSR
```

```
2275  004544  001001                    BNE     $64
2276
2277  004546                                                          ; RING DID NOT SET
2278  004546  104054                    ERROR   54                   ERRHRD 54,,FORCE
2279
2280                                                                  ;  WITH REQSEND
2281  004550                                                        ENDIF
2282  004550                          $64:
2283  004550                                                      ENDSUB
2284
2285                                                                  ; CLEAR
2286  004550                                                      BGNSUB
2287  004550  012767  004556  174332    MOV     #64$,$LPERR
2288
2289                                                                  ; CLEAR  REQSEND
2290  004556                                                        LET     @RCSR := @RCSR CLR.BY #REQSEND
2291  004556  042777  000004  174474    BIC     #REQSEND,@RCSR
2292                                                                  ; RING SHOULD FOLLOW
2293  004564                                                        IF      #RING SETIN @RCSR THEN
2294  004564  032777  040000  174466    BIT     #RING,@RCSR
2295  004572  001401                    BEQ     $65
2296                                                                  ; RING DID NOT
2297  004574                                                        ERRHRD 55,,FORCE
2298  004574  104055                    ERROR   55
2299
2300                                                                  ;  CLEAR WITH REQSEND
2301  004576                                                        ENDIF
2302  004576                          $65:
2303  004576                                                      ENDSUB
2304  004576                                                    ENDTST
2305
2306
2307
2308
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 21        TEST THAT SECREC SETS AND CLEARS
                                ;*              AS  SECXMIT    SETS AND CLEARS
                                ;;*****************************************************************
004576  000004                  TST21:  SCOPE
004600  012767  000010  174352          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004606  012767  000021  174364          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
                                                                IF      #CABLE NOTSETIN $USWR THEN
004614
004614  032767  020000  174376          BIT     #CABLE,$USWR
004622  001004                        BNE     $66
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
                                                                    EXIT TST
004624
004624  012767  000001  174326          MOV     #1,$TIMES
004632  000441                        BR      TST22           ;;;EXIT THIS TEST
                                                                ENDIF
004634
004634                          $66:
                                                                        ; SECXMIT AND
                                                                        ; SECREC ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

                                                ; CLEAR
004634                                                          BGNSUB
004634  012767  004642  174246          MOV     #64$,$LPERR

                                                                        ; CLEAR  SECXMIT
004642                                                              LET     @RCSR := @RCSR CLR.BY #SECXMIT
004642  042777  000010  174410          BIC     #SECXMIT,@RCSR

                                                                    ; SECREC SHOULD FOLLOW
004650                                                          IF      #SECREC SETIN @RCSR THEN
004650  032777  002000  174402          BIT     #SECREC,@RCSR
004656  001401                        BEQ     $67

                                                                    ; SECREC DID NOT
                                                                    ERRHRD 56,,FORCE
004660
004660  104056                        ERROR   56

                                                                    ;  CLEAR WITH SECXMIT
                                                                ENDIF
004662
004662                          $67:                            ENDSUB
004662

                                                ; SET
004662                                                          BGNSUB
004662  012767  004670  174220          MOV     #64$,$LPERR

                                                                    ; SET SECXMIT
                                                                LET     @RCSR := @RCSR SET.BY #SECXMIT
004670  052777  000010  174362          BIS     #SECXMIT,@RCSR

                                                                    ; SECREC SHOULD FOLLOW
004676                                                          IF      #SECREC NOTSETIN @RCSR THEN
004676  032777  002000  174354          BIT     #SECREC,@RCSR
004704  001001                        BNE     $70

                                                                    ; SECREC DID NOT SET
```

```
2365  004706
2366  004706  104057                        ERROR   57
2367
2368
2369  004710
2370  004710                          $70:
2371  004710
2372
2373                                                    ; CLEAR
2374  004710
2375  004710  012767  004716  174172        MOV .   #64$,$LPERR
2376
2377
2378  004716
2379  004716  042777  000010  174334        BIC     #SECXMIT,@RCSR
2380
2381  004724
2382  004724  032777  002000  174326        BIT     #SECREC,@RCSR
2383  004732  001401                        BEQ     $71
2384
2385  004734
2386  004734  104060                        ERROR   60
2387
2388
2389  004736
2390  004736                          $71:
2391  004736
2392  004736
2393
2394
2395
2396
```

```
                ERRHRD 57,,FORCE

            ; WITH SECXMIT
        ENDIF

    ENDSUB

    BGNSUB

        ; CLEAR  SECXMIT
        LET     @RCSR := @RCSR CLR.BY #SECXMIT

        ; SECREC SHOULD FOLLOW
        IF      #SECREC SETIN @RCSR THEN

            ; SECREC DID NOT
            ERRHRD 60,,FORCE

            ; CLEAR WITH SECXMIT
        ENDIF

    ENDSUB
ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 22         TEST THAT 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
                                ;*               THE (-FD) JUMPER MUST BE IN FOR THIS TEST.
                                ;;*****************************************************************
004736  000004                TST22:  SCOPE
004740  012767  000010  174212        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004746  012767  000022  174224        MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; CAN WE USE THE WRAPAROUND??
004754                                                      IF      #CABLE+FRFD NOTSETIN $USWR THEN
004754  032767  060000  174236        BIT     #CABLE+FRFD,$USWR
004762  001004                      BNE     $72
                                                                ; CAN'T TEST WITHOUT BERG OR H315
                                                                ; OR WITH (-FD) JUMPER OUT.
004764                                                          EXIT TST
004764  012767  000001  174166        MOV     #1,$TIMES
004772  000463                      BR      TST23           ;;;EXIT THIS TEST
004774                                                      ENDIF
004774                          $72:
                                                                ;MAKE SURE NOTHING UNEXPECTED HAPPENS
004774  012746  000340              MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
005000  012746  005006              MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005004  000002                      RTI                     ;;POP NEW PC AND PS
005006                          64$:
                                                        ;READ TWICE - CLEARS
005006                                                      BGNSUB
005006  012767  005014  174074        MOV     #65$,$LPERR
                                                                ; CLEAR DTR
                                                            LET     @RCSR := @RCSR CLR.BY #DTR
005014  042777  000002  174236        BIC     #DTR,@RCSR
                                                                ;WAIT 1 MILLI-SEC FOR CABLE
                                                            WAITMS  1
005022  010546                      MOV     R5,-(SP)
005024  012745  000001              MOV     #1,-(R5)
005030  004767  004712              JSR     PC,WAIT
005034  012605                      MOV     (SP)+,R5
                                                                ; READ RCSR - TO CLEAR DATAINT
005036                                                      LET     R3 := @RCSR
005036  017703  174216              MOV     @RCSR,R3
                                                                ; READ RCSR AGAIN
005042                                                      IF      #DATAINT SETIN @RCSR THEN
005042  032777  100000  174210        BIT     #DATAINT,@RCSR
005050  001401                      BEQ     $73
                                                                ; READING RCSR DID NOT CLEAR DATAINT
005052                                                          ERRHRD 61,EDATAINT
005052  104061                      ERROR   61
005054                                                      ENDIF
005054                          $73:
005054                                                  ENDSUB
```

```
                                                    ;  DTR SETTING SETS DATAINT
                                                                        BGNSUB
005054
005054  012767  005062  174026          MOV     #64$,$LPERR
                                                        ;SET DTR
                                                    LET     @RCSR  := @RCSR SET.BY #DTR
005062
005062  052777  000002  174170          BIS     #DTR,@RCSR
                                                    IF      #DATAINT NOTSETIN @RCSR THEN
005070
005070  032777  100000  174162          BIT     #DATAINT,@RCSR
005076  001001                          BNE     $74
                                                        ;SETTING DTR DID NOT SET DATAINT
                                                        ERRHRD 62,, E2DATA
005100
005100  104062                          ERROR   62
                                                    ENDIF
005102
005102                          $74:
                                                    IF      #DATAINT SETIN @RCSR THEN
005102
005102  032777  100000  174150          BIT     #DATAINT,@RCSR
005110  001401                          BEQ     $75
                                                        ;READING RCSR DID NOT CLEAR DATAINT
                                                        ERRHRD 63,EDATAINT
005112
005112  104063                          ERROR   63
                                                    ENDIF
005114
005114                          $75:
                                                                        ENDSUB
                                                    ; DTR CLEARING SETS DATAINT
                                                                        BGNSUB
005114
005114  012767  005122  173766          MOV     #64$,$LPERR
                                                        ;CLEAR DTR
005122                                              LET     @RCSR := @RCSR CLR.BY #DTR
005122  042777  000002  174130          BIC     #DTR,@RCSR
                                                    IF      #DATAINT NOTSETIN @RCSR THEN
005130
005130  032777  100000  174122          BIT     #DATAINT,@RCSR
005136  001001                          BNE     $76
                                                        ;CLEARING DTR DID NOT SET DATAINT
                                                        ERRHRD 64,,E2DATA
005140
005140  104064                          ERROR   64
                                                    ENDIF
005142
005142                          $76:
                                                                ENDSUB
005142                                          ENDTST
```

```
2499                                    ;;*****************************************************************
2500                                    ;;*****************************************************************
2501                                    ;*TEST 23        TEST THAT DATAINT SETS WHEN RING SETS
2502                                    ;*              AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
2503                                    ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
2504                                    ;;*****************************************************************
2505  005142  000004                   TST23:  SCOPE
2506  005144  012767  000010  174006           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2507  005152  012767  000023  174020           MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2508                                                                           ; CAN WE USE THE WRAPAROUND??
2509  005160                                                           IF      #CABLE+FRFD NOTSETIN $USWR THEN
2510  005160  032767  060000  174032           BIT     #CABLE+FRFD,$USWR
2511  005166  001004                           BNE     $77
2512                                                                           ; CAN'T TEST WITHOUT BERG OR H315
2513                                                                           ; OR WITH (-FR) JUMPER OUT.
2514  005170                                                           EXIT TST
2515  005170  012767  000001  173762           MOV     #1,$TIMES
2516  005176  000473                           BR      TST24           ;;;EXIT THIS.TEST
2517  005200                                                           ENDIF
2518  005200                           $77:
2519
2520                                                                           ;NO INTERRUPTS
2521  005200  012746  000340                   MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
2522  005204  012746  005212                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
2523  005210  000002                           RTI                     ;;POP NEW PC AND PS
2524  005212                           64$:
2525
2526                                                    ;START OFF WITH EVERYTHING CLEAR
2527  005212                                                           BGNSUB
2528  005212  012767  005220  173670           MOV     #65$,$LPERR
2529
2530                                                                           ;CLEAR RING
2531  005220                                                           LET     @RCSR := @RCSR CLR.BY #REQSEND
2532  005220  042777  000004  174032           BIC     #REQSEND,@RCSR
2533                                                                           ;WAIT 1 MILLI-SEC FOR CABLE
2534  005226                                                           WAITMS  1
2535  005226  010546                           MOV     R5,-(SP)
2536  005230  012745  000001                   MOV     #1,-(R5)
2537  005234  004767  004506                   JSR     PC,WAIT
2538  005240  012605                           MOV     (SP)+,R5
2539
2540  005242                                                                   ;READ ONCE
2541  005242  017703  174012                   MOV     @RCSR,R3        LET     R3 := @RCSR
2542
2543  005246                                                                   ;READ TWICE
2544  005246  032777  100000  174004           BIT     #DATAINT,@RCSR  IF      #DATAINT SETIN @RCSR THEN
2545  005254  001401                           BEQ     $100
2546
2547  005256                                                                   ;READING RCSR DID NOT CLEAR DATAINT
2548  005256  104065                           ERROR   65              ERRHRD 65, EDATAINT
2549  005260
2550  005260                           $100:                           ENDIF
2551  005260                                                   ENDSUB
2552
2553                                                   ;       SET RING --> SET DATAINT
2554  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  004440                  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  004376                  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
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 24        TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE
                                ;;*****************************************************************
005366  000004                  TST24:  SCOPE
005370  012767  000010  173562          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005376  012767  000024  173574          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
005404                                                          IF      #CABLE NOTSETIN $USWR THEN
005404  032767  020000  173606          BIT     #CABLE,$USWR
005412  001004                          BNE     $103
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
005414                                                              EXIT TST
005414  012767  000001  173536          MOV     #1,$TIMES
005422  000454                          BR      TST25           ;;;EXIT THIS TEST
005424                                                          ENDIF
005424                          $103:
                                                                        ;NO INTERRUPTS
005424  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
005430  012746  005436                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005434  000002                          RTI                     ;;POP NEW PC AND PS
005436                          64$:
                                                                ;START FRESH
                                                                        ;CLEAR SECREC
005436                                                          LET     @RCSR := @RCSR CLR.BY #SECXMIT
005436  042777  000010  173614          BIC     #SECXMIT,@RCSR
005444                                                          LET     R3 := @RCSR
005444  017703  173610                  MOV     @RCSR,R3
                                                                ;SET SECREC --> DATAINT SET
005450                                                          BGNSUB
005450  012767  005456  173432          MOV     #65$,$LPERR
                                                                        ;SET SECREC
005456                                                              LET     @RCSR := @RCSR SET.BY #SECXMIT
005456  052777  000010  173574          BIS     #SECXMIT,@RCSR
                                                                        ;WAIT 1 MILLI-SEC FOR CABLE
005464                                                              WAITMS  1
005464  010546                          MOV     R5,-(SP)
005466  012745  000001                  MOV     #1,-(R5)
005472  004767  004250                  JSR     PC,WAIT
005476  012605                          MOV     (SP)+,R5
005500                                                          IF      #DATAINT NOTSETIN @RCSR THEN
005500  032777  100000  173552          BIT     #DATAINT,@RCSR
005506  001001                          BNE     $104
                                                                        ;SETTING SECREC DID NOT SET DATAINT
005510                                                                  ERRHRD 124,, E2DATA
005510  104124                          ERROR   124
005512                                                              ENDIF
005512                          $104:
                                                                ENDSUB
                                ;CLEAR SECREC --> DATAINT SET
```

```
2661  005512
2662  005512  012767  005520  173370          MOV     #64$,$LPERR
2663
2664  005520
2665  005520  042777  000010  173532          BIC     #SECXMIT,@RCSR
2666
2667  005526
2668  005526  010546                          MOV     R5,-(SP)
2669  005530  012745  000001                  MOV     #1,-(R5)
2670  005534  004767  004206                  JSR     PC,WAIT
2671  005540  012605                          MOV     (SP)+,R5
2672  005542
2673  005542  032777  100000  173510          BIT     #DATAINT,@RCSR
2674  005550  001001                          BNE     $105
2675
2676  005552
2677  005552  104125                          ERROR   125
2678  005554
2679  005554                          $105:
2680  005554
2681  005554
2682
2683
2684
```

```
BGNSUB

    ;CLEAR SECREC
LET     @RCSR := @RCSR CLR.BY #SECXMIT

    ;WAIT 1 MILLI-SEC FOR CABLE
WAITMS  1



IF      #DATAINT NOTSETIN @RCSR THEN

    ;CLEARING SECREC DID NOT SET DATAINT
    ERRHRD 125,, E2DATA

ENDIF

ENDSUB
ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 25        TEST THAT XMIT RDY - TCSR 7 - CLEARS
                                ;*              WHEN TBUF IS LOADED WITH A CHARACTER
                                ;*              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
                                ;;*****************************************************************
005554  000004                TST25:  SCOPE
005556  012767  000001  173374        MOV     #1,$TIMES       ;;DO 1 ITERATION
005564  012767  000025  173406        MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
005572                                          IF #APTENV SETIN $ENV THEN
005572  032767  000001  173414        BIT     #APTENV,$ENV
005600  001404                        BEQ     $106
005602                                                  EXIT TEST
005602  012767  000001  173350        MOV     #1,$TIMES
005610  000454                        BR      TST26           ;;;EXIT THIS TEST
005612                                          ENDIF
005612                          $106:
005612                                                  BGNSUB
005612  012767  005620  173270        MOV     #64$,$LPERR
                                            ; LOAD TBUF WITH ONE CHARACTER
                                            ; WAIT FOR READY TO SET
                                            ; (SHOULD BE VERY SHORT WAIT
                                            ; SINCE UART DOUBLE BUFFERS ITS INPUT)

                                                        ;SEND A CHARACTER
005620                                                LET @TBUF :B= #0
005620  105077  173444                CLRB    @TBUF

                                                        ;WAIT A MAXIMUM
                                                        ;OF 50 MSEC FOR
                                                        ;XMIT RDY TO SET IN TCSR
005624                                                CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
005624  010546                        MOV     R5,-(SP)
005626  012745  177777                MOV     #SET,-(R5)
005632  016745  173426                MOV     TCSR,-(R5)
005636  012745  000200                MOV     #XMITRDY,-(R5)
005642  012745  000500                MOV     #500,-(R5)
005646  004767  003616                JSR     PC,TIMER
005652  012605                        MOV     (SP)+,R5

                                                        ;TIMER RETURNS AN ERROR IF BIT DID
                                                        ;NOT MEET CONDITION WITHIN TIME LIMIT
005654                                                IF.ERROR THEN
005654  103001                        BCC     $107

                                                        ;XMIT RDY DID NOT SET IN TCSR
005656                                                  ERRHRD 66,,DIDNOT
005656  104066                        ERROR   66
005660                                                ENDIF
005660                          $107:
005660                                                ENDSUB

005660                                                BGNSUB
005660  012767  005666  173222        MOV     #64$,$LPERR
                                            ; LOAD TBUF WITH A SECOND CHARACTER
                                            ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
                                            ; AND THEN WAIT FOR IT TO SET
```

```
2741
2742
2743  005666
2744  005666  105077  173376          CLRB    @TBUF                    ;SEND SECOND CHARACTER
                                                                      LET @TBUF :B= #0
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
2752  005704                                                              ; XMITRDY DID NOT CLEAR IN TCSR
2753  005704  104067                    ERROR   67                      ERRHRD 67,,DIDNOT
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 <#500,#XMITRDY,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  000500            MOV     #500,-(R5)
2766  005730  004767  003534            JSR     PC,TIMER
2767  005734  012605                    MOV     (SP)+,R5
2768  005736
2769  005736  103001                    BCC     $111                    IF.ERROR THEN
2770
2771  005740                                                              ;XMIT RDY DID NOT SET IN TCSR
2772  005740  104070                    ERROR   70                      ERRHRD 70,,DIDNOT
2773  005742                                                          ENDIF
2774  005742                        $111:
2775  005742                                                      ENDSUB
2776  005742                                                  ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 26        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
                                ;*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
                                ;*              AND THAT RESET CLEARS THE BIT.
                                ;;*****************************************************************
005742  000004                 TST26:  SCOPE
005744  012767  000010  173206         MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005752  012767  000026  173220         MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                                        ; SET THE MAINTENANCE BIT
005760                                                          LET @TCSR := @TCSR SET.BY #MAINT
005760  052777  000004  173276         BIS     #MAINT,@TCSR

