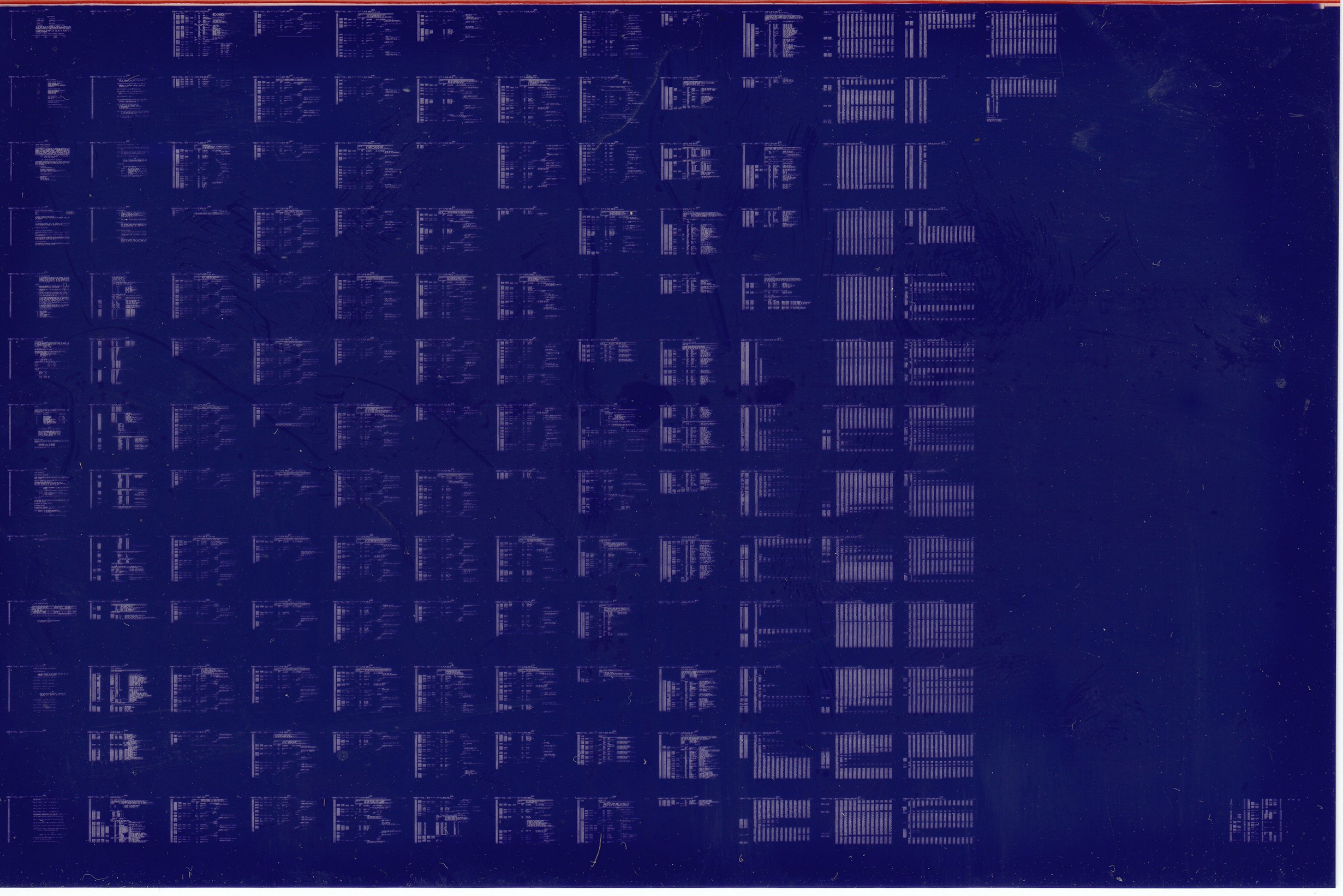
DLV11-E

OFF LINE TEST
CVDVAB0

AH-B151B-MC

FICHE 1 OF 1

JAN 1978
digital
MADE IN USA



```
                .REM    @
```

IDENTIFICATION

PRODUCT CODE: AC-B150B-MC

PRODUCT NAME: CVDVA80 DLV11-E OFFLINE TEST

PRODUCT DATE: APRIL, 1977

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-X OR BC05C-X


                                        SOFTWARE REQUIREMENTS:


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

```
                        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 'SUSWR' AND 'SDEVM' 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.

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.

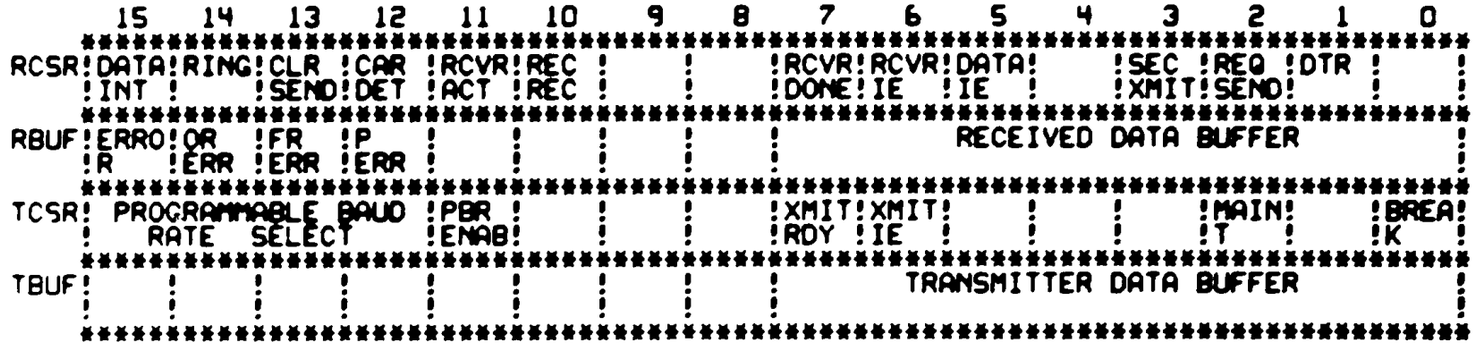
FIRST PASS-	90 SECONDS
ADDITIONAL PASSES	95 SECONDS
ADDITIONAL DEVICES	95 SECONDS

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 IS 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

```
466
```

```
                        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 1: - 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
                        ---- --
```

```
523
524
```

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


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


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

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


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


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


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


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


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

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

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

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

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

NOTE

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

ROUTINE:TIMER

THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT IN ANY REGISTER.

INPUTS:

HOWLONG	THE MAXIMUM AMOUNT OF TIME TO SPEND IN THIS ROUTINE.
WHICHBIT	A MASK WITH THE BIT(S) SET THAT ARE TO BE CHECKED
REG	A POINTER TO THE REGISTER TO BE CHECKED
SETCLR	THE DESIRED RESULTS -- EITHER SET OR CLEAR

OUTPUT:

THE 'C' BIT IS SET TO INDICATE AN ERROR BUT IT IS TESTED BY THE IF.ERROR STATEMENT.

```
                        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  ?SK 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
```

```
                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                  OVERR=          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  012362                          $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  000C030                          .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
```

```
1010  001212  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1011  001214                        $ETABLE:                ;;APT ENVIRONMENT TABLE
1012  001214      000               $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1013  001215      000               $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1014  001216  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1015  001220  071110                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1016  001222  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1017                                ;*                      BITS 15-11=CPU TYPE
1018                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1019                                ;*                              11/70=06,PDQ=07,Q=10
1020                                ;*                      BIT 10=REAL TIME CLOCK
1021                                ;*                      BIT  9=FLOATING POINT PROCESSOR
1022                                ;*                      BIT  8=MEMORY MANAGEMENT
1023  001224      000               $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1024  001225      000               $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1025                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1026                                ;*                              900 NSEC CORE=001
1027                                ;*                              300 NSEC BIPOLAR=002
1028                                ;*                              500 NSEC MOS=003
1029  001226  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1030                                ;*                       MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1031  001230      000               $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1032  001231      000               $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1033  001232  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1034  001234      000               $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1035  001235      000               $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1036  001236  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1037  001240      000               $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1038  001241      000               $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1039  001242  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1040  001244  000300                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1041  001246  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1042  001250  175610                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1043  001252  000001                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1044  001254                        $ETEND:
1045                                .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     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001342  005026                          1$:     CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001344  022706  001140                          CMP     #$WR,R6 ;;DONE?
001350  001374                                  BNE     .-6             ;;LOOP BACK IF NO
001352  012706  001100                          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
001356  012737  014304  000020                  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001364  012737  000340  000022                  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001372  012737  014104  000030                  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001400  012737  000340  000032                  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001406  012737  015236  000034                  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001414  012737  000340  000036                  MOV     #340,@#TRAPVEC+2;LEVEL 7
001422  012737  012416  000024                  MOV     #$PWRON,@#PWRVEC ;;POWER FAILURE VECTOR
001430  012737  000340  000026                  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001436  016767  010666  010656                  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
```

```
1102  001532  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1103                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1104  001534  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1105  001536  012716  001544          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1106  001542  000002                          RTI
1107  001544  012767  000176  177366  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1108  001552  012767  000174  177362          MOV     #DISPREG,DISPLAY
1109  001560  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1110
1111  001564  005067  177412                  CLR     $PASS           ;;CLEAR PASS COUNT
1112  001570  132767  000200  177417          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1113  001576  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1114  001600  012767  001216  177332          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1115  001606                          67$:
1116                                  .SBTTL  TYPE PROGRAM NAME
1117                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1118  001606  005227  177777                  INC     #-1             ;;FIRST TIME?
1119  001612  001037                          BNE     68$             ;;BRANCH IF NO
1120  001614  022737  012362  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
1121  001622  001433                          BEQ     68$             ;;BRANCH IF YES
1122  001624  104401  001672                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1123                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1124  001630  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1125  001634  001012                          BNE     70$             ;;BRANCH IF YES
1126  001636  126727  177352  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1127  001644  001406                          BEQ     70$             ;;BRANCH IF YES
1128  001646  026727  177266  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1129  001654  001005                          BNE     71$             ;;BRANCH IF NO
1130  001656  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1131  001660  000403                          BR      71$
1132  001662  112767  000001  177244  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1133  001670                          71$:
1134  001670  000410                          BR      68$             ;;GET OVER THE ASCIZ
1135                                  ;;69$:  .ASCIZ  <CRLF>#MD-ZZ-CVDVA-B#<CRLF>
1136  001712                          68$:
1137  001712                                  LET INITFLAG := #1
1138  001712  012767  000001  010344          MOV     #1,INITFLAG
1139  001720                          LOOP:
1140  001720                                  CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
1141  001720  004767  010210                  JSR     PC,CYCLE        ;                 ADDRS+2 -> NEXT VECTOR
1142                                                                  ;
1143                                                                  ;GET UNIT ADDRESS
1144  001724                                          LET     DLADD := (ADRS)+
1145  001724  012167  177324                  MOV     (ADRS)+,DLADD
1146                                                                  ;GET UNIT VECTOR
1147  001730                                          LET     DLVEC := (ADRS)
1148  001730  011167  177322                  MOV     (ADRS),DLVEC
1149  001734                                          LET     ADRS  := DLADD
1150  001734  016701  177314                  MOV     DLADD,ADRS
1151                                                                  ;RCSR = DLADD  +  0
1152  001740                                          LET     RCSR  := DLADD
1153  001740  016767  177310  177312          MOV     DLADD,RCSR
1154  001746                                          LET     RBUF  := DLADD + #2
1155  001746  016767  177302  177306          MOV     DLADD,RBUF
1156  001754  062767  000002  177300          ADD     #2,RBUF
1157  001762                                          LET     TCSR  := DLADD + #4
```

```
1158  001762  016767  177266  177274          MOV     DLADD,TCSR
1159  001770  062767  000004  177266          ADD     #4,TCSR
1160  001776
1161  001776  016767  177252  177262          MOV     DLADD,TCSRHI
1162  002004  062767  000005  177254          ADD     #5,TCSRHI
1163  002012
1164  002012  016767  177236  177250          MOV     DLADD,TBUF
1165  002020  062767  000006  177242          ADD     #6,TBUF
1166  002026
1167  002026  012705  001334                  MOV     #RSSTACK,R5
1168
1169  002032  000005                          RESET

                                LET     TCSRHI := DLADD + #5

                                LET     TBUF   := DLADD + #6

                                LET     R5     := #RSSTACK

                                ;;BRESET
```

```
                                ;;*****************************************************************
                                ;*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
                                ;;*****************************************************************
002034  000004          TST1:   SCOPE
002036  012767  000002  177114          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
002044  012767  000001  177126          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002052                                                                  LET     ADRS := DLADD
002052  016701  177176                  MOV     DLADD,ADRS
                                                                        ; SET UP INTERRUPT
002056                                                                  SETVEC  #ILLMEM,#INTSRV,#PR7
002056  010146                          MOV     R1,-(SP)
002060  012701  000004                  MOV     #ILLMEM,R1
002064  012721  011742                  MOV     #INTSRV,(R1)+
002070  012711  000340                  MOV     #PR7,(R1)
002074  012601                          MOV     (SP)+,R1
002076                                                                  LET     I := #0
002076  005067  177170                  CLR     I
002102                                                                  REPEAT
002102                          $1:
002102                                                                      BGNSUB
002102  012767  002110  177000                  MOV     #64$,$LPERR
                                                                            ;CLEAR FLAG
                                                                            LET INTFLAG := #0
002110
002110  005067  007634                  CLR     INTFLAG
                                                                            ;READ FLAG
002114  005711                          TST @ADRS
                                                                            IF INTFLAG NE #0 THEN
002116
002116  005767  007626                  TST     INTFLAG
002122  001401                          BEQ     $2
                                                                            ; FATAL ERROR
002124                                                                          ERRDF 1,,NODL
002124  104001                          ERROR   1
                                                                            ENDIF
002126                          $2:
                                                                        ENDSUB
                                                                        LET     I := I + #2
002126  062767  000002  177136          ADD     #2,I
                                                                        LET     ADRS := DLADD + I
002134
002134  016701  177114                  MOV     DLADD,ADRS
002140  066701  177126                  ADD     I,ADRS
                                                                    UNTIL I EQ #8.
002144
002144  026727  177122  000010          CMP     I,#8.
002152  001353                          BNE     $1
                                                                    CLRVEC ILLMEM
002154
002154  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
002156  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
002160  012701  000004                  MOV     #ILLMEM,R1
002164  010102                          MOV     R1,R2
002166  062702  000002                  ADD     #2,R2
002172  010221                          MOV     R2,(R1)+
002174  005011                          CLR     (R1)
002176  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
1226  002200  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
1227                                                                  ;END OF TEST
1228  002202                                                ENDTST
1229                                  ;;**********************************************************************
1230                                  ;* THE FOLLOWING 8 TESTS  TEST ALL 'READ WRITE' BITS
1231                                  ;;**********************************************************************
1232
1233
```

```
                                ;;*****************************************************************
                                ;*TEST 2         BREAK - TCSR0 SET, CLEAR, RESET
                                ;*      NOTE:  THE (H) JUMPER MUST BE REMOVED FOR THIS
                                ;*             TEST TO FUNCTION PROPERLY.
                                ;;*****************************************************************
002202  000004                  TST2:   SCOPE
002204  012767  000010  176746          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002212  012767  000002  176760          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

                                        ; SEE IF IT IS CLEAR
                                                                BGNSUB
002220  012767  002226  176662          MOV     #64$,$LPERR

                                                                  IF    #BREAK SETIN @TCSR THEN
002226  032777  000001  177030          BIT     #BREAK,@TCSR
002234  001401                          BEQ     $3

                                                                    ; BREAK DID NOT RESET IN TCSR
                                                                    ERRHRD 2,,DIDNOT
002236  104002                          ERROR   2
                                                                  ENDIF
002240                          $3:
002240                                                          ENDSUB

                                        ; TRY TO SET BREAK BIT
                                                                BGNSUB
002240  012767  002246  176642          MOV     #64$,$LPERR
                                                                  LET   @TCSR := @TCSR SET.BY #BREAK
002246  052777  000001  177010          BIS     #BREAK,@TCSR

                                                                  ; STUCK TO 0
                                                                  IF    #BREAK NOTSETIN @TCSR THEN
002254  032777  000001  177002          BIT     #BREAK,@TCSR
002262  001001                          BNE     $4

                                                                    ; BREAK DID NOT SET IN TCSR
                                                                    ERRHRD 3,,DIDNOT
002264  104003                          ERROR   3
                                                                  ENDIF
002266                          $4:
002266                                                          ENDSUB

                                        ; TRY TO CLEAR A SET BIT
                                                                BGNSUB
002266  012767  002274  176614          MOV     #64$,$LPERR
                                                                  LET   @TCSR := @TCSR CLR.BY #BREAK
002274  042777  000001  176762          BIC     #BREAK,@TCSR

                                                                  ; SHOULD HAVE CLEARED
                                                                  IF    #BREAK SETIN @TCSR THEN
002302  032777  000001  176754          BIT     #BREAK,@TCSR
002310  001401                          BEQ     $5

                                                                    ; BREAK  DID NOT CLEAR IN TCSR
                                                                    ERRHRD 4,,DIDNOT
002312  104004                          ERROR   4
                                                                  ENDIF
002314                          $5:
002314                                                          ENDSUB
```

```
                                                ; NOW SEE IF RESET CLEARS IT
                                                        BGNSUB
002314
002314  012767  002322  176566          MOV     #64$,$LPERR
                                                            LET     @TCSR := @TCSR SET.BY #BREAK
002322
002322  052777  000001  176734          BIS     #BREAK,@TCSR
                                                            ; ISSUE BUS RESET
                                                        BRESET
002330
002330  000005                          RESET
                                                            IF      #BREAK SETIN @TCSR THEN
002332
002332  032777  000001  176724          BIT     #BREAK,@TCSR
002340  001401                          BEQ     $6
                                                                ; BREAK DID NOT RESET IN TCSR
                                                                ERRHRD 5,,DIDNOT
002342
002342  104005                          ERROR   5
                                                            ENDIF
002344
$6:                                                     ENDSUB
002344                                              ENDTST
002344

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

```
                                ;;*****************************************************************
                                ;*TEST 3          MAINT - TCSR2 SET, CLEAR, RESET
                                ;;*****************************************************************
002344  000004                  TST3:   SCOPE
002346  012767  000010  176604          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002354  012767  000003  176616          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

                                                        ; SEE IF IT IS CLEAR
                                                                BGNSUB

002362
002362  012767  002370  176520          MOV     #64$,$LPERR

                                                            IF      #MAINT SETIN @TCSR THEN

002370
002370  032777  000004  176666          BIT     #MAINT,@TCSR
002376  001401                          BEQ     $7

                                                                ; MAINT DID NOT RESET IN TCSR
                                                                ERRHRD 6,,DIDNOT

002400
002400  104006                          ERROR   6
002402                                                      ENDIF
002402                          $7:
002402                                                  ENDSUB

                                                        ; TRY TO SET MAINT BIT
                                                                BGNSUB

002402
002402  012767  002410  176500          MOV     #64$,$LPERR
002410                                                      LET     @TCSR := @TCSR SET.BY #MAINT
002410  052777  000004  176646          BIS     #MAINT,@TCSR

                                                            IF  ; STUCK TO 0
                                                                #MAINT NOTSETIN @TCSR THEN

002416
002416  032777  000004  176640          BIT     #MAINT,@TCSR
002424  001001                          BNE     $10

                                                                ; MAINT DID NOT SET IN TCSR
                                                                ERRHRD 7,,DIDNOT

002426
002426  104007                          ERROR   7
002430                                                      ENDIF
002430                          $10:
002430                                                  ENDSUB

                                                        ; TRY TO CLEAR A SET BIT
                                                                BGNSUB

002430
002430  012767  002436  176452          MOV     #64$,$LPERR
                                                            LET     @TCSR := @TCSR CLR.BY #MAINT
002436
002436  042777  000004  176620          BIC     #MAINT,@TCSR

                                                            IF  ; SHOULD HAVE CLEARED
                                                                #MAINT SETIN @TCSR THEN

002444
002444  032777  000004  176612          BIT     #MAINT,@TCSR
002452  001401                          BEQ     $11

                                                                ; MAINT DID NOT CLEAR INTCSR
                                                                ERRHRD 10,,DIDNOT

002454
002454  104010                          ERROR   10
002456                                                      ENDIF
002456                          $11:
002456                                                  ENDSUB

                                                        ; NOW SEE IF RESET CLEARS IT
                                                                BGNSUB

002456
```

```
1368  002456  012767  002464  176424          MOV     #64$,$LPERR
1369
1370  002464                                                      LET     @TCSR := @TCSR SET.BY #MAINT
1371  002464  052777  000004  176572          BIS     #MAINT,@TCSR
1372                                                                ; ISSUE BUS RESET
1373  002472                                                      BRESET
1374  002472  000005                          RESET
1375  002474                                                      IF      #MAINT SETIN @TCSR THEN
1376  002474  032777  000004  176562          BIT     #MAINT,@TCSR
1377  002502  001401                          BEQ     $12
1378                                                                ; MAINT DID NOT RESET IN TCSR
1379  002504                                                        ERRHRD 11,,DIDNOT
1380  002504  104011                          ERROR   11
1381  002506                                                      ENDIF
1382  002506                          $12:
1383  002506                                                    ENDSUB
1384  002506                                                  ENDTST
1385
1386
1387
1388                              ;;*****************************************************************
```

```
                                ;;*****************************************************************
                                ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
                                ;;*****************************************************************
002506  000004          TST4:   SCOPE
002510  012767  000010  176442          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002516  012767  000004  176454          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; USE PRIORITY OF 7
002524  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
002530  012746  002536                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
002534  000002                          RTI                     ;;POP NEW PC AND PS
002536                          64$:

                                                ; SEE IF IT IS CLEAR
                                                                BGNSUB
002536  012767  002544  176344          MOV     #65$,$LPERR
                                                                IF      #XMITIE SETIN @TCSR THEN
002544  032777  000100  176512          BIT     #XMITIE,@TCSR
002552  001401                          BEQ     $13
                                                                    ; XMITIE DID NOT RESET IN TCSR
                                                                    ERRHRD 12,,DIDNOT
002554  104012                          ERROR   12
                                                                ENDIF
002556                          $13:
002556                                                          ENDSUB

                                                ; TRY TO SET XMITIE BIT
                                                                BGNSUB
002556  012767  002564  176324          MOV     #64$,$LPERR
                                                                    LET     @TCSR := @TCSR SET.BY #XMITIE
002564  052777  000100  176472          BIS     #XMITIE,@TCSR
                                                                ; STUCK TO 0
                                                                IF      #XMITIE NOTSETIN @TCSR THEN
002572  032777  000100  176464          BIT     #XMITIE,@TCSR
002600  001001                          BNE     $14
                                                                    ; XMIT DID NOT RESET IN TCSR
                                                                    ERRHRD 13,,DIDNOT
002602  104013                          ERROR   13
                                                                ENDIF
002604                          $14:
002604                                                          ENDSUB

                                                ; TRY TO CLEAR A SET BIT
                                                                BGNSUB
002604  012767  002612  176276          MOV     #64$,$LPERR
                                                                    LET     @TCSR := @TCSR CLR.BY #XMITIE
002612  042777  000100  176444          BIC     #XMITIE,@TCSR
                                                                ; SHOULD HAVE CLEARED
                                                                IF      #XMITIE SETIN @TCSR THEN
002620  032777  000100  176436          BIT     #XMITIE,@TCSR
002626  001401                          BEQ     $15
                                                                    ;  XMIT DID NOT CLEAR IN TCSR
                                                                    ERRHRD 14,,DIDNOT
002630  104014                          ERROR   14
                                                                ENDIF
002632
```

```
1445  002632                                  $15:                                                ENDSUB
1446  002632
1447
1448                                                                ; NOW SEE IF RESET CLEARS IT
1449  002632                                                        BGNSUB
1450  002632  012767  002640  176250                MOV     #64$,$LPERR
1451
1452  002640                                                            LET     @TCSR := @TCSR SET.BY #XMITIE
1453  002640  052777  000100  176416                BIS     #XMITIE,@TCSR
1454                                                                ; ISSUE BUS RESET
1455  002646                                                        BRESET
1456  002646  000005                               RESET
1457  002650                                                        IF      #XMITIE SETIN @TCSR THEN
1458  002650  032777  000100  176406                BIT     #XMITIE,@TCSR
1459  002656  001401                               BEQ     $16
1460
1461  002660                                                            ; XMIT DID NOT RESET IN TCSR
1462  002660  104015                               ERROR   15          ERRHRD 15,,DIDNOT
1463  002662                                                        ENDIF
1464  002662                                  $16:                                                ENDSUB
1465  002662                                                      ENDTST
1466  002662
1467
1468
1469
1470                                            ;;*****************************************************************
```

```
                                ;;**********************************************************************
                                ;*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.
                                ;*               THE (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK.
                                ;;**********************************************************************
002662  000004                  TSTS:   SCOPE
002664  012767  000010  176266          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
002672  012767  000005  176300          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002700                                                                  IF #FRFD NOTSETIN $USWR THEN
002700  032767  040000  176312          BIT     #FRFD,$USWR
002706  001004                          BNE     $17
002710                                                                      EXIT TST
002710  012767  000001  176242          MOV     #1,$TIMES
002716  000441                          BR      TST6            ;;;EXIT THIS TEST
002720                                                                  ENDIF
002720                          $17
                                                        ; TRY TO CLEAR DTR BIT
002720                                                                  BGNSUB
002720  012767  002726  176162          MOV     #64$,$LPERR
002726                                                                      LET     @RCSR := @RCSR CLR.BY #DTR
002726  042777  000002  176324          BIC     #DTR,@RCSR
                                                                        ; STUCK TO 0
002734                                                                  IF      #DTR SETIN @RCSR THEN
002734  032777  000002  176316          BIT     #DTR,@RCSR
002742  001401                          BEQ     $20
                                                                        ; DTR DID NOT CLEAR IN RCSR
002744                                                                  ERRHRD 16,,DIDNOT
002744  104016                          ERROR   16
002746                                                                  ENDIF
002746                          $20:
002746                                                              ENDSUB
                                                        ; TRY TO SET  DTR
002746                                                              BGNSUB
002746  012767  002754  176134          MOV     #64$,$LPERR
                                                                        LET     @RCSR := @RCSR SET.BY #DTR
002754  052777  000002  176276          BIS     #DTR,@RCSR
002762                                                                  IF      #DTR NOTSETIN @RCSR THEN
002762  032777  000002  176270          BIT     #DTR,@RCSR
002770  001001                          BNE     $21
                                                                        ; DTR DID NOT SET IN RCSR
002772                                                                  ERRHRD 17,,DIDNOT
002772  104017                          ERROR   17
002774                                                                  ENDIF
002774                          $21:
002774                                                              ENDSUB
                                                        ; TRY TO CLEAR IT AGAIN
002774                                                              BGNSUB
002774  012767  003002  176106          MOV     #64$,$LPERR
003002                                                                  LET     @RCSR := @RCSR CLR.BY #DTR
003002  042777  000002  176250          BIC     #DTR,@RCSR
```

```
1527
1528  003010
1529  003010  032777  000002  176242          BIT     #DTR,@RCSR                      ; SHOULD HAVE CLEARED IT
1530  003016  001401                          BEQ     $22                         IF      #DTR SETIN @RCSR THEN
1531
1532  003020
1533  003020  104020                          ERROR   20                              ; DTR DID NOT CLEAR IN RCSR
1534  003022                                                                          ERRHRD 20,,DIDNOT
1535  003022                          $22:                                        ENDIF
1536  003022                                                                  ENDSUB
1537  003022                                                              ENDTST
1538
1539
1540
1541                              ;;*****************************************************************
```

```
                                ;;*****************************************************************
                                ;*TEST 6          REQSEND - RCSR2         SET, CLEAR, RESET
                                ;*               THIS TEST ASSUMES THAT JUMPER -(FR) IS IN
                                ;;*****************************************************************
003022  000004                  TST6:   SCOPE
003024  012767  000010  176126          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003032  012767  000006  176140          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
003040                                                                  IF #FRFD NOTSETIN $USWR THEN
003040  032767  040000  176152          BIT     #FRFD,$USWR
003046  001004                          BNE     $23
003050                                                                      EXIT TST
003050  012767  000001  176102          MOV     #1,$TIMES
003056  000452                          BR      TST7            ;;;EXIT THIS TEST
003060                                                                  ENDIF
003060                          $23:
1557
1558                                                    ; SEE IF IT IS CLEAR
003060                                                                  BGNSUB
003060  012767  003066  176022          MOV     #64$,$LPERR
1561
003066                                                                      IF      #REQSEND SETIN @RCSR THEN
003066  032777  000004  176164          BIT     #REQSEND,@RCSR
003074  001401                          BEQ     $24
1565                                                                        ; REQSEND DID NOT RESET IN RCSR
003076                                                                      ERRHRD 21,,DIDNOT
003076  104021                          ERROR   21
003100                                                                  ENDIF
003100                          $24:
003100                                                                  ENDSUB
1571
1572                                                    ; TRY TO SET REQSEND BIT
003100                                                                  BGNSUB
003100  012767  003106  176002          MOV     #64$,$LPERR
003106                                                                      LET     @RCSR := @RCSR SET.BY #REQSEND
003106  052777  000004  176144          BIS     #REQSEND,@RCSR
1577                                                                        ; STUCK TO 0
003114                                                                      IF      #REQSEND NOTSETIN @RCSR THEN
003114  032777  000004  176136          BIT     #REQSEND,@RCSR
003122  001001                          BNE     $25
1581                                                                        ; REQSEND DID NOT SET IN RCSR
003124                                                                      ERRHRD 22,,DIDNOT
003124  104022                          ERROR   22
003126                                                                  ENDIF
003126                          $25:
003126                                                                  ENDSUB
1587
1588                                                    ; TRY TO CLEAR A SET BIT
003126                                                                  BGNSUB
003126  012767  003134  175754          MOV     #64$,$LPERR
1591
003134                                                                      LET     @RCSR := @RCSR CLR.BY #REQSEND
003134  042777  000004  176116          BIC     #REQSEND,@RCSR
1594                                                                        ; SHOULD HAVE CLEARED
003142                                                                      IF      #REQSEND SETIN @RCSR THEN
003142  032777  000004  176110          BIT     #REQSEND,@RCSR
003150  001401                          BEQ     $26
```

```
1598
1599  003152
1600  003152  104023                    ERROR   23                                  ;REQSEND DID NOT CLEAR IN RCSR
1601  003154                                                                        ERRHRD 23,,DIDNOT
1602  003154                          $26:                                        ENDIF
1603  003154                                                                    ENDSUB
1604
1605                                                          ; NOW SEE IF RESET CLEARS IT
1606  003154                                                                    BGNSUB
1607  003154  012767  003162  175726    MOV     #64$,$LPERR
1608
1609  003162                                                                    LET     @RCSR := @RCSR SET.BY #REQSEND
1610  003162  052777  000004  176070    BIS     #REQSEND,@RCSR
1611                                                                            ;ISSUE BUS RESET
1612  003170                                                                    BRESET
1613  003170  000005                    RESET
1614  003172                                                                    IF      #REQSEND SETIN @RCSR THEN
1615  003172  032777  000004  176060    BIT     #REQSEND,@RCSR
1616  003200  001401                    BEQ     $27
1617                                                                            ;REQSEND DID NOT RESET IN RCSR
1618  003202                                                                    ERRHRD 24,,DIDNOT
1619  003202  104024                    ERROR   24
1620  003204                                                                    ENDIF
1621  003204                          $27:                                    ENDSUB
1622  003204                                                                ENDTST
1623  003204
1624
1625
1626
1627                              ;;**************************************************************
```

```
                                ;;*****************************************************************
                                ;;*TEST 7          SECXMIT - RCSR3          SET, CLEAR, RESET
                                ;;*****************************************************************
003204  000004          TST7:   SCOPE
003206  012767  000010  175744          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003214  012767  000007  175756          MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                        ; SEE IF IT IS CLEAR
                                                                BGNSUB
003222
003222  012767  003230  175660          MOV     #64$,$LPERR
                                                                    IF      #SECXMIT SETIN @RCSR THEN
003230
003230  032777  000010  176022          BIT     #SECXMIT,@RCSR
003236  001401                          BEQ     $30
                                                                        ; SECXMIT DID NOT RESET IN RCSR
                                                                        ERRHRD 25,,DIDNOT
003240
003240  104025                          ERROR   25
                                                                    ENDIF
003242                          $30:
003242                                                          ENDSUB

                                                        ; TRY TO SET SECXMIT BIT
                                                                BGNSUB
003242
003242  012767  003250  175640          MOV     #64$,$LPERR
                                                                    LET     @RCSR := @RCSR SET.BY #SECXMIT
003250
003250  052777  000010  176002          BIS     #SECXMIT,@RCSR
                                                                    ; STUCK TO 0
003256                                                              IF      #SECXMIT NOTSETIN @RCSR THEN
003256  032777  000010  175774          BIT     #SECXMIT,@RCSR
003264  001001                          BNE     $31
                                                                        ; SECXMIT DID NOT SET IN RCSR
                                                                        ERRHRD 26,,DIDNOT
003266
003266  104026                          ERROR   26
                                                                    ENDIF
003270                          $31:
003270                                                          ENDSUB

                                                        ; TRY TO CLEAR A SET BIT
                                                                BGNSUB
003270
003270  012767  003276  175612          MOV     #64$,$LPERR
                                                                    LET     @RCSR := @RCSR CLR.BY #SECXMIT
003276
003276  042777  000010  175754          BIC     #SECXMIT,@RCSR
                                                                    ; SHOULD HAVE CLEARED
003304                                                              IF      #SECXMIT SETIN @RCSR THEN
003304  032777  000010  175746          BIT     #SECXMIT,@RCSR
003312  001401                          BEQ     $32
                                                                        ; SECXMIT DID NOT CLEAR IN RCSR
                                                                        ERRHRD 27,,DIDNOT
003314
003314  104027                          ERROR   27
                                                                    ENDIF
003316                          $32:
003316                                                          ENDSUB

003316                                                          BGNSUB
003316  012767  003324  175564          MOV     #64$,$LPERR
                                                        ; NOW SEE IF RESET CLEARS IT
```

```
1684
1685  003324
1686  003324  052777  000010  175726          BIS     #SECXMIT,@RCSR                  LET    @RCSR := @RCSR SET.BY #SECXMIT
1687
1688  003332                                                                          ; ISSUE BUS RESET
1689  003332  000005                          RESET                                  BRESET
1690  003334                                                                          IF     #SECXMIT SETIN @RCSR THEN
1691  003334  032777  000010  175716          BIT     #SECXMIT,@RCSR
1692  003342  001401                          BEQ     $33
1693                                                                                  ; SECXMIT DID NOT RESET IN RCSR
1694  003344                                                                          ERRHRD 30,,DIDNOT
1695  003344  104030                          ERROR   30
1696  003346                                                                          ENDIF
1697  003346                          $33:                                    ENDSUB
1698  003346                                                                ENDTST
1699  003346
1700
1701
1702
1703                                  ;;*****************************************************************
```

```
                                ;;*****************************************************************
                                ;*TEST 10        DATAIE - RCSR5 SET, CLEAR, RESET
                                ;;*****************************************************************
003346  000004                  TST10:  SCOPE
003350  012767  000010  175602          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003356  012767  000010  175614          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                        ; SEE IF IT IS CLEAR
                                                                BGNSUB
003364
003364  012767  003372  175516          MOV     #64$,$LPERR
003372                                                              IF      #DATAIE SETIN @RCSR THEN
003372  032777  000040  175660          BIT     #DATAIE,@RCSR
003400  001401                          BEQ     $34
                                                                    ; DATAIE DID NOT RESET IN RCSR
                                                                    ERRHRD 31,,DIDNOT
003402
003402  104031                          ERROR   31
003404                                                              ENDIF
003404                          $34:
003404                                                          ENDSUB
                                                        ; TRY TO SET DATAIE BIT
                                                                BGNSUB
003404
003404  012767  003412  175476          MOV     #64$,$LPERR
003412                                                              LET     @RCSR := @RCSR SET.BY #DATAIE
003412  052777  000040  175640          BIS     #DATAIE,@RCSR
                                                                    ; STUCK TO 0
003420                                                              IF      #DATAIE NOTSETIN @RCSR THEN
003420  032777  000040  175632          BIT     #DATAIE,@RCSR
003426  001001                          BNE     $35
                                                                    ; DATAIE DID NOT SET IN RCSR
                                                                    ERRHRD 32,,DIDNOT
003430
003430  104032                          ERROR   32
003432                                                              ENDIF
003432                          $35:
003432                                                          ENDSUB
                                                        ; TRY TO CLEAR A SET BIT
003432                                                          BGNSUB
003432  012767  003440  175450          MOV     #64$,$LPERR
003440                                                              LET     @RCSR := @RCSR CLR.BY #DATAIE
003440  042777  000040  175612          BIC     #DATAIE,@RCSR
                                                                    ; SHOULD HAVE CLEARED
003446                                                              IF      #DATAIE SETIN @RCSR THEN
003446  032777  000040  175604          BIT     #DATAIE,@RCSR
003454  001401                          BEQ     $36
                                                                    ; DATAIE DID NOT CLEAR IN RCSR
003456                                                              ERRHRD 33,,DIDNOT
003456  104033                          ERROR   33
003460                                                              ENDIF
003460                          $36:
003460                                                          ENDSUB
                                                        ; NOW SEE IF RESET CLEARS IT
003460                                                          BGNSUB
003460  012767  003466  175422          MOV     #64$,$LPERR
```

```
                                                                LET     @RCSR := @RCSR SET.BY #DATAIE
003466                                                                  ; ISSUE BUS RESET
003466  052777  000040  175564          BIS     #DATAIE,@RCSR   BRESET
003474                                                          IF      #DATAIE SETIN @RCSR THEN
003474  000005                          RESET
003476                                                              ; DATAIE DID NOT RESET IN RCSR
003476  032777  000040  175554          BIT     #DATAIE,@RCSR   ERRHRD 34,,DIDNOT
003504  001401                          BEQ     $37
                                                                ENDIF
003506
003506  104034                          ERROR   34
003510
003510                          $37:                                ENDSUB
003510                                                      ENDTST
003510

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

```
                        ;;*****************************************************************
                        ;;*TEST 11        RCVRIE - RCSR6 SET, CLEAR, RESET
                        ;;*****************************************************************
003510  000004          TST11:  SCOPE
003512  012767  000010  175440          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003520  012767  000011  175452          MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                ; SEE IF IT IS CLEAR
003526                                                          BGNSUB
003526  012767  003534  175354          MOV     #64$,$LPERR

003534                                                              IF      #RCVRIE SETIN @RCSR THEN
003534  032777  000100  175516          BIT     #RCVRIE,@RCSR
003542  001401                          BEQ     $40
                                                                    ; RCVRIE DID NOT RESET IN RCSR
                                                                    ERRHRD 35,,DIDNOT
003544  104035                          ERROR   35
                                                                ENDIF
003546                          $40:
003546                                                          ENDSUB

                                                ; TRY TO SET RCVRIE BIT
003546                                                          BGNSUB
003546  012767  003554  175334          MOV     #64$,$LPERR
                                                                LET     @RCSR := @RCSR SET.BY #RCVRIE
003554
003554  052777  000100  175476          BIS     #RCVRIE,@RCSR
                                                                ; STUCK TO 0
003562                                                          IF      #RCVRIE NOTSETIN @RCSR THEN
003562  032777  000100  175470          BIT     #RCVRIE,@RCSR
003570  001001                          BNE     $41
                                                                    ; RCVRIE DID NOT SET IN RCSR
                                                                    ERRHRD 36,,DIDNOT
003572  104036                          ERROR   36
                                                                ENDIF
003574                          $41:
003574                                                          ENDSUB

                                                ; TRY TO CLEAR A SET BIT
003574                                                          BGNSUB
003574  012767  003602  175306          MOV     #64$,$LPERR

