DLV11-E

OFFLINE TEST

MD-11-DVDVA-A

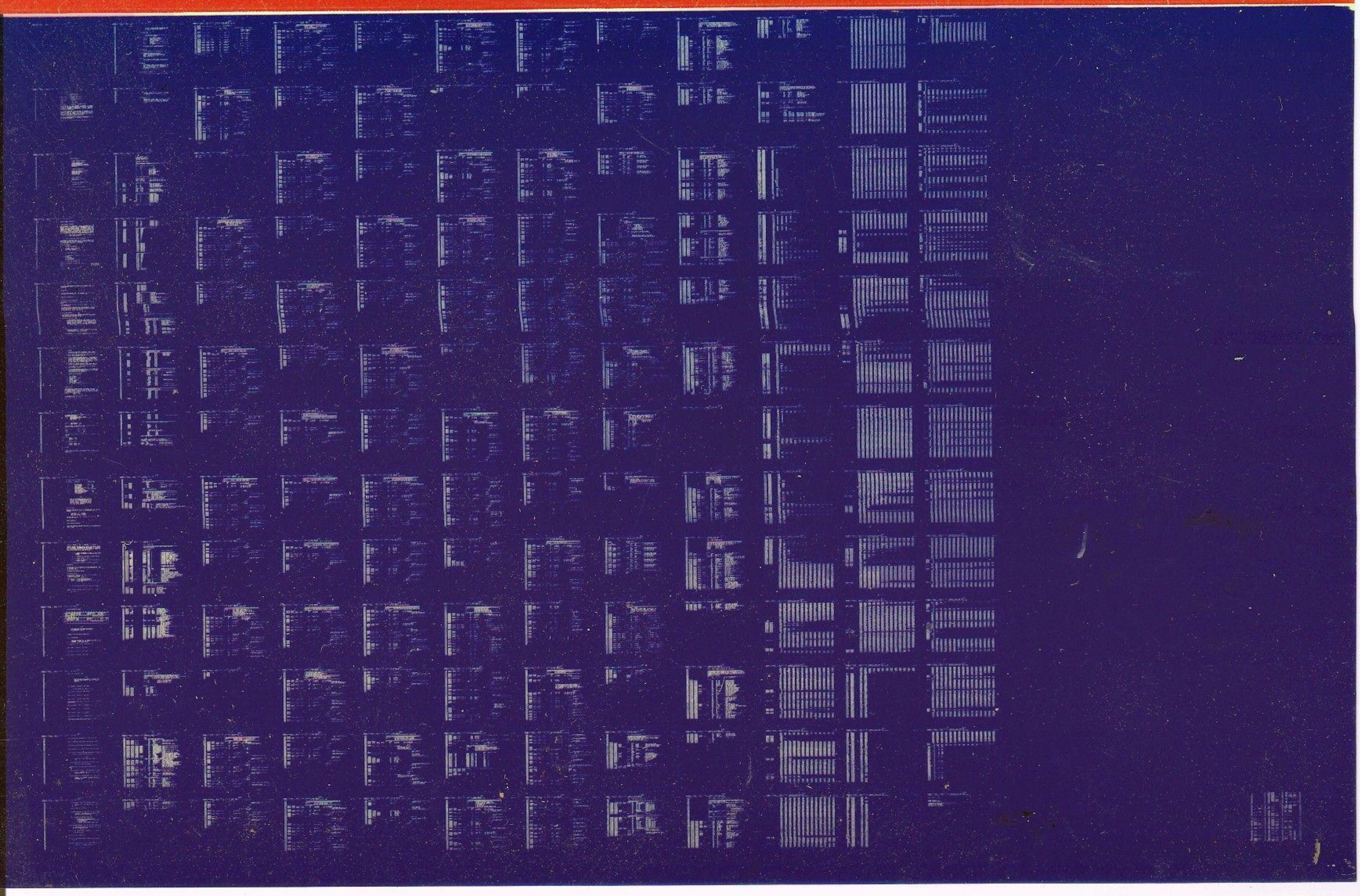
EP-DVDVA-A-DL-A

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



```
                                .REM    a
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVDVA-A-D

PRODUCT NAME: 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. POWER FAIL IS SUPPORTED.

1.2 SYSTEM REQUIREMENTS.

1. HARDWARE REQUIREMENTS:

ANY PDP-11 FAMILY PROCESSOR
4K MEMORY - MINIMUM
H315 - CABLE TURN AROUND PLUG (OR EQUIVALENT)

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 ASSUMED TO BE FULLY OPERATIONAL.

1.5 ASSUMPTIONS.

THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR HAS INITIALIZED LOCATION '$USWR' AND '$DEVM' TO THE PROPER VALUES.

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 IN QUICK VERIFY MODE.)

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) A RUBOUT WILL DELETE THE LAST INPUT VALUE AND WILL DELIMIT ALL DELETED CHARACTERS BETWEEN BACK SLASHES.

C) IF A CONTROL U <↑U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

D) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED AND WILL WAIT FOR THE OPERATOR TO ENTER THE "SWREG" DATA AGAIN USING VALID CHARACTERS.

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 (H3FS)	1 = YES
14	FR AND FD JUMPERS IN	1 = YES