005766                                                          BGNSUB
005766  012767  005774  173114         MOV     #64$,$LPERR
                                            ; SEND A CHARACTER AND LET IT WRAP AROUND

005774                                                                  LET @TBUF :B= #0
005774  105077  173270                 CLRB    @TBUF

                                                                        ; WAIT A MAXIMUM OF 50 MSEC
                                                                        ; FOR RCVR DONE TO SET IN
                                                                        ; RCSR
006000                                                                  CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
006000  010546                         MOV     R5,-(SP)
006002  012745  177777                 MOV     #SET,-(R5)
006006  016745  173246                 MOV     RCSR,-(R5)
006012  012745  000200                 MOV     #RCVRDONE,-(R5)
006016  012745  000500                 MOV     #500,-(R5)
006022  004767  003442                 JSR     PC,TIMER
006026  012605                         MOV     (SP)+,R5

                                                                            ;DIDN'T SET IN TIME
006030                                                                  IF.ERROR THEN
006030  103001                         BCC     $112

                                                                            ; RCVRDONE DID NOT SET IN RCSR
006032                                                                      ERRHRD 71,,DIDNOT
006032  104071                         ERROR   71
006034                                                                  ENDIF
006034                         $112:

006034                                                          ENDSUB

006034                                                          BGNSUB
006034  012767  006042  173046         MOV     #64$,$LPERR
                                            ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
                                            ; THIS ALSO WILL CLEAR THE MAINT. BIT
006042                                                                  BRESET
006042  000005                         RESET

006044                                                                  IF #RCVRDONE SETIN @RCSR THEN
006044  032777  000200  173206         BIT     #RCVRDONE,@RCSR
006052  001401                         BEQ     $113

                                                                            ; RCVRDONE DID NOT RESET IN RCSR.
006054                                                                      ERRHRD 72,,DIDNOT
```

```
2833  006054  104072                    ERROR   72
2834  006056                                                  ENDIF
2835  006056                  $113:
2836  006056                                                ENDSUB
2837  006056                                              ENDTST
```

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*TEST 27        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
                        ;;*****************************************************************
006056  000004          TST27:  SCOPE
006060  012767  000010  173072          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006066  012767  000027  173104          MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                                        ; SET MAINT. BIT
006074                                                          LET @TCSR := @TCSR SET.BY #MAINT
006074  052777  000004  173162          BIS     #MAINT,@TCSR
006102                                                          BGNSUB
006102  012767  006110  173000          MOV     #64$,$LPERR
                                            ; OUTPUT A CHARACTER WITH MAINTENANCE
                                            ; SET, AND WAIT FOR XMITRDY TO SET.

                                                                        ; OUTPUT A CHARACTER
006110                                                              LET @TBUF :B= #0
006110  105077  173154                  CLRB    @TBUF

                                                                    ; WAIT MAXIMUM OF 500 MSEC
                                                                    ; FOR RCVRDONE TO SET IN
                                                                    ; RCSR
006114                                                              CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
006114  010546                  MOV     R5,-(SP)
006116  012745  177777          MOV     #SET,-(R5)
006122  016745  173132          MOV     RCSR,-(R5)
006126  012745  000200          MOV     #RCVRDONE,-(R5)
006132  012745  000500          MOV     #500,-(R5)
006136  004767  003326          JSR     PC,TIMER
006142  012605                  MOV     (SP)+,R5

                                                                        ; DID IT BECAME READY?
006144                                                              IF.ERROR  THEN
006144  103001                  BCC     $114

                                                                        ;RCVRDONE DID NOT SET IN RCSR
                                                                        ERRHRD 73,, DIDNOT
006146  104073                  ERROR   73
006150                                                              ENDIF
006150                  $114:
006150                                                          ENDSUB

                                                                ; NOW THAT IT IS SET LETS SEE IF READING THE
                                                                ; BUFFER CLEARS RCVRDONE.

                                                                        ;READ BUFFER
006150                                                              LET R0 :B= @RBUF
006150  117700  173106          MOVB    @RBUF,R0

006154                                                              IF #RCVRDONE SETIN @RCSR THEN
006154  032777  000200  173076          BIT     #RCVRDONE,@RCSR
006162  001401                  BEQ     $115

                                                                        ;RCVRDONE DID NOT CLEAR IN RCSR
                                                                        ERRHRD 74,DIDNOT
006164  104074                  ERROR   74
006166                                                              ENDIF
006166                  $115:
                                                            ENDTST
```

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*TEST 30        TEST THAT RCVRACT - RCSR 11 - SETS
                        ;*              WHEN A START BIT IS RECEIVED AND
                        ;*              CLEARS WHEN RCVRDONE - RCSR 7 - SETS
                        ;;*****************************************************************
006166  000004          TST30:  SCOPE
006170  012767  000010  172762          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006176  012767  000030  172774          MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                        ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
                                        IF #APTENV SETIN $ENV THEN
006204  032767  000001  173002          BIT     #APTENV,$ENV
006212  001404                          BEQ     $116
                                                        EXIT TEST
006214  012767  000001  172736          MOV     #1,$TIMES
006222  000500                          BR      TST31           ;;;EXIT THIS TEST
                                        ENDIF
006224                  $116:
                                                        LET @TCSR := @TCSR SET.BY #MAINT
006224  052777  000004  173032          BIS     #MAINT,@TCSR
                                                                LET R0 := #CLR
006232  012700  000000                  MOV     #CLR,R0
                                                                LET R1 := #0
006236  005001                          CLR     R1
                                    ;LOAD A CHARACTER INTO TBUF
                                    ;WAIT FOR RCVRACT TO SET


                                                                ;SEND A CHARACTER
                                                            LET @TBUF :B= #0
006240  105077  173024                  CLRB    @TBUF
                                                        REPEAT
                        $117:
                                                            IF #RCVRACT SETIN @RCSR THEN
006244  032777  004000  173006                  BIT     #RCVRACT,@RCSR
006252  001403                          BEQ     $120
                                                                LET R0 := #SET
006254  012700  177777                  MOV     #SET,R0
                                                            ELSE
006260  000401                          BR      $121
                        $120:
                                                                LET R1 := R1 + #1
006262  005201                          INC     R1
                                                            ENDIF
                        $121:
                                                        UNTIL R0 EQ #SET OR R1 HI MAX
006264  020027  177777                  CMP     R0,#SET
006270  001403                          BEQ     $122
006272  020167  000124                  CMP     R1,MAX
006276  101762                          BLOS    $117
                        $122:

                                                        IF R1 HI MAX THEN
006300  020167  000116                  CMP     R1,MAX
006304  101407                          BLOS    $123
                                                            ;IT NEVER SET
                                                            ;RCVRACT DID NOT SET IN RCSR.
```

```
2950  006306
2951  006306  104075                          ERROR   75                              ERRHRD 75,, DIDNOT
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
3007  006402                                                          ;READ CHAR.
3008  006402  017700  172654          MOV     @RBUF,R0              LET R0 := @RBUF
3009
3010  006406                                                          IF #RCVRDONE SETIN @RCSR THEN
3011  006406  032777  000200  172644  BIT     #RCVRDONE,@RCSR
3012  006414  001401                  BEQ     $131
3013
3014  006416                                                              ;RCVRDONE DID NOT CLEAR IN RCSR
3015  006416  104100                  ERROR   100                         ERRHRD 100,,DIDNOT
3016  006420
3017  006420                  $131:                                   ENDIF
3018
3019  006420                                                  EXIT
3020  006420  000401          BR      TST31           ;;;EXIT THIS TEST
3021  006422  070000          MAX:70000
3022
3023  006424                                          ENDTST
3024
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 31        TEST THE OVERRUN BIT - RBUF 14
                                ;;*****************************************************************
006424  000004                  TST31:  SCOPE
006426  012767  000010  172524          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006434  012767  000031  172536          MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

006442                                                          BGNSUB
006442  012767  006450  172440          MOV     #64$,$LPERR
                                            ;OUTPUT 2 CHARACTERS WITH
                                            ;AMPLE DELAYS BETWEEN FOR RECEPTION.
                                            ;THIS SHOULD AN CAUSE OVERRUN ERROR.

                                                                ;OUTPUT 1 CHARACTER
006450                                                      LET @TBUF :B= #0
006450  105077  172614                  CLRB    @TBUF

                                                                ;GO AWAY FOR 200. M SEC
006454                                                      WAITMS 200.
006454  010546                  MOV     R5,-(SP)
006456  012745  000310          MOV     #200.,-(R5)
006462  004767  003260          JSR     PC,WAIT
006466  012605                  MOV     (SP)+,R5

                                                                ;OUTPUT 2ND CHARACTER
006470                                                      LET @TBUF :B= #0
006470  105077  172574                  CLRB    @TBUF

                                                                ;LET OVERRUN HAPPEN
006474                                                      WAITMS 200.
006474  010546                  MOV     R5,-(SP)
006476  012745  000310          MOV     #200.,-(R5)
006502  004767  003240          JSR     PC,WAIT
006506  012605                  MOV     (SP)+,R5

                                                                ;READ BUFFER AND ERROR BITS
006510                                                      LET R4 := @RBUF
006510  017704  172546                  MOV     @RBUF,R4

                                                                ;IT DIDN'T SET
006514                                                      IF #ORERR NOTSETIN R4 THEN
006514  032704  040000                  BIT     #ORERR,R4
006520  001005                  BNE     $132

                                                                ;ORERR DID NOT SET IN RBUF
006522                                                          ERRHRD 101,,DIDNOT
006522  104101                  ERROR   101

                                                                ;NO USE COMPOUNDING ERRORS
006524                                                          EXIT TST
006524  012767  000001  172426          MOV     #1,$TIMES
006532  000456                  BR      TST32           ;;;EXIT THIS TEST
                                                              ENDIF
006534
006534                          $132:
006534                                                      ENDSUB

                                ;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     $133
3086
3087                                                          ;ERROR DID NOT SET IN RBUF
3088  006550                                                  ERRHRD 102,,DIDNOT
3089  006550  104102                        ERROR   102
3090
3091                                                          ;-WHEN ORERR SET.
3092                                                          ;GET OUT NOW.
3093  006552                                                  EXIT TST
3094  006552  012767  000001  172400        MOV     #1,$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 200.
3127  006616  010546                        MOV     R5,-(SP)
3128  006620  012745  000310                MOV     #200.,-(R5)
3129  006624  004767  003116                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
3136  006642                                                  ;ORERR DID NOT CLEAR IN RBUF
                                                              ERRHRD 104,,DIDNOT
```

```
3137  006642  104104                          ERROR   104
3138
3139                                                        ;-AFTER RECEIVING ANOTHER CHAR
3140                                                        ;SKIP AROUND REST
3141  006644                                                EXIT TST
3142  006644  012767  000001  172306          MOV     #1,$TIMES
3143  006652  000406                          BR      TST32          ;;;EXIT THIS TEST
3144  006654                                                      ENDIF
3145  006654                          $135:
3146
3147  006654                                                    IF #ERROR SETIN @RBUF THEN
3148  006654  032777  100000  172400          BIT     #ERROR,@RBUF
3149  006662  001401                          BEQ     $136
3150                                                        ;ERROR DID NOT CLEAR IN RBUF
3151  006664                                                ERRHRD 105,,DIDNOT
3152  006664  104105                          ERROR   105
3153
3154  006666                                                    ENDIF
3155  006666                          $136:
3156  006666                                                  ENDSUB
3157  006666                                                  EXIT
3158  006666  000400                          BR      TST32          ;;;EXIT THIS TEST
3159                                                          .EVEN
3160  006670                                              ENDTST
3161
```

```
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 RA E
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          CLR     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:
```

```
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
```

```
                                                                ;UPDATE OLD TIME
                                                              LET OLD := NEW
007126  016767  000110  000112          MOV     NEW,OLD
                                                              LET OLD+2 := NEW+2
007134  016767  000104  000106          MOV     NEW+2,OLD+2

                                                        ENDINC  ;BAUD RATE
007142  000714                          BR      $142
007144                          $143:
                                                      LET R3 :B= $USWR+1 AND #17      ; PUT BAUD BACK
007144  116703  172051                  MOVB    $USWR+1,R3
007150  110346                          MOVB    R3,-(SP)
007152  142716  000017                  BICB    #17,(SP)
007156  142603                          BICB    (SP)+,R3
007160                                                LET @TCSRHI :B= RATES(R3)      ; LIKE HE WANTED IT
007160  116377  007170  172100          MOVB    RATES(R3),@TCSRHI

007166                                                  EXIT    ;SKIP TABLE
007166  000431                          BR      TST33           ;;;EXIT THIS TEST

007170                          RATES:  ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
                                ;UPPER BYTE OF XCSR FOR EACH BAUD RATE
                                ;** NOTE::      THE VALUE INDICATED IN THE COLUMN 'OFFSET
                                ;**             INTO TABLE' CAN BE PLACED INTO BITS<11:8>
                                ;**             OF LOCATION '$USWR' TO CAUSE THE CORROSPONDING
                                ;**             BAUD TO BE SELECTED IN THE DLV11-E UPON
                                ;**             COMPLETION OF THIS TEST.

                                                        ;       BAUD    OFFSET INTO TABLE
007170      010                 R0050:  .BYTE   010     ;       50      0
007171      030                 R0070:  .BYTE   030     ;       70      1
007172      050                 R0110:  .BYTE   050     ;       110     2
007173      070                 R0135:  .BYTE   070     ;       135     3
007174      110                 R0150:  .BYTE   110     ;       150     4
007175      130                 R0300:  .BYTE   130     ;       300     5
007176      150                 R0600:  .BYTE   150     ;       600     6
007177      170                 R0200:  .BYTE   170     ;       1200    7
007200      210                 R1800:  .BYTE   210     ;       1800    10
007201      230                 R2000:  .BYTE   230     ;       2000    11
007202      250                 R2400:  .BYTE   250     ;       2400    12
007203      270                 R3600:  .BYTE   270     ;       3600    13
007204      310                 R4800:  .BYTE   310     ;       4800    14
007205      330                 R7200:  .BYTE   330     ;       7200    15
007206      350                 R9600:  .BYTE   350     ;       9600    16
007207      370                 R10000: .BYTE   370     ;       19200   17

007210  040502  042125  051040  BAUDRATE:       .ASCIZ  /BAUD RATE DIDN'T CHANGE./
007216  052101  020105  044504
007224  047104  052047  041440
007232  040510  043516  027105
007240     000
        007242                  .EVEN
007242  000000  000000          NEW:    0,0
007246  000000  000000          OLD:    0,0
007252                                                  ENDTST
```

```
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
                                                                LET INTFLAG := #0
007270  005067  002540                  CLR     INTFLAG

                                                                        ;GET VECTOR ADDRESS
                                                                LET R3 := DLVEC
007274  016703  171756                  MOV     DLVEC,R3

                                                                        ;FOR THE TRANSMITTER
                                                                LET R3 := R3 + #4
007300  062703  000004                  ADD     #4,R3

                                                                        ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
                                                                SETVEC R3, #INTSRV, #PR7
007304  010146                  MOV     R1,-(SP)
007306  010301                  MOV     R3,R1
007310  012721  012026          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
                                                                    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
                                                                    LET @TCSR := @TCSR SET.BY #XMITIE
007350  052777  000100  171706          BIS     #XMITIE,@TCSR

                                                                        ;LET INTERRUPT HAVE TIME TO OCCUR
                                                                WAITMS 200.
007356  010546                  MOV     R5,-(SP)
007360  012745  000310          MOV     #200.,-(R5)
007364  004767  002356          JSR     PC,WAIT
007370  012605                  MOV     (SP)+,R5

                                                                        ;DID EXACTLY 1 INTERRUPT OCCUR
                                                                IF INTFLAG NE #1 THEN
007372  026727  002436  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  002426          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                  $154:
3406  007416                                                    ENDSUB
3407                                                    ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
3408  007416                                                    BGNSUB
3409  007416  012767  007424  171464  MOV     #64$,$LPERR
3410                                                                    ;CLEAR 'INTERRUPT OCCURED' FLAG
3411  007424                                                        LET INTFLAG := #0
3412  007424  005067  002404          CLR     INTFLAG
3413                                                                    ;CLEAR INTERRUPT ENABLE
3414  007430                                                        LET @TCSR := @TCSR CLR.BY #XMITIE
3415  007430  042777  000100  171626  BIC     #XMITIE,@TCSR
3416                                                                    ;NO INTERRUPTS SHOULD OCCUR.
3417  007436  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3418  007442  012746  007450          MOV     #65$,-(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  002264          JSR     PC,WAIT
3426  007462  012605                  MOV     (SP)+,R5
3427  007464                                                        IF INTFLAG NE #0 THEN
3428  007464  005767  002344          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                                                    CLRVEC R3
3440  007476  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3441  007500  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3442  007502  012701  000003          MOV     #3,R1
3443  007506  010102                  MOV     R1,R2
```

```
3444  007510  062702  000002          ADD     #2,R2
3445  007514  010221                  MOV     R2,(R1)+
3446  007516  005011                  CLR     (R1)
3447  007520  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
3448  007522  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
3449
3450  007524                                      ENDTST
3451
3452
3453
3454
3455
3456
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 34        RECEIVER INTERRUPT LOGIC TEST
                                ;*              THIS TEST COVERS ALL OF THE RECEIVER
                                ;*              SIDE OF THE INTERRUPT LOGIC, BOTH DATASET
                                ;*              AND CHARACTER MODES.
                                ;;*****************************************************************
007524  000004                  TST34:  SCOPE
007526  012767  000010  171424          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007534  012767  000034  171436          MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR INTERRUPT OCCURED FLAG
                                                                        ;SET UP RECEIVER INTER.VECTOR
                                                                SETVEC DLVEC,#INTSRV,#PR7
007542  010146                          MOV     R1,-(SP)
007544  016701  171506                  MOV     DLVEC,R1
007550  012721  012026                  MOV     #INTSRV,(R1)+
007554  012711  000340                  MOV     #PR7,(R1)
007560  012601                          MOV     (SP)+,R1
                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
007562                                                          BGNSUB
007562  012767  007570  171320          MOV     #64$,$LPERR
007570                                                          LET INTFLAG := #0
007570  005067  002240                  CLR     INTFLAG
                                                                        ;SET MAINT. BIT
007574                                                          LET @TCSR := @TCSR SET.BY #MAINT
007574  052777  000004  171462          BIS     #MAINT,@TCSR
                                                                        ;CLEAR INTERRUPTS
007602                                                              LET @RCSR := @RCSR CLR.BY #RCVRIE
007602  042777  000100  171450          BIC     #RCVRIE,@RCSR
                                                                        ;CHANGE PRIORITY
                                                                        ;..TO 0
007610  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007614  012746  007622                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
007620  000002                          RTI                     ;;POP NEW PC AND PS
007622                          65$:
                                                                        ;SEND A CHARACTER
007622                                                          LET @TBUF :B= #0
007622  105077  171442                  CLRB    @TBUF
                                                                        ;WAIT A MAXIMUM
                                                                ;OF 500 MSEC FOR
                                                                ;RCVR RDY TO SET IN RCSR
007626                                                          CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
007626  010546                          MOV     R5,-(SP)
007630  012745  177777                  MOV     #SET,-(R5)
007634  016745  171420                  MOV     RCSR,-(R5)
007640  012745  000200                  MOV     #RCVRDONE,-(R5)
007644  012745  000500                  MOV     #500,-(R5)
007650  004767  001614                  JSR     PC,TIMER
007654  012605                          MOV     (SP)+,R5
                                                                        ;SET INTERRUPT ENABLE
007656                                                          LET @RCSR := @RCSR SET.BY #RCVRIE
007656  052777  000100  171374          BIS     #RCVRIE,@RCSR
                                                                        ;LET IT COME IN.
007664                                                          WAITMS 1
007664  010546                          MOV     R5,-(SP)
```