                                                                    LET     @RCSR := @RCSR CLR.BY #RCVRIE
003602
003602  042777  000100  175450          BIC     #RCVRIE,@RCSR
                                                                ; SHOULD HAVE CLEARED
003610                                                          IF      #RCVRIE SETIN @RCSR THEN
003610  032777  000100  175442          BIT     #RCVRIE,@RCSR
003616  001401                          BEQ     $42
                                                                    ; RCVRIE DID NOT CLEAR IN RCSR
                                                                    ERRHRD 37,,DIDNOT
003620  104037                          ERROR   37
                                                                ENDIF
003622                          $42:
003622                                                          ENDSUB

                                                ; NOW SEE IF RESET CLEARS IT
003622                                                          BGNSUB
003622  012767  003630  175260          MOV     #64$,$LPERR
```

```
1836
1837   003630
1838   003630  052777  000100  175422          BIS     #RCVRIE,@RCSR
1839
1840   003636
1841   003636  000005                          RESET
1842   003640
1843   003640  032777  000100  175412          BIT     #RCVRIE,@RCSR
1844   003646  001401                          BEQ     $43
1845
1846   003650
1847   003650  104040                          ERROR   40
1848   003652
1849   003652                          $43:
1850   003652
1851   003652
1852   003652
1853
1854
1855
1856
1857                                    ;;*****************************************************************
1858                                    ;* THE FOLLOWING 4 TESTS VERIFY
1859                                    ;* THAT RESET (INIT) INITIALIZES READ ONLY BITS.
1860                                    ;;*****************************************************************
1861
1862
```

```
                LET     @RCSR := @RCSR SET.BY #RCVRIE
                ; ISSUE BUS RESET
                BRESET
                IF      #RCVRIE SETIN @RCSR THEN

                ; RCVRIE DID NOT RESET IN RCSR
                ERRHRD 40,,DIDNOT

                ENDIF

                CKLOOP
            ENDSUB
        ENDTST
```

```
                        ;;*********************************************************************
                        ;*TEST 12        TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
                        ;;*********************************************************************
003652  000004          TST12:  SCOPE
003654  012767  000010  175276          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003662  012767  000012  175310          MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
003670
003670  012767  003676  175212          MOV     #64$,$LPERR                     BGNSUB
003676                                                                              IF      #RCVRDONE SETIN @RCSR THEN
003676  032777  000200  175354          BIT     #RCVRDONE,@RCSR
003704  001402                          BEQ     $44
                                                                                        ;RCVRDONE SHOULD HAVE CLEARED BY INIT
                                                                                        ; RCVRDONE DID NOT CLEAR IN RCSR
003706                                                                                  ERRHRD 41,HRESET, DIDNOT
003706  104041                          ERROR   41
                                                                                        ;REISSUE RESET
003710                                                                                  @RESET
003710  000005                          RESET
003712                                                                              ENDIF
003712                  $44:
                                                                                        ;ALLOW LOOPING AFTER ERROR
003712                                                                              CKLOOP
003712                                                                          ENDSUB
003712                                                                  ENDTST
```

```
                                ;;*****************************************************************
                                ;*TEST 13        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
                                ;;*****************************************************************
003712  000004                  TST13:  SCOPE
003714  012767  000010  175236          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003722  012767  000013  175250          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


003730                                                  BGNSUB
003730  012767  003736  175152          MOV     #64$,$LPERR

003736                                      IF      #XMITRDY NOTSETIN @TCSR THEN
003736  032777  000200  175320          BIT     #XMITRDY,@TCSR
003744  001002                          BNE     $45


                                            ;RESET SHOULD HAVE SET BIT.
                                            ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
003746                                      ERRHRD 42,HRESET,DIDNOT
003746  104042                  ERROR   42

                                            ;ISSUE ANOTHER RESET
003750                                      HRESET
003750  000005                  RESET
003752                                  ENDIF
003752                  $45:
                                            ;ALLOW LOOPING ON ERROR
003752                                  CKLOOP
003752                              ENDSUB
003752                          ENDTST
```

```
                                ;;*****************************************************************
                                ;*TEST 14        TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.
                                ;;*****************************************************************
003752  000004                  $TST14:  SCOPE
003754  012767  000010  175176          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003762  012767  000014  175210          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
003770                                                                  BGNSUB
003770  012767  003776  175112          MOV     #64$,$LPERR
003776                                                                      IF      #DATAINT SETIN @RCSR THEN
003776  032777  100000  175254          BIT     #DATAINT,@RCSR
004004  001402                        BEQ     $46
004006                                                                          ERRHRD 43, HRESET, DIDNOT
004006  104043                        ERROR   43
                                                                            ;TESTING EFFECT OF RESET ON BIT
                                                                            ;DATAINT DID NOT CLEAR IN RCSR
                                                                            ;ALLOW A FRESH START
                                                                            BRESET
004010
004010  000005                        RESET
                                                                        ENDIF
004012                          $46:
004012                                                                  CKLOOP
004012                                                              ENDSUB
004012                                                          ENDTST
                                ;;*****************************************************************
```

```
                                ;;*****************************************************************
                                ;*TEST 15        TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT
                                ;;*****************************************************************
004012  000004          TST15:  SCOPE
004014  012767  000010  175136          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004022  012767  000015  175150          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                                        IF      #CABLE NOTSETIN $USWR THEN

004030  032767  020000  175162          BIT     #CABLE,$USWR
004036  001004                          BNE     $47
                                                                            ; CAN'T TEST WITHOUT BERG OR H315.
                                                                        EXIT TST
004040  012767  000001  175112          MOV     #1,$TIMES
004046  000411                          BR      TST16           ;;;EXIT THIS TEST
                                                                        ENDIF
004050                          $47:

                                                                        BGNSUB

004050  012767  004056  175032          MOV     #64$,$LPERR

                                                                            IF      #RCVRACT SETIN @RCSR THEN

004056  032777  004000  175174          BIT     #RCVRACT,@RCSR
004064  001402                          BEQ     $50

                                                                            ;RESET SHOULD HAVE CLEARED RCVRACT
004066                                                                      ERRHRD 44, HRESET, DIDNOT
004066  104044                          ERROR   44

                                                                            ;TESTING EFFECT OF RESET ON BIT

                                                                            ;RCVRACT DID NOT CLEAR IN RCSR

                                                                            ;ALLOW ANOTHER TRY
004070                                                                      BRESET
004070  000005                  RESET

                                                                        ENDIF
004072                          $50:
                                                                            ;ALLOW LOOPING ON ERROR
004072                                                                  CKLOOP
004072                                                              ENDSUB
004072                                                          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.
                                ;;*****************************************************************
004072  000004                  TST16:  SCOPE
004074  012767  000010  175056          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004102  012767  000016  175070          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
                                                                IF      @CABLE+FRFD NOTSETIN $USWR THEN
004110
004110  032767  060000  175102          BIT     #CABLE+FRFD,$USWR
004116  001004                          BNE     $S1
                                                                        ; CAN'T TEST WITHOUT BERG OR H315
                                                                        ; OR WITH (-FD) JUMPER OUT.
                                                                        ; OR WITH (-FR) JUMPER OUT.
                                                                    EXIT TST
004120
004120  012767  000001  175032          MOV     #1,$TIMES
004126  000441                          BR      TST17           ;;;EXIT THIS TEST
                                                                ENDIF
004130
004130                          $S1:
                                                                        ; DTR AND
                                                                        ; CARDET ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

                                                                        ; CLEAR
                                                                BGNSUB
004130
004130  012767  004136  174752          MOV     #$$,$LPERR

                                                                        ; CLEAR  DTR
                                                                LET     @RCSR := @RCSR CLR.BY #DTR
004136
004136  042777  000002  175114          BIC     #DTR,@RCSR
                                                                        ; CARDET SHOULD FOLLOW
                                                                IF      @CARDET SETIN @RCSR THEN
004144
004144  032777  010000  175106          BIT     #CARDET,@RCSR
004152  001401                          BEQ     $S2

                                                                    ; CARDET DID NOT
004154                                                              ERRHRD 45,,FORCE
004154  104045                          ERROR   45

                                                                    ;  CLEAR WITH DTR
                                                                ENDIF
004156
004156                          $S2:
004156                                                          ENDSUB

                                                                        ; SET
                                                                BGNSUB
004156
```

```
2064  004156  012767  004164  174724          MOV     #64$,$LPERR
2065
2066
2067  004164
2068  004164  052777  000002  175066          BIS     #DTR,@RCSR
2069
2070  004172
2071  004172  032777  010000  175060          BIT     #CARDET,@RCSR
2072  004200  001001                          BNE     $53
2073
2074  004202
2075  004202  104046                          ERROR   46
2076
2077
2078  004204
2079  004204                          $53:
2080  004204
2081
2082                                                    ; CLEAR
2083  004204
2084  004204  012767  004212  174676          MOV     #64$,$LPERR
2085
2086
2087  004212
2088  004212  042777  000002  175040          BIC     #DTR,@RCSR
2089
2090  004220
2091  004220  032777  010000  175032          BIT     #CARDET,@RCSR
2092  004226  001401                          BEQ     $54
2093
2094  004230
2095  004230  104047                          ERROR   47
2096
2097
2098  004232
2099  004232                          $54:
2100  004232
2101  004232
2102
2103
2104
2105
```

```
        ; SET DTR
LET     @RCSR := @RCSR SET.BY #DTR

        ; CARDET SHOULD FOLLOW
IF      #CARDET NOTSETIN @RCSR THEN

    ; CARDET DID NOT SET
    ERRHRD 46,,FORCE

    ;  WITH DTR
ENDIF

ENDSUB

BGNSUB

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

        ; CARDET SHOULD FOLLOW
IF      #CARDET SETIN @RCSR THEN

    ; CARDET DID NOT
    ERRHRD 47,,FORCE

    ;  CLEAR WITH DTR
ENDIF

        ENDSUB
ENDTST
```

```
                                ;;**************************************************************
                                ;;**************************************************************
                                ;*TEST 17         TEST THAT CLRSEND SETS AND CLEARS
                                ;*               AS  DTR   SETS AND CLEARS
                                ;*               (-FD) JUMPER MUST BE IN FOR THIS TEST TO WORK
                                ;;**************************************************************
004232  000004                  TST17:  SCOPE
004234  012767  000010  174716          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004242  012767  000017  174730          MOV     #17.$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
                                                                IF      @CABLE+FRFD NOTSETIN $USWR THEN
004250
004250  032767  060000  174742          BIT     #CABLE+FRFD,$USWR
004256  001004                          BNE     $55
                                                                        ; CAN'T TEST WITHOUT BERG OR H315
                                                                    EXIT TST
004260
004260  012767  000001  174672          MOV     #1,$TIMES
004266  000441                          BR      TST20           ;;:EXIT THIS TEST
                                                                ENDIF
004270
004270                          $55:
                                                                        ; DTR AND
                                                                        ; CLRSEND ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

                                                                        ; CLEAR
                                                                BGNSUB
004270
004270  012767  004276  174612          MOV     #64$,$LPERR

                                                                    ; CLEAR  DTR
                                                                LET   @RCSR := @RCSR CLR.BY #DTR
004276
004276  042777  000002  174754          BIC     #DTR,@RCSR
                                                                    ; CLRSEND SHOULD FOLLOW
                                                                IF    #CLRSEND SETIN @RCSR THEN
004304
004304  032777  020000  174746          BIT     #CLRSEND,@RCSR
004312  001401                          BEQ     $56

                                                                    ; CLRSEND DID NOT
004314                                                              ERRHRD 50,,FORCE
004314  104050                          ERROR   50
                                                                    ; CLEAR WITH DTR
                                                                ENDIF
004316
004316                          $56:                            ENDSUB
004316

                                                                        ; SET
                                                                BGNSUB
004316
004316  012767  004324  174564          MOV     #64$,$LPERR

                                                                    ; SET DTR
                                                                LET   @RCSR := @RCSR SET.BY #DTR
004324
004324  052777  000002  174726          BIS     #DTR,@RCSR
                                                                    ; CLRSEND SHOULD FOLLOW
                                                                IF    #CLRSEND NOTSETIN @RCSR THEN
004332
004332  032777  020000  174720          BIT     #CLRSEND,@RCSR
004340  001001                          BNE     $57
```

```
2162
2163  004342
2164  004342  104051                    ERROR   51
2165
2166
2167  004344
2168  004344                        $57:
2169  004344
2170
2171                                                ; CLEAR
2172  004344
2173  004344  012767  004352  174536    MOV     #64$,$LPERR
2174
2175
2176  004352
2177  004352  042777  000002  174700    BIC     #DTR,@RCSR
2178
2179  004360
2180  004360  032777  020000  174672    BIT     #CLRSEND,@RCSR
2181  004366  001401                    BEQ     $60
2182
2183  004370
2184  004370  104052                    ERROR   52
2185
2186
2187  004372
2188  004372                        $60:
2189  004372
2190  004372
2191
2192
2193
2194
```

```
                    ; CLRSEND DID NOT SET
                    ERRHRD 51,,FORCE

                ;  WITH DTR
            ENDIF

        ENDSUB

        BGNSUB

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

            ; CLRSEND SHOULD FOLLOW
            IF      #CLRSEND SETIN @RCSR THEN

                ; CLRSEND DID NOT
                ERRHRD 52,,FORCE

            ;  CLEAR WITH DTR
            ENDIF

        ENDSUB
ENDTST
```

```
2195                                    ;;**********************************************************************
2196                                    ;;**********************************************************************
2197                                    ;*TEST 20        TEST THAT RING SETS AND CLEARS
2198                                    ;*              AS  REQSEND    SETS AND CLEARS
2199                                    ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
2200                                    ;;**********************************************************************
2201  004372  000004                    TST20:  SCOPE
2202  004374  012767  000010  174556            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2203  004402  012767  000020  174570            MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2204                                                                            ; CAN WE USE THE WRAPAROUND??
2205  004410                                                            IF      #CABLE+FRFD NOTSETIN $USWR THEN
2206  004410  032767  060000  174602            BIT     #CABLE+FRFD,$USWR
2207  004416  001004                            BNE     $61
2208                                                                            ; CAN'T TEST WITHOUT BERG OR H315
2209                                                                            ; OR WITH (-FR) JUMPER OUT.
2210  004420                                                        EXIT TST
2211  004420  012767  000001  174532            MOV     #1,$TIMES
2212  004426  000441                            BR      TST21           ;;;EXIT THIS TEST
2213  004430                                                            ENDIF
2214  004430                                $61:
2215
2216                                                                            ; REQSEND AND
2217                                                                            ; RING ARE CONNECTED
2218
2219                                                                    ; BY THE H315 OR EQUIV.
2220
2221                                                    ; CLEAR
2222  004430                                                            BGNSUB
2223  004430  012767  004436  174452            MOV     #64$,$LPERR
2224
2225                                                                        ; CLEAR  REQSEND
2226  004436                                                        LET     @RCSR := @RCSR CLR.BY #REQSEND
2227  004436  042777  000004  174614            BIC     #REQSEND,@RCSR
2228                                                                        ; RING SHOULD FOLLOW
2229  004444                                                        IF      #RING SETIN @RCSR THEN
2230  004444  032777  040000  174606            BIT     #RING,@RCSR
2231  004452  001401                            BEQ     $62
2232                                                                        ; RING DID NOT
2233  004454                                                        ERRHRD 53,,FORCE
2234  004454  104053                            ERROR   53
2235
2236                                                                        ;  CLEAR WITH REQSEND
2237  004456                                                        ENDIF
2238  004456                                $62:
2239  004456                                                    ENDSUB
2240
2241                                                    ; SET
2242  004456                                                    BGNSUB
2243  004456  012767  004464  174424            MOV     #64$,$LPERR
2244
2245                                                                        ; SET REQSEND
2246  004464                                                        LET     @RCSR := @RCSR SET.BY #REQSEND
2247  004464  052777  000004  174566            BIS     #REQSEND,@RCSR
2248                                                                        ; RING SHOULD FOLLOW
2249  004472                                                        IF      #RING NOTSETIN @RCSR THEN
2250  004472  032777  040000  174560            BIT     #RING,@RCSR
```

```
2251  004500  001001                    BNE     $63
2252
2253  004502                                                          ; RING DID NOT SET
2254  004502  104054                    ERROR   54                   ERRHRD 54,,FORCE
2255
2256                                                                  ; WITH REQSEND
2257  004504                                                        ENDIF
2258  004504                          $63:
2259  004504                                                      ENDSUB
2260
2261                                                ; CLEAR
2262  004504                                                      BGNSUB
2263  004504  012767  004512  174376    MOV     #64$,$LPERR
2264
2265                                                                  ; CLEAR  REQSEND
2266  004512                                                        LET     @RCSR := @RCSR CLR.BY #REQSEND
2267  004512  042777  000004  174540    BIC     #REQSEND,@RCSR
2268
2269  004520                                                                ; RING SHOULD FOLLOW
2270  004520  032777  040000  174532    BIT     #RING,@RCSR          IF      #RING SETIN @RCSR THEN
2271  004526  001401                    BEQ     $64
2272
2273  004530                                                          ; RING DID NOT
2274  004530  104055                    ERROR   55                   ERRHRD 55,,FORCE
2275
2276                                                                  ; CLEAR WITH REQSEND
2277  004532                                                        ENDIF
2278  004532                          $64:
2279  004532                                                      ENDSUB
2280  004532                                                  ENDTST
2281
2282
2283
2284
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 21        TEST THAT SECREC SETS AND CLEARS
                                ;*              AS  SECXMIT    SETS AND CLEARS
                                ;;*****************************************************************
004532  000004                  TST21:  SCOPE
004534  012767  000010  174416          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004542  012767  000021  174430          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
004550                                                          IF      #CABLE NOTSETIN $USWR THEN
004550  032767  020000  174442          BIT     #CABLE,$USWR
004556  001004                        BNE     $65
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
                                                                    EXIT TST
004560
004560  012767  000001  174372          MOV     #1,$TIMES
004566  000441                        BR      TST22           ;;;EXIT THIS TEST
                                                                ENDIF
004570
004570                          $65:
                                                                        ; SECXMIT AND
                                                                        ; SECREC ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

                                                                BGNSUB
                                                ; CLEAR
004570                                                          
004570  012767  004576  174312          MOV     #64$,$LPERR
                                                                        ; CLEAR  SECXMIT
004576                                                              LET     @RCSR := @RCSR CLR.BY #SECXMIT
004576  042777  000010  174454          BIC     #SECXMIT,@RCSR
                                                                        ; SECREC SHOULD FOLLOW
004604                                                              IF      #SECREC SETIN @RCSR THEN
004604  032777  002000  174446          BIT     #SECREC,@RCSR
004612  001401                        BEQ     $66
                                                                        ; SECREC DID NOT
004614                                                              ERRHRD 56,,FORCE
004614  104056                        ERROR   56
                                                                        ;  CLEAR WITH SECXMIT
004616                                                              ENDIF
004616                          $66:
004616                                                          ENDSUB

                                                ; SET
004616                                                          BGNSUB
004616  012767  004624  174264          MOV     #64$,$LPERR
                                                                        ; SET SECXMIT
004624                                                              LET     @RCSR := @RCSR SET.BY #SECXMIT
004624  052777  000010  174426          BIS     #SECXMIT,@RCSR
                                                                        ; SECREC SHOULD FOLLOW
004632                                                              IF      #SECREC NOTSETIN @RCSR THEN
004632  032777  002000  174420          BIT     #SECREC,@RCSR
004640  001001                        BNE     $67
                                                                        ; SECREC DID NOT SET
```

```
2341  004642
2342  004642  104057                      ERROR   57
2343
2344
2345  004644
2346  004644                        $67:
2347  004644
2348
2349                                      ; CLEAR
2350  004644
2351  004644  012767  004652  174236      MOV     #64$,$LPERR
2352
2353
2354  004652
2355  004652  042777  000010  174400      BIC     #SECXMIT,@RCSR
2356
2357  004660
2358  004660  032777  002000  174372      BIT     #SECREC,@RCSR
2359  004666  001401                      BEQ     $70
2360
2361  004670
2362  004670  104060                      ERROR   60
2363
2364
2365  004672
2366  004672                        $70:
2367  004672
2368  004672
2369
2370
2371
2372
```

```
                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.
                                ;;*****************************************************************
004672  000004                TST22:  SCOPE
004674  012767  000010  174256        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004702  012767  000022  174270        MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; CAN WE USE THE WRAPAROUND??
004710                                                      IF      @CABLE+FRFD NOTSETIN $USWR THEN
004710  032767  060000  174302        BIT     #CABLE+FRFD,$USWR
004716  001004                        BNE     $71
                                                                ; CAN'T TEST WITHOUT BERG OR H315
                                                                ; OR WITH (-FD) JUMPER OUT.
004720                                                          EXIT TST
004720  012767  000001  174232        MOV     #1,$TIMES
004726  000463                        BR      TST23           ;;;EXIT THIS TEST
004730                                                      ENDIF
004730                          $71:
                                                                ;MAKE SURE NOTHING UNEXPECTED HAPPENS
004730  012746  000340                MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
004734  012746  004742                MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
004740  000002                        RTI                     ;;POP NEW PC AND PS
004742                          64$:
                                                        ;READ TWICE - CLEARS
                                                            BGNSUB
004742
004742  012767  004750  174140        MOV     #65$,$LPERR
                                                                ; CLEAR DTR
004750                                                      LET     @RCSR := @RCSR CLR.BY #DTR
004750  042777  000002  174302        BIC     #DTR,@RCSR
                                                                ;WAIT 1 MILLI-SEC FOR CABLE
004756                                                      WAITMS  1
004756  010546                        MOV     R5,-(SP)
004760  012745  000001                MOV     #1,-(R5)
004764  004767  004672                JSR     PC,WAIT
004770  012605                        MOV     (SP)+,R5
                                                                ; READ RCSR - TO CLEAR DATAINT
004772                                                      LET     R3 := @RCSR
004772  017703  174262                MOV     @RCSR,R3
                                                                ; READ RCSR AGAIN
004776                                                      IF      #DATAINT SETIN @RCSR THEN
004776  032777  100000  174254        BIT     #DATAINT,@RCSR
005004  001401                        BEQ     $72
                                                                ; READING RCSR DID NOT CLEAR DATAINT
005006                                                          ERRHRD 61,EDATAINT
005006  104061                        ERROR   61
005010                                                      ENDIF
005010                          $72:
                                ENDSUB
005010
```

```
                                                        ;  DTR SETTING SETS DATAINT
                                                                BGNSUB
005010
005010  012767  005016  174072          MOV     #64$,$LPERR

                                                            ;SET DTR
                                                        LET     @RCSR  := @RCSR SET.BY #DTR
005016
005016  052777  000002  174234          BIS     #DTR,@RCSR
                                                        IF      #DATAINT NOTSETIN @RCSR THEN
005024
005024  032777  100000  174226          BIT     #DATAINT,@RCSR
005032  001001                          BNE     $73

                                                            ;SETTING DTR DID NOT SET DATAINT
                                                            ERRHRD 62,, E2DATA
005034
005034  104062                          ERROR   62
                                                        ENDIF
005036
                                $73:
005036
                                                        IF      #DATAINT SETIN @RCSR THEN
005036  032777  100000  174214          BIT     #DATAINT,@RCSR
005044  001401                          BEQ     $74

                                                            ;READING RCSR DID NOT CLEAR DATAINT
                                                            ERRHRD 63,EDATAINT
005046
005046  104063                          ERROR   63
                                                        ENDIF
005050
                                $74:
005050
                                                                ENDSUB
                                                        ; DTR CLEARING SETS DATAINT
                                                                BGNSUB
005050
005050  012767  005056  174032          MOV     #64$,$LPERR

                                                            ;CLEAR DTR
                                                        LET     @RCSR := @RCSR CLR.BY #DTR
005056
005056  042777  000002  174174          BIC     #DTR,@RCSR
                                                      IF        #DATAINT NOTSETIN @RCSR THEN
005064
005064  032777  100000  174166          BIT     #DATAINT,@RCSR
005072  001001                          BNE     $75

                                                            ;CLEARING DTR DID NOT SET DATAINT
                                                            ERRHRD 64,,E2DATA
005074
005074  104064                          ERFOR   64
                                                        ENDIF
005076
                                $75:
005076                                                  ENDSUB
005076                                          ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 23        TEST THAT DATAINT SETS WHEN RING SETS
                                ;*              AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
                                ;*              THE (-FR) JUMPER MUST BE IN FOR THIS TEST.
                                ;;*****************************************************************
005076  000004                  TST23:  SCOPE
005100  012767  000010  174052          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005106  012767  000023  174064          MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
005114                                                          IF      #CABLE+FRFD NOTSETIN $USWR THEN
005114  032767  060000  174076          BIT     #CABLE+FRFD,$USWR
005122  001004                          BNE     $76
                                                                        ; CAN'T TEST WITHOUT BERG OR H315
                                                                        ; OR WITH (-FR) JUMPER OUT.
005124                                                              EXIT TST
005124  012767  000001  174026          MOV     #1,$TIMES
005132  000473                          BR      TST24           ;;;EXIT THIS TEST
005134                                                          ENDIF
005134                          $76:
                                                                        ;NO INTERRUPTS
005134  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
005140  012746  005146                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005144  000002                          RTI                     ;;POP NEW PC AND PS
005146                          64$:
                                                ;START OFF WITH EVERYTHING CLEAR
005146                                                          BGNSUB
005146  012767  005154  173734          MOV     #65$,$LPERR
                                                                        ;CLEAR RING
005154                                                              LET     @RCSR := @RCSR CLR.BY #REQSEND
005154  042777  000004  174076          BIC     #REQSEND,@RCSR
                                                                        ;WAIT 1 MILLI-SEC FOR CABLE
005162                                                              WAITMS  1
005162  010546                          MOV     R5,-(SP)
005164  012745  000001                  MOV     #1,-(R5)
005170  004767  004466                  JSR     PC,WAIT
005174  012605                          MOV     (SP)+,R5
                                                                        ;READ ONCE
005176                                                              LET     R3 := @RCSR
005176  017703  174056                  MOV     @RCSR,R3
                                                                        ;READ TWICE
005202                                                              IF      #DATAINT SETIN @RCSR THEN
005202  032777  100000  174050          BIT     #DATAINT,@RCSR
005210  001401                          BEQ     $77
                                                                        ;READING RCSR DID NOT CLEAR DATAINT
005212                                                                  ERRHRD 65, EDATAINT
005212  104065                          ERROR   65
005214                                                              ENDIF
005214                          $77:
005214                                                          ENDSUB

                                                ;       SET RING --> SET DATAINT
005214                                                          BGNSUB
```

```
2531  005214  012767  005222  173666          MOV     #64$,$LPERR
2532
2533                                                                    ;WE ARE SETTING RING
2534                                                                    ; SET RING
2535  005222                                                        LET     @RCSR := @RCSR SET.BY #REQSEND
2536  005222  052777  000004  174030          BIS     #REQSEND,@RCSR
2537                                                                    ;WAIT 1 MILLI-SEC FOR CABLE
2538  005230                                                        WAITMS  1
2539  005230  010546                          MOV     R5,-(SP)
2540  005232  012745  000001                  MOV     #1,-(R5)
2541  005236  004767  004420                  JSR     PC,WAIT
2542  005242  012605                          MOV     (SP)+,R5
2543  005244                                                        IF      #DATAINT NOTSETIN @RCSR THEN
2544  005244  032777  100000  174006          BIT     #DATAINT,@RCSR
2545  005252  001001                          BNE     $100
2546                                                                    ;SETTING RING DID NOT SET DATAINT
2547  005254                                                            ERRHRD 122,, E2DATA
2548  005254  104122                          ERROR   122
2549  005256                                                        ENDIF
2550  005256                          $100:
2551  005256                                                    ENDSUB
2552
2553                                                        ;CLEAR RING CAUSES NO CHANGE IN DATAINT (CLEAR)
2554  005256                                                    BGNSUB
2555  005256  012767  005264  173624          MOV     #64$,$LPERR
2556
2557                                                                    ;CLEAR RING
2558  005264                                                        LET     @RCSR := @RCSR CLR.BY #REQSEND
2559  005264  042777  000004  173766          BIC     #REQSEND,@RCSR
2560                                                                    ;WAIT 1 MILLI-SEC FOR CABLE
2561  005272                                                        WAITMS  1
2562  005272  010546                          MOV     R5,-(SP)
2563  005274  012745  000001                  MOV     #1,-(R5)
2564  005300  004767  004356                  JSR     PC,WAIT
2565  005304  012605                          MOV     (SP)+,R5
2566  005306                                                            IF      #DATAINT SETIN @RCSR THEN
2567  005306  032777  100000  173744          BIT     #DATAINT,@RCSR
2568  005314  001401                          BEQ     $101
2569                                                                        ;CLEARING RING SET DATAINT
2570  005316                                                                ERRHRD 123, CLRRNG
2571  005316  104123                          ERROR   123
2572  005320                                                            ENDIF
2573  005320                          $101:
2574  005320                                                        ENDSUB
2575  005320                                                        EXIT    ;SKIP AROUND MESSAGE
2576  005320  000400                          BR      TST24       ;;;EXIT THIS TEST
2577  005322                                                        ENDTST
2578
2579
2580
```

```
                                      ;;*************************************************************
                                      ;;*************************************************************
                                      ;*TEST 24          TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE
                                      ;;*************************************************************
005322  000004                        TST24:  SCOPE
005324  012767  000010  173626                MOV     #10,$TIMES          ;;DO 10 ITERATIONS
005332  012767  000024  173640                MOV     #24,$TESTN          ;;SET TEST NUMBER IN APT MAIL BOX
                                                                                    ; CAN WE USE THE WRAPAROUND??
005340                                                                     IF       #CABLE NOTSETIN $USWR THEN
005340  032767  020000  173652                BIT     #CABLE,$USWR
005346  001004                                BNE     $102
                                                                                    ; CAN'T TEST WITHOUT BERG OR H315.
005350                                                                        EXIT TST
005350  012767  000001  173602                MOV     #1,$TIMES
005356  000454                                BR      TST25               ;;;EXIT THIS TEST
005360                                                                     ENDIF
005360                                $102:
                                                                                    ;NO INTERRUPTS
005360  012746  000340                        MOV     #PR7,-(SP)          ;;PUT NEW PS ON STACK
005364  012746  005372                        MOV     #64$,-(SP)          ;;PUT NEW PC ON  STACK
005370  000002                                RTI                         ;;POP NEW PC AND PS
005372                                64$:


                                                                           ;START FRESH
                                                                                    ;CLEAR SECREC
005372                                                                     LET      @RCSR := @RCSR CLR.BY #SECXMIT
005372  042777  000010  173660                BIC     #SECXMIT,@RCSR
005400                                                                     LET      R3 := @RCSR
005400  017703  173654                        MOV     @RCSR,R3

                                                        ;SET SECREC --> DATAINT SET
005404                                                                     BGNSUB
005404  012767  005412  173476                MOV     #65$,$LPERR

                                                                                    ;SET SECREC
005412                                                                        LET     @RCSR := @RCSR SET.BY #SECXMIT
005412  052777  000010  173640                BIS     #SECXMIT,@RCSR
                                                                                    ;WAIT 1 MILLI-SEC FOR CABLE
005420                                                                        WAITMS  1
005420  010546                                MOV     R5,-(SP)
005422  012745  000001                        MOV     #1,-(R5)
005426  004767  004230                        JSR     PC,WAIT
005432  012605                                MOV     (SP)+,R5
005434                                                                        IF      #DATAINT NOTSETIN @RCSR THEN
005434  032777  100000  173616                BIT     #DATAINT,@RCSR
005442  001001                                BNE     $103
                                                                                    ;SETTING SECREC DID NOT SET DATAINT
005444                                                                              ERRHRD 124,, E2DATA
005444  104124                                ERROR   124
005446                                                                        ENDIF
005446                                $103:
005446                                                                     ENDSUB

                                                        ;CLEAR SECREC --> DATAINT SET
```

```
                                                                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
```