15	(NOT DEFINED)	

NOTE

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-	2 MINUTE
ADDITIONAL PASSES	2 MINUTES
ADDITIONAL DEVICES	2 MINUTES

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

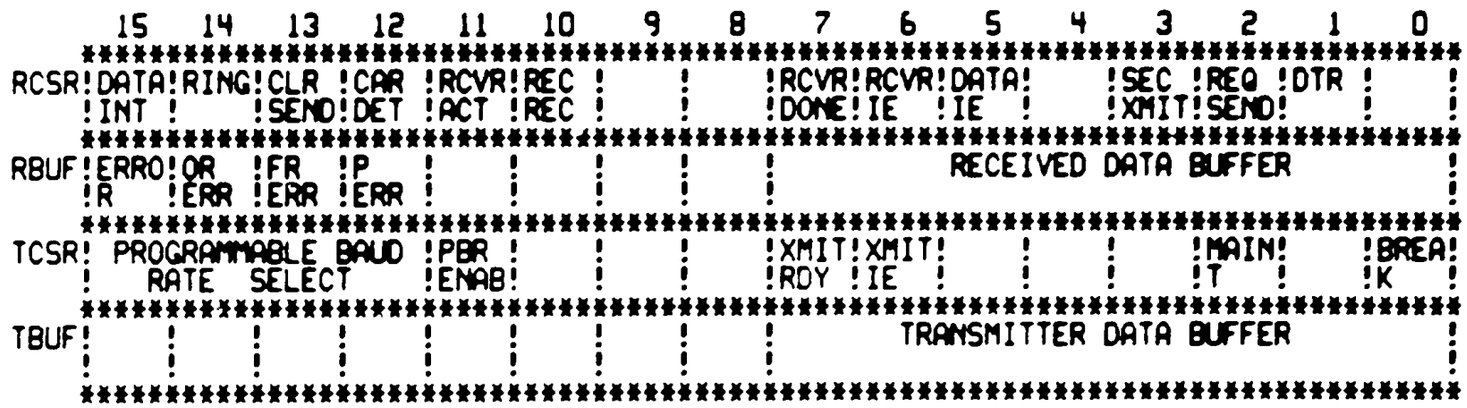
3.0 ERROR INFORMATION.

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 EXCEPT THE ERROR NUMBER,
                        WHICH IS DECIMAL.  THE ADDRESS AND VECTOR REFER TO THE FAILING
                        SLU'S 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.

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

                               THE ERROR ROUTINE SUPPORTS THE CONTROL G FUNCTION.

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

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

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

                    4.1 PERFORMANCE REPORTS.

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

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


                                                NOTE

                            THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.


                        AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS
                        TYPED:
                                "ENDPASS#+++++."
```