```
3513  007666  012745  000001          MOV     #1,-(R5)
3514  007672  004767  002050          JSR     PC,WAIT
3515 *007676  012605                  MOV     (SP)+,R5
3516  007700                                                  LET R0 := @RBUF ; CLEAR RCVRDONE
3517  007700  017700  171356          MOV     @RBUF,R0
3518                                                          ;DID HE DO IT RIGHT?
3519                                                      IF INTFLAG NE #1 THEN
3520  007704
3521  007704  026727  002124  000001  CMP     INTFLAG,#1
3522  007712  001406                  BEQ     $160
3523                                                          ;NONE OCCURED
3524  007714                                                  IF INTFLAG EQ #0 THEN
3525  007714  005767  002114          TST     INTFLAG
3526  007720  001002                  BNE     $161
3527                                                          ;RECEIVER DID NOT INTERRUPT IN TIME
3528  007722                                                      ERRHRD 111,,DIDNOT
3529  007722  104111                  ERROR   111
3530                                                          ;TWICE OR MORE
3531  007724                                                  ELSE
3532  007724  000401                  BR      $162
3533  007726                  $161:
3534                                                          ;RECEIVER INTERRUPTED TWICE
3535  007726                                                      ERRHRD 112,,TWICE
3536  007726  104112                  ERROR   112
3537  007730                                                  ENDIF
3538  007730                  $162:
3539  007730
3540  007730                  $160:                       ENDIF
3541
3542  007730                                                  ;RESET MAINT. BIT.
3543  007730  042777  000004  171326  BIC     #MAINT,@TCSR  LET @TCSR := @TCSR CLR.BY #MAINT
3544
3545  007736                                                  ; CLEAR INTERRUPT ENABLE
3546  007736  042777  000100  171314  BIC     #RCVRIE,@RCSR LET @RCSR := @RCSR CLR.BY #RCVRIE
3547  007744                                                  ENDSUB
3548
3549
3550
3551
3552
3553
3554
3555                                                      ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3556  007744                                                  BGNSUB
3557  007744  012767  007752  171136  MOV     #64$,$LPERR
3558  007752                                                  IF #CABLE NOTSETIN $USWR THEN
3559  007752  032767  020000  171240  BIT     #CABLE,$USWR
3560  007760  001004                  BNE     $163
3561                                                              ;CAN'T TEST WITHOUT A CABLE
3562  007762                                                      EXIT TST
3563  007762  012767  000001  171170  MOV     #1,$TIMES
3564  007770  000466                  BR      TST35       ;;;EXIT THIS TEST
3565  007772                                                  ENDIF
3566  007772                  $163:
3567
3568  007772                                                  ; CLEAR 'INTFLAG'
                                                          LET INTFLAG := #0
```

```
3569  007772  005067  002036                CLR     INTFLAG
3570
3571  007776                                                        ;CLEAR INTERRUPTS
3572  007776  042777  000040  171254          BIC     #DATAIE,@RCSR LET @RCSR := @RCSR CLR.BY #DATAIE
3573
3574                                                                ;CHANGE PRIORITY
3575  010004  012746  000000                  MOV     #PR0,-(SP)    ;;...TO 0
3576  010010  012746  010016                  MOV     #64$,-(SP)    ;;PUT NEW PS ON STACK
3577  010014  000002                          RTI                   ;;PUT NEW PC ON  STACK
3578  010016                          64$:                          ;;POP NEW PC AND PS
3579  010016
3580  010016  042777  000004  171234          BIC     #REQSEND,@RCSR LET @RCSR := @RCSR CLR.BY #REQSEND
3581
3582  010024                                                        ;SET INTERRUPT ENABLE
3583  010024  052777  000040  171226          BIS     #DATAIE,@RCSR LET @RCSR := @RCSR SET.BY #DATAIE
3584  010032
3585  010032  052777  000004  171220          BIS     #REQSEND,@RCSR LET    @RCSR := @RCSR SET.BY #REQSEND
3586
3587  010040                                                        ;LET IT COME IN.
3588  010040  010546                  MOV     R5,-(SP)              WAITMS 1
3589  010042  012745  000001          MOV     #1,-(R5)
3590  010046  004767  001674          JSR     PC,WAIT
3591  010052  012605                  MOV     (SP)+,R5
3592
3593
3594  010054                                                        ; DID IT DO IT RIGHT?
3595  010054  026727  001754  000001          CMP     INTFLAG,#1    IF INTFLAG NE #1 THEN
3596  010062  001406                  BEQ     $164
3597
3598  010064                                                        ;NONE OCCURED
3599  010064  005767  001744          TST     INTFLAG               IF INTFLAG EQ #0 THEN
3600  010070  001002                  BNE     $165
3601
3602  010072                                                        ;DATAINT  DID NOT INTERRUPT IN TIME
3603  010072  104113                  ERROR   113                   .   ERRHRD 113,,DIDNOT
3604
3605  010074                                                        ;TWICE OR MORE
3606  010074  000401                  BR      $166                  ELSE
3607  010076                          $165:
3608
3609  010076                                                        ; DATAINT INTERRUPTED TWICE
3610  010076  104114                  ERROR   114                      ERRHRD 114,,TWICE
3611  010100
3612  010100                          $166:                         ENDIF
3613  010100
3614  010100                          $164:                     ENDIF
3615  010100
3616  010100  042777  000040  171152          BIC     #DATAIE,@RCSR     LET @RCSR := @RCSR CLR.BY #DATAIE
3617  010106
3618  010106  042777  000004  171144          BIC     #REQSEND,@RCSR    LET @RCSR := @RCSR CLR.BY #REQSEND
3619  010114                                                        ENDSUB
3620
3621  010114                                                        LET R4 := @DLVEC
3622  010114  017704  171136          MOV     @DLVEC,R4
3623  010120                                                        CLRVEC R4
3624  010120  010146                  MOV     R1,-(SP)      ;,PUSH R1 ON STACK
```