```
2637  005446
2638  005446  012767  005454  173434          MOV     #64$,$LPERR
2639
2640  005454
2641  005454  042777  000010  173576          BIC     #SECXMIT,@RCSR
2642
2643  005462
2644  005462  010546                          MOV     R5,-(SP)
2645  005464  012745  000001                  MOV     #1,-(R5)
2646  005470  004767  004166                  JSR     PC,WAIT
2647  005474  012605                          MOV     (SP)+,R5
2648  005476
2649  005476  032777  100000  173554          BIT     #DATAINT,@RCSR
2650  005504  001001                          BNE     $104
2651
2652  005506
2653  005506  104125                          ERROR   125
2654  005510
2655  005510                          $104:
2656  005510
2657  005510
2658
2659
2660
```

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*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.
                        ;;*****************************************************************
005510  000004          TST25:  SCOPE
005512  012767  000001  173440          MOV     #1,$TIMES       ;;DO 1 ITERATION
005520  012767  000025  173452          MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                        ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
005526                                  IF #APTENV SETIN $ENV THEN
005526  032767  000001  173460          BIT     #APTENV,$ENV
005534  001404                          BEQ     $105
005536                                                  EXIT TEST
005536  012767  000001  173414          MOV     #1,$TIMES
005544  000454                          BR      TST26           ;;EXIT THIS TEST
005546                                  ENDIF
005546                  $105:
005546                                                  BGNSUB
005546  012767  005554  173334          MOV     #64$,$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
                                                        LET @TBUF :B= #0

005554                                                  ;WAIT A MAXIMUM
005554  105077  173510          CLRB    @TBUF           ;OF 50 MSEC FOR
                                                        ;XMIT RDY TO SET IN TCSR
005560                                                  CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
005560  010546                  MOV     R5,-(SP)
005562  012745  177777          MOV     #SET,-(R5)
005566  016745  173472          MOV     TCSR,-(R5)
005572  012745  000200          MOV     #XMITRDY,-(R5)
005576  012745  000500          MOV     #500,-(R5)
005602  004767  003576          JSR     PC,TIMER
005606  012605                  MOV     (SP)+,R5

                                                        ;TIMER RETURNS AN ERROR IF BIT DID
                                                        ;NOT MEET CONDITION WITHIN TIME LIMIT
005610                                                  IF.ERROR THEN
005610  103001                  BCC     $106
005612                                                  ;XMIT RDY DID NOT SET IN TCSR
005612  104066                  ERROR   66              ERRHRD 66,,DIDNOT
005614                                                  ENDIF
005614                  $106:
005614                                                  ENDSUB
005614                                                  BGNSUB
005614  012767  005622  173266          MOV     #64$,$LPERR
                                            ; LOAD TBUF WITH A SECOND CHARACTER
                                            ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
                                            ; AND THEN WAIT FOR IT TO SET
```

```
2717
2718
2719  005622
2720  005622  105077  173442          CLRB    @TBUF
2721  005626  000240                          NOP
2722
2723
2724  005630
2725  005630  032777  000200  173426  BIT     #XMITRDY,@TCSR
2726  005636  001401                  BEQ     $107
2727
2728  005640
2729  005640  104067                  ERROR   67
2730  005642
2731  005642                  $107:
2732
2733
2734
2735
2736  005642
2737  005642  010546                  MOV     R5,-(SP)
2738  005644  012745  177777          MOV     #SET,-(R5)
2739  005650  016745  173410          MOV     TCSR,-(R5)
2740  005654  012745  000200          MOV     #XMITRDY,-(R5)
2741  005660  012745  000500          MOV     #500,-(R5)
2742  005664  004767  003514          JSR     PC,TIMER
2743  005670  012605                  MOV     (SP)+,R5
2744  005672
2745  005672  103001                  BCC     $110
2746
2747  005674
2748  005674  104070                  ERROR   70
2749  005676
2750  005676                  $110:
2751  005676
2752  005676
```

```
        ;SEND SECOND CHARACTER
    LET @TBUF :B= #0

; GIVE IT TIME TO CLEAR
        ; XMITRDY SHOULD HAVE CLEARED UPON
        ; RECEIPT OF A CHARACTER
    IF #XMITRDY SETIN @TCSR THEN

        ; XMITRDY DID NOT CLEAR IN TCSR
        ERRHRD 67,,DIDNOT

    ENDIF

        ;WAIT A MAXIMUM
        ;OF 50 MSEC FOR
        ;XMIT RDY TO SET IN TCSR
    CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>

    IF.ERROR THEN

        ;XMIT RDY DID NOT SET IN TCSR
        ERRHRD 70,,DIDNOT

    ENDIF

  ENDSUB
ENDTST
```

```
2753                                    ;;*****************************************************************
2754                                    ;;*****************************************************************
2755                                    ;*TEST 26        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
2756                                    ;*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
2757                                    ;*              AND THAT RESET CLEARS THE BIT.
2758                                    ;*****************************************************************
2759  005676  000004                   TST26:  SCOPE
2760  005700  012767  000010  173252           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2761  005706  012767  000026  173264           MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2762
2763                                                                            ; SET THE MAINTENANCE BIT
2764  005714                                                                LET @TCSR := @TCSR SET.BY #MAINT
2765  005714  052777  000004  173342           BIS     #MAINT,@TCSR
2766
2767  005722                                                                BGNSUB
2768  005722  012767  005730  173160           MOV     #64$,$LPERR
2769                                                ; SEND A CHARACTER AND LET IT WRAP AROUND
2770
2771  005730                                                                    LET @TBUF :B= #0
2772  005730  105077  173334                   CLRB    @TBUF
2773
2774                                                                       ; WAIT A MAXIMUM OF 50 MSEC
2775                                                                       ; FOR RCVR DONE TO SET IN
2776                                                                       ; RCSR
2777  005734                                                               CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
2778  005734  010546                           MOV     R5,-(SP)
2779  005736  012745  177777                   MOV     #SET,-(R5)
2780  005742  016745  173312                   MOV     RCSR,-(R5)
2781  005746  012745  000200                   MOV     #RCVRDONE,-(R5)
2782  005752  012745  000500                   MOV     #500,-(R5)
2783  005756  004767  003422                   JSR     PC,TIMER
2784  005762  012605                           MOV     (SP)+,R5
2785
2786                                                                        ;DIDN'T SET IN TIME
2787  005764                                                               IF.ERROR THEN
2788  005764  103001                   BCC     $111
2789                                                                           ; RCVRDONE DID NOT SET IN RCSR
2790  005766                                                                   ERRHRD 71,,DIDNOT
2791  005766  104071                   ERROR   71
2792  005770                                                               ENDIF
2793  005770                           $111:
2794
2795  005770                                                            ENDSUB
2796
2797  005770                                                            BGNSUB
2798  005770  012767  005776  173112           MOV     #64$,$LPERR
2799                                                ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
2800                                                ; THIS ALSO WILL CLEAR THE MAINT. BIT
2801  005776                                                                BRESET
2802  005776  000005                   RESET
2803
2804  006000                                                                IF #RCVRDONE SETIN @RCSR THEN
2805  006000  032777  000200  173252           BIT     #RCVRDONE,@RCSR
2806  006006  001401                           BEQ     $112
2807                                                                           ; RCVRDONE DID NOT RESET IN RCSR.
2808  006010                                                                   ERRHRD 72,,DIDNOT
```

```
2809  006010  104072                          ERROR   72
2810  006012                                                  ENDIF
2811  006012                          $112:
2812  006012                                              ENDSUB
2813  006012                                          ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;;TEST 27        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
                                ;;**********************************************************************
006012  000004                  TST27:  SCOPE
006014  012767  000010  173136          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006022  012767  000027  173150          MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                                        ; SET MAINT. BIT
006030                                                          LET @TCSR := @TCSR SET.BY #MAINT
006030  052777  000004  173226          BIS     #MAINT,@TCSR
006036                                                          BGNSUB
006036  012767  006044  173044          MOV     #64$,$LPERR
                                            ; OUTPUT A CHARACTER WITH MAINTENANCE
                                            ; SET, AND WAIT FOR XMITRDY TO SET.

                                                                    ; OUTPUT A CHARACTER
006044                                                          LET @TBUF :B= #0
006044  105077  173220                  CLRB    @TBUF

                                                                ; WAIT MAXIMUM OF 500 MSEC
                                                                ; FOR RCVRDONE TO SET IN
                                                                ; RCSR
006050                                                          CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
006050  010546                          MOV     R5,-(SP)
006052  012745  177777                  MOV     #SET,-(R5)
006056  016745  173176                  MOV     RCSR,-(R5)
006062  012745  000200                  MOV     #RCVRDONE,-(R5)
006066  012745  000500                  MOV     #500,-(R5)
006072  004767  003306                  JSR     PC,TIMER
006076  012605                          MOV     (SP)+,R5

                                                                    ; DID IT BECAME READY?
006100                                                          IF.ERROR  THEN
006100  103001                          BCC     $113

                                                                    ;RCVRDONE DID NOT SET IN RCSR
006102                                                              ERRHRD 73,, DIDNOT
006102  104073                          ERROR   73
006104                                                          ENDIF
006104                          $113:
006104                                                      ENDSUB

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

                                                                    ;READ BUFFER
006104                                                          LET R0 :B= @RBUF
006104  117700  173152                  MOVB    @RBUF,R0

006110                                                          IF #RCVRDONE SETIN @RCSR THEN
006110  032777  000200  173142          BIT     #RCVRDONE,@RCSR
006116  001401                          BEQ     $114

                                                                    ;RCVRDONE DID NOT CLEAR IN RCSR
006120                                                              ERRHRD 74,DIDNOT
006120  104074                          ERROR   74
006122                                                          ENDIF
006122                          $114:
006122                                                  ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 30        TEST THAT RCVRACT - RCSR 11 - SETS
                                ;*              WHEN A START BIT IS RECEIVED AND
                                ;*              CLEARS WHEN RCVRDONE - RCSR 7 - SETS
                                ;;**********************************************************************
006122  000004                  TST30:  SCOPE
006124  012767  000010  173026          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006132  012767  000030  173040          MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
                                                IF #APTENV SETIN $ENV THEN
006140  032767  000001  173046          BIT     #APTENV,$ENV
006146  001404                          BEQ     $115
                                                        EXIT TEST
006150  012767  000001  173002          MOV     #1,$TIMES
006156  000500                          BR      TST31           ;;;EXIT THIS TEST
                                                ENDIF
006160                          $115:
                                                        LET @TCSR := @TCSR SET.BY #MAINT
006160  052777  000004  173076          BIS     #MAINT,@TCSR
                                                            LET R0 := #CLR
006166  012700  000000                  MOV     #CLR,R0
                                                            LET R1 := #0
006172  005001                          CLR     R1
                                            ;LOAD A CHARACTER INTO TBUF
                                            ;WAIT FOR RCVRACT TO SET

                                                        ;SEND A CHARACTER
                                                    LET @TBUF :B= #0
006174  105077  173070                  CLRB    @TBUF
                                                REPEAT
006200                          $116:
006200  032777  004000  173052          BIT     #RCVRACT,@RCSR
006206  001403                          BEQ     $117
                                                    IF #RCVRACT SETIN @RCSR THEN
006210  012700  177777                  MOV     #SET,R0
                                                        LET R0 := #SET
006214  000401                          BR      $120
                                                ELSE
006216                          $117:
                                                        LET R1 := R1 + #1
006216  005201                          INC     R1
                                                ENDIF
006220                          $120:
                                            UNTIL R0 EQ #SET OR R1 HI MAX
006220  020027  177777                  CMP     R0,#SET
006224  001403                          BEQ     $121
006226  020167  000124                  CMP     R1,MAX
006232  101762                          BLOS    $116
006234                          $121:
                                            IF R1 HI MAX THEN
006234  020167  000116                  CMP     R1,MAX
006240  101407                          BLOS    $122
                                                    ;IT NEVER SET
                                                    ;RCVRACT DID NOT SET IN RCSR.
```

```
2926  006242                                                          ERRHRD 75,, DIDNOT
2927  006242  104075                  ERROR   75
2928  006244                                                          LET R0 := @RBUF ; CLEAR BUFFER
2929  006244  017700  173012          MOV     @RBUF,R0
2930  006250                                                          EXIT TEST
2931  006250  012767  000001  172702  MOV     #1,$TIMES
2932  006256  000440                  BR      TST31          ;;;EXIT THIS TEST
                                                                      ENDIF
2933  006260
2934  006260                  $122:
2935
2936
2937                                          ;CHECK FOR TIMING OF RCVRACT. CLEARING
2938                                          ;VS RCVRDONE SETTING
2939
2940
2941  006260                                                          WHILE #RCVRACT SETIN @RCSR DO
2942  006260                  $123:
2943  006260  032777  004000  172772  BIT     #RCVRACT,@RCSR
2944  006266  001416                  BEQ     $124
2945
2946  006270                                                          IF #RCVRDONE SETIN @RCSR THEN
2947  006270  032777  000200  172762  BIT     #RCVRDONE,@RCSR
2948  006276  001411                  BEQ     $125
2949  006300                                                             IF #RCVRACT SETIN @RCSR THEN
2950  006300  032777  004000  172752  BIT     #RCVRACT,@RCSR
2951  006306  001405                  BEQ     $126
2952                                                                  ;RCVRDONE AND RCVRACT
2953                                                                      ;BOTH SET
2954  006310                                                              ERRHRD 76, DONEACT
2955  006310  104076                  ERROR   76
2956                                                                  ;NO USE CONTINUING
2957  006312                                                              EXIT TST
2958  006312  012767  000001  172640  MOV     #1,$TIMES
2959  006320  000417                  BR      TST31          ;;;EXIT THIS TEST
2960  006322                                                           ENDIF
2961  006322                  $126:
2962  006322                                                        ENDIF
2963  006322                  $125:
2964  006322                                                     ENDDO
2965  006322  000756                  BR      $123
2966  006324                  $124:
2967
2968                                                                  ;RCVRACT = 0 NOW.
2969  006324                                                          IF #RCVRDONE NOTSETIN @RCSR THEN
2970  006324  032777  000200  172726  BIT     #RCVRDONE,@RCSR
2971  006332  001001                  BNE     $127
2972                                                                      ;RCVRDONE DID NOT SET IN RCSR
2973  006334                                                              ERRHRD 77,,DIDNOT
2974  006334  104077                  ERROR   77
2975                                                                      ;SET IT BACK.
2976  006336                                                          ENDIF
2977  006336                  $127:
2978                                          ;TEST THAT READING THE RECEIVER
2979                                          ;BUFFER CLEARS RCVRDONE
2980
2981
```

```
                                                                    ;READ CHAR.
                                                                LET R0 := @RBUF
006336
006336  017700  172720          MOV     @RBUF,R0
006342
006342  032777  000200  172710  BIT     #RCVRDONE,@RCSR
006350  001401                  BEQ     $130                    IF #RCVRDONE SETIN @RCSR THEN
                                                                    ;RCVRDONE DID NOT CLEAR IN RCSR
006352                                                              ERRHRD 100,,DIDNOT
006352  104100                  ERROR   100
006354                                                          ENDIF
006354                  $130:
                                                                    EXIT
006354  000401                  BR      TST31           ;;;EXIT THIS TEST
006356  070000                  MAX:70000
                                                            ENDTST
006360
```

```
3001
3002                                    ;;**********************************************************************
3003                                    ;;**********************************************************************
3004                                    ;*TEST 31        TEST THE OVERRUN BIT - RBUF 14
3005                                    ;;**********************************************************************
3006   006360  000004                   TST31:  SCOPE
3007   006362  012767  000010  172570           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
3008   006370  012767  000031  172602           MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3009
3010   006376                                                           BGNSUB
3011   006376  012767  006404  172504           MOV     #64$,$LPERR
3012                                                    ;OUTPUT 2 CHARACTERS WITH
3013                                                    ;AMPLE DELAYS BETWEEN FOR RECEPTION.
3014                                                    ;THIS SHOULD AN CAUSE OVERRUN ERROR.
3015
3016                                                                         ;OUTPUT 1 CHARACTER
3017   006404                                                          LET @TBUF :B= #0
3018   006404  105077  172660           CLRB    @TBUF
3019                                                                         ;GO AWAY FOR 200. M SEC
3020   006410                                                          WAITMS 200.
3021   006410  010546                   MOV     R5,-(SP)
3022   006412  012745  000310           MOV     #200.,-(R5)
3023   006416  004767  003240           JSR     PC,WAIT
3024   006422  012605                   MOV     (SP)+,R5
3025
3026                                                                         ;OUTPUT 2ND CHARACTER
3027   006424                                                          LET @TBUF :B= #0
3028   006424  105077  172640           CLRB    @TBUF
3029                                                                         ;LET OVERRUN HAPPEN
3030   006430                                                          WAITMS 200.
3031   006430  010546                   MOV     R5,-(SP)
3032   006432  012745  000310           MOV     #200.,-(R5)
3033   006436  004767  003220           JSR     PC,WAIT
3034   006442  012605                   MOV     (SP)+,R5
3035
3036                                                                         ;READ BUFFER AND ERROR BITS
3037   006444                                                          LET R4 := @RBUF
3038   006444  017704  172612           MOV     @RBUF,R4
3039
3040                                                                         ;IT DIDN'T SET
3041   006450                                                          IF #ORERR NOTSETIN R4 THEN
3042   006450  032704  040000           BIT     #ORERR,R4
3043   006454  001005                   BNE     $131
3044                                                                         ;ORERR DID NOT SET IN RBUF
3045   006456                                                                ERRHRD 101,,DIDNOT
3046   006456  104101                   ERROR   101
3047
3048                                                                         ;NO USE COMPOUNDING ERRORS
3049   006460                                                                EXIT TST
3050   006460  012767  000001  172472           MOV     #1,$TIMES
3051   006466  000456                   BR      TST32           ;;;EXIT THIS TEST
3052   006470                                                                ENDIF
3053   006470                          $131:
3054   006470                                                          ENDSUB
3055
3056                                    ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
```

```
3057  006470                                                          BGNSUB
3058  006470  012767  006476  172412          MOV     #64$,$LPERR
3059  006476                                      IF #ERROR NOTSETIN R4 THEN
3060  006476  032704  100000                  BIT     #ERROR,R4
3061  006502  001005                          BNE     $132
3062
3063                                                                  ;ERROR DID NOT SET IN RBUF
3064  006504                                                          ERRHRD 102,,DIDNOT
3065  006504  104102                          ERROR   102
3066
3067                                                                  ;-WHEN ORERR SET.
3068                                                                  ;GET OUT NOW.
3069  006506                                                          EXIT TST
3070  006506  012767  000001  172444          MOV     #1,$TIMES
3071  006514  000443                          BR      TST32           ;;;EXIT THIS TEST
3072  006516                                      ENDIF
3073  006516                              $132:
3074  006516                                                          ENDSUB
3075
3076  006516                                                          BGNSUB
3077  006516  012767  006524  172364          MOV     #64$,$LPERR
3078                                              ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
3079
3080  006524                                      IF #ORERR NOTSETIN @RBUF THEN
3081  006524  032777  040000  172530          BIT     #ORERR,@RBUF
3082  006532  001002                          BNE     $133
3083
3084                                                                  ;READING RBUF CLEARED ORERR.
3085  006534                                                          ERRHRD 103,ITCLRED
3086  006534  104103                          ERROR   103
3087                                                                  ;SKIP REST OF TEST
3088  006536                                                          EXIT
3089  006536  000432                          BR      TST32           ;;;EXIT THIS TEST
3090  006540                                      ENDIF
3091  006540                              $133:
3092  006540                                                          ENDSUB
3093
3094  006540                                                          BGNSUB
3095  006540  012767  006546  172342          MOV     #64$,$LPERR
3096                                              ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
3097
3098                                                                  ;SEND A CHARACTER AROUND.
3099  006546                                                      LET @TBUF :B= #0
3100  006546  105077  172516                  CLRB    @TBUF
3101                                                                  ;LET IT CIRCULATE
3102  006552                                                      WAITMS 200.
3103  006552  010546                          MOV     R5,-(SP)
3104  006554  012745  000310                  MOV     #200.,-(R5)
3105  006560  004767  003076                  JSR     PC,WAIT
3106  006564  012605                          MOV     (SP)+,R5
3107
3108  006566                                      IF #ORERR SETIN @RBUF THEN
3109  006566  032777  040000  172466          BIT     #ORERR,@RBUF
3110  006574  001405                          BEQ     $134
3111
3112  006576                                                          ;ORERR DID NOT CLEAR IN RBUF
                                                                      ERRHRD 104,,DIDNOT
```

```
3113  006576  104104                    ERROR   104
3114
3115                                                          ;-AFTER RECEIVING ANOTHER CHAR
3116                                                          ;SKIP AROUND REST
3117  006600                                                  EXIT TST
3118  006600  012767  000001  172352      MOV     #1,$TIMES
3119  006606  000406                      BR      TST32               ;;;EXIT THIS TEST
3120  006610                                                      ENDIF
3121  006610                        $134:
3122
3123  006610                                                      IF #ERROR SETIN @RBUF THEN
3124  006610  032777  100000  172444      BIT     #ERROR,@RBUF
3125  006616  001401                      BEQ     $135
3126                                                          ;ERROR DID NOT CLEAR IN RBUF
3127  006620                                                  ERRHRD 105,,DIDNOT
3128  006620  104105                    ERROR   105
3129
3130  006622                                                      ENDIF
3131  006622                        $135:
3132  006622                                                  ENDSUB
3133  006622                                                  EXIT
3134  006622  000400                      BR      TST32               ;;;EXIT THIS TEST
3135                                                              .EVEN
3136  006624                                                  ENDTST
3137
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*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.
                                ;;**********************************************************************
006624  000004                  TST32:  SCOPE
006626  012767  000010  172324          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006634  012767  000032  172336          MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
006642                                                                  IF #PBR NOTSETIN $USWR THEN
006642  032767  000200  172350          BIT     #PBR,$USWR
006650  001004                          BNE     $136
006652                                                                      EXIT TST
006652  012767  000001  172300          MOV     #1,$TIMES
006660  000552                          BR      TST33           ;;;EXIT THIS TEST
006662                                                                  ENDIF
006662                          $136:
                                                        ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
006662                                                                  IF #APTENV SETIN $ENV THEN
006662  032767  000001  172324          BIT     #APTENV,$ENV
006670  001404                          BEQ     $137
006672                                                                      EXIT TEST
006672  012767  000001  172260          MOV     #1,$TIMES
006700  000542                          BR      TST33           ;;;EXIT THIS TEST
006702                                                                  ENDIF
006702                          $137:
006702                                                                  LET OLD := #-1
006702  012767  177777  000272          MOV     #-1,OLD
006710                                                                  LET OLD+2 := #-1
006710  012767  177777  000266          MOV     #-1,OLD+2
006716                                                                  LET @TCSR := @TCSR SET.BY #MAINT
006716  052777  000004  172340          BIS     #MAINT,@TCSR
006724                                                                      ;EACH BAUD RATE
006724  005003                          CLR     R3              INCR R3 FROM #0 TO #15. BY #1
006726  000401                          BR      $140
006730                          $141:
006730  005203                          INC     R3
006732                          $140:
006732  020327  000017                  CMP     R3,#15.
006736  003060                          BGT     $142
006740                                                                  LET R0 := @RBUF
006740  017700  172316                  MOV     @RBUF,R0
006744                                                                      ;CHANGE BAUDE RATE
006744  116377  007124  172314          MOVB    RATES(R3),@TCSRHI       LET @TCSRHI :8= RATES(R3)
006752                                                                      ;FLAG
006752  005002                          CLR     BIT             LET BIT := #0
006754                                                                      ;OUTPUT THE CHARACTER
006754  005077  172310                  CLR     @TBUF           LET @TBUF := #0
                                                                        ;INITIALIZE COUNTER
```

```
3194  006760
3195  006760  005067  000212          CLR     NEW
3196  006764
3197  006764  005067  000210          CLR     NEW+2
3198  006770
3199  006770                      $143:
3200  006770  005702                  TST     BIT
3201  006772  001014                  BNE     $144
3202  006774
3203  006774  032777  000200  172256  BIT     #RCVRDONE,@RCSR
3204  007002  001403                  BEQ     $145
3205
3206  007004
3207  007004  012702  000001          MOV     #1,BIT
3208  007010
3209  007010  000404                  BR      $146
3210  007012                      $145:
3211
3212  007012
3213  007012  005267  000160          INC     NEW
3214  007016
3215  007016  005567  000156          ADC     NEW+2
3216  007022
3217  007022                      $146:
3218
3219  007022
3220  007022  000762                  BR      $143
3221  007024                      $144:
3222
3223  007024
3224  007024  026767  000150  000152  CMP     NEW+2,OLD+2
3225  007032  103001                  BHIS    $147
3226
3227  007034
3228  007034  000412                  BR      $150
3229  007036                      $147:
3230
3231  007036
3232  007036  026767  000136  000140  CMP     NEW+2,OLD+2
3233  007044  001005                  BNE     $151
3234  007046  026767  000124  000126  CMP     NEW,OLD
3235  007054  103001                  BHIS    $151
3236
3237  007056
3238  007056  000401                  BR      $152
3239  007060                      $151:
3240
3241
3242
3243
3244  007060
3245  007060  104126                  ERROR   126
3246  007062
3247  007062                      $152:
3248  007062
3249  007062                      $150:
```

```
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
                                                    LET OLD+2 := NEW+2
                                                ENDINC  ;BAUD RATE
007062
007062  016767  000110  000112          MOV     NEW,OLD
007070
007070  016767  000104  000106          MOV     NEW+2,OLD+2
007076
007076  000714                          BR      $141
007100                          $142:
007100                                                  LET R3 :B= $USWR+1 AND #17      ; PUT BAUD BACK
007100  116703  172115                  MOVB    $USWR+1,R3
007104  110346                          MOVB    R3,-(SP)
007106  142716  000017                  BICB    #17,(SP)
007112  142603                          BICB    (SP)+,R3
007114                                                  LET @TCSRHI :B= RATES(R3)       ; LIKE HE WANTED IT
007114  116377  007124  172144          MOVB    RATES(R3),@TCSRHI
                                                EXIT    ;SKIP TABLE
007122
007122  000431                          BR      TST33           ;;;EXIT THIS TEST

007124                          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
007124      010                 R0050:  .BYTE   010     ;       50      0
007125      030                 R0070:  .BYTE   030     ;       70      1
007126      050                 R0110:  .BYTE   050     ;       110     2
007127      070                 R0135:  .BYTE   070     ;       135     3
007130      110                 R0150:  .BYTE   110     ;       150     4
007131      130                 R0300:  .BYTE   130     ;       300     5
007132      150                 R0600:  .BYTE   150     ;       600     6
007133      170                 R0200:  .BYTE   170     ;       1200    7
007134      210                 R1800:  .BYTE   210     ;       1800    10
007135      230                 R2000:  .BYTE   230     ;       2000    11
007136      250                 R2400:  .BYTE   250     ;       2400    12
007137      270                 R3600:  .BYTE   270     ;       3600    13
007140      310                 R4800:  .BYTE   310     ;       4800    14
007141      330                 R7200:  .BYTE   330     ;       7200    15
007142      350                 R9600:  .BYTE   350     ;       9600    16
007143      370                 R10000: .BYTE   370     ;       19200   17

007144  040502  042125  051040  BAUDRATE:       .ASCIZ  /BAUD RATE DIDN'T CHANGE./
007152  052101  020105  044504
007160  047104  052047  041440
007166  040510  043516  027105
007174     000
        007176                  .EVEN
007176  000000  000000          NEW:    0,0
007202  000000  000000          OLD:    0,0
                                                                ENDTST
007206
```

```
3306
3307
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*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
                                ;*****************************************************************
007206  000004                  TST33:  SCOPE
007210  012767  000010  171742          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007216  012767  000033  171754          MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
007224                                                                  LET INTFLAG := #0
007224  005067  002520                  CLR     INTFLAG

                                                                        ;GET VECTOR ADDRESS
007230                                                                  LET R3 := DLVEC
007230  016703  172022                  MOV     DLVEC,R3
                                                                        ;FOR THE TRANSMITTER
007234                                                                  LET R3 := R3 + #4
007234  062703  000004                  ADD     #4,R3
                                                                        ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
007240                                                                  SETVEC R3, #INTSRV, #PR7
007240  010146                  MOV     R1,-(SP)
007242  010301                  MOV     R3,R1
007244  012721  011742          MOV     #INTSRV,(R1)+
007250  012711  000340          MOV     #PR7,(R1)
007254  012601                  MOV     (SP)+,R1
007256                                                                  BGNSUB
007256  012767  007264  171624          MOV     #65$,$LPERR

                                                                        ;CLEAR INTERRUPT ENABLE
007264                                                                  LET @TCSR := @TCSR CLR.BY #XMITIE
007264  042777  000100  171772          BIC     #XMITIE,@TCSR

                                                                        ;SET IT TO 0
007272  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007276  012746  007304          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
007302  000002                  RTI                     ;;POP NEW PC AND PS
007304                  65$:

                                                                        ;NOW SET I.E. BIT
007304                                                                  LET @TCSR := @TCSR SET.BY #XMITIE
007304  052777  000100  171752          BIS     #XMITIE,@TCSR

                                                                        ;LET INTERRUPT HAVE TIME TO OCCUR
007312                                                                  WAITMS 200.
007312  010546                  MOV     R5,-(SP)
007314  012745  000310          MOV     #200.,-(R5)
007320  004767  002336          JSR     PC,WAIT
007324  012605                  MOV     (SP)+,R5

                                                                        ;DID EXACTLY 1 INTERRUPT OCCUR
007326                                                                  IF INTFLAG NE #1 THEN
007326  026727  002416  000001          CMP     INTFLAG,#1
007334  001406                  BEQ     $153
```

```
                                                                        ;NO - WAS IT 0 OR MORE THAN ONCE
                                                                        IF INTFLAG EQ #0 THEN
007336
007336  005767  002406                TST     INTFLAG
007342  001002                        BNE     $154
                                                                        ;TRANSMITTER DID NOT INTERRUPT IN TIME
                                                                            ERRHRD 106,,DIDNOT
007344
007344  104106                        ERROR   106
                                                                        ELSE
007346
007346  000401                        BR      $155
                                $154:
                                                                        ;TWICE
                                                                        ;TRANSMITTER INTERRUPTED TWICE
                                                                            ERRHRD 107,,TWICE
007350
007350  104107                        ERROR   107
                                                                        ENDIF
007352
                                $155:
                                                                    ENDIF
                                $153:
                                                                ENDSUB
                                                        ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
                                                                BGNSUB
007352
007352  012767  007360  171530        MOV     #64$,$LPERR
                                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
007360                                                              LET INTFLAG := #0
007360  005067  002364                CLR     INTFLAG
                                                                        ;CLEAR INTERRUPT ENABLE
007364                                                              LET @TCSR := @TCSR CLR.BY #XMITIE
007364  042777  000100  171672        BIC     #XMITIE,@TCSR
                                                                        ;NO INTERRUPTS SHOULD OCCUR.
007372  012746  000000                MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007376  012746  007404                MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
007402  000002                        RTI                     ;;POP NEW PC AND PS
                                65$:
                                                                        ;DARE IT TO HAPPEN
007404                                                              WAITMS 2
007404
007404  010546                        MOV     R5,-(SP)
007406  012745  000002                MOV     #2,-(R5)
007412  004767  002244                JSR     PC,WAIT
007416  012605                        MOV     (SP)+,R5
007420                                                              IF INTFLAG NE #0 THEN
007420  005767  002324                TST     INTFLAG
007424  001401                        BEQ     $156
                                                                        ;INTERRUPT OCCURED WITH I E CLEARED
007426                                                                  ERRHRD 110,NOTENAB
007426  104110                        ERROR   110
                                                                    ENDIF
007430
                                $156:
                                                                    BRESET
007430
007430  000005                        RESET
                                                                ENDSUB
007432                                                                  ;RESTORE VECTOR AREA
                                                                CLRVEC R3
007432
007432  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
007434  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
007436  012701  000003                MOV     #R3,R1
007442  010102                        MOV     R1,R2
```

```
3420  007444  062702  000002          ADD     #2,R2
3421  007450  010221                  MOV     R2,(R1)+
3422  007452  005011                  CLR     (R1)
3423  007454  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
3424  007456  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
3425
3426  007460                              ENDTST
3427
3428
3429
3430
3431
3432
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 34        RECEIVER INTERRUPT LOGIC TEST
                                ;*              THIS TEST COVERS ALL OF THE RECEIVER
                                ;*              SIDE OF THE INTERRUPT LOGIC, BOTH DATASET
                                ;*              AND CHARACTER MODES.
                                ;*****************************************************************
007460  000004                  TST34:  SCOPE
007462  012767  000010  171470          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007470  012767  000034  171502          MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR INTERRUPT OCCURED FLAG
                                                                        ;SET UP RECEIVER INTER.VECTOR
                                                                SETVEC DLVEC,#INTSRV,#PR7
007476
007476  010146                          MOV     R1,-(SP)
007500  016701  171552                  MOV     DLVEC,R1
007504  012721  011742                  MOV     #INTSRV,(R1)+
007510  012711  000340                  MOV     #PR7,(R1)
007514  012601                          MOV     (SP)+,R1
                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
                                                                BGNSUB
007516
007516  012767  007524  171364          MOV     #64$,$LPERR
007524                                                          LET INTFLAG := #0
007524  005067  002220                  CLR     INTFLAG
                                                                        ;SET MAINT. BIT
007530                                                          LET @TCSR := @TCSR SET.BY #MAINT
007530  052777  000004  171526          BIS     #MAINT,@TCSR
                                                                        ;CLEAR INTERRUPTS
007536                                                              LET @RCSR := @RCSR CLR.BY #RCVRIE
007536  042777  000100  171514          BIC     #RCVRIE, RCSR
                                                                        ;CHANGE PRIORITY
                                                                        ;..TO 0
007544  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007550  012746  007556                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
007554  000002                          RTI                     ;;POP NEW PC AND PS
007556                          65$:
                                        ;SEND A CHARACTER
007556                                                  LET @TBUF :B= #0
007556  105077  171506          CLRB    @TBUF
                                        ;WAIT A MAXIMUM
                                ;OF 500 MSEC FOR
                                ;RCVR RDY TO SET IN RCSR
007562                                                  CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
007562  010546                  MOV     R5,-(SP)
007564  012745  177777          MOV     #SET,-(R5)
007570  016745  171464          MOV     RCSR,-(R5)
007574  012745  000200          MOV     #RCVRDONE,-(R5)
007600  012745  000500          MOV     #500,-(R5)
007604  004767  001574          JSR     PC,TIMER
007610  012605                  MOV     (SP)+,R5
                                        ;SET INTERRUPT ENABLE
007612                                                  LET @RCSR := @RCSR SET.BY #RCVRIE
007612  052777  000100  171440  BIS     #RCVRIE,@RCSR
                                        ;LET IT COME IN.
007620                                                  WAITMS 1
007620  010546                  MOV     R5,-(SP)
```

```
3489  007622  012745  000001          MOV     #1,-(R5)
3490  007626  004767  002030          JSR     PC,WAIT
3491  007632  012605                  MOV     (SP)+,R5
3492  007634
3493  007634  017700  171422          MOV     @RBUF,R0
3494
3495
3496  007640
3497  007640  026727  002104  000001  CMP     INTFLAG,#1
3498  007646  001406                  BEQ     $157
3499
3500  007650
3501  007650  005767  002074          TST     INTFLAG
3502  007654  001002                  BNE     $160
3503
3504  007656
3505  007656  104111                  ERROR   111
3506
3507  007660
3508  007660  000401                  BR      $161
3509  007662                  $160:
3510
3511  007662
3512  007662  104112                  ERROR   112
3513  007664
3514  007664                  $161:
3515  007664
3516  007664                  $157:
3517
3518  007664
3519  007664  042777  000004  171372  BIC     #MAINT,@TCSR
3520
3521  007672
3522  007672  042777  000100  171360  BIC     #RCVRIE,@RCSR
3523  007700
3524
3525
3526
3527
3528
3529
3530
3531                                  ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3532  007700                                                  BGNSUB
3533  007700  012767  007706  171202  MOV     #64$,$LPERR
3534  007706                                                  IF #CABLE NOTSETIN $USWR THEN
3535  007706  032767  020000  171304  BIT     #CABLE,$USWR
3536  007714  001004                  BNE     $162
3537                                                              ;CAN'T TEST WITHOUT A CABLE
3538  007716                                                      EXIT'TST
3539  007716  012767  000001  171234  MOV     #1,$TIMES
3540  007724  000466                  BR      TST35           ;;;EXIT THIS TEST
3541  007726                                                  ENDIF
3542  007726                  $162:
3543                                                              ; CLEAR 'INTFLAG'
3544  007726                                                  LET INTFLAG := #0
```

```
        LET R0 := @RBUF ; CLEAR RCVRDONE

            ;DID HE DO IT RIGHT?
        IF INTFLAG NE #1 THEN

            ;NONE OCCURED
            IF INTFLAG EQ #0 THEN

            ;RECEIVER DID NOT INTERRUPT IN TIME
                ERRHRD 111,,DIDNOT

            ;TWICE OR MORE
            ELSE

            ;RECEIVER INTERRUPTED TWICE
                ERRHRD 112,,TWICE

            ENDIF

        ENDIF

            ;RESET MAINT. BIT.
    LET @TCSR := @TCSR CLR.BY #MAINT

            ; CLEAR INTERRUPT ENABLE
    LET @RCSR := @RCSR CLR.BY #RCVRIE

    ENDSUB
```

```
3545  007726  005067  002016          CLR     INTFLAG
3546                                                        ;CLEAR INTERRUPTS
3547  007732                                          LET @RCSR := @RCSR CLR.BY #DATAIE
3548  007732  042777  000040  171320  BIC     #DATAIE,@RCSR
3549                                                        ;CHANGE PRIORITY
3550                                                        ;..TO 0
3551  007740  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3552  007744  012746  007752          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3553  007750  000002                  RTI                     ;;POP NEW PC AND PS
3554  007752                  64$:
3555  007752                                          LET @RCSR := @RCSR CLR.BY #REQSEND
3556  007752  042777  000004  171300  BIC     #REQSEND,@RCSR
3557                                                        ;SET INTERRUPT ENABLE
3558  007760                                          LET @RCSR := @RCSR SET.BY #DATAIE
3559  007760  052777  000040  171272  BIS     #DATAIE,@RCSR
3560  007766                                          LET    @RCSR := @RCSR SET.BY #REQSEND
3561  007766  052777  000004  171264  BIS     #REQSEND,@RCSR
3562                                                        ;LET IT COME IN.
3563  007774                                          WAITMS 1
3564  007774  010546                  MOV     R5,-(SP)
3565  007776  012745  000001          MOV     #1,-(R5)
3566  010002  004767  001654          JSR     PC,WAIT
3567  010006  012605                  MOV     (SP)+,R5
3568
3569                                                        ; DID IT DO IT RIGHT?
3570  010010                                          IF INTFLAG NE #1 THEN
3571  010010  026727  001734  000001  CMP     INTFLAG,#1
3572  010016  001406                  BEQ     $163
3573
3574  010020                                              ;NONE OCCURED
3575  010020  005767  001724          TST     INTFLAG     IF INTFLAG EQ #0 THEN
3576  010024  001002                  BNE     $164
3577
3578  010026                                              ;DATAINT  DID NOT INTERRUPT IN TIME
3579  010026  104113                  ERROR   113             ERRHRD 113,,DIDNOT
3580                                                      ;TWICE OR MORE
3581  010030                                              ELSE
3582  010030  000401                  BR      $165 ·
3583  010032                  $164:
3584                                                      ; DATAINT INTERRUPTED TWICE
3585  010032                                                  ERRHRD 114,,TWICE
3586  010032  104114                  ERROR   114
3587  010034                                              ENDIF
3588  010034                  $165:
3589  010034                                          ENDIF
3590  010034                  $163:
3591  010034                                          LET @RCSR := @RCSR CLR.BY #DATAIE
3592  010034  042777  000040  171216  BIC     #DATAIE,@RCSR
3593  010042                                          LET @RCSR := @RCSR CLR.BY #REQSEND
3594  010042  042777  000004  171210  BIC     #REQSEND,@RCSR
3595  010050                                      ENDSUB
3596
3597  010050                                      LET R4 := @DLVEC
3598  010050  017704  171202          MOV     @DLVEC,R4
3599  010054                                      CLRVEC R4
3600  010054  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
```

```
 3601  010056  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 3602  010060  012701  000004            MOV     #R4,R1
 3603  010064  010102                    MOV     R1,R2
 3604  010066  062702  000002            ADD     #2,R2
 3605  010072  010221                    MOV     R2,(R1)+
 3606  010074  005011                    CLR     (R1)
 3607  010076  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
 3608  010100  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
 3609  010102                                            ENDTST
```

```
3610
3611
3612
3613
3614
3615  010102  000004                    ;;*****************************************************************
3616  010104  012767  000001  171046    ;;*****************************************************************
3617  010112  012767  000035  171060    ;*TEST 35       TEST ACTUAL DATA TRANSFERED
3618                                     ;*              NON-INTERRUPT MAINTENANCE BIT SET
3619  010120                            ;;*****************************************************************
3620  010120  052777  000004  171136    ;;*****************************************************************
3621                                    T35:  SCOPE
3622                                          MOV     #1,$TIMES       ;;DO 1 ITERATION
3623                                          MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3624  010126  012746  000000                                                 ;SET MAINT. BIT
3625  010132  012746  010140                                         LET @TCSR := @TCSR SET.BY #MAINT
3626  010136  000002                          BIS     #MAINT,@TCSR
3627  010140                            64$:
3628
3629  010140                                                          ;CHANGE PRIORITY
3630  010140  162705  000002                                         ;..TO 0
3631  010144  004767  001412                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3632  010150  012501                          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3633                                          RTI                     ;;POP NEW PC AND PS
3634  010152
3635  010152  017700  171104                                                 ;GET DATA MASK.
3636                                                                  CALL DATLNG OUT <R1>
3637
3638  010156                                  SUB     #1*2,R5
3639  010156  005002                          JSR     PC,DATLNG
3640  010160  000401                          MOV     (R5)+,R1
3641  010162                            $167:
3642  010162  005202                                                         ; START CLEAN
3643  010164                                                          LET R0 := @RBUF
3644  010164  020227  000377                  MOV     @RBUF,R0
3645  010170  003047
3646                                                                          ;ALL BINARY CHAR.
3647                                                                  INCR R2 FROM #0 TO #377 BY #1
3648
3649
3650  010172                                  CLR     R2
3651  010172  010546                          BR      $166
3652  010174  012745  177777           $167:
3653  010200  016745  171060                  INC     R2
3654  010204  012745  000200           $166:
3655  010210  012745  000500                  CMP     R2,#377
3656  010214  004767  001164                  BGT     $170
3657  010220  012605
3658
3659                                                                  ;TRANSMIT CHAR IN R2
3660  010222
3661  010222  110277  171042                                        CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
3662
3663  010226                                  MOV     R5,-(SP)
3664  010226  010546                          MOV     #SET,-(R5)
3665  010230  012745  177777                  MOV     TCSR,-(R5)
                                               MOV     #XMITRDY,-(R5)
                                               MOV     #500,-(R5)
                                               JSR     PC,TIMER
                                               MOV     (SP)+,R5

                                                   ;TRANSMIT IT
                                               LET @TBUF :B= R2

                                               MOVB    R2,@TBUF

                                               CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>

                                               MOV     R5,-(SP)
                                               MOV     #SET,-(R5)
```

```
3666  010234  016745  171020          MOV     RCSR,-(R5)
3667  010240  012745  000200          MOV     #RCVRDONE,-(R5)
3668  010244  012745  000500          MOV     #500.,-(R5)
3669  010250  004767  001130          JSR     PC,TIMER
3670  010254  012605                  MOV     (SP)+,R5
3671                                                          ;AND SAVE IT
3672  010256                                                LET R3 := @RBUF
3673  010256  017703  171000          MOV     @RBUF,R3
3674
3675                                                          ;COMPARE TO SEE IF WE RECEIVED IT ALL
3676
3677                                                              ;CLEAN OFF NON-DATA BITS
3678                                                              ;ON BOTH TRANSMITTED AND
3679                                                          LET R4 := R2 CLR.BY R1
3680  010262
3681  010262  010204                  MOV     R2,R4
3682  010264  040104                  BIC     R1,R4
3683  010266                                                  LET R3 := R3 CLR.BY R1
3684  010266  040103                  BIC     R1,R3
3685
3686                                                          ;RECEIVED DATA
3687  010270                                                  IF R4 NE R3 THEN
3688  010270  020403                  CMP     R4,R3
3689  010272  001405                  BEQ     $171
3690
3691  010274                                                      ;DATA COMPARE ERROR
3692  010274  104116                  ERROR   116             ERRHRD 116,COMP,SBWAS
3693  010276
3694  010276  012767  000001  170654  MOV     #1,$TIMES       EXIT TST  ; ON ERROR
3695  010304  000404                  BR      TST36   ;;;EXIT THIS TEST
3696  010306                                                  ENDIF
3697  010306                  $171:
3698  010306                                                ENDINC  ; R2
3699  010306  000725                  BR      $167
3700  010310                  $170:
3701
3702                                                          ;RESET MAINT. BIT.
3703  010310                                                LET @TCSR := @TCSR CLR.BY #MAINT
3704  010310  042777  000004  170746  BIC     #MAINT,@TCSR
3705  010316                                                ENDTST
3706
3707
3708
```

```
                                ;;**************************************************************
                                ;;**************************************************************
                                ;*TEST 36        TEST DATA THROUGH CABLE
                                ;;**************************************************************
010316  000004                  TST36:  SCOPE
010320  012767  000001  170632          MOV     #1,$TIMES       ;;DO 1 ITERATION
010326  012767  000036  170644          MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
010334                                                                  IF #CABLE NOTSETIN $USWR THEN
010334  032767  020000  170656          BIT     #CABLE,$USWR
010342  001004                          BNE     $172
                                                                            ;CAN'T TEST WITHOUT A CABLE
                                                                        EXIT TST
010344
010344  012767  000001  170606          MOV     #1,$TIMES
010352  000474                          BR      TST37           ;;;EXIT THIS TEST
010354                                                                  ENDIF
010354                          $172:
                                                                            ;DON'T USE MAINT.
010354                                                                  LET @TCSR := @TCSR CLR.BY #MAINT
010354  042777  000004  170702          BIC     #MAINT,@TCSR
                                                                            ;CHANGE PRIORITY
                                                                            ;..TO 0
010362  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
010366  012746  010374                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
010372  000002                          RTI                     ;;POP NEW PC AND PS
010374                          64$:
                                                                            ;GET DATA MASK
010374                                                                  CALL DATLNG OUT <R1>
010374  162705  000002                  SUB     #1*2,R5
010400  004767  001156                  JSR     PC,DATLNG
010404  012501                          MOV     (R5)+,R1
010406                                                                  LET R0 := @RBUF ; START CLEAN
010406  017700  170650                  MOV     @RBUF,R0
                                                                            ;BINARY COUNT PATTERN
010412                                                                  INCR R2 FROM #0 TO #377 BY #1
010412  005002                          CLR     R2
010414  000401                          BR      $173
                                $174:
010416  005202                          INC     R2
010420                          $173:
010420  020227  000377                  CMP     R2,#377
010424  003047                          BGT     $175


                                                                ;TRANSMIT THE CHAR. IN R2.

010426                                                          CALL TIMER IN <#500,#XMITRDY,TCSR,#SET>
010426  010546                  MOV     R5,-(SP)
010430  012745  177777          MOV     #SET,-(R5)
010434  016745  170624          MOV     TCSR,-(R5)
010440  012745  000200          MOV     #XMITRDY,-(R5)
010444  012745  000500          MOV     #500,-(R5)
010450  004767  000730          JSR     PC,TIMER
010454  012605                  MOV     (SP)+,R5


                                                                    ;START IT ON ITS WAY
010456                                                          LET @TBUF :B= R2
```

```
3765  010456  110277  170606          MOVB    R2,@TBUF
3766  010462
3767  010462  010546                  MOV     R5,-(SP)
3768  010464  012745  177777          MOV     #SET,-(R5)
3769  010470  016745  170564          MOV     RCSR,-(R5)
3770  010474  012745  000200          MOV     #RCVRDONE,-(R5)
3771  010500  012745  000500          MOV     #500,-(R5)
3772  010504  004767  000674          JSR     PC,TIMER
3773  010510  012605                  MOV     (SP)+,R5
3774
3775
3776  010512
3777  010512  017703  170544          MOV     @RBUF,R3
3778
3779
3780  010516
3781  010516  010204                  MOV     R2,R4
3782  010520  040104                  BIC     R1,R4
3783  010522
3784  010522  040103                  BIC     R1,R3
3785
3786
3787  010524
3788  010524  020403                  CMP     R4,R3
3789  010526  001405                  BEQ     $176
3790
3791  010530
3792  010530  104117                  ERROR   117
3793  010532
3794  010532  012767  000001  170420  MOV     #1,$TIMES
3795  010540  000401                  BR      TST37
3796  010542
3797  010542                  $176:
3798
3799  010542
3800  010542  000725                  BR      $174
3801  010544                  $175:
3802
3803
3804
3805  010544
3806
3807
3808
3809
```

```
                        CALL TIMEP IN <#500,#RCVRDONE,RCSR,#SET>




                            ;RETRIEVE
                        LET R3 := @RBUF

                            ;STRIP OFF JUNK ON BOTH
                        LET R4 := R2 CLR.BY R1

                        LET R3 := R3 CLR.BY R1

                            ;WE HAVE TROUBLE
                        IF R4 NE R3 THEN

                            ;DATA COMPARE ERROR
                            ERRHRD 117,COMP,SBWAS

                            EXIT TST  ; ON ERROR

;;;EXIT THIS TEST
                        ENDIF


                                ENDINC  ; R2


                        ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 37        FULL DATA TRANSFER WITH INTERRUPTS
                                ;*              AND MAINTENANCE MODE.
                                ;;**********************************************************************
010544  000004                  TST37:  SCOPE
010546  012767  000001  170404          MOV     #1,$TIMES       ;;DO 1 ITERATION
010554  012767  000037  170416          MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                        ; THIS TEST IS 'BREAK OR HALT' SINSATIVE.
010562                                          IF #APTENV SETIN $ENV THEN
010562  032767  000001  170424          BIT     #APTENV,$ENV
010570  001404                          BEQ     $177
                                                        EXIT TEST
010572
010572  012767  000001  170360          MOV     #1,$TIMES
010600  000550                          BR      TST40           ;;EXIT THIS TEST
010602                                  ENDIF
010602                  $177:
                                                        ;GET DATA MASK
010602                                          CALL DATLNG OUT <R3>
010602  162705  000002                  SUB     #1*2,R5
010606  004767  000750                  JSR     PC,DATLNG
010612  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
010614  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
010620  012746  010626                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
010624  000002                          RTI                     ;;POP NEW PC AND PS
010626                  64$:
                                                        ;GET VECTOR ADDRESS
010626                                          LET R1 := DLVEC
010626  016701  170424                  MOV     DLVEC,R1
                                                        ;RCVR VECTOR
010632                                          LET (R1)+ := #REC
010632  012721  011026                  MOV     #REC,(R1)+
010636                                          LET (R1)+ := #PR7
010636  012721  000340                  MOV     #PR7,(R1)+
                                                        ;POINT TO TRANSMITTER VECTOR
                                                        ;AND SET IT UP ALSO
010642                                          LET (R1)+ := #TRAN
010642  012721  010764                  MOV     #TRAN,(R1)+
010646                                          LET (R1)  := #PR7
```

```
3866  010646  012711  000340                  MOV     #PR7,(R1)
3867
3868
3869  010652                                                      ; INITIALIZE COUNTERS
3870  010652  012701  177777                  MOV     #-1,R1     LET R1 := #-1
3871
3872  010656                                                      ;RECEIVER STORAGE
3873  010656  005002                          CLR     R2         LET R2 := #0
3874
3875  010660                                                      ;# OF RECEIVED CHAR. COUNT.
3876  010660  012704  177777                  MOV     #-1,R4     LET R4 := #-1
3877
3878                                                              ; CLEAR ERROR COUNT.
3879  010664                                                     LET ERRCNT := #0
3880  010664  005067  000066                  CLR     ERRCNT
3881
3882  010670                                                     BRESET  ;SET UP ALL REGISTERS
3883  010670  000005                          RESET
3884
3885  010672                                                      ;SET UP MAINTENANCE
3886  010672  052777  000004  170364          BIS     #MAINT,@TCSR  LET @TCSR := @TCSR SET.BY #MAINT
3887
3888
3889  010700                                                      ;SET I.E. IN TRANSMITTER
3890  010700  052777  000100  170356          BIS     #XMITIE,@TCSR  LET @TCSR := @TCSR SET.BY #XMITIE
3891
3892  010706                                                      ;AND RECEIVER
3893  010706  052777  000100  170344          BIS     #RCVRIE,@RCSR  LET @RCSR := @RCSR SET.BY #RCVRIE
3894
3895
3896                                                             ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3897  010714                                                     REPEAT
3898  010714                          $200:
3899  010714                                                     UNTIL R4 EQ NUMBER OR ERRCNT GT #0
3900  010714  020467  000040                  CMP     R4,NUMBER
3901  010720  001403                          BEQ     $201
3902  010722  005767  000030                  TST     ERRCNT
3903  010726  003772                          BLE     $200
3904  010730                          $201:
3905
3906                                                              ;DATA COMPARE ERRORS.
3907  010730                                                     IF ERRCNT NE #0 THEN
3908  010730  005767  000022                  TST     ERRCNT
3909  010734  001401                          BEQ     $202
3910
3911  010736                                                      ;DATA COMPARE ERROR
3912  010736  104120                          ERROR   120        ERRHRD 120,COMP,FIRST
3913  010740                                                     ENDIF
3914  010740                          $202:
3915
3916  010740                                                     LET @TCSR := @TCSR CLR.BY #XMITIE
3917  010740  042777  000100  170316          BIC     #XMITIE,@TCSR
3918  010746                                                     LET @RCSR := @RCSR CLR.BY #XMITIE
3919  010746  042777  000100  170304          BIC     #XMITIE,@RCSR
3920
3921  010754                                                     EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
```

```
3922  010754  000462                  BR      TST40           ;;;EXIT THIS TEST
3923
3924  010756  000000                                                  ERRCNT: 0
3925  010760  000400                                                  NUMBER: 400
3926  010762     000                                                  SB:     .BYTE 0
3927  010763     000                                                  WAS:    .BYTE 0
3928
3929
3930
3931                                  ;*************************************************************
3932                                  ;TRANSMIT INTERRUPT HANDLER
3933
3934  010764                                                          BGNSRV  TRAN
3935  010764                          TRAN:
3936                                  ;*************************************************************
3937
3938                                                                      ;INCREMENT CHAR COUNT
3939  010764                                                          LET R1 := R1 + #1
3940  010764  00520C:                 INC     R1
3941                                                                  ;SET UP FOR TRANSFER
3942  010766                                                          LET HOLD := R1      CLR.BY R3
3943  010766  010167  000030          MOV     R1,HOLD
3944  010772  040367  000024          BIC     R3,HOLD
3945
3946  010776                                                              ;AND SEND.
3947  010776  016777  000020  170264  MOV     HOLD,@TBUF              LET @TBUF := HOLD
3948
3949  011004                                                              ;ALL DONE
3950  011004  020167  177750          CMP     R1,NUMBER               IF R1 EQ NUMBER THEN
3951  011010  001003                  BNE     $203
3952
3953  011012                                                              ;STOP   INTERRUPT PROCESSING
3954  011012  042777  000100  170244  BIC     #XMITIE,@TCSR           LET @TCSR := @TCSR CLR.BY #XMITIE
3955  011020                                                          ENDIF
3956  011020                          $203:
3957
3958  011020  000401                  BR      ZZZ                         ; EXIT SRV
3959
3960  011022  000000                                                      HOLD:0
3961
3962  011024                          ZZZ:                            ENDSRV
3963  011024  000002                  RTI
3964
3965
3966                                  ;*************************************************************
3967
3968                                  ;RECEIVER INTERRUPT HANDLER
3969  011026                                                          BGNSRV  REC
3970  011026                          REC:
3971                                  ;*************************************************************
3972
3973                                                                      ;COUNT THIS CHAR.
3974  011026                                                          LET R4 := R4 + #1
3975  011026  005204                  INC     R4
3976                                                                  ;GET CHAR IN + MASK IT
3977  011030                                                          LET R2 := @RBUF CLR.BY R3
```

```
 3978  011030  017702  170226          MOV     @RBUF,R2
 3979  011034  040302                  BIC     R3,R2
 3980
 3981  011036
 3982  011036  010467  000054          MOV     R4,RHLD
 3983  011042  040367  000050          BIC     R3,RHLD
 3984
 3985
 3986  011046
 3987  011046  020267  000044          CMP     R2,RHLD
 3988  011052  001412                  BEQ     $204
 3989
 3990  011054
 3991  011054  005767  177676          TST     ERRCNT
 3992  011060  001005                  BNE     $205
 3993
 3994  011062
 3995  011062  116767  000030  177672  MOVB    RHLD,SB
 3996  011070
 3997  011070  110267  177667          MOVB    R2,WAS
 3998  011074
 3999  011074                  $205:
 4000
 4001  011074
 4002  011074  005267  177656          INC     ERRCNT
 4003  011100
 4004  011100                  $204:
 4005
 4006
 4007  011100
 4008  011100  020467  177654          CMP     R4,NUMBER
 4009  011104  001003                  BNE     $206
 4010
 4011  011106
 4012  011106  042777  000100  170144  BIC     #RCVRIE,@RCSR
 4013
 4014  011114
 4015  011114                  $206:
 4016
 4017
 4018  011114  000401                  BR      ZZZZ
 4019
 4020  011116  000000
 4021  011120                  ZZZZ:
 4022  011120
 4023  011120  000002                  RTI
 4024
 4025  011122
 4026
 4027
 4028
```

```
    ;RHLD WILL CONTAIN EXPECTED INPUT
LET RHLD := R4 CLR.BY R3


    ;DO THEY COMPARE
IF R2 NE RHLD THEN

    ;FIRST ERROR
    IF ERRCNT EQ #0 THEN

    ;SAVE RECORD OF FIRST MISS
        LET SB :B= RHLD

        LET WAS :B= R2

    ENDIF

    ;COUNT IT.
    LET ERRCNT := ERRCNT + #1

ENDIF

    ;ALL DONE?
IF R4 EQ NUMBER THEN

    ;STOP RECEIVER INTERRUPTS
    LET @RCSR := @RCSR CLR.BY #RCVRIE
        ;MAIN REPEAT LOOP IS CHECKING
    ENDIF

    ;FOR 'R4 = NUMBER' ALSO
    EXIT SRV                    ;

    RHLD:0

ENDSRV

ENDTST
```