5.0 DEVICE INFORMATION TABLES.



NOTE

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

6.0 SUMMARY OF TESTS AND SPECIAL SUBROUTINES.

TEST 1 ADDRESSABILITY

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

THE FOLLOWING 8 TESTS TEST ALL 'READ WRITE' BITS

TEST 2 BREAK - TCSR0 SET, CLEAR, RESET

TEST 3 MAINT - TCSR2 SET, CLEAR, RESET

TEST 4 XMITIE - TCSR6 SET, CLEAR, RESET

```
                        TEST 5 DTR - RCSR1 SET, CLEAR
                        ---- -

                                                NOTE

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



                        TEST 6 REQSEND - RCSR2 SET, CLEAR, RESET
                        ---- -

                               THIS TEST ASSUMES THAT JUMPER FR IS IN.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

                                                NOTE

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

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

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

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


ROUTINE:WAIT
------------
        THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
        MAIN PROGRAM FOR A SPECIFIED AMOUNT OF TIME.
        THIS IS ACCOMPLISHED BY INCREMENTING A
        REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
        TO APPROXIMATE 1 MICRO 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-11-DVDVA-A
                        ;*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 ***
000110                  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                  ORERR=          BIT14           ; OVERRUN ERROR
020000                  FRERR=          BIT13           ; FRAMING ERROR
010000                  PERR=           BIT12           ; PARITY ERROR
                        ; UNUSED        BIT11
                        ; UNUSED        BIT10
                        ; UNUSED        BIT09
                        ; UNUSED        BIT08
000200                  RDATA7=         BIT07           ; \
000100                  RDATA6=         BIT06           ; !
000040                  RDATA5=         BIT05           ; !
000020                  RDATA4=         BIT04           ;  \    RECEIVED DATA BITS
000010                  RDATA3=         BIT03           ;  /
000004                  RDATA2=         BIT02           ; !
000002                  RDATA1=         BIT01           ; !
000001                  RDATA0=         BIT00           ; /

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


                        ;;*****************************************************************
                        ;               TBUF REGISTER BIT NAMES
                        ;;*****************************************************************
                        ; UNUSED        BIT15
                        ; UNUSED        BIT14
                        ; UNUSED        BIT13
                        ; UNUSED        BIT12
                        ; UNUSED        BIT11
                        ; UNUSED        BIT10
                        ; UNUSED        BIT09
                        ; UNUSED        BIT08
000200                  TDATA7=         BIT07           ; \
000100                  TDATA6=         BIT06           ; !
```

```
                000040                          TDATA5=         BIT05           ; !
                000020                          TDATA4=         BIT04           ; \ TRANSMITTER DATA BUFFER
                000010                          TDATA3=         BIT03           ; /
                000004                          TDATA2=         BIT02           ; !
                000002                          TDATA1=         BIT01           ; !
                000001                          TDATA0=         BIT00           ; /


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

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

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

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

```
                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
                                        $SVPC=.                 ;SAVE PC
000204                                  .=46
000046                                  $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOF
012214                                  .=52
000052                                  .WORD   0               ;;2)SET LOC.52 TO ZERO
000000                                  .=$SVPC                 ;; RESTORE PC
000204                          .=1000
001000                          .SBTTL  APT PARAMETER BLOCK

                                ;;******************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;******************************************************************
                                        .$X=.   ;;SAVE CURRENT LOCATION
001000                                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024                                  200     ;;FOR APT START UP
000200                                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044                                  $APTHDR ;;POINT TO APT HEADER BLOCK
001000                                  .=.$X   ;;RESET LOCATION COUNTER
001000                          ;;******************************************************************
                                ;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   ;;AD. ESS OF APT MAILBOX (BITS 0-15)
001004  000005                  $TSTM:  .WORD   5       ;;RUN TIM OF LONGEST TEST
001006  000055                  $PASTM: .WORD   45.     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000036                  $UNITM: .WORD   30.     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000030                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

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

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

```
1017  001212  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1018  001214                        $ETABLE:                ;;APT ENVIRONMENT TABLE
1019  001214      000               $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1020  001215      000               $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1021  001216  000000                $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1022  001220  071110                $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1023  001222  000000                $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1024                                ;*                      BITS 15-11=CPU TYPE
1025                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1026                                ;*                              11/70=06,PDQ=07,Q=10
1027                                ;*                      BIT 10=REAL TIME CLOCK
1028                                ;*                      BIT  9=FLOATING POINT PROCESSOR
1029                                ;*                      BIT  8=MEMORY MANAGEMENT
1030  001224      000               $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1031  001225      000               $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1032                                ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1033                                ;*                              900 NSEC CORE=001
1034                                ;*                              300 NSEC BIPOLAR=002
1035                                ;*                              500 NSEC MOS=003
1036  001226  000000                $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1037                                ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1038  001230      000               $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1039  001231      000               $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1040  001232  000000                $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1041  001234      000               $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1042  001235      000               $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1043  001236  000000                $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1044  001240      000               $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1045  001241      000               $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1046  001242  000000                $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1047  001244  000300                $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1048  001246  000000                $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1049  001250  175610                $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1050  001252  000001                $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1051  001254                        $ETEND:
1052                                .MEXIT
```