```
 3625  010122  010246                      MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 3626  010124  012701  000004              MOV     #R4,R1
 3627  010130  010102                      MOV     R1,R2
 3628  010132  062702  000002              ADD     #2,R2
 3629  010136  010221                      MOV     R2,(R1)+
 3630  010140  005011                      CLR     (R1)
 3631  010142  012602                      MOV     (SP)+,R2        ;;POP STACK INTO R2
 3632  010144  012601                      MOV     (SP)+,R1        ;;POP STACK INTO R1
 3633  010146                                              ENDTST
```

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*TEST 35        TEST ACTUAL DATA TRANSFERED
                        ;*                NON-INTERRUPT MAINTENANCE BIT SET
                        ;;*****************************************************************
010146  000004          TST35:  SCOPE
010150  012767  000001  171002          MOV     #1,$TIMES       ;;DO 1 ITERATION
010156  012767  000035  171014          MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ;SET MAINT. BIT
010164                                                      LET @TCSR := @TCSR SET.BY #MAINT
010164  052777  000004  171072          BIS     #MAINT,@TCSR

                                                                ;CHANGE PRIORITY
                                                                ;..TO 0
010172  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
010176  012746  010204                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
010202  000002                          RTI                     ;;POP NEW PC AND PS
010204                          64$:

                                                                ;GET DATA MASK.
010204                                                      CALL DATLNG OUT <R1>
010204  162705  000002                  SUB     #1*2,R5
010210  004767  001432                  JSR     PC,DATLNG
010214  012501                          MOV     (R5)+,R1

                                                                ; START CLEAN
010216                                                      LET R0 := @RBUF
010216  017700  171040                  MOV     @RBUF,R0

                                                                ;ALL BINARY CHAR.
010222                                                      INCR R2 FROM #0 TO #377 BY #1
010222  005002                          CLR     R2
010224  000401                          BR      $167
010226                          $170:
010226  005202                          INC     R2
010230                          $167:
010230  020227  000377                  CMP     R2,#377
010234  003047                          BGT     $171


                                                            ;TRANSMIT CHAR IN R2

010236                                                      CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
010236  010546                  MOV     R5,-(SP)
010240  012745  177777          MOV     #SET,-(R5)
010244  016745  171014          MOV     TCSR,-(R5)
010250  012745  000200          MOV     #XMITRDY,-(R5)
010254  012745  000500          MOV     #500,-(R5)
010260  004767  001204          JSR     PC,TIMER
010264  012605                  MOV     (SP)+,R5


                                                                ;TRANSMIT IT
010266                                                      LET @TBUF :B= R2
010266  110277  170776          MOVB    R2,@TBUF

010272                                                      CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
010272  010546                  MOV     R5,-(SP)
010274  012745  177777          MOV     #SET,-(R5)
```

```
3690  010300  016745  170754          MOV     RCSR,-(R5)
3691  010304  012745  000200          MOV     #RCVRDONE,-(R5)
3692  010310  012745  000500          MOV     #500,-(R5)
3693  010314  004767  001150          JSR     PC,TIMER
3694  010320  012605                  MOV     (SP)+,R5
3695                                                        ;AND SAVE IT
3696  010322                                                LET R3 := @RBUF
3697  010322  017703  170734          MOV     @RBUF,R3
3698
3699
3700                                                        ;COMPARE TO SEE IF WE RECEIVED IT ALL
3701
3702                                                            ;CLEAN OFF NON-DATA BITS
3703                                                            ;ON BOTH TRANSMITTED AND
3704  010326                                                LET R4 := R2 CLR.BY R1
3705  010326  010204                  MOV     R2,R4
3706  010330  040104                  BIC     R1,R4
3707  010332                                                LET R3 := R3 CLR.BY R1
3708  010332  040103                  BIC     R1,R3
3709
3710                                                        ;RECEIVED DATA
3711  010334                                                IF R4 NE R3 THEN
3712  010334  020403                  CMP     R4,R3
3713  010336  001405                  BEQ     $172
3714
3715  010340                                                    ;DATA COMPARE ERROR
3716  010340  104116                  ERROR   116               ERRHRD 116,COMP,SBWAS
3717  010342
3718  010342  012767  000001  170610  MOV     #1,$TIMES         EXIT TST  ; ON ERROR
3719  010350  000404                  BR      TST36       ;;;EXIT THIS TEST
3720  010352                                                    ENDIF
3721  010352                  $172:
3722  010352                                                ENDINC  ; R2
3723  010352  000725                  BR      $170
3724  010354                  $171:
3725
3726                                                            ;RESET MAINT. BIT.
3727  010354                                                LET @TCSR := @TCSR CLR.BY #MAINT
3728  010354  042777  000004  170702  BIC     #MAINT,@TCSR
3729  010362                                                ENDTST
3730
3731
3732
```

```
3733                                    ;;*****************************************************************
3734                                    ;;*****************************************************************
3735                                    ;*TEST 36        TEST DATA THROUGH CABLE
3736                                    ;;*****************************************************************
3737  010362  000004                    TST36:  SCOPE
3738  010364  012767  000001  170566            MOV     #1,$TIMES       ;;DO 1 ITERATION
3739  010372  012767  000036  170600            MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3740  010400                                                                    IF #CABLE NOTSETIN $USWR THEN
3741  010400  032767  020000  170612            BIT     #CABLE,$USWR
3742  010406  001004                            BNE     $173
3743                                                                              ;CAN'T TEST WITHOUT A CABLE
3744  010410                                                                    EXIT TST
3745  010410  012767  000001  170542            MOV     #1,$TIMES
3746  010416  000474                            BR      TST37           ;;;EXIT THIS TEST
3747  010420                                                                    ENDIF
3748  010420                                $173:
3749                                                                              ;DON'T USE MAINT.
3750  010420                                                                    LET @TCSR := @TCSR CLR.BY #MAINT
3751  010420  042777  000004  170636            BIC     #MAINT,@TCSR
3752                                                                              ;CHANGE PRIORITY
3753                                                                              ;..TO 0
3754  010426  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3755  010432  012746  010440                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3756  010436  000002                            RTI                     ;;POP NEW PC AND PS
3757  010440                                64$:
3758                                                                              ;GET DATA MASK
3759  010440                                                                    CALL DATLNG OUT <R1>
3760  010440  162705  000002                    SUB     #1*2,R5
3761  010444  004767  001176                    JSR     PC,DATLNG
3762  010450  012501                            MOV     (R5)+,R1
3763  010452                                                                    LET R0 := @RBUF ; START CLEAN
3764  010452  017700  170604                    MOV     @RBUF,R0
3765                                                                              ;BINARY COUNT PATTERN
3766  010456                                                                    INCR R2 FROM #0 TO #377 BY #1
3767  010456  005002                            CLR     R2
3768  010460  000401                            BR      $174
3769  010462                                $175:
3770  010462  005202                            INC     R2
3771  010464                                $174:
3772  010464  020227  000377                    CMP     R2,#377
3773  010470  003047                            BGT     $176
3774
3775
3776                                                                              ;TRANSMIT THE CHAR. IN R2.
3777
3778  010472                                                                    CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
3779  010472  010546                            MOV     R5,-(SP)
3780  010474  012745  177777                    MOV     #SET,-(R5)
3781  010500  016745  170560                    MOV     TCSR,-(R5)
3782  010504  012745  000200                    MOV     #XMITRDY,-(R5)
3783  010510  012745  000500                    MOV     #500,-(R5)
3784  010514  004767  000750                    JSR     PC,TIMER
3785  010520  012605                            MOV     (SP)+,R5
3786
3787                                                                              ;START IT ON ITS WAY
3788  010522                                                                    LET @TBUF :B= R2
```

```
3789  010522  110277  170542                MOVB    R2,@TBUF
3790  010526                                                    CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
3791  010526  010546                        MOV     R5,-(SP)
3792  010530  012745  177777                MOV     #SET,-(R5)
3793  010534  016745  170520                MOV     RCSR,-(R5)
3794  010540  012745  000200                MOV     #RCVRDONE,-(R5)
3795  010544  012745  000500                MOV     #500,-(R5)
3796  010550  004767  000714                JSR     PC,TIMER
3797  010554  012605                        MOV     (SP)+,R5
3798
3799                                                        ;RETRIEVE
3800  010556                                                LET R3 := @RBUF
3801  010556  017703  170500                MOV     @RBUF,R3
3802
3803                                                        ;STRIP OFF JUNK ON BOTH
3804  010562                                                LET R4 := R2 CLR.BY R1
3805  010562  010204                        MOV     R2,R4
3806  010564  040104                        BIC     R1,R4
3807  010566                                                LET R3 := R3 CLR.BY R1
3808  010566  040103                        BIC     R1,R3
3809
3810                                                        ;WE HAVE TROUBLE
3811  010570                                                IF R4 NE R3 THEN
3812  010570  020403                        CMP     R4,R3
3813  010572  001405                        BEQ     $177
3814
3815  010574                                                    ;DATA COMPARE ERROR
3816  010574  104117                        ERROR   117       ERRHRD 117,COMP,SBWAS
3817  010576
3818  010576  012767  000001  170354        MOV     #1,$TIMES EXIT TST  ; ON ERROR
3819  010604  000401                        BR      TST37   ;;;EXIT THIS TEST
3820  010606                                                ENDIF
3821  010606                        $177:
3822
3823  010606
3824  010606  000725                        BR      $175                ENDINC  ; R2
3825  010610                        $176:
3826
3827
3828
3829  010610                                                ENDTST
3830
3831
3832
3833
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 37        FULL DATA TRANSFER WITH INTERRUPTS
                                ;*              AND MAINTENANCE MODE.
                                ;;*****************************************************************
010610  000004                  TST37:  SCOPE
010612  012767  000001  170340          MOV     #1,$TIMES       ;;DO 1 ITERATION
010620  012767  000037  170352          MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


                                                        ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
010626                                                          IF #APTENV SETIN $ENV THEN
010626  032767  000001  170360          BIT     #APTENV,$ENV
010634  001404                          BEQ     $200
010636                                                                  EXIT TEST
010636  012767  000001  170314          MOV     #1,$TIMES
010644  000550                          BR      TST40           ;;;EXIT THIS TEST
010646                                                          ENDIF
010646                          $200:
                                                                        ;GET DATA MASK
010646                                                                  CALL DATLNG OUT <R3>
010646  162705  000002                  SUB     #1*2,R5
010652  004767  000770                  JSR     PC,DATLNG
010656  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
010660  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
010664  012746  010672                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
010670  000002                          RTI                     ;;POP NEW PC AND PS
010672                          64$:
                                                                        ;GET VECTOR ADDRESS
010672                                                                  LET R1 := DLVEC
010672  016701  170360                  MOV     DLVEC,R1
                                                                        ;RCVR VECTOR
010676                                                                  LET (R1)+ := #REC
010676  012721  011072                  MOV     #REC,(R1)+
010702                                                                  LET (R1)+ := #PR7
010702  012721  000340                  MOV     #PR7,(R1)+
                                                                        ;POINT TO TRANSMITTER VECTOR
                                                                        ;AND SET IT UP ALSO
010706                                                                  LET (R1)+ := #TRAN
010706  012721  011030                  MOV     #TRAN,(R1)+
010712                                                                  LET (R1)   := #PR7
```

```
                                                                                ; INITIALIZE COUNTERS
3890  010712  012711  000340                MOV     #PR7,(R1)                 LET R1 := #-1
3891
3892                                                                            ;RECEIVER STORAGE
3893  010716                                                                    LET R2 := #0
3894  010716  012701  177777                MOV     #-1,R1
3895                                                                            ;# OF RECEIVED CHAR. COUNT.
3896  010722                                                                    LET R4 := #-1
3897  010722  005002                        CLR     R2
3898                                                                            ; CLEAR ERROR COUNT.
3899  010724                                                                    LET ERRCNT := #0
3900  010724  012704  177777                MOV     #-1,R4
3901                                                                            BRESET  ;SET UP ALL REGISTERS
3902
3903  010730                                                                            ;SET UP MAINTENANCE
3904  010730  005067  000066                CLR     ERRCNT                    LET @TCSR := @TCSR SET.BY #MAINT
3905
3906  010734                                                                            ;SET I.E. IN TRANSMITTER
3907  010734  000005                        RESET                             LET @TCSR := @TCSR SET.BY #XMITIE
3908
3909  010736                                                                            ;AND RECEIVER
3910  010736  052777  000004  170320        BIS     #MAINT,@TCSR              LET @RCSR := @RCSR SET.BY #RCVRIE
3911
3912
3913  010744
3914  010744  052777  000100  170312        BIS     #XMITIE,@TCSR
3915
3916  010752
3917  010752  052777  000100  170300        BIS     #RCVRIE,@RCSR
3918
3919
3920                                                                            ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3921  010760                                                                    REPEAT
3922  010760                        $201:
3923  010760
3924  010760  020467  000040                CMP     R4,NUMBER                 UNTIL R4 EQ NUMBER OR ERRCNT GT #0
3925  010764  001403                        BEQ     $202
3926  010766  005767  000030                TST     ERRCNT
3927  010772  003772                        BLE     $201
3928  010774                        $202:
3929
3930                                                                                    ;DATA COMPARE ERRORS.
3931  010774                                                                    IF ERRCNT NE #0 THEN
3932  010774  005767  000022                TST     ERRCNT
3933  011000  001401                        BEQ     $203
3934                                                                                    ;DATA COMPARE ERROR
3935  011002                                                                        ERRHRD 120,COMP,FIRST
3936  011002  104120                        ERROR   120
3937  011004                                                                    ENDIF
3938  011004                        $203:
3939
3940  011004                                                                    LFT @TCSR := @TCSR CLR.BY #XMITIE
3941  011004  042777  000100  170252        BIC     #XMITIE,@TCSR
3942  011012                                                                    LET @RCSR := @RCSR CLR.BY #XMITIE
3943  011012  042777  000100  170240        BIC     #XMITIE,@RCSR
3944
3945  011020                                                                    EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
```

```
3946  011020  000462                          BR      TST40           ;;;EXIT THIS TEST
3947
3948  011022  000000                                                  ERRCNT: 0
3949  011024  000400                                                  NUMBER: 400
3950  011026     000                                                  SB:     .BYTE 0
3951  011027     000                                                  WAS:    .BYTE 0
3952
3953
3954
3955                                  ;*************************************************************
3956                                                  ;TRANSMIT INTERRUPT HANDLER
3957
3958  011030                                                          BGNSRV  TRAN
3959  011030                          TRAN:
3960                                  ;*************************************************************
3961
3962
3963  011030                                                              ;INCREMENT CHAR COUNT
3964  011030  005201                          INC     R1              LET R1 := R1 + #1
3965
3966  011032                                                              ;SET UP FOR TRANSFER
3967  011032  010167  000030                  MOV     R1,HOLD         LET HOLD := R1      CLR.BY R3
3968  011036  040367  000024                  BIC     R3,HOLD
3969
3970  011042                                                              ;AND SEND.
3971  011042  016777  000020  170220          MOV     HOLD,@TBUF      LET @TBUF := HOLD
3972
3973  011050                                                              ;ALL DONE
3974  011050  020167  177750                  CMP     R1,NUMBER       IF R1 EQ NUMBER THEN
3975  011054  001003                          BNE     $204
3976
3977  011056                                                                  ;STOP   INTERRUPT PROCESSING
3978  011056  042777  000100  170200          BIC     #XMITIE,@TCSR           LET @TCSR := @TCSR CLR.BY #XMITIE
3979  011064                                                              ENDIF
3980  011064                          $204:
3981
3982  011064  000401                          BR      ZZZ                 ; EXIT SRV
3983
3984  011066  000000                                                      HOLD:0
3985
3986  011070                          ZZZ:                            ENDSRV
3987  011070  000002                          RTI
3988
3989
3990                                  ;*************************************************************
3991
3992                                                  ;RECEIVER INTERRUPT HANDLER
3993  011072                                                                  BGNSRV  REC
3994  011072                          REC:
3995                                  ;*************************************************************
3996
3997
3998  011072                                                              ;COUNT THIS CHAR.
3999  011072  005204                          INC     R4              LET R4 := R4 + #1
4000
4001  011074                                                              ;GET CHAR IN + MASK IT
                                                                      LET R2 := @RBUF CLR.BY R3
```

```
4002  011074  017702  170162          MOV     @RBUF,R2
4003  011100  040302                  BIC     R3,R2
4004                                                        ;RHLD WILL CONTAIN EXPECTED INPUT
4005  011102                                              LET RHLD := R4 CLR.BY R3
4006  011102  010467  000054          MOV     R4,RHLD
4007  011106  040367  000050          BIC     R3,RHLD
4008
4009                                                        ;DO THEY COMPARE
4010  011112                                              IF R2 NE RHLD THEN
4011  011112  020267  000044          CMP     R2,RHLD
4012  011116  001412                  BEQ     $205
4013
4014  011120                                                  ;FIRST ERROR
4015  011120  005767  177676          TST     ERRCNT         IF ERRCNT EQ #0 THEN
4016  011124  001005                  BNE     $206
4017
4018  011126                                                  ;SAVE RECORD OF FIRST MISS
4019  011126  116767  000030  177672  MOVB    RHLD,SB            LET SB :B= RHLD
4020  011134
4021  011134  110267  177667          MOVB    R2,WAS             LET WAS :B= R2
4022  011140
4023  011140                  $206:                           ENDIF
4024
4025  011140                                                  ;COUNT IT.
4026  011140  005267  177656          INC     ERRCNT         LET ERRCNT := ERRCNT + #1
4027  011144
4028  011144                  $205:                       ENDIF
4029
4030
4031  011144                                                  ;ALL DONE?
4032  011144  020467  177654          CMP     R4,NUMBER      IF R4 EQ NUMBER THEN
4033  011150  001003                  BNE     $207
4034
4035  011152                                                      ;STOP RECEIVER INTERRUPTS
4036  011152  042777  000100  170100  BIC     #RCVRIE,@RCSR      LET @RCSR := @RCSR CLR.BY #RCVRIE
4037                                                              ;MAIN REPEAT LOOP IS CHECKING
4038  011160                                                  ENDIF
4039  011160                  $207:
4040
4041                                                          ;FOR 'R4 = NUMBER' ALSO
4042  011160  000401                  BR      ZZZZ           ;  EXIT SRV
4043
4044  011162  000000                                          RHLD:0
4045  011164                  ZZZZ:
4046  011164                                              ENDSRV
4047  011164  000002                  RTI
4048
4049  011166                                          ENDTST
4050
4051
4052
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 40        TEST BREAK GENERATION LOGIC
                                ;*              TRANSMIT KNOWN CHAR WITH BREAK SET
                                ;*              AND COMPARE RECEIVED WITH 0.
                                ;;*****************************************************************
011166  000004                  TST40:  SCOPE
011170  012767  000010  167762          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
011176  012767  000040  167774          MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
011204                                                                  IF #BRK NOTSETIN $USWR THEN
011204  032767  010000  170006          BIT     #BRK,$USWR
011212  001004                          BNE     $210
011214                                                                          EXIT TST
011214  012767  000001  167736          MOV     #1,$TIMES
011222  000452                          BR      TST41           ;;;EXIT THIS TEST
011224                                                                  ENDIF
011224                                  $210:
011224                                                                  ;SET MAINTENANCE BIT
011224                                                                  LET @TCSR := @TCSR SET.BY #MAINT
011224  052777  000004  170032          BIS     #MAINT,@TCSR
                                                                        ; CLEAR RCVRDONE JUST IN CASE
011232                                                                  LET R0 := @RBUF
011232  017700  170024                  MOV     @RBUF,R0
                                                                        ;SET BREAK BIT
011236                                                                  LET @TCSR := @TCSR SET.BY #BREAK
011236  052777  000001  170020          BIS     #BREAK,@TCSR
                                                                        ;NON-ZERO CHAR. '*'
011244                                                          LET @TBUF := #252
011244  012777  000252  170016          MOV     #252,@TBUF
011252                                                          CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
011252  010546                          MOV     R5,-(SP)
011254  012745  177777                  MOV     #SET,-(R5)
011260  016745  167774                  MOV     RCSR,-(R5)
011264  012745  000200                  MOV     #RCVRDONE,-(R5)
011270  012745  000500                  MOV     #500,-(R5)
011274  004767  000170                  JSR     PC,TIMER
011300  012605                          MOV     (SP)+,R5
011302
011302  103001                          BCC     $211            IF.ERROR THEN
                                                                        ; RECIEVER DONE DID NOT SET
011304                                                                  ERRHRD 115
011304  104115                          ERROR   115
011306                                                          ENDIF
011306                                  $211:
                                                                IFB @RBUF NE #0 THEN
011306
011306  105777  167750                  TSTB    @RBUF
011312  001401                          BEQ     $212
                                                                        ; BREAK DID NOT EQUAL 0
011314                                                                  ERRHRD 121 ,BADBRK
011314  104121                          ERROR   121
011316                                                          ENDIF
011316                                  $212:
```

```
4109  011316
4110  011316  000005                          RESET                                   BRESET  ;CLEAN UP
4111  011320                                                                        EXIT
4112  011320  000413                          BR      TST41          ;;;EXIT THIS TEST
4113  011322  051102  040505  020113                                  BADBRK: .ASCIZ  /BREAK DID NOT EQUAL 0/
4114  011330  044504  020104  047516
4115  011336  020124  050505  040525
4116  011344  020114  000060
4117
4118  011350                                                  ENDTST
```

```
4119
4120                                    ;;**********************************************************************
4121                                    ;*TEST 41        NOT A TEST - SEND BACK TO LOOP
4122                                    ;;**********************************************************************
4123  011350  000004                   TST41:  SCOPE
4124  011352  012767  000001  167600           MOV     #1,$TIMES       ;;DO 1 ITERATION
4125  011360  104401  011366                   TYPE    ,65$            ;;TYPE ASCIZ STRING
4126  011364  000404                           BR      64$             ;;GET OVER THE ASCIZ
4127                                    ;;65$:  .ASCIZ  <CRLF>*CSR: *
4128  011376                           64$:
4129  011376  016746  167652                   MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
4130  011402  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4131  011404  104401  011412                   TYPE    ,67$            ;;TYPE ASCIZ STRING
4132  011410  000405                           BR      66$             ;;GET OVER THE ASCIZ
4133                                    ;;67$:  .ASCIZ  *,VECTOR: *
4134  011424                           66$:
4135  011424  016746  167626                   MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
4136  011430  104402                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
4137  011432  104401  011440                   TYPE    ,69$            ;;TYPE ASCIZ STRING
4138  011436  000405                           BR      68$             ;;GET OVER THE ASCIZ
4139                                    ;;69$:  .ASCIZ  *,ERRORS: *
4140  011452                           68$:
4141  011452  016746  167434                   MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
4142  011456  104405                           TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
4143  011460  005067  167426                   CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
4144  011464  000167  170254                   JMP     LOOP            ; BACK UP TO THE BEGINNING
```

```
4145
4146                                    ;;BGNMOD        SUBS
4147                                    ;;*****************************************************************
4148  011470                                                    ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
4149  011470                            TIMER:
4150                                    ;* ROUTINE:TIMER
4151                                    ;*      THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
4152                                    ;*      IN ANY REGISTER.
4153                                    ;*      INPUTS:
4154                                    ;*            HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
4155                                    ;*                       THIS ROUTINE.
4156                                    ;*            WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
4157                                    ;*                         TO BE CHECKED.
4158                                    ;*            REG      A POINTER TO THE REGISTER TO BE CHECKED
4159                                    ;*            SETCLR   THE DESIRED RESULTS
4160                                    ;*                     EITHER #SET OR #CLEAR
4161                                    ;*      OUTPUT:
4162                                    ;*            THE 'C' BIT IS SET TO INDICATE AN ERROR
4163                                    ;*            BUT IT IS TESTED BY THE IF.ERROR STATEMENT
4164                                    ;*
4165                                    ;*
4166                                    ;*            NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
4167                                    ;*                    MECHANISM BETWEEN THE CALLER AND THE CALLED
4168                                    ;;*****************************************************************
4169
4170          000001                                                TRUE=   1
4171          000000                                                FALSE=  0
4172
4173  011470
4174  011470  016567  000004  000136            MOV     REG(R5),REGSAV          LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
4175  011476
4176  011476  016567  000000  000132            MOV     HOWLONG(R5),TIMSAV      LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
4177  011504
4178  011504  112767  000000  000126            MOVB    #FALSE,FLAG             LET     FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA
4179
4180
4181                                                                    ;       START OF AN INFINITE LOOP
4182
4183  011512                                                            LOOP
4184  011512                            $215:
4185                                                                            ; TEST TO SEE IF WHICHBIT IS SET
4186  011512                                                            IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
4187  011512  036577  000002  000114            BIT     WHICHBIT(R5),@REGSAV
4188  011520  001004                            BNE     $217
4189  011522                                                                LET     HOLDSC :B= #CLR
4190  011522  112767  000000  000111            MOVB    #CLR,HOLDSC
4191  011530                                                            ELSE
4192  011530  000403                            BR      $220
4193  011532                            $217:
4194  011532                                                                LET     HOLDSC :B= #SET ; REMEMBER THIS
4195  011532  112767  177777  000101            MOVB    #SET,HOLDSC
4196  011540                                                            ENDIF
4197  011540                            $220:
4198                                                                    ; NOW SEE IF THAT WAS WHAT WE WANTED
4199                                                                    IFB     HOLDSC EQ SETCLR(R5) THEN
4200  011540
```

```
4201  011540  126765  000075  000006          CMPB    HOLDSC,SETCLR(R5)
4202  011546  001003                          BNE     $221
4203                                                    ; JUST THE THING WE NEEDED
4204  011550                                            LET     FLAG :B= #TRUE
4205  011550  112767  000001  000062          MOVB    #TRUE,FLAG
4206  011556                                            ENDIF
4207  011556                          $221:
4208
4209  011556                                    EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
4210  011556  126727  000056  000001          CMPB    FLAG,#TRUE
4211  011564  001414                          BEQ     $216
4212  011566  005767  000044                  TST     TIMSAV
4213  011572  003411                          BLE     $216
4214                                                    ; ONE WAY OR THE OTHER, WE ARE DONE
4215                                            ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
4216
4217  011574                                            WAITMS  1       ;WAIT FOR 10 MILLI-SECONDS
4218  011574  010546                          MOV     R5,-(SP)
4219  011576  012745  000001                  MOV     #1,-(R5)
4220  011602  004767  000140                  JSR     PC,WAIT
4221  011606  012605                          MOV     (SP)+,R5
4222  011610                                            LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4223  011610  005367  000022                  DEC     TIMSAV
4224  011614                                    ENDLOOP                 ; CONTINUED AT THE TOP
4225  011614  000736                          BR      $215
4226  011616                          $216:
4227
4228                                    ; ONLY 2 WAYS TO GET HERE
4229                                    ;       1). WE RAN OUT OF TIME---ERROR ! !
4230                                    ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
4231
4232  011616                                    IFB     FLAG EQ #TRUE THEN
4233  011616  126727  000016  000001          CMPB    FLAG,#TRUE
4234  011624  001001                          BNE     $222
4235  011626                                            RETURN  NO.ERROR        ; GOOD
4236  011626  000405                          BR      $213
4237  011630                                    ENDIF
4238  011630                          $222:
4239  011630                                    RETURN  ERROR           ; BAD
4240  011630  000261                          SEC
4241  011632  000404                          BR      $214
4242
4243  011634  000000                          REGSAV: .WORD 0
4244  011636  000000                          TIMSAV: .WORD 0
4245  011640     000                          FLAG:   .BYTE 0
4246  011641     000                          HOLDSC: .BYTE 0
4247                                    ; WE ARE DONE   GO BACK HOME
4248  011642                                    ENDRTN
4249  011642                          $213:
4250  011642  000241                          CLC
4251  011644                          $214:
4252  011644  000207                          RTS     PC
```

```
                        ;;*****************************************************************
                        ROUTINE DATLNG <MASK>
011646                  DATLNG:
                        ;* 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 ONES RIGHT-JUSTIFIED
                        ;*              THE NUMBER OF WHICH IS DEFINED IN $USWR WORD.

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

011646                                                          LET MASK(R5) := #0              ; START
011646  005065  000000          CLR     MASK(R5)
011652                                                          LET NUMBR := $USWR AND #DATA
011652  016767  167342  000062  MOV     $USWR,NUMBR
011660  016746  000056          MOV     NUMBR,-(SP)
011664  042716  000017          BIC     #DATA,(SP)
011670  042667  000046          BIC     (SP)+,NUMBR

011674                                                          INCR I FROM #1 TO NUMBR BY #1
011674  012767  000001  167370  MOV     #1,I
011702  000402                  BR      $225
011704                  $226:
011704  005267  167362          INC     I
011710                  $225:
011710  026767  167356  000024  CMP     I,NUMBR
011716  003006                  BGT     $227
011720                                                              LET MASK(R5) := MASK(R5) SHIFT #1
011720  006365  000000          ASL     MASK(R5)
011724                                                              LET MASK(R5) := MASK(R5) SET.BY #1
011724  052765  000001  000000  BIS     #1,MASK(R5)
011732                                                          ENDINC
011732  000764                  BR      $226
011734                  $227:
011734                                                          LET MASK(R5) := COMP MASK(R5)
011734  005165  000000          COM     MASK(R5)
011740                                                  RETURN
011740  000401                  BR      $223
011742  000000                  NUMBR:0
011744                                                  ENDRTN
011744                  $223:
011744                  $224:
011744  000207                  RTS     PC
```

```
                                ;;**********************************************************************
011746                          ROUTINE WAIT    <TIME>
011746                          WAIT:
                                ;* 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.
                                ;;**********************************************************************
011746  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
011750  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
011752  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
011754                                  LET R1 := TIME(R5)
011754  016501  000000                  MOV     TIME(R5),R1
011760                                  INCRU R2 FROM #1 TO R1 BY #1
011760  012702  000001                  MOV     #1,R2
011764  000402                          BR      $232
011766                          $233:
011766  062702  000001                  ADD     #01,R2
011772                          $232:
011772  020201                          CMP     R2,R1
011774  101010                          BHI     $234
011776                                  INCR R3 FROM #0 TO #100 BY #1
011776  005003                          CLR     R3
012000  000401                          BR      $235
012002                          $236:
012002  005203                          INC     R3
012004                          $235:
012004  020327  000100                  CMP     R3,#100
012010  003001                          BGT     $237
012012                                  ENDINC
012012  000773                          BR      $236
012014                          $237:
012014                                  ENDINC
012014  000764                          BR      $233
012016                          $234:
012016  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
012020  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
012022  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
012024                          ENDRTN
012024                          $230:
012024                          $231:
012024  000207                          RTS     PC
```

```
                                .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
                                ;;*****************************************************************
012026                          INTSRV:
                                ;* 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.
                                ;;*****************************************************************

                                                        ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
012026                                                          LET INTFLAG := INTFLAG + #1
012026  005267  000002                  INC     INTFLAG
012032                                  ENDSRV                          ;THAT'S ALL
012032  000002                          RTI
012034  000000                  INTFLAG:  0
```

```
 4364
 4365  012036                          ROUTINE MYTYPE
 4366  012036                          MYTYPE:
 4367                                  ;;*****************************************************************
 4368  012036  104401  012044                TYPE    ,65$            ;;TYPE ASCIZ STRING
 4369  012042  000405                        BR      64$             ;;GET OVER THE ASCIZ
 4370                                  ;;65$:  .ASCIZ  <CRLF>*TEST # *
 4371  012056                          64$:
 4372  012056  016746  167116                MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
 4373  012062  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4374  012064  104401  012072                TYPE    ,67$            ;;TYPE ASCIZ STRING
 4375  012070  000405                        BR      66$             ;;GET OVER THE ASCIZ
 4376                                  ;;67$:  .ASCIZ  *,ERROR # *
 4377  012104                          66$:
 4378  012104  116767  167004  167064        MOVB    $ITEMB,$FATAL   ; APT FATAL ERROR NUMBER
 4379  012112  016746  167060                MOV     $FATAL,-(SP)    ;;SAVE $FATAL FOR TYPEOUT
 4380  012116  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4381  012120  104401  012126                TYPE    ,69$            ;;TYPE ASCIZ STRING
 4382  012124  000404                        BR      68$             ;;GET OVER THE ASCIZ
 4383                                  ;;69$:  .ASCIZ  *,PC = *
 4384  012136                          68$:
 4385  012136  016746  166754                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 4386  012142  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4387  012144  104401  012152                TYPE    ,71$            ;;TYPE ASCIZ STRING
 4388  012150  000404                        BR      70$             ;;GET OVER THE ASCIZ
 4389                                  ;;71$:  .ASCIZ  *,CSR: *
 4390  012162                          70$:
 4391  012162  016746  167066                MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
 4392  012166  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4393  012170  104401  012176                TYPE    ,73$            ;;TYPE ASCIZ STRING
 4394  012174  000405                        BR      72$             ;;GET OVER THE ASCIZ
 4395                                  ;;73$:  .ASCIZ  *,VECTOR: *
 4396  012210                          72$:
 4397  012210  016746  167042                MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
 4398  012214  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4399  012216                          ENDRTN
 4400  012216                          $240:
 4401  012216                          $241:
 4402  012216  000207                        RTS     PC
```

```
 4403  012220                          ROUTINE CYCLE
 4404  012220                          CYCLE:
 4405                                  ;;*****************************************************************
 4406                                  ;* ROUTINE:     CYCLE
 4407                                  ;*       THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
 4408                                  ;*       ADDRESS  OF  DLV11-E  UNDER  TEST,  ADRS +2 TO
 4409                                  ;*       POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
 4410                                  ;*      IT  KEEPS  TRACK  OF THE CURRENT DEVICE AND BIT
 4411                                  ;*      MASKS.
 4412                                  ;;*****************************************************************
 4413  012220                                                  REPEAT
 4414  012220                          $244:
 4415  012220                                                      IF BITMASK EQ #0 THEN
 4416  012220  005767  000122                  TST     BITMASK
 4417  012224  001027                          BNE     $245
 4418  012226                                                          IF INITFLAG EQ #1 THEN
 4419  012226  026727  000116  000001          CMP     INITFLAG,#1
 4420  012234  001003                          BNE     $246
 4421  012236                                                              LET INITFLAG := #0
 4422  012236  005067  000106                  CLR     INITFLAG
 4423  012242                                                          ELSE
 4424  012242  000403                          BR      $247
 4425  012244                          $246:
 4426  012244                                                              CALL $EOP   ; AS A SUBROUTINE
 4427  012244  004767  000110                  JSR     PC,$EOP
 4428
 4429  012250                          SPECIALADDRESS:                 ; BECAUSE $EOP RETURNS AS A JUMP
 4430  012250                                                              LET R0 := POP
 4431  012250  012600                          MOV     (SP)+,R0
 4432  012252                                                          ENDIF
 4433  012252                          $247:
 4434  012252                                                          LET BITMASK := #1
 4435  012252  012767  000001  000066          MOV     #1,BITMASK
 4436  012260                                                          LET $DEVCT := #1
 4437  012260  012767  000001  166716          MOV     #1,$DEVCT
 4438  012266                                                          LET ADDRESS := $BASE
 4439  012266  016767  166756  000056          MOV     $BASE,ADDRESS
 4440  012274                                                          LET VECTOR := $VECT1
 4441  012274  016767  166744  000052          MOV     $VECT1,VECTOR
 4442  012302                                                      ELSE
 4443  012302  000410                          BR      $250
 4444  012304                          $245:
 4445  012304                                                          LET R4 := #10
 4446  012304  012704  000010                  MOV     #10,R4
 4447  012310                                                          LET BITMASK := BITMASK ROTATE 1
 4448  012310  006167  000032                  ROL     BITMASK
 4449  012314                                                          LET ADDRESS  := ADDRESS + R4
 4450  012314  060467  000032                  ADD     R4,ADDRESS
 4451  012320                                                          LET VECTOR := VECTOR + R4
 4452  012320  060467  000030                  ADD     R4,VECTOR
 4453  012324                                                      ENDIF
 4454  012324                          $250:
 4455  012324                                                  UNTIL   BITMASK SETIN $DEVM
 4456  012324  036767  000016  166720          BIT     BITMASK,$DEVM
 4457  012332  001732                          BEQ     $244
 4458
```

```
                                                        LET ADRS := #ADDRESS
012334
012334  012701  012352                MOV     #ADDRESS,ADRS
012340                                                  LET $DEVCT := $DEVCT + #1
012340  005267  166640                INC     $DEVCT
012344                                                  RETURN
012344  000404                        BR      $242
012346  000000              BITMASK:  0
012350  000001              INITFLAG: 1
012352  000000              ADDRESS:  0
012354  000000              VECTOR:   0

012356                                                  ENDRTN
012356              $242:
012356              $243:
012356  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

012360                          $EOP:
012360  000004                         SCOPE
012362  005067  166514                 CLR     $TSTNM          ;;ZERO THE TEST NUMBER
012366  005067  166566                 CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
012372  005267  166604                 INC     $PASS           ;;INCREMENT THE PASS NUMBER
012376  042767  100000  166576         BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
012404  005327                         DEC     (PC)+           ;;LOOP?
012406  000001                 $EOPCT: .WORD   1
012410  003022                         BGT     $DOAGN          ;;YES
012412  012737                         MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
012414  000001                 $ENDCT: .WORD   1
012416  012406                         $EOPCT
012420  104401  012465                 TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
012424  016746  166552                 MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
012430  104405                         TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
012432  104401  012462                 TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
012436  013700  000042         $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
012442  001405                         BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
012444  000005                         RESET                   ;;CLEAR THE WORLD
012446  004710                 $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
012450  000240                         NOP                     ;;SAVE ROOM
012452  000240                         NOP                     ;;FOR
012454  000240                         NOP                     ;;ACT11
012456                          $DOAGN:
012456  000137                         JMP     @(PC)+          ;;RETURN
012460  012250                 $RTNAD: .WORD   SPECIALADDRESS
012462     377     377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
012465     015  042412  042116 $ENDMG: .ASCIZ  <15><12>/END PASS #/
012472  050040  051501  020123
012500  000043
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES

                                        ;;*****************************************************************
                                        ;POWER DOWN ROUTINE
012502  012737  012646  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
012510  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
012516  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
012520  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
012522  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
012524  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
012526  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
012530  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
012532  017746  166402                  MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
012536  010667  000110                  MOV     SP,$SAVR6       ;;SAVE SP
012542  012737  012554  000024          MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
012550  000000                          HALT
012552  000776                          BR      .-2             ;;HANG UP

                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
012554  012737  012646  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
012562  016706  000064                  MOV     $SAVR6,SP       ;;GET SP
012566  005067  000060                  CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
012572  005267  000054          1$:     INC     $SAVR6          ;;WAIT FOR THE INC
012576  001375                          BNE     1$              ;;OF  WORD
012600  012677  166334                  MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
012604  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
012606  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
012610  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
012612  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
012614  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
012616  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
012620  012737  012502  000024          MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
012626  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;PRIO:7
012634  104401                          TYPE                    ;REPORT THE POWER FAILURE
012636  012654          $PWRMG: .WORD   $POWER                  ;;POWER FAIL MESSAGE POINTER
012640  012716                  MOV     (PC)+,(SP)              ;;RESTART AT START
012642  001336          $PWRAD: .WORD   START                   ;;RESTART ADDRESS
012644  000002                  RTI
012646  000000          $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
012650  000776                  BR      .-2                     ;; BEFORE THE POWER DOWN WAS COMPLETE
012652  000000          $SAVR6: 0                               ;;PUT THE SP HERE
012654  005015  047520  042527 $POWER: .ASCIZ  <15><12>'POWER'
012662  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
                                ;*

012664  105767  166267          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
012670  100002                          BPL     1$              ;;BR IF YES
012672  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
012674  000430                          BR      3$              ;;LEAVE
012676  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
012700  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
012704  122767  000001  166302          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
012712  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
012714  132767  000100  166273          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
012722  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
012724  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
012730  004767  000774                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
012734  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
012736  132767  000040  166251  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
012744  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
012746  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
012750  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
012752  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
012754  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
012756  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
012762  000002                          RTI                     ;;RETURN
012764  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
012770  001430                          BEQ     8$
012772  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
012776  001006                          BNE     5$
013000  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
013002  104401                          TYPE                    ;;TYPE A CR AND LF
013004  001171                          $CRLF
013006  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
013012  000755                          BR      2$              ;;GET NEXT CHARACTER
013014  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
013020  126726  166132          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
013024  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
013026  016746  166122                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
013032  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
013036  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
013040  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
013044  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
4616  013050  000770                        BR      7$              ;;LOOP
4617
4618                                  ;HORIZONTAL TAB PROCESSOR
4619
4620  013052  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
4621  013056  004767  000014          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
4622  013062  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
4623  013070  001372                        BNE     9$              ;;TAB STOP
4624  013072  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
4625  013074  000724                        BR      2$              ;;GET NEXT CHARACTER
4626  013076  105777  166046        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
4627  013102  100375                        BPL     $TYPEC
4628  013104  116677  000002  166040        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4629  013112  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
4630  013120  001003                        BNE     1$              ;;BRANCH IF NO
4631  013122  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
4632  013126  000406                        BR      $TYPEX          ;;EXIT
4633  013130  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
4634  013136  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
4635  013140  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
4636  013142  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
4637  013144  000207          $TYPEX: RTS     PC
4638
```

```
                        .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.
013146  022767  000176  165764  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
013154  001074                          BNE     15$             ;;BRANCH IF NO
013156  105777  165762                  TSTB    @$TKS           ;;CHAR THERE?
013162  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
013164  117746  165756                  MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
013170  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
013174  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
013200  001062                          BNE     15$             ;;NO, RETURN TO USER
013202  126727  165726  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
013210  001456                          BEQ     15$             ;;BRANCH IF YES

013212  104401  013673                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
013216  104401  013700          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
013222  016746  164750                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
013226  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013230  104401  013711                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
013234  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
013236  005046                          CLR     -(SP)           ;;THE NEW SWR
013240  105777  165700          7$:     TSTB    @$TKS           ;;CHAR THERE?
013244  100375                          BPL     7$              ;;IF NOT TRY AGAIN

013246  117746  165674                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
013252  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII


013256  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
013262  001005                          BNE     10$             ;;BRANCH IF NOT
013264  104401  013666                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
013270  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
013274  000757                          BR      19$             ;;LET'S TRY IT AGAIN


013276  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
013302  001022                          BNE     16$             ;;BRANCH IF NO
013304  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
013310  001403                          BEQ     11$             ;;BRANCH IF YES
013312  016677  000002  165620          MOV     2(SP),@SWR      ;;SAVE NEW SWR
013320  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
013324  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
013330  126727  165601  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
013336  001003                          BNE     15$             ;;BRANCH IF NOT
013340  012777  000100  165576          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
013346  000002                  15$:    RTI                     ;;RETURN
013350  004767  177522          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
013354  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
```

```
                                        BLT     18$             ;;BRANCH IF YES
                                        CMP     (SP),#67        ;;CHAR > 7?
                                        BGT     18$             ;;BRANCH IF YES
                                        BIC     #60,(SP)+       ;;STRIP-OFF ASCII
                                        TST     2(SP)           ;;IS THIS THE FIRST CHAR
                                        BEQ     17$             ;;BRANCH IF YES
                                        ASL     (SP)            ;;NO, SHIFT PRESENT
                                        ASL     (SP)            ;;   CHAR OVER TO MAKE
                                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
                                17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
                                        BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
                                        BR      7$              ;;GET THE NEXT ONE
                                18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
                                        BR      20$             ;;SIMULATE CONTROL-U
                                .DSABL  LSB


                                ;;*******************************************************************
                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
                                ;

                                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
                                        MOV     4(SP),2(SP)     ;;SAVE THE PS
                                1$:     TSTB    @$TKS           ;;WAIT FOR
                                        BPL     1$              ;;A CHARACTER
                                        MOVB    @$TKB,4(SP)     ;;READ THE TTY
                                        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
                                        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
                                        BNE     3$              ;;BRANCH IF NO
                                2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
                                        BPL     2$              ;;LOOP UNTIL ITS THERE
                                        MOVB    @$TKB,-(SP)     ;;GET CHARACTER
                                        BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
                                        BNE     2$              ;;IF NOT DISCARD IT
                                        BR      1$              ;;YES, RESUME
                                3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
                                        BLT     4$              ;;BRANCH IF YES
                                        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
                                        BGT     4$              ;;BRANCH IF YES
                                        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
                                4$:     RTI                     ;;GO BACK TO USER
                                ;;*******************************************************************
                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

                                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
                                1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
                                2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
```

```
 4751  013562  101405                          BLOS    4$              ;;BR IF YES
 4752  013564  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4753  013566  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4754  013570  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4755  013574  001003                          BNE     3$              ;;SKIP IF NOT
 4756  013576  104401  001170          4$:     TYPE    ,$QUES          ;;TYPE A '?'
 4757  013602  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4758  013604  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 4759  013610  104401  013654                  TYPE    ,9$
 4760  013614  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 4761  013620  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 4762  013622  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4763  013626  104401  001172                  TYPE    ,$LF            ;;TYPE A LINE FEED
 4764  013632  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 4765  013634  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4766  013636  016666  000004  000002          MOV     4(SP),2(SP)     ;;       FIRST ASCII CHARACTER ON IT
 4767  013644  012766  013656  000004          MOV     #$TTYIN,4(SP)
 4768  013652  000002                          RTI                     ;;RETURN
 4769  013654     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 4770  013655     000                          .BYTE   0               ;;TERMINATOR
 4771  013656  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
 4772  013666  052536  005015     000 $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL 'U'
 4773  013673     136  006507  000012 $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL 'G'
 4774  013700  005015  053523  020122 $MSWR:  .ASCIZ  <15><12>/SWR = /
 4775  013706  020075     000
 4776  013711     040  047040  053505 $MNEW:  .ASCIZ  /  NEW = /
 4777  013716  036440  000040
```

```
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;**************************************************************
013722  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
013730  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
013736  000403                         BR      $ATYC
013740  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
013746                          $ATYC:
013746  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013750  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013752  105767  000206                 TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
013756  001450                         BEQ     5$              ;;IF NOT:  BR
013760  122767  000001  165226         CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
013766  001031                         BNE     3$              ;;IF NOT:  BR
013770  132767  000100  165217         BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
013776  001425                         BEQ     3$              ;;IF NOT:  BR
014000  017600  000004                 MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
014004  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
014012  005767  165156          1$:    TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
014016  001375                         BNE     1$              ;;IF NOT:  WAIT
014020  010067  165164                 MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
014024  105720                  2$:    TSTB    (R0)+           ;;FIND END OF MESSAGE
014026  001376                         BNE     2$
014030  166700  165154                 SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
014034  006200                         ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
014036  010067  165150                 MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
014042  012767  000004  165124         MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
014050  000413                         BR      5$
014052  017667  000004  000016  3$:    MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
014060  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
014066  016746  163704                 MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
014072  004767  176566                 JSR     PC,$TYPE        ;;CALL TYPE MACRO
014076  000000                  4$:    .WORD   0
014100                          5$:
014100  105767  000062          10$:   TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
014104  001416                         BEQ     12$             ;;IF NOT:  BR
014106  005767  165102                 TST     $ENV            ;;RUNNING UNDER APT?
014112  001413                         BEQ     12$             ;;IF NOT:  BR
014114  005767  165054          11$:   TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
014120  001375                         BNE     11$             ;;IF NOT:  WAIT
014122  017667  000004  165046         MOV     @4(SP),$FATAL   ;;GET ERROR #
014130  062766  000002  000004         ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
014136  005267  165032                 INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
014142  105067  000020          12$:   CLRB    $FFLG           ;;CLEAR FATAL FLAG
014146  105067  000013                 CLRB    $LFLG           ;;CLEAR LOG FLAG
014152  105067  000006                 CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
014156  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
014160  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
014162  000207                         RTS     PC              ;;RETURN
014164     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
014165     000          $LFLG:  .BYTE   0               ;;LOG FLAG
014166     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
014170                         .EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
 4834           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

014170                                  $ERROR:
014170  104407                                  CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
014172  105267  164705                  7$:     INCB    $ERFLG                  ;;SET THE ERROR FLAG
014176  001775                                  BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
014200  016777  164676  164734                  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
014206  032777  002000  164724                  BIT     #BIT10,@SWR             ;;BELL ON ERROR?
014214  001402                                  BEQ     1$                      ;;NO - SKIP
014216  104401  001164                          TYPE    ,$BELL                  ;;RING BELL
014222  005267  164664                  1$:     INC     $ERTTL                  ;;COUNT THE NUMBER OF ERRORS
014226  011667  164664                          MOV     (SP),$ERRPC             ;;GET ADDRESS OF ERROR INSTRUCTION
014232  162767  000002  164656                  SUB     #2,$ERRPC
014240  117767  164652  164646                  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
014246  032777  020000  164664                  BIT     #BIT13,@SWR             ;;SKIP TYPEOUT IF SET
014254  001004                                  BNE     20$                     ;;SKIP TYPEOUTS
014256  004767  175554                          JSR     PC,MYTYPE               ;;GO TO USER ERROR ROUTINE
014262  104401  001171                          TYPE    ,$CRLF
014266                                  20$:
014266  122767  000001  164720                  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
014274  001007                                  BNE     2$                      ;;NO,SKIP APT ERROR REPORT
014276  116767  164612  000004                  MOVB    $ITEMB,21$              ;;SET ITEM NUMBER AS ERROR NUMBER
014304  004767  177430                          JSR     PC,$ATY4                ;;REPORT FATAL ERROR TO APT
014310     000                  21$:    .BYTE   0
014311     000                          .BYTE   0
014312  000777                  22$:    BR      22$                     ;;APT ERROR LOOP
014314  005777  164620          2$:     TST     @SWR                    ;;HALT ON ERROR
014320  100002                          BPL     3$                      ;;SKIP IF CONTINUE
014322  000000                          HALT                            ;;HALT ON ERROR!
014324  104407                          CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
014326  032777  001000  164604  3$:     BIT     #BIT09,@SWR             ;;LOOP ON ERROR SWITCH SET?
014334  001402                          BEQ     4$                      ;;BR IF NO
014336  016716  164546                  MOV     $LPERR,(SP)             ;;FUDGE RETURN FOR LOOPING
014342  005767  164614          4$:     TST     $ESCAPE                 ;;CHECK FOR AN ESCAPE ADDRESS
014346  001402                          BEQ     5$                      ;;BR IF NONE
014350  016716  164606                  MOV     $ESCAPE,(SP)            ;;FUDGE RETURN ADDRESS FOR ESCAPE
014354                                  5$:
014354  022737  012446  000042                  CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
014362  001001                          BNE     6$                      ;;BRANCH IF NO
014364  000000                          HALT                            ;;YES
014366                                  6$:
014366  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

014370                                  $SCOPE:
014370  104407                                 CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
014372  032777  040000  164540          1$:    BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
014400  001114                                 BNE     $OVER                   ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
014402  000416                          $XTSTR: BR      6$                      ;;IF RUNNING ON THE ''XOR'' TESTER CHANGE
                                                                                ;;THIS INSTRUCTION TO A ''NOP'' (NOP=240)
014404  013746  000004                          MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
014410  012737  014430  000004                  MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
014416  005737  177060                         TST     @#177060                ;;TIME OUT ON XOR?
014422  012637  000004                         MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
014426  000463                                 BR      $SVLAD                  ;;GO TO THE NEXT TEST
014430  022626                          5$:    CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
014432  012637  000004                         MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
014436  000423                                 BR      7$                      ;;LOOP ON THE PRESENT TEST
014440                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
014440  032777  000400  164472                  BIT     #BIT08,@SWR             ;;LOOP ON SPEC. TEST?
014446  001404                                 BEQ     2$                      ;;BR IF NO
014450  127767  164464  164424                 CMPB    @SWR,$TSTNM             ;;ON THE RIGHT TEST?    SWR<7:0>
014456  001465                                 BEQ     $OVER                   ;;BR IF YES
014460  105767  164417                  2$:    TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
014464  001421                                 BEQ     3$                      ;;BR IF NO
014466  126767  164423  164407                 CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
014474  101015                                 BHI     3$                      ;;BR IF NO
014476  032777  001000  164434                 BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
014504  001404                                 BEQ     4$                      ;;BR IF NO
014506  016767  164376  164372          7$:    MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
014514  000446                                 BR      $OVER
014516  105067  164361                  4$:    CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
014522  005067  164432                         CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
014526  000415                                 BR      1$                      ;;ESCAPE TO THE NEXT TEST
014530  032777  004000  164402          3$:    BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
014536  001011                                 BNE     1$                      ;;BR IF YES
014540  005767  164436                         TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
014544  001406                                 BEQ     1$                      ;;      INHIBIT ITERATIONS
014546  005267  164332                         INC     $ICNT                   ;;INCREMENT ITERATION COUNT
014552  026767  164402  164324                 CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
014560  002024                                 BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
014562  012767  000001  164314          1$:    MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
014570  016767  000052  164362                 MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
014576  105267  164300                  $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
014602  116767  164274  164370                 MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
```

```
4945  014610  011667  164272                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
4946  014614  011667  164270                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
4947  014620  005067  164336                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
4948  014624  112767  000001  164263          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
4949  014632  016777  164244  164302  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
4950  014640  016716  164242                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
4951  014644  000002                          RTI                     ;;FIXES PS
4952  014646  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

014650                          $TYPDS:
014650  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
014652  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
014654  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
014656  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
014660  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
014662  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
014666  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
014672  100004                          BPL     1$              ;;BR IF INPUT IS POS.
014674  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
014676  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
014704  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
014706  012703  015064                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
014712  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
014716  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
014720  016001  015054                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
014724  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
014726  002402                          BLT     4$              ;;BR IF DONE
014730  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
014732  000774                          BR      3$
014734  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
014736  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
014740  001002                          BNE     5$              ;;FALL THROUGH IF 0
014742  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
014744  100407                          BMI     7$              ;;BR IF YES
014746  106316                  5$:     ASLB    (SP)            ;;MSD?
014750  103003                          BCC     6$              ;;BR IF NO
014752  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
014760  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
014764  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
014770  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
014772  005720                          TST     (R0)+           ;;JUST INCREMENTING
014774  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
015000  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
015002  003002                          BGT     8$              ;;GO TO EXIT
015004  010502                          MOV     R5,R2           ;;GET THE LSD
015006  000764                          BR      6$              ;;GO CHANGE TO ASCII
015010  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
015012  100003                          BPL     9$              ;;BR IF NO
015014  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
015022  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
015024  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
015026  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
015030  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
5009  015032  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
5010  015034  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
5011  015036  104401  015064                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
5012  015042  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
5013  015050  012616                        MOV     (SP)+,(SP)
5014  015052  000002                        RTI                     ;;RETURN TO USER
5015  015054  023420                $DTBL:  10000.
5016  015056  001750                        1000.
5017  015060  000144                        100.
5018  015062  000012                        10.
5019  015064  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

015074  017646  000000  $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
015100  116667  000001  000211 MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
015106  112667  000207         MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
015112  062716  000002         ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
015116  000406                 BR      $TYPON
015120  112767  000001  000171 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
015126  112767  000006  000165 MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
015134  112767  000005  000154 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
015142  010346                 MOV     R3,-(SP)        ;;SAVE R3
015144  010446                 MOV     R4,-(SP)        ;;SAVE R4
015146  010546                 MOV     R5,-(SP)        ;;SAVE R5
015150  116704  000145         MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
015154  005404                 NEG     R4
015156  062704  000006         ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
015162  110467  000132         MOVB    R4,$OMODE       ;;SAVE IT FOR USE
015166  116704  000125         MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
015172  016605  000012         MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
015176  005003                 CLR     R3              ;;CLEAR THE OUTPUT WORD
015200  006105         1$:     ROL     R5              ;;ROTATE MSB INTO "C"
015202  000404                 BR      3$              ;;GO DO MSB
015204  006105         2$:     ROL     R5              ;;FORM THIS DIGIT
015206  006105                 ROL     R5
015210  006105                 ROL     R5
015212  010503                 MOV     R5,R3
015214  006103         3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
015216  105367  000076         DECB    $OMODE          ;;TYPE THIS DIGIT?
015222  100016                 BPL     7$              ;;BR IF NO
015224  042703  177770         BIC     #177770,R3      ;;GET RID OF JUNK
015230  001002                 BNE     4$              ;;TEST FOR 0
015232  005704                 TST     R4              ;;SUPPRESS THIS 0?
015234  001403                 BEQ     5$              ;;BR IF YES
```

```
 5076  015236  005204                    4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 5077  015240  052703  000060                    BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 5078  015244  052703  000040            5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 5079  015250  110367  000040                    MOVB    R3,8$           ;;SAVE FOR TYPING
 5080  015254  104401  015314                    TYPE    ,8$             ;;GO TYPE THIS DIGIT
 5081  015260  105367  000032            7$:     DECB    $OCNT           ;;COUNT BY 1
 5082  015264  003347                            BGT     2$              ;;BR IF MORE TO DO
 5083  015266  002402                            BLT     6$              ;;BR IF DONE
 5084  015270  005204                            INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 5085  015272  000744                            BR      2$              ;;GO DO THE LAST DIGIT
 5086  015274  012605                    6$:     MOV     (SP)+,R5        ;;RESTORE R5
 5087  015276  012604                            MOV     (SP)+,R4        ;;RESTORE R4
 5088  015300  012603                            MOV     (SP)+,R3        ;;RESTORE R3
 5089  015302  016666  000002  000004            MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 5090  015310  012616                            MOV     (SP)+,(SP)
 5091  015312  000002                            RTI                     ;;RETURN
 5092  015314     000                    8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 5093  015315     000                            .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 5094  015316     000                $OCNT:      .BYTE   0               ;;OCTAL DIGIT COUNTER
 5095  015317     000                $OFILL:     .BYTE   0               ;;ZERO FILL SWITCH
 5096  015320  000000                $OMODE:     .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                .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.

015322  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
015324  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
015330  005740                          TST     -(R0)           ;;BACKUP BY 2
015332  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
015334  006300                          ASL     R0              ;;POSITION FOR INDEXING
015336  016000  015356                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
015342  000200                          RTS     R0              ;;GO TO ROUTINE


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

015344  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
015346  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
015354  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
015356  015344                  $TRPAD: .WORD   $TRAP2
015360  012664                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
015362  015120                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
015364  015074                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
015366  015134                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
015370  014650                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

015372  013216                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

015374  013146                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
015376  013430                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
015400  013550                          $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  012352         4439*  4450*  4460   4467#
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         4590   4834#
APTENV= 000001         2697   2906   3183   3846   4583   4790   4832#   4866
APTSIZ= 000200         1128   4831#
APTSPO= 000100         4585   4792   4833#
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  011322         4113#
BAUD  = 007400          921#
BAUDRA  007210         3320#
```

```
BITMAS  012346          4416    4435*   4448*   4456    4465#
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     4919
BIT09  = 001000          777#    787     4877    4927
BIT1   = 000002          795#
BIT10  = 002000          776#    835     4854
BIT11  = 004000          775#    834     874     4934
BIT12  = 010000          774#    833     853     873
BIT13  = 020000          773#    832     852     872     4861
BIT14  = 040000          772#    831     851     871     4905
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    4081
BRK    = 010000          922#    1260    4063
CABLE  = 020000          923#    1996    2050    2141    2230    2319    2410    2510    2614    3559    3741
CARDET= 010000           833#    2075    2095    2115
CKSWR  = 104407          4850    4876    4904    5136#
CLR    = 000000          824#    2916    4190
CLRSEN= 020000           832#    2164    2184    2204
COMSPD= 000100           916#
CR     = 000015          711#    4629    4639
CRLF   = 000200          712#    1152    4128    4371    4600    4639
CYCLE   012220          1157    4404#
DATA   = 000017          913#    4274    4275
DATAIE= 000040           840#    1739    1752    1755    1769    1772    1786    1791    3572    3583    3616
DATAIN= 100000           830#    1966    2443    2461    2470    2487    2544    2568    2591    2651    2673
DATLNG  011646          3655    3761    3856    4256#
DDISP  = 177570          718#     998    1116
DISPLA  001142           998#    1116*   1124*   4853*   4949*
DISPRE  000174           933#    1124
DLADD   001254          1078#    1161*   1166    1169    1171    1174    1177    1180    1196    1228    4129    4391
DLADDR= 175610           820#    1078    1080    1081    1082    1083    1084
DLVEC   001256          1079#    1164*   3350    3471    3622    3879    4135    4397
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  011022          3904*   3926    3932    3948#   4015    4026*
ERROR  = 100000          850#    3084    3148
ERRVEC= 000004           799#    1113    1114*   1125*   4910    4911*   4913*   4916*
```

```
EVENOD= 000040            915#
FALSE = 000000           4171#   4178
FLAG    011640           4178*   4205*   4210    4233    4245#
FRERR = 020000            852#
FRFD  = 040000            924#   1507    1574    2050    2141    2230    2410    2510
GNS   = ******            932    1151    4127    4133    4139    4370    4376    4383    4389    4395    5128    5129    5130
                         5131    5132    5134    5136    5137    5138
GTSWR = 104406           1146    5134#
HOLD    011066           3967*   3968*   3971    3984#
HOLDSC  011641           4190*   4195*   4201    4246#
HOWLON= 000000           4150#   4176
HT    = 000011            709#   4598    4639
I       001272           1085#   1205*   1226*   1229    1231    4279*   4282*   4284
ILLMEM= 000004            811#   1200    1236
INITFL  012350           1154*   4419    4422*   4466#
INTFLA  012034           1212*   1217    3346*   3386    3390    3412*   3428    3479*   3521    3525    3569*   3595    3599
                         4360*   4363#
INTSRV  012026           1201    3358    3472    4350#
IOTVEC= 000020            804#   1097*   1098*
LF    = 000012            710#   4633    4639
LOOP    001744           1155#   4144
MAINT = 000004            884#   1348    1361    1364    1378    1381    1395    1400    2789    2848    2914    3196    3482
                         3543    3644    3728    3751    3910    4072
MASK  = 000000           4257#   4271*   4287*   4289*   4294*
MAX     006422           2942    2946    3021#
MYTYPE  012036           4366#   4863
NEW     007242           3219*   3221*   3237*   3239*   3248    3256    3258    3276    3278    3326#
NUMBER  011024           3924    3949#   3974    4032
NUMBR   011742           4273*   4274    4276*   4284    4297#
OLD     007246           3192*   3194*   3248    3256    3258    3276*   3278*   3327#
ORERR = 040000            851#   3066    3105    3133
PARITY= 000020            914#
PBAUDS= 004000            874#
PBAUD0= 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    3488    3575    3648    3754    3873
PR1   = 000040            734#
PR2   = 000100            735#
PR3   = 000140            736#
PR4   = 000200            737#
PR5   = 000240            738#
PR6   = 000300            739#
PR7   = 000340            740#   1202    1420    2421    2521    2624    3359    3473    3884    3890
PS    = 177776            713#    714
PSW   = 177776            714#
PWRVEC= 000024            805#   1103*   1104*   4519*   4520*   4529*   4535*   4547*   4548*
RATES   007170           3210    3289    3294#
RBUF    001262           1081#   1171*   1172*   2883    2953    3008    3062    3105    3133    3148    3207    3517    3659
                         3697    3764    3801    4002    4076    4102
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   3485*  3502   3509*  3546*  3572*  3580*  3583*  3585*  3616*
                    3618*  3690   3793   3917*  3943*  4036*  4088
RCVRAC= 004000       834#  2011   2928   2967   2974
RCVRDO= 000200       838#  1901   2805   2829   2864   2886   2971   2994   3011   3227   3503   3691   3794
                    4089
RCVRIE= 000100       839#  1815   1828   1831   1845   1848   1862   1867   3485   3509   3546   3917   4036
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      4752   5137#
RDLIN = 104411      5138#
REC     011072      3882   3994#
REG   = 000004      4150#  4174
REGSAV  011634      4174*  4187   4243#
REQSEN= 000004       843#  1587   1600   1603   1617   1620   1634   1639   2251   2271   2291   2532   2560
                    2583   3580   3585   3618
RESVEC= 000010       800#
RHLD    011162      4006*  4007*  4011   4019   4044#
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      011026      3950#  4019*
SECREC= 002000       835#  2342   2362   2382
SECXMI= 000010       842#  1663   1676   1679   1693   1696   1710   1715   2339   2359   2379   2633   2643
                    2665
SET   = 177777       823#  2719   2762   2803   2862   2931   2940   3501   3676   3689   3780   3792   4087
                    4195
SETCLR= 000006      4150#  4201
SPECIA  012250      4429#  4510
STACK = 001100       704#  1095
```

```
START   001336          936    1088#   4552
STKLMT= 177774          715#
SWR     001140          997#   1093    1115*   1117    1123*   1130*   1144    4527    4540*   4649    4686*   4854    4861
                       4873    4877    4905    4919    4921    4927    4934
SWREG   000176          934#   1123    1144    4649    4662
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          801#
TBUF    001270         1084#   1180*   1181*   2713*   2744*   2796*   2856*   2924*   3042*   3052*   3124*   3216*   3495*
                       3685*   3789*   3971*   4084*
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*   3482*   3543*   3644*   3677
                       3728*   3751*   3781    3910*   3914*   3941*   3978*   4072*   4081*
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         4306#   4317
TIMER   011470         2723    2766    2807    2866    3505    3680    3693    3784    3796    4091    4149#
TIMSAV  011636         4176*   4212    4223*   4244#
TKVEC = 000060          808#
TPVEC = 000064          809#
TRAN    011030         3888    3959#
TRAPVE= 000034          807#   1101*   1102*
TRTVEC= 000014          802#
TRUE  = 000001         4170#   4205    4210    4233
```

```
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   007524          3464#
TST35   010146          3564    3639#
TST36   010362          3719    3737#
TST37   010610          3746    3819    3839#
TST4    002552          1416#
TST40   011166          3850    3946    4059#
TST41   011350          4067    4112    4123#
TST5    002726          1503#
TST6    003066          1511    1570#
TST7    003250          1578    1655#
TYPDS = 104405          4142    4499    5132#
TYPE  = 104401          1138    4125    4131    4137    4368    4374    4381    4387    4393    4497    4500    4549    4603
                        4660    4661    4664    4677    4688    4707    4756    4759    4763    4856    4864    5011    5080
                        5128#
TYPOC = 104402          4130    4136    4373    4380    4386    4392    4398    4663    5129#
TYPON = 104404          5131#
TYPOS = 104403          5130#
VECTOR  012354          4441*   4452*   4468#
WAIT    011746          2436    2537    2565    2588    2648    2670    3047    3057    3129    3381    3425    3514    3590
                        4220    4305#
WAS     011027          3951#   4021*
WHICHB= 000002          4150#   4187
XMITIE= 000100           880#   1430    1443    1446    1460    1463    1477    1482    3365    3375    3415    3914    3941
                        3943    3978
XMITRD= 000200           879#   1934    2721    2749    2764    3678    3782
ZZZ     011070          3982    3986#
ZZZZ    011164          4042    4045#
$APTHD  001000           957     963#
$ASTAT= ****** U        4812    4827
$ATYC   013746          4783    4785#
$ATY1   013722          4781#
$ATY3   013730          4588    4782#
$ATY4   013740          4784#   4869
```

```
$AUTOB  001134   994#  1148*  4657  4778
$BASE   001250  1058#  4439
$BDADR  001122   989#
$BDDAT  001126   991#
$BELL   001164  1009#  4856  4889
$BGNLE= 177777     1#
$CHARC  013142  4605*  4615*  4622  4631*  4636#
$CKSWR  013146  4649#  5136
$CMTAG  001100   977#  1090  1091  1099  1105  1106  1107
$CM3  = 000000  1007#
$CNTLG  013673  4660  4773#
$CNTLU  013666  4677  4772#
$CPUOP  001222  1032#
$CRLF   001171  1011#  4604  4639  4688  4772  4864  4889
$DBLK   015064  4977  5011  5019#
$DEVCT  001204  1023#  4437*  4462*
$DEVM   001252  1059#  4456
$DOAGN  012456  4493  4502  4508#
$DTBL   015054  4980  5015#
$ENDAD  012446   943  1136  4504#  4884
$ENDCT  012414  1105  4495#
$ENDMG  012465  4497  4512#
$ENULL  012462  4500  4511#
$ENV    001214  1028#  1142  2697  2906  3183  3846  4583  4790  4814  4866
$ENVM   001215  1029#  1128  4585  4590  4792
$EOP    012360  4427  4485#
$EOPCT  012406  1105*  4492#  4496
$ERFLG  001103   980#  4851*  4889  4894  4923  4925  4931*  4953
$ERMAX  001115   986#  1108*  4925  4948*  4953
$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#   3443#   3445#   3446#   3447    3470#   3471#   3472#
                    3473#   3474#   3475    3477#   3478    3479#   3480    3482#   3483    3485#   3486    3495#   3496
                    3500#   3509#   3510    3512#   3517#   3518    3543#   3544    3546#   3547    3557#   3558    3563#
                    3564    3569#   3570    3572#   3573    3580#   3581    3583#   3584    3585#   3586    3588#   3616#
                    3617    3618#   3619    3622#   3623    3626#   3627#   3629#   3630#   3631    3644#   3645    3654#
                    3659#   3660    3663#   3664    3666    3670    3675#   3685#   3686    3688#   3697#   3698    3705#
                    3707    3708#   3709    3718#   3719    3728#   3729    3745#   3746    3751#   3752    3760#   3764#
                    3765    3767#   3768    3770    3774    3779#   3789#   3790    3791#   3801#   3802    3805#   3807
                    3808#   3809    3818#   3819    3849#   3850    3855#   3879#   3880    3882#   3883    3884#   3885
                    3888#   3889    3890#   3891    3894#   3895    3897#   3898    3900#   3901    3904#   3905    3910#
                    3911    3914#   3915    3917#   3918    3941#   3942    3943#   3944    3964#   3965    3967#   3969
                    3971#   3972    3978#   3979    3999#   4000    4002#   4004    4006#   4008    4019#   4020    4021#
                    4022    4026#   4027    4036#   4037    4066#   4067    4072#   4073    4076#   4077    4081#   4082
                    4084#   4085    4086#   4174#   4175    4176#   4177    4178#   4179    4190#   4191    4195#   4196
                    4205#   4206    4218#   4223#   4224    4236#   4237    4240#   4242    4271#   4272    4273#   4277
                    4279#   4280    4282    4286    4287#   4288    4289#   4290    4294#   4295    4317#   4318    4319#
                    4320    4322    4326    4327#   4328    4330    4334    4360#   4361    4422#   4423    4427#   4431#
                    4432    4435#   4436    4437#   4438    4439#   4440    4441#   4442    4446#   4447    4448#   4449
                    4450#   4451    4452#   4453    4460#   4461    4462#   4463
$ERROR  014170      1099    4849#
$ERRPC  001116       987#   4385    4858*   4859*   4860    4889
$ERRTB  001254      1076#
$ERTTL  001112       984#   4141    4143*   4857*   4889
$ESCAP  001162      1008#   1107*   4880    4882    4889    4947*
$ETABL  001214      1027#
$ETEND  001254       969    1060#
$FATAL  001176      1020#   4378*   4379    4818*
$FFLG   014166      4781*   4784*   4812    4821*   4829#
$FILLC  001156      1005#   4608    4639
$FILLS  001155      1004#   4639
$F$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    3523    3527    3561    3597    3601    3714    3743    3814    3848
                    3934    3976    4013    4017    4034    4065    4104    4189    4203    4235    4418    4421
$F$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    3523    3527    3561    3597    3601    3714    3743    3814    3848
                    3934    3976    4013    4017    4034    4065    4104    4189    4203    4235    4418    4421
$F$BLA= 000170         1#
$F$CAS= 000150         1#
$F$DEC= 000220         1#
```

```
$F$GOO= 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  3066  3084  3105  3133  3148
                 3174  3183  3223  3224  3227  3248  3256  3386  3390  3428  3521  3525  3559
                 3595  3599  3712  3741  3812  3846  3932  3974  4011  4015  4032  4063  4094
                 4102  4187  4201  4233  4416  4419
$F$IF  = 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  3521  3525  3532  3538  3540  3559  3566  3595  3599
                 3606  3612  3614  3712  3721  3741  3748  3812  3821  3846  3852  3932  3938
                 3974  3980  4011  4015  4023  4028  4032  4039  4063  4069  4094  4099  4102
                 4108  4187  4192  4197  4201  4207  4233  4238  4416  4419  4424  4433  4443
                 4454
$F$INC= 000210      1#  3199  3281  3663  3723  3767  3824  4279  4291  4319  4327  4335  4338
$F$LOO= 000200      1#  4184  4210  4225
$F$NAM= 000160      1#
$F$NO  = 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  3521  3525  3559  3595  3599  3712  3741  3812
                 3846  3932  3974  4011  4015  4032  4063  4187  4416  4419
$F$OR  = 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  3523  3527  3561  3597  3601  3714  3743  3814
                 3848  3934  3976  4013  4017  4034  4065  4104  4189  4203  4235  4418  4421
$F$RTN= 000300      1#  4150  4249  4257  4299  4306  4344  4367  4400  4405  4471
$F$SEL= 000140      1#
$F$UNT= 000130      1#  1207  1231  2926  2940  3922  3924  4414  4456
$F$WHI= 000120      1#  2966  2967  2989  3223  3224  3244  3258
$F$YES= 000402      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    3521    3525    3559    3595    3599    3712    3741    3812
                    3846    3932    3974    4011    4015    4032    4063    4102    4187    4201    4233    4416    4419
$GDADR  001120       988#
$GDDAT  001124       990#
$GET42  012436      4501#
$GTSWR  013216      4661#   5134
$HD   = 000000       689
$HIBTS  001000       964#
$ICNT   001104       981#   4938*   4939    4941*   4952
$IFLEV= 177777         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#
                    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#
                    3241#   3248#   3256#   3271#   3273#   3386#   3390#   3403#   3405#   3428#   3434#   3521#   3525#
                    3538#   3540#   3559#   3566#   3595#   3599#   3612#   3614#   3712#   3721#   3741#   3748#   3812#
                    3821#   3846#   3852#   3932#   3938#   3974#   3980#   4011#   4015#   4023#   4028#   4032#   4039#
                    4063#   4069#   4094#   4099#   4102#   4108#   4187#   4197#   4201#   4207#   4233#   4238#   4416#
                    4419#   4433#   4454#
$ILLUP  012646      4519    4535    4554#
$INTAG  001135       995#   4689    4778
$ISK0 = 000001      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#   2938
                    2946#   2958    2971#   2987    2994#   3001    3011#   3017    3066#   3077    3084#   3097    3105#
                    3115    3133#   3145    3148#   3155    3174#   3180    3183#   3189    3227#   3241    3248#   3273
                    3386#   3405    3428#   3434    3521#   3540    3559#   3566    3595#   3614    3712#   3721    3741#
                    3748    3812#   3821    3846#   3852    3932#   3938    3974#   3980    4011#   4028    4032#   4039
                    4063#   4069    4094#   4099    4102#   4108    4187#   4197    4201#   4207    4233#   4238    4416#
                    4454
$ISK1 = 000001      2974#   2985    3256#   3271    3390#   3403    3525#   3538    3599#   3612    4015#   4023    4419#
                    4433
```

```
$ITEMB  001114   985#  4378  4860*  4868  4889
$LF     001172  1012#  4639  4763   4772  4889
$LFLG   014165  4822*  4828#
$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  3522  3523  3526  3527  3532
                3533  3534  3538   3539  3540  3541  3560  3561  3566  3567  3596  3597  3600
                3601  3606  3607   3608  3612  3613  3614  3615  3664  3665  3666  3667  3668
                3669  3670  3713   3714  3721  3722  3723  3724  3725  3742  3743  3748  3749
                3768  3769  3770   3771  3772  3773  3774  3813  3814  3821  3822  3824  3825
                3826  3847  3848   3852  3853  3922  3923  3925  3926  3927  3928  3929  3933
                3934  3938  3939   3975  3976  3980  3981  4012  4013  4016  4017  4023  4024
                4028  4029  4033   4034  4039  4040  4064  4065  4069  4070  4094  4095  4099
                4100  4103  4104   4108  4109  4149  4184  4185  4188  4189  4192  4193  4194
                4197  4198  4202   4203  4207  4208  4211  4212  4213  4214  4225  4226  4227
                4234  4235  4236   4237  4238  4239  4241  4242  4249  4250  4251  4252  4256
                4280  4281  4282   4283  4284  4285  4286  4291  4292  4293  4296  4297  4299
                4300  4301  4305   4320  4321  4322  4323  4324  4325  4326  4328  4329  4330
                4331  4332  4333   4334  4335  4336  4337  4338  4339  4340  4344  4345  4346
                4366  4400  4401   4402  4404  4414  4415  4417  4418  4420  4421  4424  4425
                4426  4433  4434   4443  4444  4445  4454  4455  4457  4458  4464  4465  4471
                4472  4473
$LPADR  001106   982#  1109*  4929*  4945*  4950  4952
$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* 3477* 3557* 4879  4929  4946* 4952
$LSTCN= 177777        1#  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  3521  3523
                    3525  3527  3533  3534  3538  3540  3559  3561  3566  3595  3597  3599  3601
                    3607  3608  3612  3614  3663  3665  3666  3667  3670  3712  3714  3721  3723
                    3724  3741  3743  3748  3767  3769  3770  3771  3774  3812  3814  3821  3824
                    3825  3846  3848  3852  3922  3923  3924  3929  3932  3934  3938  3974  3976
                    3980  4011  4013  4015  4017  4023  4028  4032  4034  4039  4063  4065  4069
                    4094  4095  4099  4102  4104  4108  4150  4184  4185  4187  4189  4193  4194
                    4197  4201  4203  4207  4225  4226  4233  4235  4238  4249  4257  4279  4281
                    4282  4283  4286  4291  4292  4299  4306  4319  4321  4322  4323  4326  4327
                    4329  4330  4331  4334  4335  4336  4338  4339  4344  4367  4400  4405  4414
                    4415  4416  4418  4419  4421  4425  4426  4433  4444  4445  4454  4456  4471
$LSTIN= 000000        1#  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	3442	3443	3444	3445	3446	3447	3470	3471
3472	3473	3474	3475	3477	3478	3479	3480	3482	3483	3485	3486	3495
3496	3500	3501	3502	3503	3504	3505	3506	3507	3509	3510	3512	3513
3514	3515	3516	3517	3518	3521	3522	3523	3525	3526	3527	3532	3533
3543	3544	3546	3547	3557	3558	3559	3560	3561	3563	3564	3569	3570
3572	3573	3580	3581	3583	3584	3585	3586	3588	3589	3590	3591	3592
3595	3596	3597	3599	3600	3601	3606	3607	3616	3617	3618	3619	3622
3623	3626	3627	3628	3629	3630	3631	3644	3645	3654	3655	3656	3657
3659	3660	3663	3664	3665	3666	3667	3668	3669	3670	3675	3676	3677
3678	3679	3680	3681	3682	3685	3686	3688	3689	3690	3691	3692	3693

H 11

3694 3695 3697 3698 3705 3706 3707 3708 3709 3712 3713 3714 3718
3719 3723 3724 3728 3729 3741 3742 3743 3745 3746 3751 3752 3760
3761 3762 3763 3764 3765 3767 3768 3769 3770 3771 3772 3773 3774
3779 3780 3781 3782 3783 3784 3785 3786 3789 3790 3791 3792 3793
3794 3795 3796 3797 3798 3801 3802 3805 3806 3807 3808 3809 3812
3813 3814 3818 3819 3824 3825 3846 3847 3848 3849 3850 3855 3856
3857 3858 3879 3880 3882 3883 3884 3885 3888 3889 3890 3891 3894
3895 3897 3898 3900 3901 3904 3905 3910 3911 3914 3915 3917 3918
3924 3925 3926 3927 3928 3932 3933 3934 3941 3942 3943 3944 3964
3965 3967 3968 3969 3971 3972 3974 3975 3976 3978 3979 3999 4000
4002 4003 4004 4006 4007 4008 4011 4012 4013 4015 4016 4017 4019
4020 4021 4022 4026 4027 4032 4033 4034 4036 4037 4063 4064 4065
4066 4067 4072 4073 4076 4077 4081 4082 4084 4085 4086 4087 4088
4089 4090 4091 4092 4093 4094 4095 4102 4103 4104 4174 4175 4176
4177 4178 4179 4187 4188 4189 4190 4191 4192 4193 4195 4196 4201
4202 4203 4205 4206 4210 4211 4212 4213 4214 4218 4219 4220 4221
4222 4223 4224 4225 4226 4233 4234 4235 4236 4237 4240 4241 4242
4250 4251 4252 4253 4271 4272 4273 4274 4275 4276 4277 4279 4280
4281 4282 4283 4284 4285 4286 4287 4288 4289 4290 4291 4292 4294
4295 4296 4297 4301 4302 4317 4318 4319 4320 4321 4322 4323 4324
4325 4326 4327 4328 4329 4330 4331 4332 4333 4334 4335 4336 4338
4339 4346 4347 4360 4361 4402 4403 4416 4417 4418 4419 4420 4421
4422 4423 4424 4425 4427 4428 4431 4432 4435 4436 4437 4438 4439
4440 4441 4442 4443 4444 4446 4447 4448 4449 4450 4451 4452 4453
4456 4457 4458 4460 4461 4462 4463 4464 4465 4473 4474

$LSTST= 177777 1# 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
1689 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 3521 3523 3525 3527 3532 3533 3534 3538 3540 3559 3561
3566 3595 3597 3599 3601 3606 3607 3608 3612 3614 3663 3665 3666
3667 3670 3712 3714 3721 3723 3724 3741 3743 3748 3767 3769 3770
3771 3774 3812 3814 3821 3824 3825 3846 3848 3852 3922 3923 3924
3932 3934 3938 3974 3976 3980 4011 4013 4015 4017 4023 4028 4032
4034 4039 4063 4065 4069 4094 4095 4099 4102 4104 4108 4150 4184

I 11