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*TEST 40        TEST BREAK GENERATION LOGIC
                        ;*               TRANSMIT KNOWN CHAR WITH BREAK SET
                        ;*               AND COMPARE RECEIVED WITH 0.
                        ;;*****************************************************************
011122  000004          TST40:  SCOPE
011124  012767  000010  170026          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
011132  012767  000040  170040          MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;SET MAINTENANCE BIT
                                                                LET @TCSR := @TCSR SET.BY #MAINT
011140  052777  000004  170116          BIS     #MAINT,@TCSR
                                                                        ; CLEAR RCVRDONE JUST IN CASE
                                                                LET R0 := @RBUF
011146  017700  170110                  MOV     @RBUF,R0
                                                                        ;SET BREAK BIT
                                                                LET @TCSR := @TCSR SET.BY #BREAK
011152  052777  000001  170104          BIS     #BREAK,@TCSR
                                                                        ;NON-ZERO CHAR. '*'
                                                                LET @TBUF := #252
011160  012777  000252  170102          MOV     #252,@TBUF
                                                                CALL TIMER IN <#500,#RCVRDONE,RCSR,#SET>
011166  010546                          MOV     R5,-(SP)
011170  012745  177777                  MOV     #SET,-(R5)
011174  016745  170060                  MOV     RCSR,-(R5)
011200  012745  000200                  MOV     #RCVRDONE,-(R5)
011204  012745  000500                  MOV     #500,-(R5)
011210  004767  000170                  JSR     PC,TIMER
011214  012605                          MOV     (SP)+,R5
                                                                IF.ERROR THEN
011216  103001                          BCC     $207
                                                                        ; RECIEVER DONE DID NOT SET
                                                                    ERRHRD 115
011220  104115                          ERROR   115
                                                                ENDIF
011222  000000          $207:
                                                                IFB @RBUF NE #0 THEN
011222  105777  170034                  TSTB    @RBUF
011226  001401                          BEQ     $210
                                                                        ; BREAK DID NOT EQUAL 0
                                                                    ERRHRD 121 ,BADBRK
011230  104121                          ERROR   121
                                                                ENDIF
011232  000000          $210:
                                                                BRESET  ;CLEAN UP
011232  000005                          RESET
                                                                EXIT
011234  000413                          BR      TST41           ;;;EXIT THIS TEST
011236  051102  040505  020113          BADBRK: .ASCIZ  /BREAK DID NOT EQUAL 0/
011244  044504  020104  047516
011252  020124  050505  040525
011260  020114  000060
```