ERROR POINTER TABLE SEQ 0024

```
                                .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
```

```
1079  001336                                  START:
1080                                          .SBTTL  INITIALIZE THE COMMON TAGS
1081                                          ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1082  001336  012706  001100                         MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1083  001342  005026                                 CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1084  001344  022706  001140                         CMP     #SWR,R6 ;;DONE?
1085  001350  001374                                 BNE     .-6             ;;LOOP BACK IF NO
1086  001352  012706  001100                         MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1087                                          ;;INITIALIZE A FEW VECTORS
1088  001356  012737  014132  000020                 MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1089  001364  012737  000340  000022                 MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1090  001372  012737  013732  000030                 MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1091  001400  012737  000340  000032                 MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1092  001406  012737  015064  000034                 MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1093  001414  012737  000340  000036                 MOV     #340,@#TRAPVEC+2;LEVEL 7
1094  001422  012737  012252  000024                 MOV     #$PWRON,@#PWRVEC ;;POWER FAILURE VECTOR
1095  001430  012737  000340  000026                 MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1096  001436  016767  010520  010510                 MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
1097  001444  005067  177510                        CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1098  001450  005067  177506                        CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1099  001454  112767  000001  177433                 MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1100  001462  012767  001462  177416                 MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1101  001470  012767  001470  177412                 MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1102                                          ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1103                                          ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1104  001476  013746  000004                         MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1105  001502  012737  001536  000004                 MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1106  001510  012767  177570  177422                 MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1107  001516  012767  177570  177416                 MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1108  001524  022777  177777  177406                 CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1109  001532  001012                                 BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1110                                                                          ;;AND  THE HARDWARE SWR IS NOT = -1
1111  001534  000403                                 BR      65$             ;;BRANCH IF NO TIMEOUT
1112  001536  012716  001544          64$:          MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1113  001542  000002                                 RTI
1114  001544  012767  000176  177366  65$:          MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1115  001552  012767  000174  177362                 MOV     #DISPREG,DISPLAY
1116  001560  012637  000004          66$:          MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1117
1118  001564  005067  177412                        CLR     $PASS           ;;CLEAR PASS COUNT
1119  001570  132767  000200  177417                 BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1120  001576  001403                                 BEQ     67$             ;;YES,USE NON-APT SWITCH
1121  001600  012767  001216  177332                 MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1122  001606                          67$:
1123                                          .SBTTL  TYPE PROGRAM NAME
1124                                          ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1125  001606  005227  177777                        INC     #-1             ;;FIRST TIME?
1126  001612  001037                                 BNE     68$             ;;BRANCH IF NO
1127  001614  022737  012214  000042                 CMP     #$ENDAD,@#42    ;;ACT-11?
1128  001622  001433                                 BEQ     68$             ;;BRANCH IF YES
1129  001624  104401  001672                        TYPE    ,69$            ;;TYPE ASCIZ STRING
1130                                          .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1131  001630  005737  000042                        TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1132  001634  001012                                 BNE     70$             ;;BRANCH IF YES
1133  001636  126727  177352  000001                 CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1134  001644  001406                                 BEQ     70$             ;;BRANCH IF YES
```

```
1135  001646  026727  177266  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1136  001654  001005                          BNE     71$             ;;BRANCH IF NO
1137  001656  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1138  001660  000403                          BR      71$
1139  001662  112767  000001  177244  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDIC^TOR