```
                    4185    4187    4189    4192    4193    4194    4197    4201    4203    4207    4210    4225    4226
                    4233    4235    4238    4249    4257    4279    4281    4282    4283    4286    4291    4292    4299
                    4306    4319    4321    4322    4323    4326    4327    4329    4330    4331    4334    4335    4336
                    4338    4339    4344    4367    4400    4405    4414    4415    4416    4418    4419    4421    4424
                    4425    4426    4433    4443    4444    4445    4454    4456    4471
$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    3533    3534    3538    3539    3540    3541    3566    3567    3607    3608    3612
                    3613    3614    3615    3665    3666    3667    3668    3721    3722    3724    3725    3748    3749
                    3769    3770    3771    3772    3821    3822    3825    3826    3852    3853    3922    3923    3928
                    3929    3938    3939    3980    3981    4023    4024    4028    4029    4039    4040    4069    4070
                    4099    4100    4108    4109    4149    4150    4184    4185    4193    4194    4197    4198    4207
                    4208    4226    4227    4238    4239    4249    4250    4251    4252    4256    4257    4281    4282
                    4283    4284    4292    4293    4299    4300    4301    4305    4306    4321    4322    4323    4324
                    4329    4330    4331    4332    4336    4337    4339    4340    4344    4345    4346    4366    4367
                    4400    4401    4402    4404    4405    4414    4415    4425    4426    4433    4434    4444    4445
                    4454    4455    4471    4472    4473
$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    3466    3641    3739    3841    4061    4583    4866
                    4944
$MAMS1  001224      1039#
$MAMS2  001230      1047#
$MAMS3  001234      1050#
$MAMS4  001240      1053#
$MBADR  001002       965#
$MCALL= 000000         1#
$MFLG   014164      4782*   4788    4823*   4827#
$MNEW   013711      4664    4776#
$MSGAD  001210      1025#   4798*   4801
$MSGLG  001212      1026#   4803*
$MSGTY  001174      1019#   4796    4804*   4816    4820*
$MSWR   013700      4661    4774#
$MTYP1  001225      1040#
$MTYP2  001231      1048#
$MTYP3  001235      1051#
$MTYP4  001241      1054#
```

J 11

$MXCNT 014646 4942 4952#
$NEST = 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# 2928# 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#
3521# 3525# 3532 3538# 3540# 3559# 3566# 3595# 3599# 3606 3612# 3614# 3663#
3712# 3721# 3723# 3741# 3748# 3767# 3812# 3821# 3824# 3846# 3852# 3922# 3924#
3932# 3938# 3974# 3980# 4011# 4015# 4023# 4028# 4032# 4039# 4063# 4069# 4094#
4099# 4102# 4108# 4150# 4184# 4187# 4192 4197# 4201# 4207# 4210 4225# 4233#
4238# 4249# 4257# 4279# 4291# 4299# 4306# 4319# 4327# 4335# 4338# 4344# 4367#
4400# 4405# 4414# 4416# 4419# 4424 4433# 4443 4454# 4456# 4471#
$NSKO = 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 2965# 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 3521# 3540 3559# 3566 3595# 3614 3663# 3723 3741# 3748 3767#
3824 3846# 3852 3922# 3924 3932# 3938 3974# 3980 4011# 4028 4032# 4039
4063# 4069 4094# 4099 4102# 4108 4150# 4249 4257# 4299 4306# 4344 4367#
4400 4405# 4471
$NSK1 = 000130 1217# 1223 2928# 2933 2938 2971# 2987 3223# 3244 3248# 3252 3273 3390#
3396 3403 3525# 3532 3538 3599# 3606 3612 3712# 3721 3812# 3821 4015#
4023 4184# 4210 4225 4233# 4238 4279# 4291 4319# 4338 4414# 4456
$NSK2 = 000110 2974# 2985 3227# 3233 3241 3256# 3262 3271 4187# 4192 4197 4201# 4207
4327# 4335 4416# 4443 4454
$NSK3 = 000110 4419# 4424 4433
$NULL 001154 1003# 4610 4639
$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 3458# 3460 3635# 3637 3734# 3835# 3837 4054# 4056 4120#
$OCNT 015316 5052* 5081* 5094#
$OMODE 015320 5047* 5051* 5056 5059* 5070* 5096#
$OVER 014632 4906 4922 4930 4940 4949#
$PASS 001202 1022# 1127* 4489* 4490* 4498 4511 4936 4953