```
 4085
 4086  011264                                                  ENDTST
```

```
 4087
 4088                                   ;;**********************************************************************
 4089                                   ;*TEST 41        NOT A TEST - SEND BACK TO LOOP
 4090                                   ;;**********************************************************************
 4091  011264  000004                  TST41:  SCOPE
 4092  011266  012767  000001  167664          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4093  011274  104401  011302                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 4094  011300  000404                          BR      64$             ;;GET OVER THE ASCIZ
 4095                                  ;;65$:   .ASCIZ  <CRLF>*CSR: *
 4096  011312                          64$:
 4097  011312  016746  167736                  MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
 4098  011316  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4099  011320  104401  011326                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 4100  011324  000405                          BR      66$             ;;GET OVER THE ASCIZ
 4101                                  ;;67$:   .ASCIZ  *,VECTOR: *
 4102  011340                          66$:
 4103  011340  016746  167712                  MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
 4104  011344  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4105  011346  104401  011354                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 4106  011352  000405                          BR      68$             ;;GET OVER THE ASCIZ
 4107                                  ;;69$:   .ASCIZ  *,ERRORS: *
 4108  011366                          68$:
 4109  011366  016746  167520                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
 4110  011372  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 4111  011374  005067  167512                  CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
 4112  011400  000167  170314                  JMP     LOOP            ; BACK UP TO THE BEGINNING
```

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

000001                                                  TRUE=   1
000000                                                  FALSE=  0

                                                        LET     REGSAV := REG(R5) ; GET POINTER TO REGIST

011404  016567  000004  000136          MOV     REG(R5),REGSAV
                                                        LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR

011412  016567  000000  000132          MOV     HOWLONG(R5),TIMSAV
                                                        LET     FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA

011420  112767  000000  000126          MOVB    #FALSE,FLAG


                                        ;       START OF AN INFINITE LOOP

011426                                  LOOP
011426                          $213:
                                            ; TEST TO SEE IF WHICHBIT IS SET
011426  036577  000002  000114              IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
011434  001004          BIT     WHICHBIT(R5),@REGSAV
                        BNE     $215
011436                                          LET     HOLDSC :B= #CLR
011436  112767  000000  000111      MOVB    #CLR,HOLDSC
011444                                      ELSE
011444  000403          BR      $216
011446                          $215:
011446                                          LET     HOLDSC :B= #SET ; REMEMBER THIS
011446  112767  177777  000101      MOVB    #SET,HOLDSC
011454                                      ENDIF
011454                          $216:

                                            ; NOW SEE IF THAT WAS WHAT WE WANTED
011454                                      IFB     HOLDSC EQ SETCLR(R5) THEN
```

```
4169  011454  126765  000075  000006          CMPB    HOLDSC,SETCLR(R5)
4170  011462  001003                          BNE     $217
4171                                                    ; JUST THE THING WE NEEDED
                                                              LET     FLAG :B= #TRUE
4172  011464                                                  ENDIF
4173  011464  112767  000001  000062          MOVB    #TRUE,FLAG
4174  011472
4175  011472                          $217:
4176                                                  EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
4177  011472
4178  011472  126727  000056  000001          CMPB    FLAG,#TRUE
4179  011500  001414                          BEQ     $214
4180  011502  005767  000044                  TST     TIMSAV
4181  011506  003411                          BLE     $214
4182                                                    ; ONE WAY OR THE OTHER, WE ARE DONE
4183                                                ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
4184
4185  011510                                            WAITMS  1       ;WAIT FOR 10 MILLI-SECONDS
4186  011510  010546                          MOV     R5,-(SP)
4187  011512  012745  000001                  MOV     #1,-(R5)
4188  011516  004767  000140                  JSR     PC,WAIT
4189  011522  012605                          MOV     (SP)+,R5
4190  011524
4191  011524  005367  000022                  DEC     TIMSAV                  LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4192  011530                                                ENDLOOP                   ; CONTINUED AT THE TOP
4193  011530  000736                          BR      $213
4194  011532                          $214:
4195
4196                                          ; ONLY 2 WAYS TO GET HERE
4197                                          ;       1). WE RAN OUT OF TIME---ERROR ! !
4198                                          ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
4199                                                  IFB     FLAG EQ #TRUE THEN
4200  011532
4201  011532  126727  000016  000001          CMPB    FLAG,#TRUE
4202  011540  001001                          BNE     $220
4203  011542                                            RETURN  NO.ERROR    ; GOOD
4204  011542  000405                          BR      $211                    ENDIF
4205  011544
4206  011544                          $220:                                   RETURN  ERROR           ; BAD
4207  011544
4208  011544  000261                          SEC
4209  011546  000404                          BR      $212
4210
4211  011550  000000                                                  REGSAV: .WORD 0
4212  011552  000000                                                  TIMSAV: .WORD 0
4213  011554     000                                                  FLAG:   .BYTE 0
4214  011555     000                                                  HOLDSC: .BYTE 0
4215                                                  ; WE ARE DONE  GO BACK HOME
4216  011556                                                  ENDRTN
4217  011556                          $211:
4218  011556  000241                          CLC
4219  011560                          $212:
4220  011560  000207                          RTS     PC
```

```
                                ;;*****************************************************************
                                ROUTINE DATLNG (MASK)
                                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.

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

011562                                                          LET MASK(R5) := #0              ; START
011562  005065  000000                  CLR     MASK(R5)
011566                                                          LET NUMBR := $USWR AND #DATA
011566  016767  167426  000062          MOV     $USWR,NUMBR
011574  016746  000056                  MOV     NUMBR,-(SP)
011600  042716  000017                  BIC     #DATA,(SP)
011604  042667  000046                  BIC     (SP)+,NUMBR

011610                                                          INCR I FROM #1 TO NUMBR BY #1
011610  012767  000001  167454          MOV     #1,I
011616  000402                  BR      $223
011620                          $224:
011620  005267  167446                  INC     I
011624                          $223:
011624  026767  167442  000024          CMP     I,NUMBR
011632  003006                  BGT     $225

                                                                    LET MASK(R5) := MASK(R5) SHIFT #1
011634  006365  000000                  ASL     MASK(R5)
                                                                        LET MASK(R5) := MASK(R5) SET.BY #1
011640  052765  000001  000000          BIS     #1,MASK(R5)
                                                                ENDINC
011646  000764                  BR      $224
011650                          $225:
                                                                LET MASK(R5) := COMP MASK(R5)
011650  005165  000000                  COM     MASK(R5)
                                                        RETURN
011654  000401                  BR      $221
011656  000000                  NUMBR:0
                                                    ENDRTN
011660                          $221:
011660                          $222:
011660  000207                  RTS     PC
```

```
 4270
 4271                                   ;;*****************************************************************
 4272  011662                           ROUTINE WAIT    (TIME)
 4273  011662                           WAIT:
 4274                                   ;* ROUTINE:WAIT
 4275                                   ;*       THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
 4276                                   ;*       MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
 4277                                   ;*       THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
 4278                                   ;*       REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
 4279                                   ;*       TO APPROXIMATE 1 MILLI SEC.
 4280                                   ;;*****************************************************************
 4281  011662  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 4282  011664  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 4283  011666  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 4284  011670                                  LET R1 := TIME(R5)
 4285  011670  016501  000000                  MOV     TIME(R5),R1
 4286  011674                                  INCRU R2 FROM #1 TO R1 BY #1
 4287  011674  012702  000001                  MOV     #1,R2
 4288  011700  000402                          BR      $230
 4289  011702                          $231:
 4290  011702  062702  000001                  ADD     #01,R2
 4291  011706                          $230:
 4292  011706  020201                          CMP     R2,R1
 4293  011710  101010                          BHI     $232
 4294  011712                                          INCR R3 FROM #0 TO #100 BY #1
 4295  011712  005003                          CLR     R3
 4296  011714  000401                          BR      $233
 4297  011716                          $234:
 4298  011716  005203                          INC     R3
 4299  011720                          $233:
 4300  011720  020327  000100                  CMP     R3,#100
 4301  011724  003001                          BGT     $235
 4302  011726                                          ENDINC
 4303  011726  000773                          BR      $234
 4304  011730                          $235:
 4305  011730                                  ENDINC
 4306  011730  000764                          BR      $231
 4307  011732                          $232:
 4308  011732  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
 4309  011734  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
 4310  011736  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 4311  011740                          ENDRTN
 4312  011740                          $226:
 4313  011740                          $227:
 4314  011740  000207                          RTS     PC
```

```
                                .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
                                ;;*****************************************************************
011742                          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
                                                                        LET INTFLAG := INTFLAG + #1
011742                                  INC     INTFLAG
011742  005267  000002                 ENDSRV                          ;THAT'S ALL
011746                                  RTI
011746  000002                 INTFLAG:  0
011750  000000
```

```
                                ROUTINE MYTYPE
                                MYTYPE:
                                ;;*****************************************************************
                                        TYPE    ,65$                    ;;TYPE ASCIZ STRING
                                        BR      64$                     ;;GET OVER THE ASCIZ
 011752                         ;;65$:  .ASCIZ  <CRLF>*TEST # *
                                64$:
 011752  104401  011760                MOV     $TESTN,-(SP)            ;;SAVE $TESTN FOR TYPEOUT
 011756  000405                        TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
                                        TYPE    ,67$                    ;;TYPE ASCIZ STRING
                                        BR      66$                     ;;GET OVER THE ASCIZ
 011772                         ;;67$:  .ASCIZ  *,ERROR # *
 011772  016746  167202        66$:    MOV     $ITEMB,$FATAL           ; APT FATAL ERROR NUMBER
 011776  104402                        MOV     $FATAL,-(SP)            ;;SAVE $FATAL FOR TYPEOUT
 012000  104401  012006                TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 012004  000405                        TYPE    ,69$                    ;;TYPE ASCIZ STRING
                                        BR      68$                     ;;GET OVER THE ASCIZ
 012020                         ;;69$:  .ASCIZ  *,PC = *
 012020  116767  167070  167150 68$:    MOV     $ERRPC,-(SP)            ;;SAVE $ERRPC FOR TYPEOUT
 012026  016746  167144                TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 012032  104402                        TYPE    ,71$                    ;;TYPE ASCIZ STRING
 012034  104401  012042                BR      70$                     ;;GET OVER THE ASCIZ
 012040  000404                ;;71$:  .ASCIZ  *,CSR: *
                                70$:
 012052                                 MOV     DLADD,-(SP)             ;;SAVE DLADD FOR TYPEOUT
 012052  016746  167040                TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 012056  104402                        TYPE    ,73$                    ;;TYPE ASCIZ STRING
 012060  104401  012066                BR      72$                     ;;GET OVER THE ASCIZ
 012064  000404                ;;73$:  .ASCIZ  *,VECTOR: *
                                72$:
 012076                                 MOV     DLVEC,-(SP)             ;;SAVE DLVEC FOR TYPEOUT
 012076  016746  167152                TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 012102  104402                ENDRTN
 012104  104401  012112        $236:
 012110  000405                $237:
                                        RTS     PC
 012124                         
 012124  016746  167126         
 012130  104402                 
 012132                         
 012132                         
 012132  000207                 
```

```
ROUTINE CYCLE
CYCLE:
;;*****************************************************************
;* ROUTINE:     CYCLE
;*       THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
;*       ADDRESS  OF  DLV11-E  UNDER  TEST.  ADRS +2 TO
;*       POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
;*      IT  KEEPS  TRACK  OF THE CURRENT DEVICE AND BIT
;*      MASKS.
;;*****************************************************************
                                        REPEAT
$242:
                                            IF BITMASK EQ #0 THEN
        TST     BITMASK
        BNE     $243
                                                IF INITFLAG EQ #1 THEN
        CMP     INITFLAG,#1
        BNE     $244
                                                    LET INITFLAG := #0
        CLR     INITFLAG
                                                ELSE
        BR      $245
$244:
                                                    CALL $EOP   ; AS A SUBROUTINE
        JSR     PC,$EOP
                                ; BECAUSE $EOP RETURNS AS A JUMP
SPECIALADDRESS:                             LET R0 := POP
        MOV     (SP)+,R0
                                                ENDIF
$245:
                                                LET BITMASK := #1
        MOV     #1,BITMASK
                                                LET $DEVCT := #1
        MOV     #1,$DEVCT
                                                LET ADDRESS := $BASE
        MOV     $BASE,ADDRESS
                                                LET VECTOR := $VECT1
        MOV     $VECT1,VECTOR
                                            ELSE
        BR      $246
$243:
                                                LET R4 := #10
        MOV     #10,R4
                                                LET BITMASK := BITMASK ROTATE 1
        ROL     BITMASK
                                                LET ADDRESS  := ADDRESS + R4
        ADD     R4,ADDRESS
                                                LET VECTOR := VECTOR + R4
        ADD     R4,VECTOR
                                            ENDIF
$246:
                                    UNTIL   BITMASK SETIN $DEVM
        BIT     BITMASK,$DEVM
        BEQ     $242
```

```
4427  012250                                                  LET ADRS := #ADDRESS
4428  012250  012701  012266                        MOV     #ADDRESS,ADRS
4429  012254                                                  LET $DEVCT := $DEVCT + #1
4430  012254  005267  166724                        INC     $DEVCT
4431  012260                                                  RETURN
4432  012260  000404                                BR      $240
4433  012262  000000                        BITMASK: 0
4434  012264  000001                        INITFLAG: 1
4435  012266  000000                        ADDRESS: 0
4436  012270  000000                        VECTOR:  0
4437
4438  012272                                                  ENDRTN
4439  012272                                $240:
4440  012272                                $241:
4441  012272  000207                                RTS     PC
4442
4443
```

```
                                .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
                        
012274                  $EOP:
                                SCOPE
012274  000004
012276  005067  166600          CLR     $TSTNM          ;;ZERO THE TEST NUMBER
012302  005067  166652          CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
012306  005267  166670          INC     $PASS           ;;INCREMENT THE PASS NUMBER
012312  042767  100000  166662  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
012320  005327                  DEC     (PC)+           ;;LOOP?
012322  000001          $EOPCT: .WORD   1
012324  003022                  BGT     $DOAGN          ;;YES
012326  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
012330  000001          $ENDCT: .WORD   1
012332  012322                  $EOPCT
012334  104401  012401          TYPE    ,$ENDMG         ;;TYPE "END PASS #"
012340  016746  166636          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
012344  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
012346  104401  012376          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
012352  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
012356  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
012360  000005                  RESET                   ;;CLEAR THE WORLD
012362  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
012364  000240                  NOP                     ;;SAVE ROOM
012366  000240                  NOP                     ;;FOR
012370  000240                  NOP                     ;;ACT11
012372                  $DOAGN:
012372  000137                  JMP     @(PC)+          ;;RETURN
012374  012164          $RTNAD: .WORD   SPECIALADDRESS
012376     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
012401     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
012406  050040  051501  020123
012414  000043
```

```
                                        .SBTTL  POWER DOWN AND UP ROUTINES
                                        ;;*****************************************************************
                                        ;POWER DOWN ROUTINE
012416  012737  012562  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
012424  012737  000340  000026         MOV     #340,@#PWRVEC+2  ;;PRIO:7
012432  010046                        MOV     R0,-(SP)         ;;PUSH R0 ON STACK
012434  010146                        MOV     R1,-(SP)         ;;PUSH R1 ON STACK
012436  010246                        MOV     R2,-(SP)         ;;PUSH R2 ON STACK
012440  010346                        MOV     R3,-(SP)         ;;PUSH R3 ON STACK
012442  010446                        MOV     R4,-(SP)         ;;PUSH R4 ON STACK
012444  010546                        MOV     R5,-(SP)         ;;PUSH R5 ON STACK
012446  017746  166466                MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
012452  010667  000110                MOV     SP,$SAVR6        ;;SAVE SP
012456  012737  012470  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
012464  000000                        HALT
012466  000776                        BR      .-2              ;;HANG UP
                                        ;;*****************************************************************
                                        ;POWER UP ROUTINE
012470  012737  012562  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
012476  016706  000064                MOV     $SAVR6,SP        ;;GET SP
012502  005067  000060                CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
012506  005267  000054         1$:    INC     $SAVR6           ;;WAIT FOR THE INC
012512  001375                        BNE     1$               ;;OF  WORD
012514  012677  166420                MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
012520  012605                        MOV     (SP)+,R5         ;;POP STACK INTO R5
012522  012604                        MOV     (SP)+,R4         ;;POP STACK INTO R4
012524  012603                        MOV     (SP)+,R3         ;;POP STACK INTO R3
012526  012602                        MOV     (SP)+,R2         ;;POP STACK INTO R2
012530  012601                        MOV     (SP)+,R1         ;;POP STACK INTO R1
012532  012600                        MOV     (SP)+,R0         ;;POP STACK INTO R0
012534  012737  012416  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
012542  012737  000340  000026        MOV     #340,@#PWRVEC+2  ;;PRIO:7
012550  104401                        TYPE                     ;REPORT THE POWER FAILURE
012552  012570         $PWRMG: .WORD  $POWER                   ;;POWER FAIL MESSAGE POINTER
012554  012716                        MOV     (PC)+,(SP)       ;;RESTART AT START
012556  001336         $PWRAD: .WORD  START                    ;;RESTART ADDRESS
012560  000002                        RTI
012562  000000         $ILLUP: HALT                            ;;THE POWER UP SEQUENCE WAS STARTED
012564  000776                        BR      .-2              ;;  BEFORE THE POWER DOWN WAS COMPLETE
012566  000000         $SAVR6: 0                               ;;PUT THE SP HERE
012570  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
012576  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
                                ;*

012600  105767  166353          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
012604  100002                          BPL     1$              ;;BR IF YES
012606  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
012610  000430                          BR      3$              ;;LEAVE
012612  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
012614  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
012620  122767  000001  166366          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
012626  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
012630  132767  000100  166357          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
012636  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
012640  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
012644  004767  000774                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
012650  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
012652  132767  000040  166335  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
012660  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
012662  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
012664  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
012666  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
012670  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
012672  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
012676  000002                          RTI                     ;;RETURN
012700  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
012704  001430                          BEQ     8$
012706  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
012712  001006                          BNE     5$
012714  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
012716  104401                          TYPE                    ;;TYPE A CR AND LF
012720  001171                          $CRLF
012722  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
012726  000755                          BR      2$              ;;GET NEXT CHARACTER
012730  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
012734  126726  166216          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
012740  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
012742  016746  166206                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
012746  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
012752  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
012754  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
012760  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
4584  012764  000770                        BR      7$              ;;LOOP
4585
4586                                  ;HORIZONTAL TAB PROCESSOR
4587
4588  012766  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
4589  012772  004767  000014          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
4590  012776  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
4591  013004  001372                        BNE     9$              ;;TAB STOP
4592  013006  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
4593  013010  000724                        BR      2$              ;;GET NEXT CHARACTER
4594  013012  105777  166132        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
4595  013016  100375                        BPL     $TYPEC
4596  013020  116677  000002  166124        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4597  013026  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
4598  013034  001003                        BNE     1$              ;;BRANCH IF NO
4599  013036  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
4600  013042  000406                        BR      $TYPEX          ;;EXIT
4601  013044  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
4602  013052  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
4603  013054  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
4604  013056  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
4605  013060  000207          $TYPEX: RTS     PC
4606
```

```
                                        .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.
013062  022767  000176  166050  $CKSWR: CMP     #$WREG,SWR      ;;IS THE SOFT-SWR SELECTED?
013070  001074                        BNE     15$             ;;BRANCH IF NO
013072  105777  166046                TSTB    @$TKS           ;;CHAR THERE?
013076  100071                        BPL     '5$             ;;IF NO, DON'T WAIT AROUND
013100  117746  166042                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
013104  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
013110  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
013114  001062                        BNE     15$             ;;NO, RETURN TO USER
013116  126727  166012  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
013124  001456                        BEQ     15$             ;;BRANCH IF YES
                                        ;
013126  104401  013607                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
013132  104401  013614        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
013136  016746  165034                MOV     $WREG,-(SP)     ;;SAVE $WREG FOR TYPEOUT
013142  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013144  104401  013625                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
013150  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
013152  005046                        CLR     -(SP)           ;;THE NEW SWR
013154  105777  165764        7$:     TSTB    @$TKS           ;;CHAR THERE?
013160  100375                        BPL     7$              ;;IF NOT TRY AGAIN
                                        ;
013162  117746  165760                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
013166  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                        ;
                                        ;
013172  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
013176  001005                        BNE     10$             ;;BRANCH IF NOT
013200  104401  013602                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
013204  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
013210  000757                        BR      19$             ;;LET'S TRY IT AGAIN
                                        ;
013212  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
013216  001022                        BNE     16$             ;;BRANCH IF NO
013220  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
013224  001403                        BEQ     11$             ;;BRANCH IF YES
013226  016677  000002  165704        MOV     2(SP),@SWR      ;;SAVE NEW SWR
013234  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
013240  104401  001171        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
013244  126727  165665  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
013252  001003                        BNE     15$             ;;BRANCH IF NOT
013254  012777  000100  165662        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
013262  000002                15$:    RTI                     ;;RETURN
013264  004767  177522        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
013270  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?
```

```
 4719  013476  101405                          BLOS    4$              ;;BR IF YES
 4720  013500  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 4721  013502  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4722  013504  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4723  013510  001003                          BNE     3$              ;;SKIP IF NOT
 4724  013512  104401  001170          4$:     TYPE    $QUES           ;;TYPE A '?'
 4725  013516  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4726  013520  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 4727  013524  104401  013570                  TYPE    9$
 4728  013530  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 4729  013534  001356                          BNE     2$              ;;LOOP IF NOT RETURN
 4730  013536  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4731  013542  104401  001172                  TYPE    $LF             ;;TYPE A LINE FEED
 4732  013546  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 4733  013550  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4734  013552  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 4735  013560  012766  013572  000004          MOV     #$TTYIN,4(SP)
 4736  013566  000002                          RTI                     ;;RETURN
 4737  013570     00C                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 4738  013571     000                          .BYTE   0               ;;TERMINATOR
 4739  013572  000010               $TTYIN:    .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
 4740  013602  052536  005015  000  $CNTLU:    .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
 4741  013607     136  006507  000  $CNTLG:    .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
 4742  013614  005015  053523  020  $MSWR:     .ASCIZ  <15><12>/SWR = /
 4743  013622  020075     000
 4744  013625     040  047040  053  $MNEW:     .ASCIZ  /  NEW = /
 4745  013632  036440  000040
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;*****************************************************************
013636  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
013644  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
013652  000403                  BR      $ATYC
013654  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
013662                          $ATYC:
013662  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013664  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013666  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
013672  001450                          BEQ     5$              ;;IF NOT:  BR
013674  122767  000001  165312          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
013702  001031                          BNE     3$              ;;IF NOT:  BR
013704  132767  000100  165303          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
013712  001425                          BEQ     3$              ;;IF NOT:  BR
013714  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
013720  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
013726  005767  165242          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
013732  001375                          BNE     1$              ;;IF NOT:  WAIT
013734  010067  165250                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
013740  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
013742  001376                          BNE     2$
013744  166700  165240                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
013750  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
013752  010067  165234                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
013756  012767  000004  165210          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
013764  000413                          BR      5$
013766  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
013774  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
014002  016746  163770                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
014006  004767  176566                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
014012  000000                  4$:     .WORD   0
014014                          5$:
014014  105767  000062          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
014020  001416                          BEQ     12$             ;;IF NOT:  BR
014022  005767  165166                  TST     $ENV            ;;RUNNING UNDER APT?
014026  001413                          BEQ     12$             ;;IF NOT:  BR
014030  005767  165140          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
014034  001375                          BNE     11$             ;;IF NOT:  WAIT
014036  017667  000004  165132          MOV     @4(SP),$FATAL   ;;GET ERROR #
014044  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
014052  005267  165116                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
014056  105067  000020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
014062  105067  000013                  CLRB    $LFLG           ;;CLEAR LOG FLAG
014066  105067  000006                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
014072  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
014074  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
014076  000207                          RTS     PC              ;;RETURN
014100     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
014101     000          $LFLG:  .BYTE   0               ;;LOG FLAG
014102     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
014104                          .EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
 4802           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

014104                          $ERROR:
014104  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
014106  105267  164771          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
014112  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
014114  016777  164762  165020          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
014122  032777  002000  165010          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
014130  001402                          BEQ     1$              ;;NO - SKIP
014132  104401  001164                  TYPE    ,$BELL          ;;RING BELL
014136  005267  164750          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
014142  011667  164750                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
014146  162767  000002  164742          SUB     #2,$ERRPC
014154  117767  164736  164732          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
014162  032777  020000  164750          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
014170  001004                          BNE     20$             ;;SKIP TYPEOUTS
014172  004767  175554                  JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
014176  104401  001171                  TYPE    ,$CRLF
014202                          20$:
014202  122767  000001  165004          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
014210  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
014212  116767  164676  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
014220  004767  177430                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
014224     000                  21$:    .BYTE   0
014225     000                          .BYTE   0
014226  000777                  22$:    BR      22$             ;;APT ERROR LOOP
014230  005777  164704          2$:     TST     @SWR            ;;HALT ON ERROR
014234  100002                          BPL     3$              ;;SKIP IF CONTINUE
014236  000000                          HALT                    ;;HALT ON ERROR!
014240  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
014242  032777  001000  164670  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
014250  001402                          BEQ     4$              ;;BR IF NO
014252  016716  164632                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
014256  005767  164700          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
014262  001402                          BEQ     5$              ;;BR IF NONE
014264  016716  164672                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
014270                          5$:
014270  022737  012362  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
014276  001001                          BNE     6$              ;;BRANCH IF NO
014300  000000                          HALT                    ; YES
014302                          6$:
014302  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

014304                          $SCOPE:
                                        CKSWR                           ;;TEST  FOR CHANGE IN SOFT-SWR
014304  104407                  1$:     BIT     #BIT14,@SWR             ;;LOOP ON PRESENT TEST?
014306  032777  040000  164624          BNE     $OVER                   ;;YES IF SW14=1
014314  001114                  ;#####START OF CODE FOR THE XOR TESTER#####
014316  000416                  $XTSTR: BR      6$                      ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
014320  013746  000004                  MOV     @#ERRVEC,-(SP)          ;;SAVE THE CONTENTS OF THE ERROR VECTOR
014324  012737  014344  000004          MOV     #5$,@#ERRVEC            ;;SET FOR TIMEOUT
014332  005737  177060                  TST     @#177060                ;;TIME OUT ON XOR?
014336  012637  000004                  MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
014342  000463                          BR      $SVLAD                  ;;GO TO THE NEXT TEST
014344  022626                  5$:     CMP     (SP)+,(SP)+             ;;CLEAR THE STACK AFTER A TIME OUT
014346  012637  000004                  MOV     (SP)+,@#ERRVEC          ;;RESTORE THE ERROR VECTOR
014352  000423                          BR      7$                      ;;LOOP ON THE PRESENT TEST
014354                          6$:;#####END OF CODE FOR THE XOR TESTER#####
014354  032777  000400  164556          BIT     #BIT08,@SWR             ;;LOOP ON SPEC. TEST?
014362  001404                          BEQ     2$                      ;;BR IF NO
014364  127767  164550  164510          CMPB    @SWR,$TSTNM             ;;ON THE RIGHT TEST?    SWR<7:0>
014372  001465                          BEQ     $OVER                   ;;BR IF YES
014374  105767  164503          2$:     TSTB    $ERFLG                  ;;HAS AN ERROR OCCURRED?
014400  001421                          BEQ     3$                      ;;BR IF NO
014402  126767  164507  164473          CMPB    $ERMAX,$ERFLG           ;;MAX. ERRORS FOR THIS TEST OCCURRED?
014410  101015                          BHI     3$                      ;;BR IF NO
014412  032777  001000  164520          BIT     #BIT09,@SWR             ;;LOOP ON ERROR?
014420  001404                          BEQ     4$                      ;;BR IF NO
014422  016767  164462  164456  7$:     MOV     $LPERR,$LPADR           ;;SET LOOP ADDRESS TO LAST SCOPE
014430  000446                          BR      $OVER
014432  105067  164445          4$:     CLRB    $ERFLG                  ;;ZERO THE ERROR FLAG
014436  005067  164516                  CLR     $TIMES                  ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
014442  000415                          BR      1$                      ;;ESCAPE TO THE NEXT TEST
014444  032777  004000  164466  3$:     BIT     #BIT11,@SWR             ;;INHIBIT ITERATIONS?
014452  001011                          BNE     1$                      ;;BR IF YES
014454  005767  164522                  TST     $PASS                   ;;IF FIRST PASS OF PROGRAM
014460  001406                          BEQ     1$                      ;;      INHIBIT ITERATIONS
014462  005267  164416                  INC     $ICNT                   ;;INCREMENT ITERATION COUNT
014466  026767  164466  164410          CMP     $TIMES,$ICNT            ;;CHECK THE NUMBER OF ITERATIONS MADE
014474  002024                          BGE     $OVER                   ;;BR IF MORE ITERATION REQUIRED
014476  012767  000001  164400  1$:     MOV     #1,$ICNT                ;;REINITIALIZE THE ITERATION COUNTER
014504  016767  000052  164446          MOV     $MXCNT,$TIMES           ;;SET NUMBER OF ITERATIONS TO DO
014512  105267  164364          $SVLAD: INCB    $TSTNM                  ;;COUNT TEST NUMBERS
014516  116767  164360  164454          MOVB    $TSTNM,$TESTN           ;;SET TEST NUMBER IN APT MAILBOX
```

```
 4913  014524  011667  164356                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 4914  014530  011667  164354                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 4915  014534  005067  164422                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 4916  014540  112767  000001  164347         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 4917  014546  016777  164330  164366 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 4918  014554  016716  164326                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 4919  014560  000002                         RTI                     ;;FIXES PS
 4920  014562  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

014564                          $TYPDS:
014564  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
014566  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
014570  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
014572  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
014574  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
014576  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
014602  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
014606  100004                         BPL     1$              ;;BR IF INPUT IS POS.
014610  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
014612  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
014620  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
014622  012703  015000                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
014626  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
014632  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
014634  016001  014770                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
014640  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
014642  002402                         BLT     4$              ;;BR IF DONE
014644  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
014646  000774                         BR      3$
014650  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
014652  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
014654  001002                         BNE     5$              ;;FALL THROUGH IF 0
014656  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
014660  100407                         BMI     7$              ;;BR IF YES
014662  106316                 5$:     ASLB    (SP)            ;;MSD?
014664  103003                         BCC     6$              ;;BR IF NO
014666  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
014674  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
014700  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
014704  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
014706  005720                         TST     (R0)+           ;;JUST INCREMENTING
014710  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
014714  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
014716  003002                         BGT     8$              ;;GO TO EXIT
014720  010502                         MOV     R5,R2           ;;GET THE LSD
014722  000764                         BR      6$              ;;GO CHANGE TO ASCII
014724  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
014726  100003                         BPL     9$              ;;BR IF NO
014730  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
014736  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
014740  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
014742  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
014744  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 4977  014746  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 4978  014750  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 4979  014752  104401  015000                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4980  014756  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4981  014764  012616                        MOV     (SP)+,(SP)
 4982  014766  000002                        RTI                     ;;RETURN TO USER
 4983  014770  023420                $DTBL:  10000.
 4984  014772  001750                        1000.
 4985  014774  000144                        100.
 4986  014776  000012                        10.
 4987  015000  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

015010  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
015014  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
015022  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
015026  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
015032  000406                          BR      $TYPON
015034  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
015042  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
015050  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
015056  010346                          MOV     R3,-(SP)        ;;SAVE R3
015060  010446                          MOV     R4,-(SP)        ;;SAVE R4
015062  010546                          MOV     R5,-(SP)        ;;SAVE R5
015064  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
015070  005404                          NEG     R4
015072  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
015076  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
015102  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
015106  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
015112  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
015114  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
015116  000404                          BR      3$              ;;GO DO MSB
015120  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
015122  006105                          ROL     R5
015124  006105                          ROL     R5
015126  010503                          MOV     R5,R3
015130  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
015132  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
015136  100016                          BPL     7$              ;;BR IF NO
015140  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
015144  001002                          BNE     4$              ;;TEST FOR 0
015146  005704                          TST     R4              ;;SUPPRESS THIS 0?
015150  001403                          BEQ     5$              ;;BR IF YES
```

```
5044  015152  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5045  015154  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5046  015160  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5047  015164  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
5048  015170  104401  015230                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
5049  015174  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
5050  015200  003347                  BGT     2$                      ;;BR IF MORE TO DO
5051  015202  002402                  BLT     6$                      ;;BR IF DONE
5052  015204  005204                  INC     R4                      ;;INSURE LAST DIGIT ISN'T A BLANK
5053  015206  000744                  BR      2$                      ;;GO DO THE LAST DIGIT
5054  015210  012605          6$:     MOV     (SP)+,R5                ;;RESTORE R5
5055  015212  012604                  MOV     (SP)+,R4                ;;RESTORE R4
5056  015214  012603                  MOV     (SP)+,R3                ;;RESTORE R3
5057  015216  016666  000002  000004  MOV     2(SP),4(SP)             ;;SET THE STACK FOR RETURNING
5058  015224  012616                  MOV     (SP)+,(SP)
5059  015226  000002                  RTI                             ;;RETURN
5060  015230     000          8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
5061  015231     000                  .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
5062  015232     000          $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
5063  015233     000          $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
5064  015234  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.

015236  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
015240  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
015244  005740                          TST     -(R0            ;;BACKUP BY 2
015246  111000                          MOVB    (R0), 0         ;;GET RIGHT BYTE OF TRAP
015250  006300                          ASL     R0              ;;POSITION FOR INDEXING
015252  016000  015272                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
015256  000200                          RTS     R0              ;;GO TO ROUTINE

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

015260  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
015262  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
015270  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
015272  015260                  $TRPAD: .WORD   $TRAP2
015274  012600                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
015276  015034                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
015300  015010                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
015302  015050                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
015304  014564                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

015306  013132                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

015310  013062                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
015312  013344                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
015314  013464                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
        000001                  .END
```

```
ABASE = 175610            1#   1001    1042
ACDW1 = 000000          1001
ACDW2 = 000000          1001
ACPUOP= 000000          1001    1016
ADDRES  012266          4407*   4418*   4428    4435*
ADDW0 = 000000          1001
ADDW1 = 000000          1001
ADDW10= 000000          1001
ADDW11= 000000          1001
ADDW12= 000000          1001
ADDW13= 000000          1001
ADDW14= 000000          1001
ADDW15= 000000          100.
ADDW2 = 000000          1001
ADDW3 = 000000          1001
ADDW4 = 000000          1001
ADDW5 = 000000          1001
ADDW6 = 000000          1001
ADDW7 = 000000          1001
ADDW8 = 000000          1001
ADDW9 = 000000          1001
ADEVCT= 000000          1001    1007
ADEVM = 000001            1#    1001    1043
AENV  = 000000          1001    1012
AENVM = 000000          1001    1013
AFATAL= 000000          1001    1004
AMADR1= 000000          1001    1029
AMADR2= 000000          1001    1033
AMADR3= 000000          1001    1036
AMADR4= 000000          1001    1039
AMAMS1= 000000          1001    1023
AMAMS2= 000000          1001    1031
AMAMS3= 000000          1001    1034
AMAMS4= 000000          1001    1037
AMSGAD= 000000          1001    1009
AMSGLG= 000000          1001    1010
AMSGTY= 000000          1001    1003
AMTYP1= 000000          1001    1024
AMTYP2= 000000          1001    1032
AMTYP3= 000000          1001    1035
AMTYP4= 000000          1001    1038
APASS = 000000          1001    1006
APRIOR= 000000          1001
APTCSU= 000040          4558    4802#
APTENV= 000001          2673    2882    3159    3822    4551    4758    4800#   4834
APTSIZ= 000200          1112    4799#
APTSPO= 000100          4553    4760    4801#
ASHREG= 000000          1001    1014
ATESTN= 000000          1001    1005
AUNIT = 000000          1001    1008
AUSWR = 071110            1#    1001    1015
AVECT1= 000300            1#    1001    1040
AVECT2= 000000          1001    1041
BADBRK  011236          4081#
BAUD  = 007400           905#
BAUDRA  007144          3296#
```

```
BITMAS  012262          4384    4403#   4416#   4424    4433#
BIT0   = 000001          780#
BIT00  = 000001          770#    780     849     870     891
BIT01  = 000002          769#    779     828     848     890
BIT02  = 000004          768#    778     827     847     868     889
BIT03  = 000010          767#    777     826     846     888
BIT04  = 000020          766#    776     845     887
BIT05  = 000040          765#    775     824     844     886
BIT06  = 000100          764#    774     823     843     864     885
BIT07  = 000200          763#    773     822     842     863     884
BIT08  = 000400          762#    772     4887
BIT09  = 001000          761#    771     4845    4895
BIT1   = 000002          779#
BIT10  = 002000          760#    819     4822
BIT11  = 004000          759#    818     858     4902
BIT12  = 010000          758#    817     837     857
BIT13  = 020000          757#    816     836     856     4829
BIT14  = 040000          756#    815     835     855     4873
BIT15  = 100000          755#    814     834     854
BIT2   = 000004          778#
BIT3   = 000010          777#
BIT4   = 000020          776#
BIT5   = 000040          775#
BIT6   = 000100          774#
BIT7   = 000200          773#
BIT8   = 000400          772#
BIT9   = 001000          771#
BPTVEC= 000014           787#
BREAK  = 000001          870#    1248    1261    1264    1278    1281    1295    1300    4049
BRK    = 010000          906#
CABLE  = 020000          907#    1972    2026    2117    2206    2295    2386    2486    2590    3535    3717
CARDET= 010000           817#    2051    2071    2091
CKSWR  = 104407          4818    4844    4872    5104#
CLR    = 000000           808#    2892    4158
CLRSEN= 020000           816#    2140    2160    2180
COMSPD= 000100           900#
CR     = 000015          695#    4597    4607
CRLF   = 000200          696#    1136    4096    4339    4568    4607
CYCLE   012134          1141    4372#
DATA   = 000017          897#    4242    4243
DATAIE= 000040           824#    1715    1728    1731    1745    1748    1762    1767    3548    3559    3592
DATAIN= 100000           814#    1942    2419    2437    2446    2463    2520    2544    2567    2627    2649
DATLNG  011562          3631    3737    3832    4224#
DDISP  = 177570          702#     982    1100
DISPLA  001142           982#    1100*   1108*   4821*   4917*
DISPRE  000174           917#    1108
DLADD   001254          1062#    1145*   1150    1153    1155    1158    1161    1164    1180    1212    4097    4359
DLADDR= 175610           804#    1062    1064    1065    1066    1067    1068
DLVEC   001256          1063#    1148*   3326    3447    3598    3855    4103    4365
DSWR   = 177570          701#     981    1099
DTR    = 000002          828#    1494    1497    1511    1513    1526    1529    2048    2068    2088    2137    2157    2177
                        2407    2435    2461
EMTVEC= 000030           790#    1083*   1084*
ERRCNT  010756          3880*   3902    3908    3924#   3991    4002*
ERROR  = 100000           834#    3060    3124
ERRVEC= 000004           783#    1097    1098*   1109*   4878    4879*   4881*   4884*
```

```
EVENOD= 000040            899#
FALSE = 000000           4139#   4146
FLAG    011554           4146#   4173*   4178    4201    4213#
FRERR = 020000            836#
FRFD  = 040000            908#   1483    1550    2026    2117    2206    2386    2486
GNS   = ****** U          916    1135    4095    4101    4107    4338    4344    4351    4357    4363    5096    5097    5098
                         5099    5100    5102    5104    5105    5106
GTSWR = 104406           1130    5102#
HOLD    011022           3943*   3944*   3947    3960#
HOLDSC  011555           4158*   4163*   4169    4214#
HOWLON= 000000           4118#   4144
HT    = 000011            693#   4566    4607
I       001272           1069#   1189*   1210*   1213    1215    4247*   4250*   4252
ILLMEM= 000004            795#   1184    1220
INITFL  012264           1138#   4387    4390*   4434#
INTFLA  011750           1196*   1201    3322*   3362    3366    3388*   3404    3455*   3497    3501    3545*   3571    3575
                         4328*   4331#
INTSRV  011742           1185    3334    3448    4318#
IOTVEC= 000020            788#   1081*   1082*
LF    = 000012            694#   4601    4607
LOOP    001720           1139#   4112
MAINT = 000004            868#   1324    1337    1340    1354    1357    1371    1376    2765    2824    2890    3172    3453
                         3519    3620    3704    3727    3886    4040
MASK  = 000000           4225#   4239*   4255*   4257*   4262*
MAX     006356           2918    2922    2997#
MYTYPE  011752           4334#   4831
NEW     007176           3195*   3197*   3213*   3215*   3224    3232    3234    3252    3254    3302#
NUMBER  010760           3900    3925#   3950    4008
NUMBR   011656           4241*   4242    4244*   4252    4265#
OLD     007202           3168*   3170*   3224    3232    3234    3252*   3254*   3303#
OERR  = 040000            835#   3042    3081    3109
PARITY= 000020            898#
PBAUDS= 004000            858#
PBAUD0= 010000            857#
PBAUD1= 020000            856#
PBAUD2= 040000            855#
PBAUD3= 100000            854#
PBR   = 000200            901#   3150
PERR  = 010000            837#
PIRQ  = 177772            700#
PIRQVE= 000240            794#
PR0   = 000000            717#   3344    3393    3464    3551    3624    3730    3849
PR1   = 000040            718#
PR2   = 000100            719#
PR3   = 000140            720#
PR4   = 000200            721#
PR5   = 000240            722#
PR6   = 000300            723#
PR7   = 000340            724#   1186    1396    2397    2497    2600    3335    3449    3860    3866
PS    = 177776            697#    698
PSW   = 177776            698#
PWRVEC= 000024            789#   1087*   1088*   4487*   4488*   4497*   4503*   4515*   4516*
RATES   007124           3186    3265    3270#
RBUF    001262           1065#   1155*   1156*   2859    2929    2984    3038    3081    3109    3124    3183    3493    3635
                         3673    3740    3777    3978    4044    4070
RCSR    001260           1064#   1153*   1494*   1497    1511*   1513    1526*   1529    1563    1576*   1579    1593*   1596
```

```
                    1610*  1615   1639   1652*  1655   1669*  1672   1686*  1691   1715   1728*  1731   1745*
                    1748   1762*  1767   1791   1804*  1807   1821*  1824   1838*  1843   1877   1942   1987
                    2048*  2051   2068*  2071   2088*  2091   2137*  2140   2157*  2160   2177*  2180   2227*
                    2230   2247*  2250   2267*  2270   2315*  2318   2335*  2338   2355*  2358   2407*  2416
                    2419   2435*  2437   2446   2461*  2463   2508*  2517   2520   2536*  2544   2559*  2567
                    2609*  2611   2619*  2627   2641*  2649   2780   2805   2839   2862   2904   2943   2947
                    2950   2970   2987   3203   3461*  3478   3485*  3522*  3548*  3556*  3559*  3561*  3592*
                    3594*  3666   3769   3893*  3919*  4012*  4056
RCVRAC= 004000       818*  1987   2904   2943   2950
RCVRDO= 000200       822*  1877   2781   2805   2840   2862   2947   2970   2987   3203   3479   3667   3770
                    4057
RCVRIE= 000100       823*  1791   1804   1807   1821   1824   1838   1843   3461   3485   3522   3893   4012
RDATA0= 000001       849*
RDATA1= 000002       848*
RDATA2= 000004       847*
RDATA3= 000010       846*
RDATA4= 000020       845*
RDATA5= 000040       844*
RDATA6= 000100       843*
RDATA7= 000200       842*
RDCHR = 104410      4720   5105*
RDLIN = 104411      5106*
REC     011026      3858   3970*
REG   = 000004      4118*  4142
REGSAV  011550      4142*  4155   4211*
REQSEN= 000004       827*  1563   1576   1579   1593   1596   1610   1615   2227   2247   2267   2508   2536
                    2559   3556   3561   3594
RESVEC= 000010       784*
RHLD    011116      3982*  3983*  3987   3995   4020*
RING  = 040000       815*  2230   2250   2270
R0050   007124      3279*
R0070   007125      3280*
R0110   007126      3281*
R0135   007127      3282*
R0150   007130      3283*
R0200   007133      3286*
R0300   007131      3284*
R0600   007132      3285*
R10000  007143      3294*
R1800   007134      3287*
R2000   007135      3288*
R2400   007136      3289*
R3600   007137      3290*
R4800   007140      3291*
RSSTAC  001334      1071*  1167
R7200   007141      3292*
R9600   007142      3293*
S8      010762      3926*  3995*
SECREC= 002000       819*  2318   2338   2358
SECXMI= 000010       826*  1639   1652   1655   1669   1672   1686   1691   2315   2335   2355   2609   2619
                    2641
SET   = 177777       807*  2695   2738   2779   2838   2907   2916   3477   3652   3665   3756   3768   4055
                    4163
SETCLR= 000006      4118*  4169
SPECIA  012164      4397*  4478
STACK = 001100       688*  1079
```

```
START   001336          920    1072#   4520
STKLMT= 177774          699#
SWR     001140          981#   1077    1099*   1101    1107*   1114*   1128    4495    4508*   4617    4654*   4822    4829
                       4841    4845    4873    4887    4889    4895    4902
SWREG   000176          918#   1107    1128    4617    4630
SW0    = 000001         752#
SW00   = 000001         742#    752
SW01   = 000002         741#    751
SW02   = 000004         740#    750
SW03   = 000010         739#    749
SW04   = 000020         738#    748
SW05   = 000040         737#    747
SW06   = 000100         736#    746
SW07   = 000200         735#    745
SW08   = 000400         734#    744
SW09   = 001000         733#    743
SW1    = 000002         751#
SW10   = 002000         732#
SW11   = 004000         731#
SW12   = 010000         730#
SW13   = 020000         729#
SW14   = 040000         728#
SW15   = 100000         727#
SW2    = 000004         750#
SW3    = 000010         749#
SW4    = 000020         748#
SW5    = 000040         747#
SW6    = 000100         746#
SW7    = 000200         745#
SW8    = 000400         744#
SW9    = 001000         743#
TBITVE= 000014          785#
TBUF    001270         1068#   1164*   1165*   2689*   2720*   2772*   2832*   2900*   3018*   3028*   3100*   3192*   3471*
                       3661*   3765*   3947*   4052*
TCSR    001264         1066#   1158*   1159*   1248    1261*   1264    1278*   1281    1295*   1300    1324    1337*   1340
                       1354*   1357    1371*   1376    1406    1419*   1422    1436*   1439    1453*   1458    1910    2696
                       2725    2739    2765*   2824*   2890*   3172*   3341*   3351*   3391*   3458*   3519*   3620*   3653
                       3704*   3727*   3757    3886*   3890*   3917*   3954*   4040*   4049*
TCSRHI  001266         1067#   1161*   1162*   3186*   3265*
TDATA0= 000001          891#
TDATA1= 000002          890#
TDATA2= 000004          889#
TDATA3= 000010          888#
TDATA4= 000020          887#
TDATA5= 000040          886#
TDATA6= 000100          885#
TDATA7= 000200          884#
TIME   = 000000        4274#   4285
TIMER   011404         2699    2742    2783    2842    3481    3656    3669    3760    3772    4059    4117#
TIMSAV  011552         4144*   4180    4191*   4212#
TKVEC  = 000060          792#
TPVEC  = 000064          793#
TRAN    010764         3864    3935#
TRAPVE= 000034           791#   1085*   1086*
TRTVEC= 000014           786#
TRUE   = 000001        4138#   4173    4178    4201
```

```
TST1     002034          1176#
TST10    003346          1707#
TST11    003510          1783#
TST12    003652          1866#
TST13    003712          1899#
TST14    003752          1932#
TST15    004012          1966#
TST16    004072          1977    2021#
TST17    004232          2033    2112#
TST2     002202          1239#
TST20    004372          2122    2201#
TST21    004532          2212    2290#
TST22    004672          2300    2381#
TST23    005076          2392    2481#
TST24    005322          2492    2576    2585#
TST25    005510          2595    2667#
TST26    005676          2677    2759#
TST27    006012          2818#
TST3     002344          1315#
TST30    006122          2876#
TST31    006360          2886    2932    2959    2996    3006#
TST32    006624          3051    3071    3089    3119    3134    3146#
TST33    007206          3154    3163    3268    3317#
TST34    007460          3440#
TST35    010102          3540    3615#
TST36    010316          3695    3713#
TST37    010544          3722    3795    3815#
TST4     002506          1392#
TST40    011122          3826    3922    4035#
TST41    011264          4080    4091#
TST5     002662          1479#
TST6     003022          1487    1546#
TST7     003204          1554    1631#
TYPDS  = 104405          4110    4467    5100#
TYPE   = 104401          1122    4093    4099    4105    4336    4342    4349    4355    4361    4465    4468    4517    4571
                         4628    4629    4632    4645    4656    4675    4724    4727    4731    4824    4832    4979    5048
                         5096#
TYPOC  = 104402          4098    4104    4341    4348    4354    4360    4366    4631    5097#
TYPON  = 104404          5099#
TYPOS  = 104403          5098#
VECTOR   012270          4409#   4420#   4436#
WAIT     011662          2412    2513    2541    2564    2624    2646    3023    3033    3105    3357    3401    3490    3566
                         4188    4273#
WAS      010763          3927#   3997#
WHICHB=  000002          4118#   4155
XMITIE=  000100           864#   1406    1419    1422    1436    1439    1453    1458    3341    3351    3391    3890    3917
                         3919    3954
XMITRD=  000200           863#   1910    2697    2725    2740    3654    3758
ZZZ      011024          3958    3962#
ZZZZ     011120          4018    4021#
$APTHD   001000           941     947#
$ASTAT=  ****** U        4780    4795
$ATYC    013662          4751    4753#
$ATY1    013636          4749#
$ATY3    013644          4556    4750#
$ATY4    013654          4752#   4837
```

```
$AUTOB  001134            978#   1132#  4625   4746
$BASE   001250           1042#   4407
$BOADR  001122            973#
$BOOAT  001126            975#
$BELL   001164            993#   4824   4857
$BGNLE= 177777              1#
$CHARC  0:3056           4573#   4583#  4590   4599#  4604#
$CKSUM  013062           4617#   5104
$CMTAG  001100            961#   1074   1075   1083   1089   1090   1091
$CM3    = 000000          991#
$CNTLG  013607           4628    4741#
$CNTLU  013602           4645    4740#
$CPUOP  001222           1016#
$CRLF   001171            995#   4572   4607   4656   4740   4832   4857
$DBLK   015000           4945    4979   4987#
$DEVCT  001204           1007#   4405#  4430#
$DEVM   001252           1043#   4424
$DOAGN  012372           4461    4470   4476#
$DTBL   014770           4948    4983#
$ENDAD  012362            927    1120   4472#  4852
$ENDCT  012330           1089    4463#
$ENDMG  012401           4465    4480#
$ENULL  012376           4468    4479#
$ENV    001214           1012#   1126   2673   2882   3159   3822   4551   4758   4782   4834
$ENVM   001215           1013#   1112   4553   4558   4760
$EOP    012274           4395    4453#
$EOPCT  012322           1089#   4460#  4464
$ERFLG  001103            964#   4819#  4857   4862   4891   4893   4899#  4921
$ERMAX  001115            970#   1092#  4893   4916#  4921
$ERRFL= 000000           1138#   1139   1141#  1145#  1146   1148#  1149   1150#  1151   1153#  1154   1155#  1157
                         1158#   1160   1161#  1163   1164#  1166   1167#  1168   1180#  1181   1183#  1184#  1185#
                         1186#   1187#  1188   1189#  1190   1193#  1194   1196#  1197   1210#  1211   1212#  1214
                         1220#   1221#  1223#  1224#  1225   1245#  1246   1259#  1260   1261#  1262   1275#  1276
                         1278#   1279   1292#  1293   1295#  1296   1321#  1322   1335#  1336   1337#  1338   1351#
                         1352    1354#  1355   1368#  1369   1371#  1372   1403#  1404   1417#  1418   1419#  1420
                         1433#   1434   1436#  1437   1450#  1451   1453#  1454   1486#  1487   1492#  1493   1494#
                         1495    1508#  1509   1511#  1512   1524#  1525   1526#  1527   1553#  1554   1560#  1561
                         1574#   1575   1576#  1577   1590#  1591   1593#  1594   1607#  1608   1610#  1611   1636#
                         1637    1650#  1651   1652#  1653   1666#  1667   1669#  1670   1682#  1683   1686#  1687
                         1712#   1713   1726#  1727   1728#  1729   1742#  1743   1745#  1746   1759#  1760   1762#
                         1763    1788#  1789   1802#  1803   1804#  1805   1818#  1819   1821#  1822   1835#  1836
                         1838#   1839   1875#  1876   1907#  1908   1940#  1941   1976#  1977   1984#  1985   2032#
                         2033    2044#  2045   2048#  2049   2064#  2065   2068#  2069   2084#  2085   2088#  2089
                         2121#   2122   2133#  2134   2137#  2138   2153#  2154   2157#  2158   2173#  2174   2177#
                         2178    2211#  2212   2223#  2224   2227#  2228   2243#  2244   2247#  2248   2263#  2264
                         2267#   2268   2299#  2300   2311#  2312   2315#  2316   2331#  2332   2335#  2336   2351#
                         2352    2355#  2356   2391#  2392   2404#  2405   2407#  2408   2410#  2416#  2417   2431#
                         2432    2435#  2436   2457#  2458   2461#  2462   2491#  2492   2504#  2505   2508#  2509
                         2511#   2517#  2518   2531#  2532   2536#  2537   2539#  2555#  2556   2559#  2560   2562#
                         2594#   2595   2609#  2610   2611#  2612   2615#  2616   2619#  2620   2622#  2638#  2639
                         2641#   2642   2644#  2676#  2677   2681#  2682   2689#  2690   2694#  2713#  2714   2720#
                         2721    2737#  2765#  2766   2768#  2769   2772#  2773   2778#  2798#  2799   2824#  2825
                         2826#   2827   2832#  2833   2837#  2859#  2860   2885#  2886   2890#  2891   2892#  2893
                         2894#   2895   2900#  2901   2907#  2908   2912#  2913   2929#  2930   2931#  2932   2958#
                         2959    2984#  2985   3011#  3012   3018#  3019   3021#  3028#  3029   3031#  3038#  3039
                         3050#   3051   3058#  3059   3070#  3071   3077#  3078   3095#  3096   3100#  3101   3103#
```

```
                    3118#   3119    3153#   3154    3162#   3163    3168#   3169    3170#   3171    3172#   3173    3175#
                    3176    3178    3182    3183#   3184    3186#   3187    3189#   3190    3192#   3193    3195#   3196
                    3197#   3198    3207#   3208    3213#   3214    3215#   3216    3252#   3253    3254#   3255    3260#
                    3264    3265#   3266    3322#   3323    3326#   3327    3329#   3330    3332#   3333#   3334#   3335#
                    3336#   3337    3338#   3339    3341#   3342    3351#   3352    3355#   3385#   3386    3388#   3389
                    3391#   3392    3399#   3418#   3419#   3421#   3422#   3423    3446#   3447#   3448#   3449#   3450#
                    3451    3453#   3454    3455#   3456    3458#   3459    3461#   3462    3471#   3472    3476#   3485#
                    3486    3488#   3493#   3494    3519#   3520    3522#   3523    3533#   3534    3539#   3540    3545#
                    3546    3548#   3549    3556#   3557    3559#   3560    3561#   3562    3564#   3592#   3593    3594#
                    3595    3598#   3599    3602#   3603#   3605#   3606#   3607    3620#   3621    3630#   3635#   3636
                    3639#   3640    3642    3646    3651#   3661#   3662    3664#   3673#   3674    3681#   3683    3684#
                    3685    3694#   3695    3704#   3705    3721#   3722    3727#   3728    3736#   3740#   3741    3743#
                    3744    3746    3750    3755#   3765#   3766    3767#   3777#   3778    3781#   3783    3784#   3785
                    3794#   3795    3825#   3826    3831#   3855#   3856    3858#   3859    3860#   3861    3864#   3865
                    3866#   3867    3870#   3871    3873#   3874    3876#   3877    3880#   3881    3886#   3887    3890#
                    3891    3893#   3894    3917#   3918    3919#   3920    3940#   3941    3943#   3945    3947#   3948
                    3954#   3955    3975#   3976    3978#   3980    3982#   3984    3995#   3996    3997#   3998    4002#
                    4003    4012#   4013    4040#   4041    4044#   4045    4049#   4050    4052#   4053    4054#   4142#
                    4143    4144#   4145    4146#   4147    4158#   4159    4163#   4164    4173#   4174    4186#   4191#
                    4192    4204#   4205    4208#   4210    4239#   4240    4241#   4245    4247#   4248    4250    4254
                    4255#   4256    4257#   4258    4262#   4263    4285#   4286    4287#   4288    4290    4294    4295#
                    4296    4298    4302    4328#   4329    4390#   4391    4395#   4399#   4400    4403#   4404    4405#
                    4406    4407#   4408    4409#   4410    4414#   4415    4416#   4417    4418#   4419    4420#   4421
                    4428#   4429    4430#   4431
$ERROR  014104      1083    4817#
$ERRPC  001116       971#   4353    4826*   4827*   4828    4857
$ERRTB  001254      1060#
$ERTTL  001112       968#   4109    4111*   4825*   4857
$ESCAP  001162       992#   1091*   4848    4850    4857    4915*
$ETABL  001214      1011#
$ETEND  001254       953    1044#
$FATAL  001176      1004#   4346*   4347    4786*
$FFLG   014102      4749*   4752*   4780    4789*   4797#
$FILLC  001156       989#   4576    4607
$FILLS  001155       988#   4607
$FSAND= 000310         1#   1203    1250    1266    1283    1302    1326    1342    1359    1378    1408    1424    1441
                    1460    1485    1499    1515    1531    1552    1565    1581    1598    1617    1641    1657    1674
                    1693    1717    1733    1750    1769    1793    1809    1826    1845    1879    1912    1944    1974
                    1989    2028    2053    2073    2093    2119    2142    2162    2182    2208    2232    2252    2272
                    2297    2320    2340    2360    2388    2421    2439    2448    2465    2488    2522    2546    2569
                    2592    2629    2651    2675    2727    2807    2864    2884    2906    2924    2945    2949    2952
                    2972    2989    3044    3062    3083    3111    3126    3152    3161    3202    3205    3226    3234
                    3364    3368    3406    3499    3503    3537    3573    3577    3690    3719    3790    3824    3910
                    3952    3989    3993    4010    4072    4157    4171    4203    4386    4389
$FSBAD= 000401         1#   1203    1250    1266    1283    1302    1326    1342    1359    1378    1408    1424    1441
                    1460    1485    1499    1515    1531    1552    1565    1581    1598    1617    1641    1657    1674
                    1693    1717    1733    1750    1769    1793    1809    1826    1845    1879    1912    1944    1974
                    1989    2028    2053    2073    2093    2119    2142    2162    2182    2208    2232    2252    2272
                    2297    2320    2340    2360    2388    2421    2439    2448    2465    2488    2522    2546    2569
                    2592    2629    2651    2675    2727    2807    2864    2884    2906    2924    2945    2949    2952
                    2972    2989    3044    3062    3083    3111    3126    3152    3161    3202    3205    3226    3236
                    3364    3368    3406    3499    3503    3537    3573    3577    3690    3719    3790    3824    3910
                    3952    3989    3993    4010    4072    4157    4171    4203    4386    4389
$FSBLA= 000170         1#
$FSCAS= 000150         1#
$FSDEC= 000220         1#
```

```
$FSGOO= 000400      1#  1201  1248  1264  1281  1300  1324  1340  1357  1376  1406  1422  1439
                    1458  1483  1497  1513  1529  1550  1563  1579  1596  1615  1639  1655  1672
                    1691  1715  1731  1748  1767  1791  1807  1824  1843  1877  1910  1942  1972
                    1987  2026  2051  2071  2091  2117  2140  2160  2180  2206  2230  2250  2270
                    2295  2318  2338  2358  2386  2419  2437  2446  2463  2486  2520  2544  2567
                    2590  2627  2649  2673  2704  2725  2745  2788  2805  2846  2862  2882  2904
                    2922  2942  2943  2947  2950  2970  2987  3042  3060  3081  3109  3124  3150
                    3159  3199  3200  3203  3224  3232  3362  3366  3404  3497  3501  3535  3571
                    3575  3688  3717  3788  3822  3908  3950  3987  3991  4008  4062  4070  4155
                    4169  4201  4384  4387
$FSIF = 000110      1#  1201  1207  1248  1254  1264  1270  1281  1287  1300  1306  1324  1330
                    1340  1346  1357  1363  1376  1382  1406  1412  1422  1428  1439  1445  1458
                    1464  1483  1489  1497  1503  1513  1519  1529  1535  1550  1556  1563  1569
                    1579  1585  1596  1602  1615  1621  1639  1645  1655  1661  1672  1678  1691
                    1697  1715  1721  1731  1737  1748  1754  1767  1773  1791  1797  1807  1813
                    1824  1830  1843  1849  1877  1888  1910  1921  1942  1955  1972  1979  1987
                    2002  2026  2035  2051  2059  2071  2079  2091  2099  2117  2124  2140  2148
                    2160  2168  2180  2188  2206  2214  2230  2238  2250  2258  2270  2278  2295
                    2302  2318  2326  2338  2346  2358  2366  2386  2394  2419  2425  2437  2443
                    2446  2452  2463  2469  2486  2494  2520  2526  2544  2550  2567  2573  2590
                    2597  2627  2633  2649  2655  2673  2679  2704  2709  2725  2731  2745  2750
                    2788  2793  2805  2811  2846  2851  2862  2868  2882  2888  2904  2909  2914
                    2922  2934  2947  2950  2961  2963  2970  2977  2987  2993  3042  3053  3060
                    3073  3081  3091  3109  3121  3124  3131  3150  3156  3159  3165  3203  3209
                    3217  3224  3228  3232  3234  3238  3247  3249  3362  3366  3372  3379  3381
                    3404  3410  3497  3501  3508  3514  3516  3535  3542  3571  3575  3582  3588
                    3590  3688  3697  3717  3724  3788  3797  3822  3828  3908  3914  3950  3956
                    3987  3991  3999  4004  4008  4015  4062  4067  4070  4076  4155  4160  4165
                    4169  4175  4201  4206  4384  4387  4392  4401  4411  4422
$FSINC= 000210      1#  3175  3257  3639  3699  3743  3800  4247  4259  4287  4295  4303  4306
$FSLOO= 000200      1#  4152  4178  4193
$FSNAM= 000160      1#
$FSNO = 000403      1#  1201  1248  1264  1281  1300  1324  1340  1357  1376  1406  1422  1439
                    1458  1483  1497  1513  1529  1550  1563  1579  1596  1615  1639  1655  1672
                    1691  1715  1731  1748  1767  1791  1807  1824  1843  1877  1910  1942  1972
                    1987  2026  2051  2071  2091  2117  2140  2160  2180  2206  2230  2250  2270
                    2295  2318  2338  2358  2386  2419  2437  2446  2463  2486  2520  2544  2567
                    2590  2627  2649  2673  2725  2805  2862  2882  2904  2922  2943  2947  2950
                    2970  2987  3042  3060  3081  3109  3124  3150  3159  3200  3203  3224  3232
                    3234  3362  3366  3404  3497  3501  3535  3571  3575  3688  3717  3788  3822
                    3908  3950  3987  3991  4008  4155  4384  4387
$FSOR = 000320      1#  1203  1250  1266  1283  1302  1326  1342  1359  1378  1408  1424  1441
                    1460  1485  1499  1515  1531  1552  1565  1581  1598  1617  1641  1657  1674
                    1693  1717  1733  1750  1769  1793  1809  1826  1845  1879  1912  1944  1974
                    1989  2028  2053  2073  2093  2119  2142  2162  2182  2208  2232  2252  2272
                    2297  2320  2340  2360  2388  2421  2439  2448  2465  2488  2522  2546  2569
                    2592  2629  2651  2675  2727  2807  2864  2884  2906  2924  2945  2949  2952
                    2972  2989  3044  3062  3083  3111  3126  3152  3161  3202  3205  3226  3234
                    3236  3364  3368  3406  3499  3503  3537  3573  3577  3690  3719  3790  3824
                    3910  3952  3989  3993  4010  4072  4157  4171  4203  4386  4389
$FSRTN= 000300      1#  4118  4217  4225  4267  4274  4312  4335  4368  4373  4439
$FSSEL= 000140      1#
$FSUNT= 000130      1#  1191  1215  2902  2916  3898  3900  4382  4424
$FSWHI= 000120      1#  2942  2943  2965  3199  3200  3220  3234
$FSYES= 000402      1#  1201  1248  1264  1281  1300  1324  1340  1357  1376  1406  1422  1439
                    1458  1483  1497  1513  1529  1550  1563  1579  1596  1615  1639  1655  1672
```

```
                    1691    1715    1731    1748    1767    1791    1807    1824    1843    1877    1910    1942    1972
                    1987    2026    2051    2071    2091    2117    2140    2160    2180    2206    2230    2250    2270
                    2295    2318    2338    2358    2386    2419    2437    2463    2486    2520    2544    2567
                    2590    2627    2649    2673    2725    2805    2862    2882    2904    2922    2943    2947    2950
                    2970    2987    3042    3060    3081    3109    3124    3150    3159    3200    3203    3224    3232
                    3234    3362    3366    3404    3497    3501    3535    3571    3575    3688    3717    3788    3822
                    3908    3950    3987    3991    4008    4070    4155    4169    4201    4384    4387
$GDADR  001120       972#
$GDDAT  001124       974#
$GET42  012352      4469#
$GTSWR  013132      4629#   5102
$HD     000000       673
$HIBTS  001000       948#
$ICNT   001104       965#   4906*   4907    4909*   4920
$IFLEV= 177777         1#   1201#   1207#   1248#   1254#   1264#   1270#   1281#   1287#   1300#   1306#   1324#   1330#
                    1340#   1346#   1357#   1363#   1376#   1382#   1406#   1412#   1422#   1428#   1439#   1445#   1458#
                    1464#   1483#   1489#   1497#   1503#   1513#   1519#   1529#   1535#   1550#   1556#   1563#   1569#
                    1579#   1585#   1596#   1602#   1615#   1621#   1639#   1645#   1655#   1661#   1672#   1678#   1691#
                    1697#   1715#   1721#   1731#   1737#   1748#   1754#   1767#   1773#   1791#   1797#   1807#   1813#
                    1824#   1830#   1843#   1849#   1877#   1888#   1910#   1921#   1942#   1955#   1972#   1979#   1987#
                    2002#   2026#   2035#   2051#   2059#   2071#   2079#   2091#   2099#   2117#   2124#   2140#   2148#
                    2160#   2168#   2180#   2188#   2206#   2214#   2230#   2238#   2250#   2258#   2270#   2278#   2295#
                    2302#   2318#   2326#   2338#   2346#   2358#   2366#   2386#   2394#   2419#   2425#   2437#   2443#
                    2446#   2452#   2463#   2469#   2486#   2494#   2520#   2526#   2544#   2550#   2567#   2573#   2590#
                    2597#   2627#   2633#   2649#   2655#   2673#   2679#   2704#   2709#   2725#   2731#   2745#   2750#
                    2788#   2793#   2805#   2811#   2846#   2851#   2862#   2868#   2882#   2888#   2904#   2914#   2922#
                    2934#   2947#   2963#   2970#   2977#   2987#   2993#   3042#   3053#   3060#   3073#   3081#   3091#
                    3109#   3121#   3124#   3131#   3150#   3156#   3159#   3165#   3203#   3217#   3224#   3249#   3362#
                    3381#   3404#   3410#   3497#   3516#   3535#   3542#   3571#   3590#   3688#   3697#   3717#   3724#
                    3788#   3797#   3822#   3828#   3908#   3914#   3950#   3956#   3987#   4004#   4008#   4015#   4062#
                    4067#   4070#   4076#   4155#   4165#   4169#   4175#   4201#   4206#   4384#   4387#   4401#   4422#
$ILLUP  012562      4487    4503    4522#
$INTAG  001135       979#   4657    4746
$ISKO = 000001      1201#   1207    1248#   1254    1264#   1270    1281#   1287    1300#   1306    1324#   1330    1340#
                    1346    1357#   1363    1376#   1382    1406#   1412    1422#   1428    1439#   1445    1458#   1464
                    1483#   1489    1497#   1503    1513#   1519    1529#   1535    1550#   1556    1563#   1569    1579#
                    1585    1596#   1602    1615#   1621    1639#   1645    1655#   1661    1672#   1678    1691#   1697
                    1715#   1721    1731#   1737    1748#   1754    1767#   1773    1791#   1797    1807#   1813    1824#
                    1830    1843#   1849    1877#   1888    1910#   1921    1942#   1955    1972#   1979    1987#   2002
                    2026#   2035    2051#   2059    2071#   2079    2091#   2099    2117#   2124    2140#   2148    2160#
                    2168    2180#   2188    2206#   2214    2230#   2238    2250#   2258    2270#   2278    2295#   2302
                    2318#   2326    2338#   2346    2358#   2366    2386#   2394    2419#   2425    2437#   2443    2446#
                    2452    2463#   2469    2486#   2494    2520#   2526    2544#   2550    2567#   2573    2590#   2597
                    2627#   2633    2649#   2655    2673#   2679    2704#   2709    2725#   2731    2745#   2750    2788#
                    2793    2805#   2811    2846#   2851    2862#   2868    2882#   2888    2904#   2914    2922#   2934
                    2947#   2963    2970#   2977    2987#   2993    3042#   3053    3060#   3073    3081#   3091    3109#
                    3121    3124#   3131    3150#   3156    3159#   3165    3203#   3217    3224#   3249    3362#   3381
                    3404#   3410    3497#   3516    3535#   3542    3571#   3590    3688#   3697    3717#   3724    3788#
                    3797    3822#   3828    3908#   3914    3950#   3956    3987#   4004    4008#   4015    4062#   4067
                    4070#   4076    4155#   4165    4169#   4175    4201#   4206    4384#   4422
$ISK1 = 000001      2950#   2961    3232#   3247    3366#   3379    3501#   3514    3575#   3588    3991#   3999    4387#
                    4401
$ITEMB  001114       969#   4346    4828#   4836    4857
$LF     001172       996#   4607    4731    4740    4857
$LFLG   014101      4790#   4796#
```