```
$PASTM  001006          967#
$POWER  012654         4550    4557#
$PWRAD  012642         4552#
$PWRDN  012502         1103    4519#   4547
$PWRMG  012636         4550#
$PWRUP  012554         4529    4535#
$QUES   001170         1010#   4639    4707    4756    4772    4889
$RDCHR  013430         4720#   5137
$RDDEC= ****** U       5139
$RDLIN  013550         4748#   5138
$RDOCT= ****** U       5139
$RDSZ = 000010         4741#
$RTNAD  012460         4510#
$R2A  = ****** U       5139
$SAVLE= 177777            1#   2989#   2990#   3202#   3206#   3244#   3245#   3666#   3670#   3770#   3774#   4225#   4226#
                       4282#   4286#   4322#   4326#   4330#   4334#
$SAVRE= ****** U       5139
$SAVR6  012652         4528*   4536    4537*   4538*   4556#
$SCOPE  014370         1097    4903#
$SETUP= 000137         1088#   1096    1097    1099    1101    1103    1105    1106    1107    1109    1136    1139    4487
                       4644    4778    4850    4876    4884    4904
$SSKO = 000236         2989#   2990    3202#   3206    3244#   3245    3666#   3670    3770#   3774    4225#   4226    4282#
                       4286    4322#   4326    4330#   4334
$STUP = 177777         1088#
$SVLAD  014576         4914    4943#
$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    3465    3640    3738    3840    4060    4124    4482    4488    4503
                       4509    4511    4553    4841    4842    4843    4844    4845    4854    4861    4873    4877    4889
                       4895    4896    4897    4898    4899    4905    4917    4919    4920    4923    4924    4925    4932
                       4933    4934    4946    4949    4952
$SWREG  001216         1030#   1130
$SWRMK= 000000          699     700    4899    4900    4921
$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#   3388#   3392#   3397#   3398#   3403#   3405#   3430#   3434#   3523#   3527#   3533#
                       3534#   3538#   3540#   3561#   3566#   3597#   3601#   3607#   3608#   3612#   3614#   3665#   3667#
                       3670#   3714#   3721#   3723#   3724#   3743#   3748#   3769#   3771#   3774#   3814#   3821#   3824#
                       3825#   3848#   3852#   3923#   3924#   3934#   3938#   3976#   3980#   4013#   4017#   4023#   4028#
                       4034#   4039#   4065#   4069#   4095#   4099#   4104#   4108#   4185#   4189#   4193#   4194#   4197#
```

```
                    4203#  4207#  4210   4225#  4235#  4238#  4281#  4283#  4286#  4291#  4292#  4321#  4323#
                    4326#  4329#  4331#  4334#  4335#  4336#  4338#  4339#  4415#  4418#  4421#  4425#  4426#
                    4433#  4444#  4445#  4454#  4456#
$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   3068#
                    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#  3522   3523#  3526   3527#  3532   3534#  3560   3561#  3596   3597#
                    3600   3601#  3606   3608#  3664   3665#  3666#  3669   3670#  3713   3714#  3742   3743#
                    3768   3769#  3770#  3773   3774#  3813   3814#  3847   3848#  3922   3923#  3925   3928
                    3929#  3933   3934#  3975   3976#  4012   4013#  4016   4017#  4033   4034#  4064   4065#
                    4094   4095#  4103   4104#  4150#  4184   4185#  4188   4189#  4192   4194#  4202   4203#
                    4234   4235#  4257#  4280   4281#  4282#  4285   4286#  4306#  4320   4321#  4322#  4325
                    4326#  4328   4329#  4330#  4333   4334#  4367#  4405#  4414   4415#  4417   4418#  4420
                    4421#  4424   4426#  4443   4445#
$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#  3442#
                    3443#  3445#  3446#  3447#  3470#  3471#  3472#  3473#  3474#  3475#  3477#  3478#  3479#
                    3480#  3482#  3483#  3485#  3486#  3495#  3496#  3509#  3510#  3517#  3518#  3532#  3533#
                    3538#  3540#  3543#  3544#  3546#  3547#  3557#  3558#  3563#  3564#  3566#  3569#  3570#
                    3572#  3573#  3580#  3581#  3583#  3584#  3585#  3586#  3606#  3607#  3612#  3614#  3616#
                    3617#  3618#  3619#  3622#  3623#  3626#  3627#  3629#  3630#  3631#  3644#  3645#  3659#
                    3660#  3667#  3670#  3685#  3686#  3697#  3698#  3705#  3707#  3708#  3709#  3718#  3719#
                    3721#  3723#  3724#  3728#  3729#  3745#  3746#  3748#  3751#  3752#  3764#  3765#  3771#
                    3774#  3789#  3790#  3801#  3802#  3805#  3807#  3808#  3809#  3818#  3819#  3821#  3824#
                    3825#  3849#  3850#  3852#  3879#  3880#  3882#  3883#  3884#  3885#  3888#  3889#  3890#
                    3891#  3894#  3895#  3897#  3898#  3900#  3901#  3904#  3905#  3910#  3911#  3914#  3915#
                    3917#  3918#  3924#  3927    3938#  3941#  3942#  3943#  3944#  3964#  3965#  3967#  3969#
                    3971#  3972#  3978#  3979#  3980#  3999#  4000#  4002#  4004#  4006#  4008#  4019#  4020#
                    4021#  4022#  4023#  4026#  4027#  4028#  4036#  4037#  4039#  4066#  4067#  4069#  4072#
                    4073#  4076#  4077#  4081#  4082#  4084#  4085#  4099#  4108#  4174#  4175#  4176#  4177#
                    4178#  4179#  4190#  4191#  4192#  4193#  4195#  4196#  4197#  4205#  4206#  4207#  4210#
                    4211    4213   4223#  4224#  4225#  4226#  4238#  4249#  4271#  4272#  4273#  4277#  4283#
                    4286#  4287#  4288#  4289#  4290#  4291#  4292#  4294#  4295#  4299#  4317#  4318#  4323#
                    4326#  4331#  4334#  4335#  4336#  4338#  4339#  4344#  4360#  4361#  4400#  4422#  4423#
                    4424#  4425#  4431#  4432#  4433#  4435#  4436#  4437#  4438#  4439#  4440#  4441#  4442#
                    4443#  4444#  4446#  4447#  4448#  4449#  4450#  4451#  4452#  4453#  4454#  4456#  4457
                    4460#  4461#  4462#  4463#  4471#
$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*  3466*  3641*  3739*  3841*  4061*  4372   4944*
$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*  3465*  3563*  3640*
                    3718*  3738*  3745*  3818*  3840*  3849*  4060*  4066*  4124*  4488*  4932*  4939   4942*
                    4952
$TKB    001146      1000#  4642   4653   4670   4724   4730
$TKS    001144       999#  4642   4651   4667   4691*  4722   4728
$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   3458   3465#  3466   3564   3635   3640#  3641   3719   3734
                    3738#  3739   3746   3819   3835   3840#  3841   3850   3946   4054   4060#  4061   4067
                    4112   4120   4124#
$TPB    001152      1002#  4628*  4639
```