$LOCTA= 177777

```
                  1#   1191  1192  1202  1203  1207  1208  1216  1217  1249  1250  1254  1255
  1265  1266  1270  1271  1282  1283  1287  1288  1301  1302  1306  1307  1325
  1326  1330  1331  1341  1342  1346  1347  1358  1359  1363  1364  1377  1378
  1382  1383  1407  1408  1412  1413  1423  1424  1428  1429  1440  1441  1445
  1446  1459  1460  1464  1465  1484  1485  1489  1490  1498  1499  1503  1504
  1514  1515  1519  1520  1530  1531  1535  1536  1551  1552  1556  1557  1564
  1565  1569  1570  1580  1581  1585  1586  1597  1598  1602  1603  1616  1617
  1621  1622  1640  1641  1645  1646  1656  1657  1661  1662  1673  1674  1678
  1679  1692  1693  1697  1698  1716  1717  1721  1722  1732  1733  1737  1738
  1749  1750  1754  1755  1768  1769  1773  1774  1792  1793  1797  1798  1808
  1809  1813  1814  1825  1826  1830  1831  1844  1845  1849  1850  1878  1879
  1888  1889  1911  1912  1921  1922  1943  1944  1955  1956  1973  1974  1979
  1980  1988  1989  2002  2003  2027  2028  2035  2036  2052  2053  2059  2060
  2072  2073  2079  2080  2092  2093  2099  2100  2118  2119  2124  2125  2141
  2142  2148  2149  2161  2162  2168  2169  2181  2182  2188  2189  2207  2208
  2214  2215  2231  2232  2238  2239  2251  2252  2258  2259  2271  2272  2278
  2279  2296  2297  2302  2303  2319  2320  2326  2327  2339  2340  2346  2347
  2359  2360  2366  2367  2387  2388  2394  2395  2420  2421  2425  2426  2438
  2439  2443  2444  2447  2448  2452  2453  2464  2465  2469  2470  2487  2488
  2494  2495  2521  2522  2526  2527  2545  2546  2550  2551  2568  2569  2573
  2574  2591  2592  2597  2598  2628  2629  2633  2634  2650  2651  2655  2656
  2674  2675  2679  2680  2704  2705  2709  2710  2726  2727  2731  2732  2745
  2746  2750  2751  2788  2789  2793  2794  2806  2807  2811  2812  2846  2847
  2851  2852  2863  2864  2868  2869  2883  2884  2888  2889  2902  2903  2905
  2906  2909  2910  2911  2914  2915  2917  2918  2919  2920  2921  2923  2924
  2934  2935  2942  2943  2944  2945  2948  2949  2951  2952  2961  2962  2963
  2964  2965  2966  2967  2971  2972  2977  2978  2988  2989  2993  2994  3043
  3044  3053  3054  3061  3062  3073  3074  3082  3083  3091  3092  3110  3111
  3121  3122  3125  3126  3131  3132  3151  3152  3156  3157  3160  3161  3165
  3166  3176  3177  3178  3179  3180  3181  3182  3199  3200  3201  3202  3204
  3205  3209  3210  3211  3217  3218  3220  3221  3222  3225  3226  3228  3229
  3230  3233  3234  3235  3236  3238  3239  3240  3247  3248  3249  3250  3257
  3258  3259  3363  3364  3367  3368  3372  3373  3374  3379  3380  3381  3382
  3405  3406  3410  3411  3498  3499  3502  3503  3508  3509  3510  3514  3515
  3516  3517  3536  3537  3542  3543  3572  3573  3576  3577  3582  3583  3584
  3588  3589  3590  3591  3640  3641  3642  3643  3644  3645  3646  3689  3690
  3697  3698  3699  3700  3701  3718  3719  3724  3725  3744  3745  3746  3747
  3748  3749  3750  3789  3790  3797  3798  3800  3801  3802  3823  3824  3828
  3929  3898  3899  3901  3902  3903  3904  3905  3909  3910  3914  3915  3951
  3952  3956  3957  3988  3989  3992  3993  3999  4000  4004  4005  4009  4010
  4015  4016  4062  4063  4067  4068  4071  4072  4076  4077  4117  4152  4153
  4156  4157  4160  4161  4162  4165  4166  4170  4171  4175  4176  4179  4180
  4181  4182  4193  4194  4195  4202  4203  4204  4205  4206  4207  4209  4210
  4217  4218  4219  4220  4224  4248  4249  4250  4251  4252  4253  4254  4259
  4260  4261  4264  4265  4267  4268  4269  4273  4288  4289  4290  4291  4292
  4293  4294  4296  4297  4298  4299  4300  4301  4302  4303  4304  4305  4306
  4307  4308  4312  4313  4314  4334  4368  4369  4370  4372  4382  4383  4385
  4386  4388  4389  4392  4393  4394  4401  4402  4411  4412  4413  4422  4423
  4425  4426  4432  4433  4439  4440  4441

$LPADR  001106   966#  1093#  4897#  4913#  4918  4920
$LPERR  001110   967#  1094#  1193#  1245#  1259#  1275#  1292#  1321#  1335#  1351#  1368#  1403#  1417#
                1433#  1450#  1492#  1508#  1524#  1560#  1574#  1590#  1607#  1636#  1650#  1666#  1682#
                1712#  1726#  1742#  1759#  1788#  1802#  1818#  1835#  1875#  1907#  1940#  1984#  2044#
                2064#  2084#  2133#  2153#  2173#  2223#  2243#  2263#  2311#  2331#  2351#  2404#  2431#
                2457#  2504#  2531#  2555#  2615#  2638#  2681#  2713#  2768#  2798#  2826#  3011#  3058#
                3077#  3095#  3338#  3385#  3453#  3533#  4847  4897  4914#  4920
```

$LSTCN= 177777

1# 1191 1192 1201 1203 1207 1215 1248 1250 1254 1264 1266 1270
1281 1283 1287 1300 1302 1306 1324 1326 1330 1340 1342 1346 1357
1359 1363 1376 1378 1382 1406 1408 1412 1422 1424 1428 1439 1441
1445 1458 1460 1464 1483 1485 1489 1497 1499 1503 1513 1515 1519
1529 1531 1535 1550 1552 1556 1563 1565 1569 1579 1581 1585 1596
1598 1602 1615 1617 1621 1639 1641 1645 1655 1657 1661 1672 1674
1678 1691 1693 1697 1715 1717 1721 1731 1733 1737 1748 1750 1754
1767 1769 1773 1791 1793 1797 1807 1809 1813 1824 1826 1830 1843
1845 1849 1877 1879 1888 1910 1912 1921 1942 1944 1955 1972 1974
1979 1987 1989 2002 2026 2028 2035 2051 2053 2059 2071 2073 2079
2091 2093 2099 2117 2119 2124 2140 2142 2148 2160 2162 2168 2180
2182 2188 2206 2208 2214 2230 2232 2238 2250 2252 2258 2270 2272
2278 2295 2297 2302 2318 2320 2326 2338 2340 2346 2358 2360 2366
2386 2388 2394 2419 2421 2425 2437 2439 2443 2446 2448 2452 2463
2465 2469 2486 2488 2494 2520 2522 2526 2544 2546 2550 2567 2569
2573 2590 2592 2597 2627 2629 2633 2649 2651 2655 2673 2675 2679
2704 2705 2709 2725 2727 2731 2745 2746 2750 2788 2789 2793 2805
2807 2811 2846 2847 2851 2862 2864 2868 2882 2884 2888 2902 2903
2904 2906 2910 2911 2914 2916 2921 2922 2924 2934 2942 2943 2945
2947 2949 2950 2952 2961 2963 2965 2966 2970 2972 2977 2987 2989
2993 3042 3044 3053 3060 3062 3073 3081 3083 3091 3109 3111 3121
3124 3126 3131 3150 3152 3156 3159 3161 3165 3175 3177 3178 3179
3182 3199 3200 3202 3203 3205 3210 3211 3217 3220 3221 3224 3226
3229 3230 3232 3236 3239 3240 3247 3249 3257 3258 3362 3364 3366
3368 3373 3374 3379 3381 3404 3406 3410 3497 3499 3501 3503 3509
3510 3514 3516 3535 3537 3542 3571 3573 3575 3577 3583 3584 3588
3590 3639 3641 3642 3643 3646 3688 3690 3697 3699 3700 3717 3719
3724 3743 3745 3746 3747 3750 3788 3790 3797 3800 3801 3822 3824
3828 3898 3899 3900 3905 3908 3910 3914 3950 3952 3956 3987 3989
3991 3993 3999 4004 4008 4010 4015 4062 4063 4067 4070 4072 4076
4118 4152 4153 4155 4157 4161 4162 4165 4169 4171 4175 4193 4194
4201 4203 4206 4217 4225 4247 4249 4250 4251 4254 4259 4260 4267
4274 4287 4289 4290 4291 4294 4295 4297 4298 4299 4302 4303 4304
4306 4307 4312 4335 4368 4373 4382 4383 4384 4386 4387 4389 4393
4394 4401 4412 4413 4422 4424 4439

$LSTIN= 000000

1# 1138 1139 1141 1142 1145 1146 1148 1149 1150 1151 1153 1154
1155 1156 1157 1158 1159 1160 1161 1162 1163 1164 1165 1166 1167
1168 1180 1181 1183 1184 1185 1186 1187 1188 1189 1190 1193 1194
1196 1197 1201 1202 1203 1210 1211 1212 1213 1214 1215 1216 1217
1220 1221 1222 1223 1224 1225 1245 1246 1248 1249 1250 1259 1260
1261 1262 1264 1265 1266 1275 1276 1278 1279 1281 1282 1283 1292
1293 1295 1296 1300 1301 1302 1321 1322 1324 1325 1326 1335 1336
1337 1338 1340 1341 1342 1351 1352 1354 1355 1357 1358 1359 1368
1369 1371 1372 1376 1377 1378 1403 1404 1406 1407 1408 1417 1418
1419 1420 1422 1423 1424 1433 1434 1436 1437 1439 1440 1441 1450
1451 1453 1454 1458 1459 1460 1483 1484 1485 1486 1487 1492 1493
1494 1495 1497 1498 1499 1508 1509 1511 1512 1513 1514 1515 1524
1525 1526 1527 1529 1530 1531 1550 1551 1552 1553 1554 1560 1561
1563 1564 1565 1574 1575 1576 1577 1579 1580 1581 1590 1591 1593
1594 1596 1597 1598 1607 1608 1610 1611 1615 1616 1617 1636 1637
1639 1640 1641 1650 1651 1652 1653 1655 1656 1657 1666 1667 1669
1670 1672 1673 1674 1682 1683 1686 1687 1691 1692 1693 1712 1713
1715 1716 1717 1726 1727 1728 1729 1731 1732 1733 1742 1743 1745
1746 1748 1749 1750 1759 1760 1762 1763 1767 1768 1769 1788 1789
1791 1792 1793 1802 1803 1804 1805 1807 1808 1809 1818 1819 1821
1822 1824 1825 1826 1835 1836 1838 1839 1843 1844 1845 1875 1876

1877	1878	1879	1907	1908	1910	1911	1912	1940	1941	1942	1943	1944
1972	1973	1974	1976	1977	1984	1985	1987	1988	1989	2026	2027	2028
2032	2033	2044	2045	2048	2049	2051	2052	2053	2064	2065	2068	2069
2071	2072	2073	2084	2085	2088	2089	2091	2092	2093	2117	2118	2119
2121	2122	2133	2134	2137	2138	2140	2141	2142	2153	2154	2157	2158
2160	2161	2162	2173	2174	2177	2178	2180	2181	2182	2206	2207	2208
2211	2212	2223	2224	2227	2228	2230	2231	2232	2243	2244	2247	2248
2250	2251	2252	2263	2264	2267	2268	2270	2271	2272	2295	2296	2297
2299	2300	2311	2312	2315	2316	2318	2319	2320	2331	2332	2335	2336
2338	2339	2350	2351	2352	2355	2356	2358	2359	2360	2386	2387	2388
2391	2392	2404	2405	2407	2408	2410	2411	2412	2413	2414	2416	2417
2419	2420	2421	2431	2432	2435	2436	2437	2438	2439	2446	2447	2448
2457	2458	2461	2462	2463	2464	2465	2486	2487	2488	2491	2492	2504
2505	2508	2509	2511	2512	2513	2514	2515	2517	2518	2520	2521	2522
2531	2532	2536	2537	2539	2540	2541	2542	2543	2544	2545	2546	255
2556	[illegible]	[illegible]	[illegible]	[illegible]	64	2565	2566	2567	2568	2569	2590	25
2592	[illegible]	[illegible]	[illegible]	[illegible]	11	2612	2615	2616	2619	2620	2622	26 3
2624	[illegible]	[illegible]	[illegible]	[illegible]	29	2638	2639	2641	2642	2644	2645	2646
2647	[illegible]	[illegible]	[illegible]	[illegible]	3	2674	2675	2676	2677	2681	2682	2689
2690	[illegible]	[illegible]	[illegible]	[illegible]	3	2699	2700	2701	2704	2705	2713	2714
2720	2721	[illegible]	[illegible]	[illegible]	[illegible]	2738	2739	2740	2741	2742	2743	2744
2745	2746	[illegible]	[illegible]	[illegible]	69	2772	2773	2778	2779	2780	[illegible]	2782
2783	2784	[illegible]	[illegible]	[illegible]	[illegible]	2799	2805	2806	2807	2824	2825	2826
2827	2832	2833	[illegible]	[illegible]	39	2840	2841	2842	2843	2844	2846	2847
2859	2860	[illegible]	[illegible]	[illegible]	12	2883	2884	2885	2886	2890	2891	2892
2893	2894	[illegible]	[illegible]	[illegible]	4	2905	2906	2907	2908	2909	2910	2912
2913	2916	2917	2918	2919	0	2922	2923	2924	2929	2930	2931	2932
2943	2944	2945	2947	2948	2949	2950	2951	2952	2958	2959	2965	2966
2970	2971	2972	2984	2985	2987	2988	2989	3011	3012	3018	3019	3021
3022	3023	3024	3025	3028	3029	3031	3032	3033	3034	3035	3038	3039
3042	3043	3044	3050	3051	3058	3059	3060	3061	3062	3070	3071	3077
3078	3081	3082	3083	3095	3096	3100	3101	3103	3104	3105	3106	3107
3109	3110	3111	3118	3119	3124	3125	3126	3150	3151	3152	3153	3154
3159	3160	3161	3162	3163	3168	3169	3170	3171	3172	3173	3175	3176
3177	3178	3179	3180	3181	3182	3183	3184	3186	3187	3189	3190	3192
3193	3195	3196	3197	3198	3200	3201	[illegible]02	3203	3204	3205	3207	3208
3209	3210	3213	3214	3215	3216	3220	[illegible]21	3224	3225	3226	3228	3229
3232	3233	3234	3235	3236	3238	3239	3252	3253	3254	3255	3257	3258
3260	3261	3262	3263	3264	3265	3266	3322	3323	3326	3327	3329	3330
3332	3333	3334	3335	3336	3337	3338	3339	3341	3342	3351	3352	3355
3356	3357	3358	3359	3362	3363	3364	3366	3367	3368	3372	3373	3385
3386	3388	3389	3391	3392	3399	3400	3401	3402	3403	3404	3405	3406
3418	3419	3420	3421	3422	3423	3446	3447	3448	3449	3450	3451	3453
3454	3455	3456	3458	3459	3461	3462	3471	3472	3476	3477	3478	3479
3480	3481	3482	3483	3485	3486	3488	3489	3490	3491	3492	3493	3494
3497	3498	3499	3501	3502	3503	3508	3509	3519	3520	3522	3523	3533
3534	3535	3536	3537	3539	3540	3545	3546	3548	3549	3556	3557	3559
3560	3561	3562	3564	3565	3566	3567	3568	3571	3572	3573	3575	3576
3577	3582	3583	3592	3593	3594	3595	3598	3599	3602	3603	3604	3605
3606	3607	3620	3621	3630	3631	3632	3533	3635	3636	3639	3640	3641
3642	3643	3644	3645	3646	3651	3652	3653	3654	3655	3656	3657	3658
3661	3662	3664	3665	3666	3667	3668	3669	3670	3671	3673	3674	3681
3682	3683	3684	3685	3688	3689	3690	3694	3695	3699	3700	3704	3705
3717	3718	3719	3721	3722	3727	3728	3736	3737	3738	3739	3740	3741
3743	3744	3745	3746	3747	3748	3749	3750	3755	3756	3757	3758	3759
3760	3761	3762	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774

```
                    3777    3778    3781    3782    3783    3784    3785    3788    3789    3790    3794    3795    3800
                    3801    3822    3823    3824    3825    3826    3831    3832    3833    3834    3855    3856    3858
                    3859    3860    3861    3864    3865    3866    3867    3870    3871    3873    3874    3876    3877
                    3880    3881    3886    3887    3890    3891    3893    3894    3900    3901    3902    3903    3904
                    3908    3909    3910    3917    3918    3919    3920    3940    3941    3943    3944    3945    3947
                    3948    3950    3951    3952    3954    3955    3975    3976    3978    3979    3980    3982    3983
                    3984    3987    3988    3989    3991    3992    3993    3995    3996    3997    3998    4002    4003
                    4008    4009    4010    4012    4013    4040    4041    4044    4045    4049    4060    4052    4053
                    4054    4055    4056    4057    4058    4059    4060    4061    4062    4063    4070    4071    4072
                    4142    4143    4144    4145    4146    4147    4155    4156    4157    4158    4159    4160    4161
                    4163    4164    4169    4170    4171    4173    4174    4178    4179    4180    4181    4182    4186
                    4187    4188    4189    4190    4191    4192    4193    4194    4201    4202    4203    4204    4205
                    4208    4209    4210    4218    4219    4220    4221    4239    4240    4241    4242    4243    4244
                    4245    4247    4248    4249    4250    4251    4252    4253    4254    4255    4256    4257    4258
                    4259    4260    4262    4263    4264    4265    4269    4270    4285    4286    4287    4288    4289
                    4290    4291    4292    4293    4294    4295    4296    4297    4298    4299    4300    4301    4302
                    4303    4304    4306    4307    4314    4315    4328    4329    4370    4371    4384    4385    4386
                    4387    4388    4389    4390    4391    4392    4393    4395    4396    4399    4400    4403    4404
                    4405    4406    4407    4408    4409    4410    4411    4412    4414    4415    4416    4417    4418
                    4419    4420    4421    4424    4425    4426    4428    4429    4430    4431    4432    4433    4441
                    4442
$LSTST= 177777       1#  1191    1192    1201    1203    1207    1215    1248    1250    1254    1264    1266    1270
                    1281    1283    1287    1300    1302    1306    1324    1326    1330    1340    1342    1346    1357
                    1359    1363    1376    1378    1382    1406    1408    1412    1422    1424    1428    1439    1441
                    1445    1458    1460    1464    1483    1485    1489    1497    1499    1503    1513    1515    1519
                    1529    1531    1535    1550    1552    1556    1563    1565    1569    1579    1581    1585    1596
                    1598    1602    1615    1617    1621    1639    1641    1645    1655    1657    1661    1672    1674
                    1678    1691    1693    1697    1715    1717    1721    1731    1733    1737    1748    1750    1754
                    1767    1769    1773    1791    1793    1797    1807    1809    1813    1824    1826    1830    1843
                    1845    1849    1877    1879    1888    1910    1912    1921    1942    1944    1955    1972    1974
                    1979    1987    1989    2002    2026    2028    2035    2051    2053    2059    2071    2073    2079
                    2091    2093    2099    2117    2119    2124    2140    2142    2148    2160    2162    2168    2180
                    2182    2188    2206    2208    2214    2230    2232    2238    2250    2252    2258    2270    2272
                    2278    2295    2297    2302    2318    2320    2326    2338    2340    2346    2358    2360    2366
                    2386    2388    2394    2419    2421    2425    2437    2439    2443    2446    2448    2452    2463
                    2465    2469    2486    2488    2494    2520    2522    2526    2544    2546    2550    2567    2569
                    2573    2590    2592    2597    2627    2629    2633    2649    2651    2655    2673    2675    2679
                    2704    2705    2709    2725    2727    2731    2745    2746    2750    2788    2789    2793    2805
                    2807    2811    2846    2847    2851    2862    2864    2868    2882    2884    2888    2902    2903
                    2904    2906    2909    2910    2911    2914    2916    2922    2924    2934    2942    2943    2945
                    2947    2949    2950    2952    2961    2963    2965    2966    2970    2972    2977    2987    2989
                    2993    3042    3044    3053    3060    3062    3073    3081    3083    3091    3109    3111    3121
                    3124    3126    3131    3150    3152    3156    3159    3161    3165    3175    3177    3178    3179
                    3182    3199    3200    3202    3203    3205    3209    3210    3211    3217    3220    3221    3224
                    3226    3228    3229    3230    3232    3236    3238    3239    3240    3247    3249    3257    3258
                    3362    3364    3366    3368    3372    3373    3374    3379    3381    3404    3406    3410    3497
                    3499    3501    3503    3508    3509    3510    3514    3516    3535    3537    3542    3571    3573
                    3575    3577    3582    3583    3584    3588    3590    3639    3641    3642    3643    3646    3688
                    3690    3697    3699    3700    3717    3719    3724    3743    3745    3746    3747    3750    3788
                    3790    3797    3800    3801    3822    3824    3828    3898    3899    3900    3908    3910    3914
                    3950    3952    3956    3987    3989    3991    3993    3999    4004    4008    4010    4015    4062
                    4063    4067    4070    4072    4076    4118    4152    4153    4155    4157    4160    4161    4162
                    4165    4169    4171    4175    4178    4193    4194    4201    4203    4206    4217    4225    4247
                    4249    4250    4251    4254    4259    4260    4267    4274    4287    4289    4290    4291    4294
                    4295    4297    4298    4299    4302    4303    4304    4306    4307    4312    4335    4368    4373
                    4382    4383    4384    4386    4387    4389    4392    4393    4394    4401    4411    4412    4413
```

```
                    4422    4424    4439
$LSTTA= 000000       1#  1191    1192    1207    1208    1254    1255    1270    1271    1287    1288    1306    1307
                    1330    1331    1346    1347    1363    1364    1382    1383    1412    1413    1428    1429    1445
                    1446    1464    1465    1489    1490    1503    1504    1519    1520    1535    1536    1556    1557
                    1569    1570    1585    1586    1602    1603    1621    1622    1645    1646    1661    1662    1678
                    1679    1697    1698    1721    1722    1737    1738    1754    1755    1773    1774    1797    1798
                    1813    1814    1830    1831    1849    1850    1888    1889    1921    1922    1955    1956    1979
                    1980    2002    2003    2035    2036    2059    2060    2079    2080    2099    2100    2124    2125
                    2148    2149    2168    2169    2188    2189    2214    2215    2238    2239    2258    2259    2278
                    2279    2302    2303    2326    2327    2346    2347    2366    2367    2394    2395    2425    2426
                    2443    2444    2452    2453    2469    2470    2494    2495    2526    2527    2550    2551    2573
                    2574    2597    2598    2633    2634    2655    2656    2679    2680    2709    2710    2731    2732
                    2750    2751    2793    2794    2811    2812    2851    2852    2868    2869    2888    2889    2902
                    2903    2910    2911    2914    2915    2920    2921    2934    2935    2942    2943    2961    2962
                    2963    2964    2966    2967    2977    2978    2993    2994    3053    3054    3073    3074    3091
                    3092    3121    3122    3131    3132    3156    3157    3165    3166    3177    3178    3179    3180
                    3199    3200    3210    3211    3217    3218    3221    3222    3229    3230    3239    3240    3247
                    3248    3249    3250    3258    3259    3373    3374    3379    3380    3381    3382    3410    3411
                    3509    3510    3514    3515    3516    3517    3542    3543    3583    3584    3588    3589    3590
                    3591    3641    3642    3643    3644    3697    3698    3700    3701    3724    3725    3745    3746
                    3747    3748    3797    3798    3801    3802    3828    3829    3898    3899    3904    3905    3914
                    3915    3956    3957    3999    4000    4004    4005    4015    4016    4067    4068    4076    4077
                    4117    4118    4152    4153    4161    4162    4165    4166    4175    4176    4194    4195    4206
                    4207    4217    4218    4219    4220    4224    4225    4249    4250    4251    4252    4260    4261
                    4267    4268    4269    4273    4274    4289    4290    4291    4292    4297    4298    4299    4300
                    4304    4305    4307    4308    4312    4313    4314    4334    4335    4368    4369    4370    4372
                    4373    4382    4383    4393    4394    4401    4402    4412    4413    4422    4423    4439    4440
                    4441
$MADR1  001226      1029#
$MADR2  001232      1033#
$MADR3  001236      1036#
$MADR4  001242      1039#
$MAIL   001174       949     953    1002#   1111    1126    1178    1241    1317    1394    1481    1548    1633    1709
                    1785    1868    1901    1934    1968    2023    2114    2203    2292    2383    2483    2587    2669
                    2761    2820    2878    3008    3148    3319    3442    3617    3715    3817    4037    4551    4834
                    4912
$MAMS1  001224      1023#
$MAMS2  001230      1031#
$MAMS3  001234      1034#
$MAMS4  001240      1037#
$MBADR  001002       949#
$MCALL= 000000         1#
$MFLG   014100      4750#   4756    4791#   4795#
$MNEW   013625      4632    4744#
$MSGAD  001210      1009#   4766#   4769
$MSGLG  001212      1010#   4771#
$MSGTY  001174      1003#   4764    4772#   4784    4788#
$MSWP   013614      4629    4742#
$MTYP1  001225      1024#
$MTYP2  001231      1032#
$MTYP3  001235      1035#
$MTYP4  001241      1038#
$MXCNT  014562      4910    4920#
$NESTL= 177777         1#  1191#   1201#   1207#   1215#   1248#   1254#   1264#   1270#   1281#   1287#   1300#   1306#
                    1324#   1330#   1340#   1346#   1357#   1363#   1376#   1382#   1406#   1412#   1422#   1428#   1439#
                    1445#   1458#   1464#   1483#   1489#   1497#   1503#   1513#   1519#   1529#   1535#   1550#   1556#
```

```
                    1563#  1569#  1579#  1585#  1596#  1602#  1615#  1621#  1639#  1645#  1655#  1661#  1672#
                    1678#  1691#  1697#  1715#  1721#  1731#  1737#  1748#  1754#  1767#  1773#  1791#  1797#
                    1807#  1813#  1824#  1830#  1843#  1849#  1877#  1888#  1910#  1921#  1942#  1955#  1972#
                    1979#  1987#  2002#  2026#  2035#  2051#  2059#  2071#  2079#  2091#  2099#  2117#  2124#
                    2140#  2148#  2160#  2168#  2180#  2188#  2206#  2214#  2230#  2238#  2250#  2258#  2270#
                    2278#  2295#  2302#  2318#  2326#  2338#  2346#  2358#  2366#  2386#  2394#  2419#  2425#
                    2437#  2443#  2446#  2452#  2463#  2469#  2486#  2494#  2520#  2526#  2544#  2550#  2567#
                    2573#  2590#  2597#  2627#  2633#  2649#  2655#  2673#  2679#  2704#  2709#  2725#  2731#
                    2745#  2750#  2788#  2793#  2805#  2811#  2846#  2851#  2862#  2868#  2882#  2888#  2902#
                    2904#  2909    2914#  2916#  2922#  2934#  2942#  2947#  2950#  2961#  2963#  2965#  2970#
                    2977#  2987#  2993#  3042#  3053#  3060#  3073#  3081#  3091#  3109#  3121#  3124#  3131#
                    3150#  3156#  3159#  3165#  3175#  3199#  3203#  3209    3217#  3220#  3224#  3228    3232#
                    3238    3247#  3249#  3257#  3362#  3366#  3372    3379#  3381#  3404#  3410#  3497#  3501#
                    3508    3514#  3516#  3535#  3542#  3571#  3575#  3582    3588#  3590#  3639#  3688#  3697#
                    3699#  3717#  3724#  3743#  3788#  3797#  3800#  3822#  3828#  3898#  3900#  3908#  3914#
                    3950#  3956#  3987#  3991#  3999#  4004#  4008#  4015#  4062#  4067#  4070#  4076#  4118#
                    4152#  4155#  4160    4165#  4169#  4175#  4178    4193#  4201#  4206#  4217#  4225#  4247#
                    4259#  4267#  4274#  4287#  4295#  4303#  4306#  4312#  4335#  4368#  4373#  4382#  4384#
                    4387#  4392    4401#  4411    4422#  4424#  4439#
$NSK0 = 000300      1191#  1215    1248#  1254    1264#  1270    1281#  1287    1300#  1306    1324#  1330    1340#
                    1346    1357#  1363    1376#  1382    1406#  1412    1422#  1428    1439#  1445    1458#  1464
                    1483#  1489    1497#  1503    1513#  1519    1529#  1535    1550#  1556    1563#  1569    1579#
                    1585    1596#  1602    1615#  1621    1639#  1645    1655#  1661    1672#  1678    1691#  1697
                    1715#  1721    1731#  1737    1748#  1754    1767#  1773    1791#  1797    1807#  1813    1824#
                    1830    1843#  1849    1877#  1888    1910#  1921    1942#  1955    1972#  1979    1987#  2002
                    2026#  2035    2051#  2059    2071#  2079    2091#  2099    2117#  2124    2140#  2148    2160#
                    2168    2180#  2188    2206#  2214    2230#  2238    2250#  2258    2270#  2278    2295#  2302
                    2318#  2326    2338#  2346    2358#  2366    2386#  2394    2419#  2425    2437#  2443    2446#
                    2452    2463#  2469    2486#  2494    2520#  2526    2544#  2550    2567#  2573    2590#  2597
                    2627#  2633    2649#  2655    2673#  2679    2704#  2709    2725#  2731    2745#  2750    2788#
                    2793    2805#  2811    2846#  2851    2862#  2868    2882#  2888    2902#  2916    2922#  2934
                    2942#  2965    2970#  2977    2987#  2993    3042#  3053    3060#  3073    3081#  3091    3109#
                    3121    3124#  3131    3150#  3156    3159#  3165    3175#  3257    3362#  3381    3404#  3410
                    3497#  3516    3535#  3542    3571#  3590    3639#  3699    3717#  3724    3743#  3800    3822#
                    3828    3898#  3900    3908#  3914    3950#  3956    3987#  4004    4008#  4015    4062#  4067
                    4070#  4076    4118#  4217    4225#  4267    4274#  4312    4335#  4368    4373#  4439
$NSK1 = 000130      1201#  1207    2904#  2909    2914    2947#  2963    3199#  3220    3224#  3228    3249    3366#
                    3372    3379    3501#  3508    3514    3575#  3582    3588    3688#  3697    3788#  3797    3991#
                    3999    4152#  4178    4193    4201#  4206    4247#  4259    4287#  4306    4382#  4424
$NSK2 = 000110      2950#  2961    3203#  3209    3217    3232#  3238    3247    4155#  4160    4165    4169#  4175
                    4295#  4303    4384#  4411    4422
$NSK3 = 000110      4387#  4392    4401
$NULL   001154       987#  4578    4607
$NWTST= 000001      1170#  1172    1234#  1236    1312#  1389#  1471#  1473    1542#  1544    1628#  1704#  1780#
                    1863#  1896#  1929#  1963#  2016#  2018    2107#  2109    2196#  2198    2286#  2288    2374#
                    2376    2476#  2478    2582#  2662#  2664    2754#  2756    2815#  2871#  2873    3003#  3140#
                    3142    3309#  3311    3434#  3436    3611#  3613    3710#  3811#  3813    4030#  4032    4088#

$OCNT   015232      5020#  5049#  5062#
$OMODE  015234      5015#  5019#  5024    5027#  5038#  5064#
$OVER   014546      4874    4890    4898    4908    4917#
$PASS   001202      1006#  1111#  4457#  4458#  4466    4479    4904    4921
$PASTM  001006       951#
$POWER  012570      4518    4525#
$PWRAD  012556      4520#
$PWRDN  012416      1087    4487#  4515
$PWRMG  012552      4518#
```

```
$PWRUP  012470          4497    4503#
$QUES   001170           994#   4607    4675    4724    4740    4857
$RDCHR  013344          4688#   5105
$RDDEC= ****** U        5107
$RDLIN  013464          4716#   5106
$RDOCT= ****** U        5107
$RDSZ = 000010          4709#
$RTNAD  012374          4478#
$R2A  = ****** U        5107
$SAVLE= 177777             1#   2965#   2966#   3178#   3182#   3220#   3221#   3642#   3646#   3746#   3750#   4193#   4194#
                        4250#   4254#   4290#   4294#   4298#   4302#
$SAVRE= ****** U        5107
$SAVR6  012566          4496#   4504    4505#   4506#   4524#
$SCOPE  014304          1081    4871#
$SETUP= 000137          1072#   1080    1081    1083    1085    1087    1089    1090    1091    1093    1120    1123    4455
                        4612    4746    4818    4844    4852    4872
$SSKO = 000234          2965#   2966    3178#   3182    3220#   3221    3642#   3646    3746#   3750    4193#   4194    4250#
                        4254    4290#   4294    4298#   4302
$STUP = 177777          1072#
$SVLAD  014512          4882    4911#
$SVPC = 000204           925#    930
$SWR  = 167400             1#    673     677     678     679     680     681     682     683     991     992     993    1090
                        1091    1093    1094    1177    1240    1316    1393    1480    1547    1632    1708    1784    1967
                        1900    1933    1967    2022    2113    2202    2291    2382    2482    2586    2668    2760    2819
                        2877    3007    3147    3318    3441    3616    3714    3816    4036    4092    4450    4456    4471
                        4477    4479    4521    4809    4810    4811    4812    4813    4822    4829    4841    4845    4857
                        4863    4864    4865    4866    4867    4873    4885    4887    4888    4891    4892    4893    4900
                        4901    4902    4914    4917    4920
$SWREG  001216          1014#   1114
$SWRMK= 000000           683     684    4867    4868    4889
$TAGLE= 177777             1#   1192#   1203#   1207#   1215#   1250#   1254#   1266#   1270#   1283#   1287#   1302#   1306#
                        1326#   1330#   1342#   1346#   1359#   1363#   1378#   1382#   1408#   1412#   1424#   1428#   1441#
                        1445#   1460#   1464#   1485#   1489#   1499#   1503#   1515#   1519#   1531#   1535#   1552#   1556#
                        1565#   1569#   1581#   1585#   1598#   1602#   1617#   1621#   1641#   1645#   1657#   1661#   1674#
                        1678#   1693#   1697#   1717#   1721#   1733#   1737#   1750#   1754#   1769#   1773#   1793#   1797#
                        1809#   1813#   1826#   1830#   1845#   1849#   1879#   1888#   1912#   1921#   1944#   1955#   1974#
                        1979#   1989#   2002#   2028#   2035#   2053#   2059#   2073#   2079#   2093#   2099#   2119#   2124#
                        2142#   2148#   2162#   2168#   2182#   2188#   2208#   2214#   2232#   2238#   2252#   2258#   2272#
                        2278#   2297#   2302#   2320#   2326#   2340#   2346#   2360#   2366#   2388#   2394#   2421#   2425#
                        2439#   2443#   2448#   2452#   2465#   2469#   2488#   2494#   2522#   2526#   2546#   2550#   2569#
                        2573#   2592#   2597#   2620#   2633#   2651#   2655#   2675#   2679#   2705#   2709#   2727#   2731#
                        2746#   2750#   2789#   2793#   2807#   2811#   2847#   2851#   2864#   2868#   2884#   2888#   2903#
                        2906#   2910#   2911#   2914#   2916#   2924#   2934#   2943#   2945#   2949#   2952#   2961#   2963#
                        2965#   2972#   2977#   2989#   2993#   3044#   3053#   3062#   3073#   3083#   3091#   3111#   3121#
                        3126#   3131#   3152#   3156#   3161#   3165#   3177#   3179#   3182#   3200#   3202#   3205#   3210#
                        3211#   3217#   3220#   3226#   3229#   3230#   3236#   3239#   3240#   3247#   3249#   3257#   3258#
                        3364#   3368#   3373#   3374#   3379#   3381#   3406#   3410#   3499#   3503#   3509#   3510#   3514#
                        3516#   3537#   3542#   3573#   3577#   3583#   3584#   3588#   3590#   3641#   3643#   3646#   3690#
                        3697#   3699#   3700#   3719#   3724#   3745#   3747#   3750#   3790#   3797#   3800#   3801#   3824#
                        3828#   3899#   3900#   3910#   3914#   3952#   3956#   3989#   3993#   3999#   4004#   4010#   4015#
                        4063#   4067#   4072#   4076#   4153#   4157#   4161#   4162#   4165#   4171#   4175#   4178    4193#
                        4203#   4206#   4249#   4251#   4254#   4239#   4260#   4289#   4291#   4294#   4297#   4299#   4302#
                        4303#   4304#   4306#   4307#   4383#   4386#   4389#   4393#   4394#   4401#   4412#   4413#   4422#
                        4424#
$TAGNU= 000247             1#   1191    1192#   1202    1203#   1249    1250#   1265    1266#   1282    1283#   1301    1302#
                        1325    1326#   1341    1342#   1358    1359#   1377    1378#   1407    1408#   1423    1424#   1440
```

```
                    3101#  3118#  3119#  3121#  3131#  3153#  3154#  3156#  3162#  3163#  3165#  3168#  3169#
                    3170#  3171#  3172#  3173#  3179#  3182#  3183#  3184#  3186#  3187#  3189#  3190#  3192#
                    3193#  3195#  3196#  3197#  3198#  3207#  3208#  3209#  3210#  3213#  3214#  3215#  3216#
                    3217#  3220#  3221#  3228#  3229#  3238#  3239#  3247#  3249#  3252#  3253#  3254#  3255#
                    3257#  3258#  3260#  3264#  3265#  3266#  3322#  3323#  3326#  3327#  3329#  3330#  3332#
                    3333#  3334#  3335#  3336#  3337#  3338#  3339#  3341#  3342#  3351#  3352#  3372#  3373#
                    3379#  3381#  3385#  3386#  3388#  3389#  3391#  3392#  3410#  3418#  3419#  3421#  3422#
                    3423#  3446#  3447#  3448#  3449#  3450#  3451#  3453#  3454#  3455#  3456#  3458#  3459#
                    3461#  3462#  3471#  3472#  3485#  3486#  3493#  3494#  3508#  3509#  3514#  3516#  3519#
                    3520#  3522#  3523#  3533#  3534#  3539#  3540#  3542#  3545#  3546#  3548#  3549#  3556#
                    3557#  3559#  3560#  3561#  3562#  3582#  3583#  3588#  3590#  3592#  3593#  3594#  3595#
                    3598#  3599#  3602#  3603#  3605#  3606#  3607#  3620#  3621#  3635#  3636#  3643#  3646#
                    3661#  3662#  3673#  3674#  3681#  3683#  3684#  3685#  3694#  3695#  3697#  3699#  3700#
                    3704#  3705#  3721#  3722#  3724#  3727#  3728#  3740#  3741#  3747#  3750#  3765#  3766#
                    3777#  3778#  3781#  3783#  3784#  3785#  3794#  3795#  3797#  3800#  3801#  3825#  3826#
                    3828#  3855#  3856#  3858#  3859#  3860#  3861#  3864#  3865#  3866#  3867#  3870#  3871#
                    3873#  3874#  3876#  3877#  3880#  3881#  3886#  3887#  3890#  3891#  3893#  3894#  3900#
                    3903   3914#  3917#  3918#  3919#  3920#  3940#  3941#  3943#  3945#  3947#  3948#  3954#
                    3955#  3956#  3975#  3976#  3978#  3980#  3982#  3984#  3995#  3996#  3997#  3998#  3999#
                    4002#  4003#  4004#  4012#  4013#  4015#  4040#  4041#  4044#  4045#  4049#  4050#  4052#
                    4053#  4067#  4076#  4142#  4143#  4144#  4145#  4146#  4147#  4158#  4159#  4160#  4161#
                    4163#  4164#  4165#  4173#  4174#  4175#  4178#  4179   4181   4191#  4192#  4193#  4194#
                    4206#  4217#  4239#  4240#  4241#  4245#  4251#  4254#  4255#  4256#  4257#  4258#  4259#
                    4260#  4262#  4263#  4267#  4285#  4286#  4291#  4294#  4299#  4302#  4303#  4304#  4306#
                    4307#  4312#  4328#  4329#  4368#  4390#  4391#  4392#  4393#  4399#  4400#  4401#  4403#
                    4404#  4405#  4406#  4407#  4408#  4409#  4410#  4411#  4412#  4414#  4415#  4416#  4417#
                    4418#  4419#  4420#  4421#  4422#  4424#  4425   4428#  4429#  4430#  4431#  4439#
$TESTN  001200      1005#  1178#  1241#  1317#  1394#  1481#  1548#  1633#  1709#  1785#  1868#  1901#  1934#
                    1968#  2023#  2114#  2203#  2292#  2383#  2483#  2587#  2669#  2761#  2820#  2878#  3008#
                    3148#  3319#  3442#  3617#  3715#  3817#  4037#  4340   4912#
$TIMES  001160       991#  1090#  1177#  1240#  1316#  1393#  1480#  1486#  1547#  1553#  1632#  1708#  1784#
                    1867#  1900#  1933#  1967#  1976#  2022#  2032#  2113#  2121#  2202#  2211#  2291#  2299#
                    2382#  2391#  2482#  2491#  2586#  2594#  2668#  2676#  2760#  2819#  2877#  2885#  2931#
                    2958#  3007#  3050#  3070#  3118#  3147#  3153#  3162#  3318#  3441#  3539#  3616#  3694#
                    3714#  3721#  3794#  3816#  3825#  4036#  4092#  4456#  4900#  4907   4910#  4920
$TKB    001146       984#  4610   4621   4638   4692   4698
$TKS    001144       983#  4610   4619   4635   4659#  4690   4696
$TN   = 000042         1#   673   1170   1177#  1178   1234   1240#  1241   1312   1316#  1317   1389   1393#
                    1394   1471   1480#  1481   1487   1542   1547#  1548   1554   1628   1632#  1633   1704
                    1708#  1709   1780   1784#  1785   1863   1867#  1868   1896   1900#  1901   1929   1933#
                    1934   1963   1967#  1968   1977   2016   2022#  2023   2033   2107   2113#  2114   2122
                    2196   2202#  2203   2212   2286   2291#  2292   2300   2374   2382#  2383   2392   2476
                    2482#  2483   2492   2576   2582   2586#  2587   2595   2662   2668#  2669   2677   2754
                    2760#  2761   2815   2819#  2820   2871   2877#  2878   2886   2932   2959   2996   3003
                    3007#  3008   3051   3071   3089   3119   3134   3140   3147#  3148   3154   3163   3268
                    3309   3318#  3319   3434   3441#  3442   3540   3611   3616#  3617   3695   3710   3714#
                    3715   3722   3795   3811   3816#  3817   3826   3922   4030   4036#  4037   4080   4088
                    4092#
$TPB    001152       986#  4596#  4607
$TPFLG  001157       990#  4545   4607
$TPS    001150       985#  4594   4607
$TRAP   015236      1085   5073#
$TRAP2  015260      5084#  5095
$TRP  = 000012      5088#  5097#  5098#  5099#  5100#  5101#  5102   5103#  5104   5105#  5106#  5107#
$TRPAD  015272      5078   5095#
$TSKO = 000242      1192#  1215   1250#  1254   1266#  1270   1283#  1287   1302#  1306   1326#  1330   1342#
```

```
                    1346   1359#  1363   1378#  1382   1408#  1412   1424#  1428   1441#  1445   1460#  1464
                    1485#  1489   1499#  1503   1515#  1519   1531#  1535   1552#  1556   1565#  1569   1581#
                    1585   1598#  1602   1617#  1621   1641#  1645   1657#  1661   1674#  1678   1693#  1697
                    1717#  1721   1733#  1737   1750#  1754   1769#  1773   1793#  1797   1809#  1813   1826#
                    1830   1845#  1849   1879#  1888   1912#  1921   1944#  1955   1974#  1979   1989#  2002
                    2028#  2035   2053#  2059   2073#  2079   2093#  2099   2119#  2124   2142#  2148   2162#
                    2168   2182#  2188   2208#  2214   2232#  2238   2252#  2258   2272#  2278   2297#  2302
                    2320#  2326   2340#  2346   2360#  2366   2388#  2394   2421#  2425   2439#  2443   2448#
                    2452   2465#  2469   2488#  2494   2522#  2526   2546#  2550   2569#  2573   2592#  2597
                    2629#  2633   2651#  2655   2675#  2679   2705#  2709   2727#  2731   2746#  2750   2789#
                    2793   280?#  2811   2847#  2851   2864#  2868   2884#  2?38   2903#  2916   2924#  2934
                    2943#  2965   2972#  2977   2989#  2993   3044#  3053   3062#  3073   3083#  3091   3111#
                    3121   3126#  3131   3152#  3156   3161#  3165   3177#  3179   3182#  3258   3364#  3381
                    3406#  3410   3499#  3516   3537#  3542   3573#  3590   3641#  3643   3646#  3700   3719#
                    3724   3745#  3747   3750#  3801   3824#  3828   3899#  3900   3910#  3914   3952#  3956
                    3989#  4004   4010#  4015   4063#  4067   4072#  4076   4153#  4193   4203#  4206   4249#
                    4251   4254#  4260   4289#  4291   4294#  4307   4383#  4424
$TSK1 = 000246      1203#  1207   2906#  2910   2911#  2914   2945#  2965   3182#  3257   3368#  3373   3374#
                    3379   3503#  3509   3510#  3514   3577#  3583   3584#  3588   3646#  3699   3750#  3800
                    3993#  3999   4153#  4178   4193   4254#  4259   4294#  4306   4386#  4412   4413#  4422
$TSK2 = 000245      2949#  2963   3200#  3220   3226#  3229   3230#  3249   3690#  3697   3790#  3797   4157#
                    4161   4162#  4165   4171#  4175   4297#  4299   4302#  4304   4389#  4393   4394#  4401
$TSK3 = 000234      2952#  2961   3202#  3220   3236#  3239   3240#  3247   4302#  4303
$TSK4 = 000146      3205#  3210   3211#  3217
$TSTM   001004       950#
$TSTNM  001102       963#  4455#  4821   4857   4862   4889   4911#  4912   4917   4921
$TTYIN  013572      4717   4718   4735   4739#
$TYPBN= ****** U    5101
$TYPOS  014564      4933#  5100
$TYPE   012600      4545#  4777   5088   5096
$TYPEC  013012      4575   4582   4589   4594#  4595   4661
$TYPEX  013060      4600   4602   4605#
$TYPOC  015034      5018#  5097
$TYPON  015050      5017   5020#  5099
$TYPOS  015010      5013#  5098
$UNIT   001206      1008#
$UNITM  001010       952#
$USWR   001220      1015#  1483   1550   1972   2026   2117   2206   2295   2386   2486   2590   3150   3260
                    3535   3717   4241
$VECT1  001244      1040#  4409
$VECT2  001246      1041#
$XTSTR  014316      4876#
$YESNO= 000001      1156#  1157#  1159#  1160#  1162#  1163#  1165#  1166#  1210#  1211#  1213#  1214#  1222#
                    1223#  1261#  1262#  1278#  1279#  1295#  1296#  1337#  1338#  1354#  1355#  1371#  1372#
                    1419#  1420#  1436#  1437#  1453#  1454#  1494#  1495#  1511#  1512#  1526#  1527#  1576#
                    1577#  1593#  1594#  1610#  1611#  1652#  1653#  1669#  1670#  1686#  1687#  1728#  1729#
                    1745#  1746#  1762#  1763#  1804#  1805#  1821#  1822#  1838#  1839#  2048#  2049#  2068#
                    2069#  2088#  2089#  2137#  2138#  2157#  2158#  2177#  2178#  2227#  2228#  2247#  2248#
                    2267#  2268#  2315#  2316#  2335#  2336#  2355#  2356#  2407#  2408#  2435#  2436#  2461#
                    2462#  2508#  2509#  2536#  2537#  2559#  2560#  2609#  2610#  2619#  2620#  2641#  2642#
                    2765#  2766#  2824#  2825#  2890#  2891#  2912#  2913#  3172#  3173#  3178#  3179#  3213#
                    3214#  3215#  3216#  3261#  3264#  3329#  3330#  3341#  3342#  3351#  3352#  3391#  3392#
                    3420#  3421#  3458#  3459#  3461#  3462#  3485#  3486#  3519#  3520#  3522#  3523#  3548#
                    3549#  3556#  3557#  3559#  3560#  3561#  3562#  3592#  3593#  3594#  3595#  3604#  3605#
                    3620#  3621#  3642#  3643#  3682#  3683#  3684#  3685#  3704#  3705#  3727#  3728#  3746#
                    3747#  3782#  3783#  3784#  3785#  3886#  3887#  3890#  3891#  3893#  3894#  3917#  3918#
```

$$ARGC= 000000
$$BYTE= 000403
$$DST = 000067
$$FLAG= 000001
$$FROM= 000000
$$GET4= 000000
$$LOC = 012246

```
3919#  3920#  3940#  3941#  3944#  3945#  3954#  3955#  3975#  3976#  3979#  3980#  3983#
3984#  4002#  4003#  4012#  4013#  4040#  4041#  4049#  4050#  4191#  4192#  4242#  4245#
4250#  4251#  4255#  4256#  4257#  4258#  4262#  4263#  4290#  4291#  4298#  4299#  4328#
4329#  4416#  4417#  4418#  4419#  4420#  4421#  4430#  4431#
4118#  4225#  4274#  4335#  4373#
1201#  1248#  1264#  1281#  1300#  1324#  1340#  1357#  1376#  1406#  1422#  1439#  1458#
1483#  1497#  1513#  1529#  1550#  1563#  1579#  1596#  1615#  1639#  1655#  1672#  1691#
1715#  1731#  1748#  1767#  1791#  1807#  1824#  1843#  1877#  1910#  1942#  1972#  1987#
2026#  2051#  2071#  2091#  2117#  2140#  2160#  2180#  2206#  2230#  2250#  2270#  2295#
2318#  2338#  2358#  2386#  2419#  2437#  2446#  2463#  2486#  2520#  2544#  2567#  2590#
2627#  2649#  2673#  2725#  2805#  2862#  2882#  2904#  2922#  2943#  2947#  2950#  2970#
2987#  3042#  3060#  3081#  3109#  3124#  3150#  3159#  3200#  3203#  3224#  3232#  3234#
3362#  3366#  3404#  3497#  3501#  3535#  3571#  3575#  3688#  3717#  3788#  3822#  3908#
3950#  3987#  3991#  4008#  4070#  4155#  4169#  4201#  4384#  4387#
3261#  4242#
1201#  1203   1207#  1246#  1250   1254#  1264#  1266   1270#  1281#  1283   1287#  1300#
1302   1306#  1324#  1326   1330#  1340#  1342   1346#  1357#  1359   1363#  1376#  1378
1382#  1406#  1408   1412#  1422#  1424   1428#  1439#  1441   1445#  1458#  1460   1464#
1483#  1485   1489#  1497#  1499   1503#  1513#  1515   1519#  1529#  1531   1535#  1550#
1552   1556#  1563#  1565   1569#  1579#  1581   1585#  1596#  1598   1602#  1615#  1617
1621#  1639#  1641   1645#  1655#  1657   1661#  1672#  1674   1678#  1691#  1693   1697#
1715#  1717   1721#  1731#  1733   1737#  1748#  1750   1754#  1767#  1769   1773#  1791#
1793   1797#  1807#  1809   1813#  1824#  1826   1830#  1843#  1845   1849#  1877#  1879
1888#  1910#  1912   1921#  1942#  1944   1955#  1972#  1974   1979#  1987#  1989   2002#
2026#  2028   2035#  2051#  2053   2059#  2071#  2073   2079#  2091#  2093   2099#  2117#
2119   2124#  2140#  2142   2148#  2160#  2162   2168#  2180#  2182   2188#  2206#  2208
2214#  2230#  2232   2238#  2250#  2252   2258#  2270#  2272   2278#  2295#  2297   2302#
2318#  2320   2326#  2338#  2340   2346#  2358#  2360   2366#  2386#  2388   2394#  2419#
2421   2425#  2437#  2439   2443#  2446#  2448   2452#  2463#  2465   2469#  2486#  2488
2494#  2520#  2522   2526#  2544#  2546   2550#  2567#  2569   2573#  2590#  2592   2597#
2627#  2629   2633#  2649#  2651   2655#  2673#  2675   2679#  2704#  2709#  2725#  2727
2731#  2745#  2750#  2788#  2793#  2805#  2807   2811#  2846#  2851#  2862#  2864   2868#
2882#  2884   2888#  2904#  29 ~   2914#  2922#  2924   2934#  2942#  2943#  2945   2947#
2949   2950#  2952   2961#  2963#  2970#  2972   2977#  2987#  2989   2993#  3042#  3044
3053#  3060#  3062   3073#  3081#  3083   3091#  3109#  3111   3121#  3124#  3126   3131#
3150#  3152   3156#  3159#  3161   3165#  3199#  3200#  3202   3203#  3205   3217#  3224#
3226   3232#  3234#  3236   3247#  3249#  3362#  3364   3366#  3368   3379#  3381#  3404#
3406   3410#  3497#  3499   3501#  3503   3514#  3516#  3535#  3537   3542#  3571#  3573
3575#  3577   3588#  3590#  3688#  3690   3697#  3717#  3719   3724#  3788#  3790   3797#
3822#  3824   3828#  3908#  3910   3914#  3950#  3952   3956#  3987#  3989   3991#  3993
3999#  4004#  4008#  4010   4015#  4062#  4067#  4070#  4072   4076#  4155#  4157   4165#
4169#  4171   4175#  4201#  4203   4206#  4384#  4386   4387#  4389   4401#  4422#
1141#  2410#  2511#  2539#  2562#  2622#  2644#  2694#  2737#  2778#  2837#  3021#  3031#
3103#  3355#  3399#  3476#  3488#  3564#  3630#  3651#  3664#  3736#  3755#  3767#  3831#
4054#  4186#  4395#
4471#
1202#  1203   1216#  1217   1249#  1250   1265#  1266   1282#  1283   1301#  1302   1325#
1326   1341#  1342   1358#  1359   1377#  1378   1407#  1408   1423#  1424   1440#  1441
1459#  1460   1484#  1485   1498#  1499   1514#  1515   1530#  1531   1551#  1552   1564#
1565   1580#  1581   1597#  1598   1616#  1617   1640#  1641   1656#  1657   1673#  1674
1692#  1693   1716#  1717   1732#  1733   1749#  1750   1768#  1769   1792#  1793   1808#
1809   1825#  1826   1844#  1845   1878#  1879   1911#  1912   1943#  1944   1973#  1974
1988#  1989   2027#  2028   2052#  2053   2072#  2073   2092#  2093   2118#  2119   2141#
2142   2161#  2162   2181#  2182   2207#  2208   2231#  2232   2251#  2252   2271#  2272
2296#  2297   2319#  2320   2339#  2340   2359#  2360   2387#  2388   2420#  2421   2438#
2439   2447#  2448   2464#  2465   2487#  2488   2521#  2522   2545#  2546   2568#  2569
```

```
                                2591#  2592    2628#  2629    2650#  2651    2674#  2675    2704#  2705    2726#  2727    2745#
                                2746   2788#   2789   2806#   2807   2846#   2847   2863#   2864   2883#   2884   2905#   2906
                                2919#  2920    2923#  2924    2944#  2945    2948#  2949    2951#  2952    2971#  2972    2988#
                                2989   3043#   3044   3061#   3062   3082#   3083   3110#   3111   3125#   3126   3151#   3152
                                3160#  3161    3201#  3202    3204#  3205    3225#  3226    3233#  3234    3235#  3236    3363#
                                3364   3367#   3368   3405#   3406   3498#   3499   3502#   3503   3536#   3537   3572#   3573
                                3576#  3577    3689#  3690    3718#  3719    3789#  3790    3823#  3824    3903#  3904    3909#
                                3910   3951#   3952   3988#   3989   3992#   3993   4009#   4010   4062#   4063   4071#   4072
                                4156#  4157    4170#  4171    4202#  4203    4385#  4386    4388#  4389    4425#  4426
$$LOCN= 000000                  4118#  4220    4225#  4269    4274#  4314    4335#  4370    4373#  4441
$$RETU= 000000                  4118#  4204#   4208#  4218    4225#  4268    4274#  4313    4335#  4369    4373#  4440
$$RTN1= 000240                  4118#  4204    42.7   4225#   4264   4267    4274#  4312    4335#  4368    4373#  4432    4439
$$RTN2= 000241                  4118#  4209    4219   4225#   4268   4274#   4313   4335#   4369   4373#   4440
$$SRC = 000027                  3261#  4242#
$$TO  = 000000                  1141#  1142    2410#  2413    2511#  2514    2539#  2542    2562#  2565    2622#  2625    2644#
                                2647   2694#   2700   2737#   2743   2778#   2784   2837#   2843   3021#   3024   3031#   3034
                                3103#  3106    3355#  3358    3399#  3402    3476#  3482    3488#  3491    3564#  3567    3630#
                                3631   3632    3651#  3657    3664#  3670    3736#  3737    3738   3755#   3761   3767#   3773
                                3831#  3832    3833   4054#   4060   4186#   4189   4395#   4396
$OFILL  015233                  5014#  5018#   5028   5063#
$1      002102                  1191#  1216
$10     002430                  1341   1346#
$100    005256                  2545   2550#
$101    005320                  2568   2573#
$102    005360                  2591   2597#
$103    005446                  2628   2633#
$104    005510                  2650   2655#
$105    005546                  2674   2679#
$106    005614                  2704   2709#
$107    005642                  2726   2731#
$11     002456                  1358   1363#
$110    005676                  2745   2750#
$111    005770                  2788   2793#
$112    006012                  2806   2811#
$113    006104                  2846   2851#
$114    006122                  2863   2868#
$115    006160                  2883   2888#
$116    006200                  2902#  2919
$117    006216                  2905   2910#
$12     002506                  1377   1382#
$120    006220                  2909   2914#
$121    006234                  2917   2920#
$122    006260                  2923   2934#
$123    006260                  2942#  2965
$124    006324                  2944   2966#
$125    006322                  2948   2963#
$126    006322                  2951   2961#
$127    006336                  2971   2977#
$13     002556                  1407   1412#
$130    006354                  2988   2993#
$131    006470                  3043   3053#
$132    006516                  3061   3073#
$133    006540                  3082   3091#
$134    006610                  3110   3121#
$135    006622                  3125   3131#
$136    006662                  3151   3156#
```

$137	006702	3160	3165#	
$14	002604	1423	1428#	
$140	006732	3176	3179#	
$141	006730	3177#	3257	
$142	007100	3181	3258#	
$143	006770	3199#	3220	
$144	007024	3201	3221#	
$145	007012	3204	3210#	
$146	007022	3209	3217#	
$147	007036	3225	3229#	
$15	002632	1440	1445#	
$150	007062	3228	3249#	
$151	007060	3233	3235	3239#
$152	007062	3238	3247#	
$153	007352	3363	3381#	
$154	007350	3367	3373#	
$155	007352	3372	3379#	
$156	007430	3405	3410#	
$157	007664	3498	3516#	
$16	002662	1459	1464#	
$160	007662	3502	3509#	
$161	007664	3508	3514#	
$162	007726	3536	3542#	
$163	010034	3572	3580#	
$164	010032	3576	3583#	
$165	010034	3582	3588#	
$166	010164	3640	3643#	
$167	010162	3641#	3699	
$17	002720	1484	1489#	
$170	010310	3645	3700#	
$171	010306	3689	3697#	
$172	010354	3718	3724#	
$173	010420	3744	3747#	
$174	010416	3745#	3800	
$175	010544	3749	3801#	
$176	010542	3789	3797#	
$177	010602	3823	3828#	
$2	002126	1202	1207#	
$20	002746	1498	1503#	
$200	010714	3898#	3903	
$201	010730	3901	3904#	
$202	010740	3909	3914#	
$203	011020	3951	3956#	
$204	011100	3988	4004#	
$205	011074	3992	3999#	
$206	011114	4009	4015#	
$207	011222	4062	4067#	
$21	002774	1514	1519#	
$210	011232	4071	4076#	
$211	011556	4204	4217#	
$212	011560	4209	4219#	
$213	011426	4152#	4193	
$214	011532	4179	4181	4194#
$215	011446	4156	4161#	
$216	011454	4160	4165#	
$217	011472	4170	4175#	

```
$22      003022            1530    1535#
$220     011544            4202    4206#
$221     011660            4264    4267#
$222     011660            4268#
$223     011624            4248    4251#
$224     011620            4249#   4259
$225     011650            4253    4260#
$226     011740            4312#
$227     011740            4313#
$23      003060            1551    1556#
$230     011706            4288    4291#
$231     011702            4289#   4306
$232     011732            4293    4307#
$233     011720            4296    4299#
$234     011716            4297#   4303
$235     011730            4301    4304#
$236     012132            4368#
$237     012132            4369#
$24      003100            1564    1569#
$240     012272            4432    4439#
$241     012272            4440#
$242     012134            4382#   4425
$243     012220            4385    4412#
$244     012160            4388    4393#
$245     012166            4392    4401#
$246     012240            4411    4422#
$25      003126            1580    1585#
$26      003154            1597    1602#
$27      003204            1616    1621#
$3       002240            1249    1254#
$30      003242            1640    1645#
$31      003270            1656    1661#
$32      003316            1673    1678#
$33      003346            1692    1697#
$34      003404            1716    1721#
$35      003432            1732    1737#
$36      003460            1749    1754#
$37      003510            1768    1773#
$4       002266            1265    1270#
$40      003546            1792    1797#
$40CAT=  ****** U          4831    4873
$41      003574            1808    1813#
$42      003622            1825    1830#
$43      003652            1844    1849#
$44      003712            1878    1888#
$45      003752            1911    1921#
$46      004012            1943    1955#
$47      004050            1973    1979#
$5       002314            1282    1287#
$50      004072            1988    2002#
$51      004130            2027    2035#
$52      004156            2052    2059#
$53      004204            2072    2079#
$54      004232            2092    2099#
$55      004270            2118    2124#
$56      004316            2141    2148#
```

```
$57     004344          2161    2168#
$6      002344          1301    1306#
$60     004372          2181    2188#
$61     004430          2207    2214#
$62     004456          2231    2238#
$63     004504          2251    2258#
$64     004532          2271    2278#
$65     004570          2296    2302#
$66     004616          2319    2326#
$67     004644          2339    2346#
$7      002402          1325    1330#
$70     004672          2359    2366#
$71     004730          2387    2394#
$72     005010          2420    2425#
$73     005036          2438    2443#
$74     005050          2447    2452#
$75     005076          2464    2469#
$76     005134          2487    2494#
$77     005214          2521    2526#
.       = 015316         912#    916#     925     926#    928#    930#    931#     937     938#    940#    942#    960#     997
                        1070#   1078     1093    1094    1202    1203    1216    1217    1249    1250    1265    1266    1282
                        1283    1301     1302    1325    1326    1341    1342    1358    1359    1377    1378    1407    1408
                        1423    1424     1440    1441    1459    1460    1484    1485    1498    1499    1514    1515    1530
                        1531    1551     1552    1564    1565    1580    1581    1597    1598    1616    1617    1640    1641
                        1656    1657     1673    1674    1692    1693    1716    1717    1732    1733    1749    1750    1768
                        1769    1792     1793    1808    1809    1825    1826    1844    1845    1878    1879    1911    1912
                        1943    1944     1973    1974    1988    1989    2027    2028    2052    2053    2072    2073    2092
                        2093    2118     2119    2141    2142    2161    2162    2181    2182    2207    2208    2231    2232
                        2251    2252     2271    2272    2296    2297    2319    2320    2339    2340    2359    2360    2387
                        2388    2420     2421    2438    2439    2447    2448    2464    2465    2487    2488    2521    2522
                        2545    2546     2568    2569    2591    2592    2628    2629    2650    2651    2674    2675    2704
                        2705    2726     2727    2745    2746    2788    2789    2806    2807    2846    2847    2863    2864
                        2883    2884     2905    2906    2919    2920    2923    2924    2944    2945    2948    2949    2951
                        2952    2971     2972    2988    2989    3043    3044    3061    3062    3082    3083    3110    3111
                        3125    3126     3151    3152    3160    3161    3201    3202    3204    3205    3225    3226    3233
                        3234    3235     3236    3301#   3363    3364    3367    3368    3405    3406    3498    3499    3502
                        3503    3536     3537    3572    3573    3576    3577    3689    3690    3718    3719    3789    3790
                        3823    3824     3903    3904    3909    3910    3951    3952    3988    3989    3992    3993    4009
                        4010    4062     4063    4071    4072    4096#   4156    4157    4170    4171    4202    4203    4339#
                        4352#   4358#    4385    4386    4388    4389    4425    4426    4479    4483    4499    4523    4607
                        4610    4739#    4740    4746    4798#   4857    4920    4921    4987#
.$ASTA= ****** U        4750    4753
.$X     = 001000         937#    942
```

```
BEGIN       1#
BGNHRD      1#
BGNHW       1#
BGNINI      1#
BGNMOD      1#  4114
BGNMSG      1#
BGNSFT      1#
BGNSRV      1#  3934  3969  4318
BGNSUB      1#  1192  1244  1258  1274  1291  1320  1334  1350  1367  1402  1416  1432  1449  1491
          1507  1523  1559  1573  1589  1606  1635  1649  1665  1681  1711  1725  1741  1758  1787
          1801  1817  1834  1874  1906  1939  1983  2043  2063  2083  2132  2152  2172  2222  2242
          2262  2310  2330  2350  2403  2430  2456  2503  2530  2554  2614  2637  2680  2712  2767
          2797  2825  3010  3057  3076  3094  3337  3384  3452  3532
BGNSW       1#
BRESET      1#  1168  1297  1373  1455  1612  1688  1764  184C  1885  1918  1952  1999  2801  3411
          3882  4077
CALL        1#  1140  2410  2511  2539  2562  2622  2644  2693  2736  2777  2836  3021  3031  3103
          3355  3399  3475  3488  3564  3629  3650  3663  3735  3754  3766  3830  4053  4186  4394
CASE        1#
CkLOOP      1#  1850  1890  1923  1956  2004
CLRVEC      1#  1217  3415  3599
COMMEN    795#
DECR        1#
DECRU       1#
DEFAUL      1#
DEVREG      1#
DEVTYP      1#
DISPAT      1#
ELSE        1#  2908  3208  3227  3237  3371  3507  3581  4159  4391  4410
END         1#
ENDCLN      1#
ENDCOM    795#
ENDDEC      1#
ENDDO       1#  2964  3219
ENDHRD      1#
ENDHW       1#
ENDIF       1#  1206  1253  1269  1286  1305  1329  1345  1362  1381  1411  1427  1444  1463  1488
          1502  1518  1534  1555  1568  1584  1601  1620  1644  1660  1677  1696  1720  1736  1753
          1772  1796  1812  1829  1848  1887  1920  1954  1978  2001  2034  2058  2078  2098  2123
          2147  2167  2187  2213  2237  2257  2277  2301  2325  2345  2365  2393  2424  2442  2451
          2468  2493  2525  2549  2572  2596  2632  2654  2678  2708  2730  2749  2792  2810  2850
          2867  2887  2913  2933  2960  2962  2976  2992  3052  3072  3090  3120  3130  3155  3164
          3216  3246  3248  3378  3380  3409  3513  3515  3541  3587  3589  3696  3723  3796  3827
          3913  3955  3998  4003  4014  4066  4075  4164  4174  4205  4400  4421
ENDINC      1#  3256  3698  3799  4258  4302  4305
ENDINI      1#
ENDLOO      1#  4192
ENDMOD      1#
ENDMSG      1#
ENDRTN      1#  4216  4266  4311  4367  4438
ENDSEL      1#
ENDSFT      1#
ENDSRV      1#  3962  4022  4329
ENDSUB      1#  1208  1255  1271  1288  1307  1331  1347  1364  1383  1413  1429  1446  1465  1504
          1520  1536  1570  1586  1603  1622  1646  1662  1679  1698  1722  1738  1755  1774  1798
          1814  1831  1851  1891  1924  1957  2005  2060  2080  2100  2149  2169  2189  2239  2259
```

```
            2279    2327    2347    2367    2427    2453    2470    2527    2551    2574    2634    2656    2710    2751    2795
            2812    2852    3054    3074    3092    3132    3382    3413    3523    3595
ENDSW          1#
ENDTST         1#   1228    1308    1384    1466    1537    1623    1699    1775    1852    1892    1925    1958    2006    2101
            2190    2280    2368    2471    2577    2657    2752    2813    2869    2999    3136    3304    3426    3609    3705
            3805    4025    4086
EQUALS         1#
ERRDF          1#   1204
ERRHRD         1#   1251    1267    1284    1303    1327    1343    1360    1379    1409    1425    1442    1461    1500    1516
            1532    1566    1582    1599    1618    1642    1658    1675    1694    1718    1734    1751    1770    1794    1810
            1827    1846    1882    1915    1945    1991    2054    2074    2094    2143    2163    2183    2233    2253    2273
            2321    2341    2361    2422    2440    2449    2466    2523    2547    2570    2630    2652    2706    2728    2747
            2790    2808    2848    2865    2926    2954    2973    2990    3045    3064    3085    3112    3127    3244    3369
            3376    3407    3504    3511    3578    3585    3691    3791    3911    4064    4073
ERROR        689#   1205    1252    1268    1285    1304    1328    1344    1361    1380    1410    1426    1443    1462    1501
            1517    1533    1567    1583    1600    1619    1643    1659    1676    1695    1719    1735    1752    1771    1795
            1811    1828    1847    1883    1916    1946    1992    2055    2075    2095    2144    2164    2184    2234    2254
            2274    2322    2342    2362    2423    2441    2450    2467    2524    2548    2571    2631    2653    2707    2729
            2748    2791    2809    2849    2866    2927    2955    2974    2991    3046    3065    3086    3113    3128    3245
            3370    3377    3408    3505    3512    3579    3586    3692    3792    3912    4065    4074
ESCAPE       795#
EXIF           1#
EXIFB          1#   4177
EXIT           1#   1485    1552    1975    2031    2120    2210    2298    2390    2490    2575    2593    2675    2884    2930
            2957    2995    3049    3069    3088    3117    3133    3152    3161    3267    3538    3693    3720    3793    3824
            3921    4079
GETPRI       795#
GETSWR       795#   1123#
GPHARD         1#
GPRMA          1#
GPRMD          1#
GPRML          1#
HEADER         1#
IF             1#   1200    1247    1263    1280    1299    1323    1339    1356    1375    1405    1421    1438    1457    1482
            1496    1512    1528    1549    1562    1578    1595    1614    1638    1654    1671    1690    1714    1730    1747
            1766    1790    1806    1823    1842    1876    1909    1941    1971    1986    2025    2050    2070    2090    2116
            2139    2159    2179    2205    2229    2249    2269    2294    2317    2337    2357    2385    2418    2436    2445
            2462    2485    2519    2543    2566    2589    2626    2648    2672    2724    2804    2861    2881    2903    2921
            2946    2949    2969    2986    3041    3059    3080    3108    3123    3149    3158    3202    3223    3231    3361
            3365    3403    3496    3500    3534    3570    3574    3687    3716    3787    3821    3907    3949    3986    3990
            4007    4154    4383    4386
IFB            1#   4069    4168    4200
IFCOND         1#
IF.ERR         1#   2703    2744    2787    2845    4061
IF.NO.         1#
INCR           1#   3174    3638    3742    4246    4294
INCRU          1#   4286
INLINE         1#
LASTAD         1#
LEAVE          1#
LET            1#   1137    1144    1147    1149    1152    1154    1157    1160    1163    1166    1179    1183    1184    1185
            1186    1187    1188    1193    1195    1209    1211    1220    1221    1223    1224    1245    1259    1260    1275
            1277    1292    1294    1321    1335    1336    1351    1353    1368    1370    1403    1417    1418    1433    1435
            1450    1452    1486    1492    1493    1508    1510    1524    1525    1553    1560    1574    1575    1590    1592
            1607    1609    1636    1650    1651    1666    1668    1682    1685    1712    1726    1727    1742    1744    1759
            1761    1788    1802    1803    1818    1820    1835    1837    1875    1907    1940    1976    1984    2032    2044
```

```
        2047    2064    2067    2084    2087    2121    2133    2136    2153    2156    2173    2176    2211    2223    2226
        2243    2246    2263    2266    2299    2311    2314    2331    2334    2351    2354    2391    2404    2406    2415
        2431    2434    2457    2460    2491    2504    2507    2516    2531    2535    2555    2558    2594    2608    2610
        2615    2618    2638    2640    2676    2681    2688    2713    2719    2764    2768    2771    2798    2823    2826
        2831    2858    2885    2889    2891    2893    2899    2906    2911    2928    2931    2958    2983    3011    3017
        3027    3037    3050    3058    3070    3077    3095    3099    3118    3153    3162    3167    3169    3171    3182
        3185    3188    3191    3194    3196    3206    3212    3214    3251    3253    3259    3264    3321    3325    3328
        3332    3333    3334    3335    3336    3338    3340    3350    3385    3387    3390    3418    3419    3421    3422
        3446    3447    3448    3449    3450    3453    3454    3457    3460    3470    3484    3492    3518    3521    3533
        3539    3544    3547    3555    3558    3560    3591    3593    3597    3602    3603    3605    3606    3619    3634
        3660    3672    3680    3683    3694    3703    3721    3726    3739    3764    3776    3780    3783    3794    3825
        3854    3857    3859    3863    3865    3869    3872    3875    3879    3885    3889    3892    3916    3918    3939
        3942    3946    3953    3974    3977    3981    3994    3996    4001    4011    4039    4043    4048    4051    4141
        4143    4145    4157    4162    4172    4190    4238    4240    4254    4256    4261    4284    4327    4389    4398
        4402    4404    4406    4408    4413    4415    4417    4419    4427    4429
LOCAL     1#
LOOP      1#    4151
MSG    1170#   1172    1234#   1236    1471#   1473    1542#   1544    2016#   2018    2107#   2109    2196#   2198    2286#
       2288    2374#   2376    2476#   2478    2662#   2664    2754#   2756    2871#   2873    3140#   3142    3309#   3311
       3434#   3436    3611#   3613    3811#   3813    4030#   4032
MULT    795#
NEWTST  795#   1170    1234    1312    1389    1471    1542    1628    1704    1780    1863    1896    1929    1963    2016
       2107    2196    2286    2374    2476    2582    2662    2754    2815    2871    3003    3140    3309    3434    3611
       3710    3811    4030    4088
NOLOCA    1#
POINTE    1#
POP     795#   1225    3423    3607    4308    4508    4509    4792    4793    4974
PRINTB    1#
PUSH    795#   1218    3416    3600    4281    4489    4495    4753    4755    4776    4933
REPEAT    1#   1190    2901    3897    4381
REPORT    1#    795#
RETURN    1#   4203    4207    4263    4431
ROUTIN    1#   4116    4223    4272    4333    4371
SAVR14    1#
SCOPE   690#   1176    1239    1315    1392    1479    1546    1631    1707    1783    1866    1899    1932    1966    2021
       2112    2201    2290    2381    2481    2585    2667    2759    2818    2876    3006    3146    3317    3440    3615
       3713    3815    4035    4091    4454
SELECT    1#
SETPRI  795#   1396    2397    2497    2600    3344    3393    3464    3551    3624    3730    3849
SETTRA 5088#   5097    5098    5099    5100    5102    5104    5105    5106
SETUP   795#   1072
SETVEC    1#   1182    3331    3445
SKIP    795#   1487    1554    1977    2033    2122    2212    2300    2392    2492    2576    2595    2677    2886    2932
       2959    2996    3051    3071    3089    3119    3134    3154    3163    3268    3540    3695    3722    3795    3826
       3922    4080
SLASH   795#
SPACE   795#
STARS   795#    811     813     831     833     851     853     873     875     894     909     923     934     936     943
        956     997    1000    1170    1175    1229    1231    1234    1238    1311    1312    1314    1388    1389    1391
       1470    1471    1478    1541    1542    1545    1627    1628    1630    1703    1704    1706    1779    1780    1782
       1857    1860    1863    1865    1896    1898    1929    1931    1962    1963    1965    2008    2012    2016    2020
       2106    2107    2111    2195    2196    2200    2285    2286    2289    2373    2374    2380    2475    2476    2480
       2581    2582    2584    2661    2662    2666    2753    2754    2758    2814    2815    2817    2870    2871    2875
       3002    3003    3005    3139    3140    3145    3308    3309    3316    3433    3434    3439    3610    3611    3614
       3709    3710    3712    3810    3811    3814    4029    4030    4034    4088    4090    4115    4136    4222    4236
       4271    4280    4317    4324    4335    4373    4380    4446    4485    4501    4530    4609    4612    4680    4709
```

```
        4748    4805    4859    4923    4990    5067
STRUCT     1#
SWRSU    795#  1095#
TRMTRP  5088#
TYPBIN   795#
TYPDEC   795#  4109    4466
TYPNAM   795#  1116
TYPNUM   795#
TYPOCS   795#
TYPOCT   795#  4097    4103    4340    4347    4353    4359    4365    4630
TYPTXT   795#  4093    4099    4105    4336    4342    4349    4355    4361
UNTIL      1#  1214    2915    3899    4423
UNTILB     1#
WAITMS     1#  2409    2510    2538    2561    2621    2643    3020    3030    3102    3354    3398    3487    3563    4185
WHILE      1#  2941    3198
WHILEB     1#
$ADDON     1#  1191    1192    1201    1203    1248    1250    1264    1266    1281    1283    1300    1302    1324    1326
        1340    1342    1357    1359    1376    1378    1406    1408    1422    1424    1439    1441    1458    1460    1483
        1485    1497    1499    1513    1515    1529    1531    1550    1552    1563    1565    1579    1581    1596    1598
        1615    1617    1639    1641    1655    1657    1672    1674    1691    1693    1715    1717    1731    1733    1748
        1750    1767    1769    1791    1793    1807    1809    1824    1826    1843    1845    1877    1879    1910    1912
        1942    1944    1972    1974    1987    1989    2026    2028    2051    2053    2071    2073    2091    2093    2117
        2119    2140    2142    2160    2162    2180    2182    2206    2208    2230    2232    2250    2252    2270    2272
        2295    2297    2318    2320    2338    2340    2358    2360    2386    2388    2419    2421    2437    2439    2446
        2448    2463    2465    2486    2488    2520    2522    2544    2546    2567    2569    2590    2592    2627    2629
        2649    2651    2673    2675    2704    2705    2725    2727    2745    2746    2788    2789    2805    2807    2846
        2847    2862    2864    2882    2884    2902    2903    2904    2906    2911    2921    2922    2924    2942    2943
        2945    2947    2949    2950    2952    2965    2970    2972    2987    2989    3042    3044    3060    3062    3081
        3083    3109    3111    3124    3126    3150    3152    3159    3161    3175    3177    3178    3182    3199    3200
        3202    3203    3205    3211    3220    3224    3226    3230    3232    3236    3240    3362    3364    3366    3368
        3374    3404    3406    3497    3499    3501    3503    3510    3535    3537    3571    3573    3575    3577    3584
        3639    3641    3642    3646    3688    3690    3717    3719    3743    3745    3746    3750    3788    3790    3822
        3824    3898    3899    3905    3908    3910    3950    3952    3987    3989    3991    3993    4008    4010    4062
        4063    4070    4072    4118    4152    4153    4155    4157    4162    4169    4171    4193    4201    4203    4225
        4247    4249    4250    4254    4274    4287    4289    4290    4294    4295    4297    4298    4302    4335    4373
        4382    4383    4384    4386    4387    4389    4394    4413
$AND       1#  3232
$BRANC     1#  1202    1216    1249    1265    1282    1301    1325    1341    1358    1377    1407    1423    1440    1459
        1484    1498    1514    1530    1551    1564    1580    1597    1616    1640    1656    1673    1692    1716    1732
        1749    1768    1792    1808    1825    1844    1878    1911    1943    1973    1988    2027    2052    2072    2092
        2118    2141    2161    2181    2207    2231    2251    2271    2296    2319    2339    2359    2387    2420    2438
        2447    2464    2487    2521    2545    2568    2591    2628    2650    2674    2704    2726    2745    2788    2806
        2846    2863    2883    2905    2909    2917    2919    2923    2944    2948    2951    2965    2971    2988    3043
        3061    3082    3110    3125    3151    3160    3176    3181    3201    3204    3209    3220    3225    3228    3233
        3235    3238    3257    3363    3367    3372    3405    3498    3502    3508    3536    3572    3576    3582    3640
        3645    3689    3699    3718    3744    3749    3789    3800    3823    3901    3903    3909    3951    3988    3992
        4009    4062    4071    4156    4160    4170    4179    4181    4193    4202    4204    4209    4248    4253    4259
        4264    4288    4293    4296    4301    4303    4306    4385    4388    4392    4411    4425    4432
$BRCOD     1#  2916    3180    3644    3748    3900    4178    4180    4252    4292    4300
$CALL      1#  1141    2410    2511    2539    2562    2622    2644    2694    2737    2778    2837    3021    3031    3103
        3355    3399    3476    3488    3564    3630    3651    3664    3736    3755    3767    3831    4054    4186    4395
$CHECK     1#  1201    1248    1264    1281    1300    1324    1340    1357    1376    1406    1422    1439    1458    1483
        1497    1513    1529    1550    1563    1579    1596    1615    1639    1655    1672    1691    1715    1731    1748
        1767    1791    1807    1824    1843    1877    1910    1942    1972    1987    2026    2051    2071    2091    2117
        2140    2160    2180    2206    2230    2250    2270    2295    2318    2338    2358    2386    2419    2437    2446
        2463    2486    2520    2544    2567    2590    2627    2649    2673    2725    2805    2862    2882    2904    2922
```

```
                2943    2947    2950    2970    2987    3042    3060    3081    3109    3124    3150    3159    3200    3203    3224
                3232    3362    3366    3404    3497    3501    3535    3571    3575    3688    3717    3788    3822    3908    3950
                3987    3991    4008    4070    4155    4169    4201    4384    4387
$CHK1        1#  1138    1145    1148    1150    1153    1167    1180    1183    1184    1185    1186    1187    1189    1193
                1196    1220    1223    1224    1245    1259    1275    1292    1321    1335    1351    1368    1403    1417    1433
                1450    1486    1492    1508    1524    1553    1560    1574    1590    1607    1636    1650    1666    1682    1712
                1726    1742    1759    1788    1802    1818    1835    1875    1907    1940    1976    1984    2032    2044    2064
                2084    2121    2133    2153    2173    2211    2223    2243    2263    2299    2311    2331    2351    2391    2404
                2416    2431    2457    2491    2504    2517    2531    2555    2594    2611    2615    2638    2676    2681    2689
                2713    2720    2768    2772    2798    2826    2832    2859    2885    2892    2894    2900    2907    2929    2931
                2958    2984    3011    3018    3028    3038    3050    3058    3070    3077    3095    3100    3118    3153    3162
                3168    3170    3175    3178    3183    3186    3189    3192    3195    3197    3207    3252    3254    3265    3322
                3326    3332    3333    3334    3335    3336    3338    3385    3388    3418    3421    3422    3446    3447    3448
                3449    3450    3453    3455    3471    3493    3533    3539    3545    3598    3602    3605    3606    3635    3639
                3642    3661    3673    3694    3721    3740    3743    3746    3765    3777    3794    3825    3855    3858    3860
                3864    3866    3870    3873    3876    3880    3947    3995    3997    4044    4052    4142    4144    4146    4158
                4163    4173    4239    4247    4250    4262    4285    4287    4290    4295    4298    4390    4399    4403    4405
                4407    4409    4414    4428
$CKOP2       1#  1155    1158    1161    1164    1210    1212    1221    1261    1278    1295    1337    1354    1371    1419
                1436    1453    1494    1511    1526    1576    1593    1610    1652    1669    1686    1728    1745    1762    1804
                1821    1838    2048    2068    2088    2137    2157    2177    2227    2247    2267    2315    2335    2355    2407
                2435    2461    2508    2536    2559    2609    2619    2641    2765    2824    2890    2912    3172    3213    3215
                3260    3329    3341    3351    3391    3419    3458    3461    3485    3519    3522    3548    3556    3559    3561
                3592    3594    3603    3620    3681    3684    3704    3727    3781    3784    3886    3890    3893    3917    3919
                3940    3943    3954    3975    3978    3982    4002    4012    4040    4049    4191    4241    4255    4257    4328
                4416    4418    4420    4430
$CKR6        1#  3261    4242
$CMND        1#  1201    1248    1264    1281    1300    1324    1340    1357    1376    1406    1422    1439    1458    1483
                1497    1513    1529    1550    1563    1579    1596    1615    1639    1655    1672    1691    1715    1731    1748
                1767    1791    1807    1824    1843    1877    1910    1942    1972    1987    2026    2051    2071    2091    2117
                2140    2160    2180    2206    2230    2250    2270    2295    2318    2338    2358    2386    2419    2437    2446
                2463    2486    2520    2544    2567    2590    2627    2649    2673    2725    2805    2862    2882    2904    2922
                2943    2947    2950    2970    2987    3042    3060    3081    3109    3124    3150    3159    3200    3203    3224
                3232    3234    3362    3366    3404    3497    3501    3535    3571    3575    3688    3717    3788    3822    3908
                3950    3987    3991    4008    4070    4155    4169    4201    4384    4387
$COMPA       1#  1201    1248    1264    1281    1300    1324    1340    1357    1376    1406    1422    1439    1458    1483
                1497    1513    1529    1550    1563    1579    1596    1615    1639    1655    1672    1691    1715    1731    1748
                1767    1791    1807    1824    1843    1877    1910    1942    1972    1987    2026    2051    2071    2091    2117
                2140    2160    2180    2206    2230    2250    2270    2295    2318    2338    2358    2386    2419    2437    2446
                2463    2486    2520    2544    2567    2590    2627    2649    2673    2704    2725    2745    2788    2805    2846
                2862    2882    2904    2922    2943    2947    2950    2970    2987    3042    3060    3081    3109    3124    3150
                3159    3175    3200    3203    3224    3232    3362    3366    3404    3497    3501    3535    3571    3575    3639
                3688    3717    3743    3788    3822    3908    3950    3987    3991    4008    4062    4070    4155    4169    4201
                4247    4287    4295    4384    4387
$COUNT       1#  1141    2410    2511    2539    2562    2622    2644    2694    2737    2778    2837    3021    3031    3103
                3355    3399    3476    3488    3564    3630    3651    3664    3736    3755    3767    3831    4054    4186    4395
$DO          1#  2943    3200
$ELSE        1#
$ERRMS       1#
$EXIFA       1#
$EXIFO       1#
$EXIF2       1#  4178
$EXIF3       1#
$GENBR       1#  1202    1216    1249    1265    1282    1301    1325    1341    1358    1377    1407    1423    1440    1459
                1484    1498    1514    1530    1551    1564    1580    1597    1616    1640    1656    1673    1692    1716    1732
                1749    1768    1792    1808    1825    1844    1878    1911    1943    1973    1988    2027    2052    2072    2092
```