```
$TPFLG  001157          1006#   4577    4639
$TPS    001150          1001#   4626    4639
$TRAP   015322          1101    5105#
$TRAP2  015344          5116#   5127
$TRP  = 000012          5120#   5129#   5130#   5131#   5132#   5133#   5134    5135#   5136    5137#   5138#   5139#
$TRPAD  015356          5110    5127#
$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#   3282
                        3388#   3405    3430#   3434    3523#   3540    3561#   3566    3597#   3614    3665#   3667    3670#
                        3724    3743#   3748    3769#   3771    3774#   3825    3848#   3852    3923#   3924    3934#   3938
                        3976#   3980    4013#   4028    4034#   4039    4065#   4069    4095#   4099    4104#   4108    4185#
                        4225    4235#   4238    4281#   4283    4286#   4292    4321#   4323    4326#   4339    4415#   4456
$TSK1 = 000250          1219#   1223    2930#   2934    2935#   2938    2969#   2989    3206#   3281    3392#   3397    3398#
                        3403    3527#   3533    3534#   3538    3601#   3607    3608#   3612    3670#   3723    3774#   3824
                        4017#   4023    4185#   4210    4225    4286#   4291    4326#   4338    4418#   4444    4445#   4454
$TSK2 = 000247          2973#   2987    3224#   3244    3250#   3253    3254#   3273    3714#   3721    3814#   3821    4189#
                        4193    4194#   4197    4203#   4207    4329#   4331    4334#   4336    4421#   4425    4426#   4433
$TSK3 = 000236          2976#   2985    3226#   3244    3260#   3263    3264#   3271    4334#   4335
$TSK4 = 000147          3229#   3234    3235#   3241
$TSTM   001004           966#
$TSTNM  001102           979#   4487*   4853    4889    4894    4921    4943*   4944    4949    4953
$TTYIN  013656          4749    4750    4767    4771#
$TYPBN= ****** U        5133
$TYPDS  014650          4965#   5132
$TYPE   012664          4577#   4809    5120    5128
$TYPEC  013076          4607    4614    4621    4626#   4627    4693
$TYPEX  013144          4632    4634    4637#
$TYPOC  015120          5050#   5129
$TYPON  015134          5049    5052#   5131
$TYPOS  015074          5045#   5130
$UNIT   001206          1024#
$UNITM  001010           968#
$USWR   001220          1031#   1260    1507    1574    1996    2050    2141    2230    2319    2410    2510    2614    3174
                        3284    3559    3741    4063    4273
$VECT1  001244          1056#   4441
$VECT2  001246          1057#
$XTSTR  014402          4908#
$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#
```

```
                    2486#  2532#  2533#  2560#  2561#  2583#  2584#  2633#  2634#  2643#  2644#  2665#  2666#
                    2789#  2790#  2848#  2849#  2914#  2915#  2936#  2937#  3196#  3197#  3202#  3203#  3237#
                    3238#  3239#  3240#  3285#  3288#  3353#  3354#  3365#  3366#  3375#  3376#  3415#  3416#
                    3444#  3445#  3482#  3483#  3485#  3486#  3509#  3510#  3543#  3544#  3546#  3547#  3572#
                    3573#  3580#  3581#  3583#  3584#  3585#  3586#  3616#  3617#  3618#  3619#  3628#  3629#
                    3644#  3645#  3666#  3667#  3706#  3707#  3708#  3709#  3728#  3729#  3751#  3752#  3770#
                    3771#  3806#  3807#  3808#  3809#  3910#  3911#  3914#  3915#  3917#  3918#  3941#  3942#
                    3943#  3944#  3964#  3965#  3968#  3969#  3978#  3979#  3999#  4000#  4003#  4004#  4007#
                    4008#  4026#  4027#  4036#  4037#  4072#  4073#  4081#  4082#  4223#  4224#  4274#  4277#
                    4282#  4283#  4287#  4288#  4289#  4290#  4294#  4295#  4322#  4323#  4330#  4331#  4360#
                    4361#  4448#  4449#  4450#  4451#  4452#  4453#  4462#  4463#
$$ARGC= 000000      4150#  4257#  4306#  4367#  4405#
$$BYTE= 000403      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#  3521#  3525#  3559#  3595#  3599#  3712#  3741#  3812#  3846#
                    3932#  3974#  4011#  4015#  4032#  4063#  4102#  4187#  4201#  4233#  4416#  4419#
$$DST = 000067      3285#  4274#
$$FLAG= 000001      1217#  1219   1223#  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#
                    2733#  2749#  2751   2755#  2769#  2774#  2812#  2817#  2829#  2831   2835#  2870#  2875#
                    2886#  2888   2892#  2906#  2908   2912#  2928#  2930   2938#  2946#  2948   2958#  2966#
                    2967#  2969   2971#  2973   2974#  2976   2985#  2987#  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#  3223#  3224#  3226   3227#
                    3229   3241#  3248#  3250   3256#  3258#  3260   3271#  3273#  3386#  3388   3390#  3392
                    3403#  3405#  3428#  3430   3434#  3521#  3523   3525#  3527   3538#  3540#  3559#  3561
                    3566#  3595#  3597   3599#  3601   3612#  3614#  3712#  3714   3721#  3741#  3743   3748#
                    3812#  3814   3821#  3846#  3848   3852#  3932#  3934   3938#  3974#  3976   3980#  4011#
                    4013   4015#  4017   4023#  4028#  4032#  4034   4039#  4063#  4065   4069#  4094#  4099#
                    4102#  4104   4108#  4187#  4189   4197#  4201#  4203   4207#  4233#  4235   4238#  4416#
                    4418   4419#  4421   4433#  4454#
$$FROM= 000000      1157#  2434#  2535#  2563#  2586#  2646#  2668#  2718#  2761#  2802#  2861#  3045#  3055#
                    3127#  3379#  3423#  3500#  3512#  3588#  3654#  3675#  3688#  3760#  3779#  3791#  3855#
                    4086#  4218#  4427#
$$GET4= 000000      4503#
$$LOC = 012332      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   3387#  3388   3391#  3392   3429#  3430   3522#  3523   3526#  3527   3560#  3561
                    3596#  3597   3600#  3601   3713#  3714   3742#  3743   3813#  3814   3847#  3848   3927#
                    3928   3933#  3934   3975#  3976   4012#  4013   4016#  4017   4033#  4034   4064#  4065
                    4094#  4095   4103#  4104   4188#  4189   4202#  4203   4234#  4235   4417#  4418   4420#
                    4421   4457#  4458
$$LOCN= 000000      4150#  4252   4257#  4301   4306#  4346   4367#  4402   4405#  4473
$$RETU= 000000      4150#  4236#  4240#  4250   4257#  4300   4306#  4345   4367#  4401   4405#  4472
$$RTN1= 000242      4150#  4236   4249   4257#  4296   4299   4306#  4344   4367#  4400   4405#  4464   4471
$$RTN2= 000243      4150#  4241   4251   4257#  4300   4306#  4345   4367#  4401   4405#  4472
$$SRC = 000027      3285#  4274#
$$TO  = 000000      1157#  1158   2434#  2437   2535#  2538   2563#  2566   2586#  2589   2646#  2649   2668#
                    2671   2718#  2724   2761#  2767   2802#  2808   2861#  2867   3045#  3048   3055#  3058
                    3127#  3130   3379#  3382   3423#  3426   3500#  3506   3512#  3515   3588#  3591   3654#
                    3655   3656   3675#  3681   3688#  3694   3760#  3761   3762   3779#  3785   3791#  3797
                    3855#  3856   3857   4086#  4092   4218#  4221   4427#  4428
$OFILL  015317      5046*  5050*  5060   5095#
$1      002126      1207#  1232
$10     002446      1349   1354#
$100    005260      2545   2550#
$101    005322      2569   2574#
$102    005364      2592   2597#
$103    005424      2615   2621#
$104    005512      2652   2657#
$105    005554      2674   2679#
$106    005612      2698   2703#
$107    005660      2728   2733#
$11     002474      1365   1370#
$110    005706      2750   2755#
$111    005742      2769   2774#
$112    006034      2812   2817#
$113    006056      2830   2835#
$114    006150      2870   2875#
$115    006166      2887   2892#
$116    006224      2907   2912#
$117    006244      2926#  2943
$12     002522      1382   1387#
$120    006262      2929   2934#
$121    006264      2933   2938#
$122    006300      2941   2944#
$123    006324      2947   2958#
$124    006324      2966#  2989
$125    006370      2968   2990#
$126    006366      2972   2987#
```

$127	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	007730	3522	3540#	
$161	007726	3526	3533#	
$162	007730	3532	3538#	
$163	007772	3560	3566#	
$164	010100	3596	3614#	
$165	010076	3600	3607#	
$166	010100	3606	3612#	
$167	010230	3664	3667#	
$17	002726	1483	1488#	
$170	010226	3665#	3723	
$171	010354	3669	3724#	
$172	010352	3713	3721#	
$173	010420	3742	3748#	
$174	010464	3768	3771#	
$175	010462	3769#	3824	
$176	010610	3773	3825#	
$177	010606	3813	3821#	
$2	002152	1218	1223#	
$20	002764	1508	1513#	
$200	010646	3847	3852#	
$201	010760	3922#	3927	
$202	010774	3925	3928#	
$203	011004	3933	3938#	
$204	011064	3975	3980#	
$205	011144	4012	4028#	
$206	011140	4016	4023#	
$207	011160	4033	4039#	

```
$21     003012          1522    1527#
$210    011224          4064    4069#
$211    011306          4094    4099#
$212    011316          4103    4108#
$213    011642          4236    4249#
$214    011644          4241    4251#
$215    011512          4184#   4225
$216    011616          4211    4213    4226#
$217    011532          4188    4193#
$22     003040          1538    1543#
$220    011540          4192    4197#
$221    011556          4202    4207#
$222    011630          4234    4238#
$223    011744          4296    4299#
$224    011744          4300#
$225    011710          4280    4283#
$226    011704          4281#   4291
$227    011734          4285    4292#
$23     003066          1554    1559#
$230    012024          4344#
$231    012024          4345#
$232    011772          4320    4323#
$233    011766          4321#   4338
$234    012016          4325    4339#
$235    012004          4328    4331#
$236    012002          4329#   4335
$237    012014          4333    4336#
$24     003124          1575    1580#
$240    012216          4400#
$241    012216          4401#
$242    012356          4464    4471#
$243    012356          4472#
$244    012220          4414#   4457
$245    012304          4417    4444#
$246    012244          4420    4425#
$247    012252          4424    4433#
$25     003144          1588    1593#
$250    012324          4443    4454#
$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        4863    4905
$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#
.     = 015402           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
                        3522    3523    3526    3527    3560    3561    3596    3597    3600    3601    3713    3714    3742
                        3743    3813    3814    3847    3848    3927    3928    3933    3934    3975    3976    4012    4013
                        4016    4017    4033    4034    4064    4065    4094    4095    4103    4104    4128#   4188    4189
                        4202    4203    4234    4235    4371#   4384#   4390#   4417    4418    4420    4421    4457    4458
                        4511    4515    4531    4555    4639    4642    4771#   4772    4778    4830#   4889    4952    4953
                        5019#
.$ASTA= ****** U        4782    4785
.$X   = 001000           953#    958
```

. ABS. 015402 000

ERRORS DETECTED: 0

CVDV.BIN,CVDV.LST/CRF:SYM/SOL/NL:TOC=CVDVAC.MAC,CVDVAC.SML,CVDVAC.P11
RUN-TIME: 107 116 5 SECONDS
RUN-TIME RATIO: 853/229=3.7
CORE USED: 43K (85 PAGES)