	2118	2141	2161	2181	2207	2231	2251	2271	2296	2319	2339	2359	2387	2420	2438
	2447	2464	2487	2521	2545	2568	2591	2628	2650	2674	2704	2726	2745	2788	2806
	2846	2863	2883	2905	2909	2917	2919	2923	2944	2948	2951	2965	2971	2988	3043
	3061	3082	3110	3125	3151	3160	3176	3181	3201	3204	3209	3220	3225	3228	3233
	3235	3238	3257	3363	3367	3372	3405	3498	3502	3508	3536	3572	3576	3582	3640
	3645	3689	3699	3718	3744	3749	3789	3800	3823	3901	3903	3909	3951	3988	3992
	4009	4062	4071	4156	4160	4170	4179	4181	4193	4202	4204	4209	4248	4253	4259
	4264	4288	4293	4296	4301	4303	4306	4385	4388	4392	4411	4425	4432		
$GENTA	1#	1191	1207	1254	1270	1287	1306	1330	1346	1363	1382	1412	1428	1445	1464
	1489	1503	1519	1535	1556	1569	1585	1602	1621	1645	1661	1678	1697	1721	1737
	1754	1773	1797	1813	1830	1849	1888	1921	1955	1979	2002	2035	2059	2079	2099
	2124	2148	2168	2188	2214	2238	2258	2278	2302	2326	2346	2366	2394	2425	2443
	2452	2469	2494	2526	2550	2573	2597	2633	2655	2679	2709	2731	2750	2793	2811
	2851	2868	2888	2902	2910	2914	2920	2934	2942	2961	2963	2966	2977	2993	3053
	3073	3091	3121	3131	3156	3165	3177	3179	3199	3210	3217	3221	3229	3239	3247
	3249	3258	3373	3379	3381	3410	3509	3514	3516	3542	3583	3588	3590	3641	3643
	3697	3700	3724	3745	3747	3797	3801	3828	3898	3904	3914	3956	3999	4004	4015
	4067	4076	4152	4161	4165	4175	4194	4206	4217	4219	4249	4251	4260	4267	4268
	4289	4291	4297	4299	4304	4307	4312	4313	4368	4369	4382	4393	4401	4412	4422
	4439	4440													
$IF	1#	1201	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458	1483
	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731	1748
	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091	2117
	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437	2446
	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904	2922
	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150	3159	3203	3224	3232	3362
	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788	3822	3908	3950	3987	3991
	4008	4070	4155	4169	4201	4384	4387								
$IFCOD	1#	1201	1215	1248	1264	1281	1300	1324	1340	1357	1376	1406	1422	1439	1458
	1483	1497	1513	1529	1550	1563	1579	1596	1615	1639	1655	1672	1691	1715	1731
	1748	1767	1791	1807	1824	1843	1877	1910	1942	1972	1987	2026	2051	2071	2091
	2117	2140	2160	2180	2206	2230	2250	2270	2295	2318	2338	2358	2386	2419	2437
	2446	2463	2486	2520	2544	2567	2590	2627	2649	2673	2725	2805	2862	2882	2904
	2918	2922	2943	2947	2950	2970	2987	3042	3060	3081	3109	3124	3150	3159	3200
	3203	3224	3232	3234	3362	3366	3404	3497	3501	3535	3571	3575	3688	3717	3788
	3822	3902	3908	3950	3987	3991	4008	4070	4155	4169	4201	4384	4387	4424	
$IFCON	1#	2704	2745	2788	2846	4062									
$IFOPR	1#	1202	1216	1249	1265	1282	1301	1325	1341	1358	1377	1407	1423	1440	1459
	1484	1498	1514	1530	1551	1564	1580	1597	1616	1640	1656	1673	1692	1716	1732
	1749	1768	1792	1808	1825	1844	1878	1911	1943	1973	1988	2027	2052	2072	2092
	2118	2141	2161	2181	2207	2231	2251	2271	2296	2319	2339	2359	2387	2420	2438
	2447	2464	2487	2521	2545	2568	2591	2628	2650	2674	2704	2726	2745	2788	2806
	2846	2863	2883	2905	2919	2923	2944	2948	2951	2971	2988	3043	3061	3082	3110
	3125	3151	3160	3201	3204	3225	3233	3235	3363	3367	3405	3498	3502	3536	3572
	3576	3689	3718	3789	3823	3903	3909	3951	3988	3992	4009	4062	4071	4156	4170
	4202	4385	4388	4425											
$LET	1#	1138	1145	1148	1150	1153	1155	1158	1161	1164	1167	1180	1183	1184	1185
	1186	1187	1189	1193	1196	1210	1212	1220	1221	1223	1224	1245	1259	1261	1275
	1278	1292	1295	1321	1335	1337	1351	1354	1368	1371	1403	1417	1419	1433	1436
	1450	1453	1486	1492	1494	1508	1511	1524	1526	1553	1560	1574	1576	1590	1593
	1607	1610	1636	1650	1652	1666	1669	1682	1686	1712	1726	1728	1742	1745	1759
	1762	1788	1802	1804	1818	1821	1835	1838	1875	1907	1940	1976	1984	2032	2044
	2048	2064	2068	2084	2088	2121	2133	2137	2153	2157	2173	2177	2211	2223	2227
	2243	2247	2263	2267	2299	2311	2315	2331	2335	2351	2355	2391	2404	2407	2416
	2431	2435	2457	2461	2491	2504	2508	2517	2531	2536	2555	2559	2594	2609	2611
	2615	2619	2638	2641	2676	2681	2689	2713	2720	2765	2768	2772	2798	2824	2826

```
                2832    2859    2885    2890    2892    2894    2900    2907    2912    2929    2931    2958    2984    3011    3018
                3028    3038    3050    3058    3070    3077    3095    3100    3118    3153    3162    3168    3170    3172    3183
                3186    3189    3192    3195    3197    3207    3213    3215    3252    3254    3260    3265    3322    3326    3329
                3332    3333    3334    3335    3336    3338    3341    3351    3385    3388    3391    3418    3419    3421    3422
                3446    3447    3448    3449    3450    3453    3455    3458    3461    3471    3485    3493    3519    3522    3533
                3539    3545    3548    3556    3559    3561    3592    3594    3598    3602    3603    3605    3606    3620    3635
                3661    3673    3681    3684    3694    3704    3721    3727    3740    3765    3777    3781    3784    3794    3825
                3855    3858    3860    3864    3866    3870    3873    3876    3880    3886    3890    3893    3917    3919    3940
                3943    3947    3954    3975    3978    3982    3995    3997    4002    4012    4040    4044    4043    4052    4142
                4144    4146    4158    4163    4173    4191    4239    4241    4255    4257    4262    4285    4328    4390    4399
                4403    4405    4407    4409    4414    4416    4418    4420    4428    4430
$LPCNT       1#     3175    3639    3743    4247    4287    4295
$OPADD       1#     1156    1159    1162    1165    1210    1213    1222    2912    3178    3213    3215    3329    3420    3604
                3642    3746    3940    3975    4002    4250    4290    4298    4328    4418    4420    4430
$OPAND       1#     3261    4242
$OPCD1       1#     1156    1159    1162    1165    1210    1213    1222    1261    1278    1295    1337    1354    1371    1419
                1436    1453    1494    1511    1526    1576    1593    1610    1652    1669    1686    1728    1745    1762    1804
                1821    1838    2048    2068    2088    2137    2157    2177    2227    2247    2267    2315    2335    2355    2407
                2435    2461    2508    2536    2559    2609    2619    2641    2765    2824    2890    2912    3172    3178    3213
                3215    3261    3329    3341    3351    3391    3420    3458    3461    3485    3519    3522    3548    3556    3559
                3561    3592    3594    3604    3620    3642    3682    3684    3704    3727    3746    3782    3784    3886    3890
                3893    3917    3919    3940    3944    3954    3975    3979    3983    4002    4012    4040    4049    4191    4242
                4250    4255    4257    4262    4290    4298    4328    4416    4418    4420    4430
$OPCD2       1#     4255    4262    4416
$OPCOD       1#     1156    1159    1162    1165    1210    1213    1222    1261    1278    1295    1337    1354    1371    1419
                1436    1453    1494    1511    1526    1576    1593    1610    1652    1669    1686    1728    1745    1762    1804
                1821    1838    2048    2068    2088    2137    2157    2177    2227    2247    2267    2315    2335    2355    2407
                2435    2461    2508    2536    2559    2609    2619    2641    2765    2824    2890    2912    3172    3178    3213
                3215    3261    3329    3341    3351    3391    3420    3458    3461    3485    3519    3522    3548    3556    3559
                3561    3592    3594    3604    3620    3642    3682    3684    3704    3727    3746    3782    3784    3886    3890
                3893    3917    3919    3940    3944    3954    3975    3979    3983    4002    4012    4040    4049    4191    4242
                4250    4255    4257    4262    4290    4298    4328    4416    4418    4420    4430
$OPCOM       1#     4262
$OPDEF       1#     1138    1141    1145    1148    1150    1153    1155    1156    1158    1159    1161    1162    1164    1165
                1167    1180    1183    1184    1185    1186    1187    1189    1193    1196    1201    1202    1210    1212    1213
                1215    1216    1220    1221    1222    1223    1224    1245    1248    1249    1259    1261    1264    1265    1275
                1278    1281    1282    1292    1295    1300    1301    1321    1324    1325    1335    1337    1340    1341    1351
                1354    1357    1358    1368    1371    1376    1377    1403    1406    1407    1417    1419    1422    1423    1433
                1436    1439    1440    1450    1453    1458    1459    1483    1484    1486    1492    1494    1497    1498    1508
                1511    1513    1514    1524    1526    1529    1530    1550    1551    1553    1560    1563    1564    1574    1576
                1579    1580    1590    1593    1596    1597    1607    1610    1615    1616    1636    1639    1640    1650    1652
                1655    1656    1666    1669    1672    1673    1682    1686    1691    1692    1712    1715    1716    1726    1728
                1731    1732    1742    1745    1748    1749    1759    1762    1767    1768    1788    1791    1792    1802    1804
                1807    1808    1818    1821    1824    1825    1835    1838    1843    1844    1875    1877    1878    1907    1910
                1911    1940    1942    1943    1972    1973    1976    1984    1987    1988    2026    2027    2032    2044    2048
                2051    2052    2064    2068    2071    2072    2084    2088    2091    2092    2117    2118    2121    2133    2137
                2140    2141    2153    2157    2160    2161    2173    2177    2180    2181    2206    2207    2211    2223    2227
                2230    2231    2243    2247    2250    2251    2263    2267    2270    2271    2295    2296    2299    2311    2315
                2318    2319    2331    2335    2338    2339    2351    2355    2358    2359    2386    2387    2391    2404    2407
                2410    2411    2412    2413    2416    2419    2420    2431    2435    2437    2438    2446    2447    2457    2461
                2463    2464    2486    2487    2491    2504    2508    2511    2512    2513    2514    2517    2520    2521    2531
                2536    2539    2540    2541    2542    2544    2545    2555    2559    2562    2563    2564    2565    2567    2568
                2590    2591    2594    2609    2611    2615    2619    2622    2623    2624    2625    2627    2628    2638    2641
                2644    2645    2646    2647    2649    2650    2673    2674    2676    2681    2689    2694    2695    2696    2697
                2698    2699    2700    2704    2713    2720    2725    2726    2737    2738    2739    2740    2741    2742    2743
                2745    2765    2768    2772    2778    2779    2780    2781    2782    2783    2784    2788    2798    2805    2806
```

```
                2824    2826    2832    2837    2838    2839    2840    2841    2842    2843    2846    2859    2862    2863    2882
                2883    2885    2890    2892    2894    2900    2904    2905    2907    2909    2912    2916    2917    2918    2919
                2922    2923    2929    2931    2943    2944    2947    2948    2950    2951    2958    2965    2970    2971    2984
                2987    2988    3011    3018    3021    3022    3023    3024    3028    3031    3032    3033    3034    3038    3042
                3043    3050    3058    3060    3061    3070    3077    3081    3082    3095    3100    3103    3104    3105    3106
                3109    3110    3118    3124    3125    3150    3151    3153    3159    3160    3162    3168    3170    3172    3175
                3176    3178    3180    3181    3183    3186    3189    3192    3195    3197    3200    3201    3203    3204    3207
                3209    3213    3215    3220    3224    3225    3228    3232    3233    3234    3235    3238    3252    3254    3257
                3260    3261    3262    3263    3265    3322    3326    3329    3332    3333    3334    3335    3336    3338    3341
                3351    3355    3356    3357    3358    3362    3363    3366    3367    3372    3385    3388    3391    3399    3400
                3401    3402    3404    3405    3418    3419    3420    3421    3422    3446    3447    3448    3449    3450    3453
                3455    3458    3461    3471    3476    3477    3478    3479    3480    3481    3482    3485    3488    3489    3490
                3491    3493    3497    3498    3501    3502    3508    3519    3522    3533    3535    3536    3539    3545    3548
                3556    3559    3561    3564    3565    3566    3567    3571    3572    3575    3576    3582    3592    3594    3598
                3602    3603    3604    3605    3606    3620    3630    3631    3632    3635    3639    3640    3642    3644    3645
                3651    3652    3653    3654    3655    3656    3657    3661    3664    3665    3666    3667    3668    3669    3670
                3673    3681    3682    3684    3688    3689    3694    3699    3704    3717    3718    3721    3727    3736    3737
                3738    3740    3743    3744    3746    3748    3749    3755    3756    3757    3758    3759    3760    3761    3765
                3767    3768    3769    3770    3771    3772    3773    3777    3781    3782    3784    3788    3789    3794    3800
                3822    3823    3825    3831    3832    3833    3855    3858    3860    3864    3866    3870    3873    3876    3880
                3886    3890    3893    3900    3901    3902    3903    3908    3909    3917    3919    3940    3943    3944    3947
                3950    3951    3954    3975    3978    3979    3982    3983    3987    3988    3991    3992    3995    3997    4002
                4008    4009    4012    4040    4044    4049    4052    4054    4055    4056    4057    4058    4059    4060    4062
                4070    4071    4142    4144    4146    4155    4156    4158    4160    4163    4169    4170    4173    4178    4179
                4180    4181    4186    4187    4188    4189    4191    4193    4201    4202    4204    4208    4209    4218    4220
                4239    4241    4242    4243    4244    4247    4248    4250    4252    4253    4255    4257    4259    4262    4264
                4269    4285    4287    4288    4290    4292    4293    4295    4296    4298    4300    4301    4303    4306    4314
                4328    4370    4384    4385    4387    4388    4390    4392    4395    4399    4403    4405    4407    4409    4411
                4414    4416    4418    4420    4424    4425    4428    4430    4432    4441

$OPEQU          1#
$OPNAN          1#
$OPNEG          1#
$OPNOR          1#
$OPNOT          1#      1278    1354    1436    1494    1526    1593    1669    1745    1821    2048    2088    2137    2177    2227
                2267    2315    2355    2407    2461    2508    2559    2609    2641    3341    3391    3461    3519    3522    3548
                3556    3592    3594    3682    3684    3704    3727    3782    3784    3917    3919    3944    3954    3979    3983
                4012
$OPOR           1#      1261    1295    1337    1371    1419    1453    1511    1576    1610    1652    1686    1728    1762    1804
                1838    2068    2157    2247    2335    2435    2536    2619    2765    2824    2890    3172    3351    3458    3485
                3559    3561    3620    3866    3890    3893    4040    4049    4257
$OPROT          1#      4416
$OPRO           1#      1138    1145    1148    1150    1153    1167    1180    1183    1184    1185    1186    1187    1189    1193
                1196    1220    1223    1224    1245    1259    1275    1292    1321    1335    1351    1368    1403    1417    1433
                1450    1486    1492    1508    1524    1553    1560    1574    1590    1607    1636    1650    1666    1682    1712
                1726    1742    1759    1788    1802    1818    1835    1875    1907    1940    1976    1984    2032    2044    2064
                2084    2121    2133    2153    2173    2211    2223    2243    2263    2299    2311    2331    2351    2391    2404
                2416    2431    2457    2491    2504    2517    2531    2555    2594    2611    2615    2638    2676    2681    2689
                2713    2720    2768    2772    2798    2826    2832    2859    2885    2892    2894    2900    2907    2929    2931
                2958    2984    3011    3018    3028    3038    3050    3058    3070    3077    3095    3100    3118    3153    3162
                3168    3170    3175    3183    3186    3189    3192    3195    3197    3207    3252    3254    3265    3322    3326
                3332    3333    3334    3335    3336    3338    3385    3388    3418    3421    3422    3446    3447    3448    3449
                3450    3453    3455    3471    3493    3533    3539    3545    3598    3602    3605    3606    3635    3639    3661
                3673    3694    3721    3740    3743    3765    3777    3794    3825    3855    3858    3860    3864    3866    3870
                3873    3876    3880    3947    3995    3997    4044    4052    4142    4144    4146    4158    4163    4173    4239
                4247    4285    4287    4295    4390    4399    4403    4405    4407    4409    4414    4428
$OPR1           1#      3178    3642    3746    4250    4262    4290    4298
```

```
$OPR2          1#  1155  1158  1161  1164  1210  1212  1221  1261  1278  1295  1337  1354  1371  1419
             1436  1453  1494  1511  1526  1576  1593  1610  1652  1669  1686  1728  1745  1762  1804
             1821  1838  2048  2068  2088  2137  2157  2177  2227  2247  2267  2315  2335  2355  2407
             2435  2461  2508  2536  2559  2609  2619  2641  2765  2824  2890  2912  3172  3213  3215
             3260  3329  3341  3351  3391  3419  3458  3461  3485  3519  3522  3548  3556  3559  3561
             3592  3594  3603  3620  3681  3684  3704  3727  3781  3784  3886  3890  3893  3917  3919
             3940  3943  3954  3975  3978  3982  4002  4012  4040  4049  4191  4241  4255  4257  4328
             4416  4418  4420  4430
$OPSHF         1#  4255
$OPSUB         1#  4191
$OPSWB         1#
$OPXOR         1#
$OR            1#
$PUT           1#  2411  2512  2540  2563  2623  2645  2695  2738  2779  2838  3022  3032  3104  3356
             3400  3477  3489  3565  3652  3665  3756  3768  4055  4187
$STRUC         1#
$SUBON         1#  1207  1215  1254  1270  1287  1306  1330  1346  1363  1382  1412  1428  1445  1464
             1489  1503  1519  1535  1556  1569  1585  1602  1621  1645  1661  1678  1697  1721  1737
             1754  1773  1797  1813  1830  1849  1888  1921  1955  1979  2002  2035  2059  2079  2099
             2124  2148  2168  2188  2214  2238  2258  2278  2302  2326  2346  2366  2394  2425  2443
             2452  2469  2494  2526  2550  2573  2597  2633  2655  2679  2709  2731  2750  2793  2811
             2851  2868  2888  2910  2914  2916  2934  2961  2963  2965  2966  2977  2993  3053  3073
             3091  3121  3131  3156  3165  3179  3182  3210  3217  3220  3221  3229  3239  3247  3249
             3257  3258  3373  3379  3381  3410  3509  3514  3516  3542  3583  3588  3590  3643  3646
             3697  3699  3700  3724  3747  3750  3797  3800  3801  3828  3900  3914  3956  3999  4004
             4015  4067  4076  4161  4165  4175  4193  4194  4206  4217  4251  4254  4259  4260  4267
             4291  4294  4299  4302  4303  4304  4306  4307  4312  4368  4393  4401  4412  4422  4424
             4439
$THEN          1#  1201  1248  1264  1281  1300  1324  1340  1357  1376  1406  1422  1439  1458  1483
             1497  1513  1529  1550  1563  1579  1596  1615  1639  1655  1672  1691  1715  1731  1748
             1767  1791  1807  1824  1843  1877  1910  1942  1972  1987  2026  2051  2071  2091  2117
             2140  2160  2180  2206  2230  2250  2270  2295  2318  2338  2358  2386  2419  2437  2446
             2463  2486  2520  2544  2567  2590  2627  2649  2673  2725  2805  2862  2882  2904  2922
             2947  2950  2970  2987  3042  3060  3081  3109  3124  3150  3159  3203  3224  3234  3362
             3366  3404  3497  3501  3535  3571  3575  3688  3717  3788  3822  3908  3950  3987  3991
             4008  4070  4155  4169  4201  4384  4387
$TILA          1#
$TILO          1#
$UNTL2         1#  2916  3900
$UNTL3         1#
$WHILE         1#  2942  3199
$$CMRE       954#
$$CMTM       954#
$$DEFA         1#
$$ENDS         1#
$$ERRO         1#
$$ESCA       795#
$$GEN          1#  1191  1207  1254  1270  1287  1306  1330  1346  1363  1382  1412  1428  1445  1464
             1489  1503  1519  1535  1556  1569  1585  1602  1621  1645  1661  1678  1697  1721  1737
             1754  1773  1797  1813  1830  1849  1888  1921  1955  1979  2002  2035  2059  2079  2099
             2124  2148  2168  2188  2214  2238  2258  2278  2302  2326  2346  2366  2394  2425  2443
             2452  2469  2494  2526  2550  2573  2597  2633  2655  2679  2709  2731  2750  2793  2811
             2851  2868  2888  2902  2910  2914  2920  2934  2942  2961  2963  2966  2977  2993  3053
             3073  3091  3121  3131  3156  3165  3177  3179  3199  3210  3217  3221  3229  3239  3247
             3249  3258  3373  3379  3381  3410  3509  3514  3516  3542  3583  3588  3590  3641  3643
             3697  3700  3724  3745  3747  3797  3801  3828  3898  3904  3914  3956  3999  4004  4015
```

```
                4067    4076    4117    4152    4161    4165    4175    4194    4206    4217    4219    4224    4243    4251    4260
                4267    4268    4273    4289    4291    4297    4299    4304    4307    4312    4313    4334    4368    4369    4372
                4382    4393    4401    4412    4422    4439    4440
$$GETS        1#  1207    1215    1254    1270    1287    1306    1330    1346    1363    1382    1412    1428    1445    1464
                1489    1503    1519    1535    1556    1569    1585    1602    1621    1645    1661    1678    1697    1721    1737
                1754    1773    1797    1813    1830    1849    1888    1921    1955    1979    2002    2035    2059    2079    2099
                2124    2148    2168    2188    2214    2238    2258    2278    2302    2326    2346    2366    2394    2425    2443
                2452    2469    2494    2526    2550    2573    2597    2633    2655    2679    2709    2731    2750    2793    2811
                2851    2868    2888    2909    2910    2914    2916    2934    2961    2963    2965    2966    2977    2993    3053
                3073    3091    3121    3131    3156    3165    3179    3182    3209    3210    3217    3220    3221    3228    3229
                3238    3239    3247    3249    3257    3258    3372    3373    3379    3381    3410    3508    3509    3514    3516
                3542    3582    3583    3588    3590    3643    3646    3697    3699    3700    3724    3747    3750    3797    3800
                3801    3828    3900    3914    3956    3999    4004    4015    4067    4076    4160    4161    4165    4175    4179
                4193    4194    4206    4217    4251    4254    4259    4260    4267    4291    4294    4299    4302    4303    4304
                4306    4307    4312    4368    4392    4393    4401    4411    4412    4422    4424    4439
$$GETT        1#  2909    3209    3228    3238    3372    3508    3582    4160    4178    4392    4411
$$LPCN        1#  3178    3642    3746    4250    4290    4298
$$NEWT      795#  1170    1234    1312    1389    1471    1542    1628    1704    1780    1863    1896    1929    1963    2016
                2107    2196    2276    2374    2476    2582    2662    2754    2815    2871    3003    3140    3309    3434    3611
                3710    3811    4030    4088
$$POP         1#  1207    1215    1254    1270    1287    1306    1330    1346    1363    1382    1412    1428    1445    1464
                1489    1503    1519    1535    1556    1569    1585    1602    1621    1645    1661    1678    1697    1721    1737
                1754    1773    1797    1813    1830    1849    1888    1921    1955    1979    2002    2035    2059    2079    2099
                2124    2148    2168    2188    2214    2238    2258    2278    2302    2326    2346    2366    2394    2425    2443
                2452    2469    2494    2526    2550    2573    2597    2633    2655    2679    2709    2731    2750    2793    2811
                2851    2868    2888    2910    2914    2916    2934    2961    2963    2965    2966    2977    2993    3053    3073
                3091    3121    3131    3156    3165    3179    3182    3210    3217    3220    3221    3229    3239    3247    3249
                3257    3258    3273    3379    3381    3410    3509    3514    3516    3542    3583    3588    3590    3643    3646
                3697    3699    3700    3724    3747    3750    3797    3800    3801    3828    3900    3914    3956    3999    4004
                4015    4067    4076    4161    4165    4175    4193    4194    4206    4217    4251    4254    4259    4260    4267
                4291    4294    4299    4302    4303    4304    4306    4307    4312    4368    4393    4401    4412    4422    4424
                4439
$$PUSH        1#  1191    1192    1201    1203    1248    1250    1264    1266    1281    1283    1300    1302    1324    1326
                1340    1342    1357    1359    1376    1378    1406    1408    1422    1424    1439    1441    1458    1460    1483
                1485    1497    1499    1513    1515    1529    1531    1550    1552    1563    1565    1579    1581    1596    1598
                1615    1617    1639    1641    1655    1657    1672    1674    1691    1693    1715    1717    1731    1733    1748
                1750    1767    1769    1791    1793    1807    1809    1824    1826    1843    1845    1877    1879    1910    1912
                1942    1944    1972    1974    1987    1989    2026    2028    2051    2053    2071    2073    2091    2093    2117
                2119    2140    2142    2160    2162    2180    2182    2206    2208    2230    2232    2250    2252    2270    2272
                2295    2297    2318    2320    2338    2340    2358    2360    2386    2388    2419    2421    2437    2439    2446
                2448    2463    2465    2486    2488    2520    2522    2544    2546    2567    2569    2590    2592    2627    2629
                2649    2651    2673    2675    2704    2705    2725    2727    2745    2746    2788    2789    2805    2807    2846
                2847    2862    2864    2882    2884    2902    2903    2904    2906    2911    2922    2924    2942    2943    2945
                2947    2949    2950    2952    2965    2970    2972    2987    2989    3042    3044    3060    3062    3081    3083
                3109    3111    3124    3126    3150    3152    3159    3161    3175    3177    3178    3182    3199    3200    3202
                3203    3205    3211    3220    3224    3226    3230    3232    3236    3240    3362    3364    3366    3368    3374
                3404    3406    3497    3499    3501    3503    3510    3535    3537    3571    3573    3575    3577    3584    3639
                3641    3642    3646    3688    3690    3717    3719    3743    3745    3746    3750    3788    3790    3822    3824
                3898    3899    3908    3910    3950    3952    3987    3989    3991    3993    4008    4010    4062    4063    4070
                4072    4118    4152    4153    4155    4157    4162    4169    4171    4193    4201    4203    4225    4247    4249
                4250    4254    4274    4287    4289    4290    4294    4295    4297    4298    4302    4335    4373    4382    4383
                4384    4386    4387    4389    4394    4413
$$SELE        1#
$$SET      5088#  5097    5098    5099    5100    5102    5104    5105    5106
$$SETM     1111#
$$SETS        1#  1191    1192    1201    1203    1248    1250    1264    1266    1281    1283    1300    1302    1324    1326
```

```
                1340    1342    1357    1359    1376    1378    1406    1408    1422    1424    1439    1441    1458    1460    1483
                1485    1497    1499    1513    1515    1529    1531    1550    1552    1563    1565    1579    1581    1596    1598
                1615    1617    1639    1641    1655    1657    1672    1674    1691    1693    1715    1717    1731    1733    1748
                1750    1767    1769    1791    1793    1807    1809    1824    1826    1843    1845    1877    1879    1910    1912
                1942    1944    1972    1974    1987    1989    2026    2028    2051    2053    2071    2073    2091    2093    2117
                2119    2140    2142    2160    2162    2180    2182    2206    2208    2230    2232    2250    2252    2270    2272
                2295    2297    2318    2320    2338    2340    2358    2360    2386    2388    2419    2421    2437    2439    2446
                2448    2463    2465    2486    2488    2520    2522    2544    2546    2567    2569    2590    2592    2627    2629
                2649    2651    2673    2675    2704    2705    2725    2727    2745    2746    2788    2789    2805    2807    2846
                2847    2862    2864    2882    2884    2902    2903    2904    2906    2911    2922    2924    2942    2943    2945
                2947    2949    2950    2952    2965    2970    2972    2987    2989    3042    3044    3060    3062    3081    3083
                3109    3111    3124    3126    3150    3152    3159    3161    3175    3177    3178    3182    3199    3200    3202
                3203    3205    3211    3220    3224    3226    3230    3232    3236    3240    3362    3364    3366    3368    3374
                3404    3406    3497    3499    3501    3503    3510    3535    3537    3571    3573    3575    3577    3584    3639
                3641    3642    3646    3688    3690    3717    3719    3743    3745    3746    3750    3788    3790    3822    3824
                3898    3899    3908    3910    3950    3952    3987    3989    3991    3993    4008    4010    4062    4063    4070
                4072    4118    4152    4153    4155    4157    4162    4169    4171    4193    4201    4203    4225    4247    4249
                4250    4254    4274    4287    4289    4290    4294    4295    4297    4298    4302    4335    4373    4382    4383
                4384    4386    4387    4389    4394    4413
$$SETT     1#
$$SKIP   795#    1487    1554    1977    2033    2122    2212    2300    2392    2492    2576    2595    2677    2886    2932
                2959    2996    3051    3071    3089    3119    3134    3154    3163    3268    3540    3695    3722    3795    3826
                3922    4080
.EQUAT     1#     685
.HEADE     1#     663
.SETUP     1#    1072
.SWRHI     1#     673
.SWRLO   684#
.$ACTI     1#     921
.$APTB     1#     998#
.$APTH     1#     932
.$APTY     1#    4746
.$CATC     1#     910
.$CMTA     1#     954
.$EOP      1#    4444
.$ERRO     1#    4803
.$POWE     1#    4483
.$READ     1#    4607
.$SCOP     1#    4857
.$TRAP     1#    5065
.$TYPD     1#    4921
.$TYPE     1#    4528
.$TYPO     1#    4988

. ABS.   015316      000

 ERRORS DETECTED:  0

 CVDVAB,CVDVAB.SEQ=CVDVAB.MAC,CVDVAB.P11
 RUN-TIME:  101  93.6 SECONDS
 RUN-TIME RATIO: 391/202=1.9
 CORE USED:  34K  (67 PAGES)
```