1140  001670                        71$:
1141  001670  000410                          BR      68$             ;;GET OVER THE ASCIZ
1142                                  ;;69$:   .ASCIZ  <CRLF>#MD-11-DVDVA-A#<CRLF>
1143  001712                        68$:
```

```
1144  001712                                        LET INITFLAG := #1
1145  001712  012767  000001  010176        MOV     #1,INITFLAG
1146  001720                              LOOP:
1147  001720                              CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
1148  001720  004767  010042              JSR     PC,CYCLE
1149                                                ;               ADDRS+2 -> NEXT VECTOR
1150                                                                ;GET UNIT ADDRESS
1151  001724                                                LET     DLADD := (ADRS)+
1152  001724  012167  177324              MOV     (ADRS)+,DLADD
1153                                                                ;GET UNIT VECTOR
1154  001730                                                LET     DLVEC := (ADRS)
1155  001730  011167  177322              MOV     (ADRS),DLVEC
1156  001734                                                LET     ADRS  := DLADD
1157  001734  016701  177314              MOV     DLADD,ADRS
1158                                                                ;RCSR = DLADD  +  0
1159  001740                                                LET     RCSR  := DLADD
1160  001740  016767  177310  177312      MOV     DLADD,RCSR
1161  001746                                                LET     RBUF  := DLADD + #2
1162  001746  016767  177302  177306      MOV     DLADD,RBUF
1163  001754  062767  000002  177300      ADD     #2,RBUF
1164  001762                                                LET     TCSR  := DLADD + #4
1165  001762  016767  177266  177274      MOV     DLADD,TCSR
1166  001770  062767  000004  177266      ADD     #4,TCSR
1167  001776                                                LET     TCSRHI := DLADD + #5
1168  001776  016767  177252  177262      MOV     DLADD,TCSRHI
1169  002004  062767  000005  177254      ADD     #5,TCSRHI
1170  002012                                                LET     TBUF  := DLADD + #6
1171  002012  016767  177236  177250      MOV     DLADD,TBUF
1172  002020  062767  000006  177242      ADD     #6,TBUF
1173  002026                                                LET     R5    := #R5STACK
1174  002026  012705  001334              MOV     #R5STACK,R5
1175                                                                ;;BRESET
1176  002032  000005                      RESET
```

```
1177                                    ;;*****************************************************************
1178                                    ;*TEST 1         ADDRESSABILITY
1179                                    ;*              THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
1180                                    ;*              THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-E RESPONDS
1181                                    ;*              TO THAT ADDRESS SPACE
1182                                    ;*****************************************************************
1183  002034  000004                   $TST1:  SCOPE
1184  002036  012767  000002  177114           MOV     #2,$TIMES       ;;DO 2 ITERATIONS
1185  002044  012767  000001  177126           MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1186  002052                                                                   LET     ADRS := DLADD
1187  002052  016701  177176                   MOV     DLADD,ADRS
1188                                                                     ; SET UP INTERRUPT
1189  002056                                                           SETVEC  #ILLMEM,#INTSRV,#PR7
1190  002056  010146                           MOV     R1,-(SP)
1191  002060  012701  000004                   MOV     #ILLMEM,R1
1192  002064  012721  011574                   MOV     #INTSRV,(R1)+
1193  002070  012711  000340                   MOV     #PR7,(R1)
1194  002074  012601                           MOV     (SP)+,R1
1195  002076                                                           LET     I := #0
1196  002076  005067  177170                   CLR     I
1197  002102                                                           REPEAT
1198  002102                           $1:
1199  002102
1200  002102  012767  002110  177000           MOV     #64$,$LPERR
1201                                                                       BGNSUB
1202  002110                                                                   ;CLEAR FLAG
1203  002110  005067  007466                   CLR     INTFLAG                LET INTFLAG := #0
1204
1205                                                                       ;READ FLAG
1206  002114  005711                           TST @ADRS
1207  002116                                                               IF INTFLAG NE #0 THEN
1208  002116  005767  007460                   TST     INTFLAG
1209  002122  001401                           BEQ     $2
1210                                                                     ; FATAL ERROR
1211  002124                                                                 ERRDF 1,,NODL
1212  002124  104001                           ERROR   1
1213  002126                                                               ENDIF
1214  002126                           $2:
1215  002126                                                           ENDSUB
1216  002126                                                           LET     I := I + #2
1217  002126  062767  000002  177136           ADD     #2,I
1218  002134                                                           LET     ADRS := DLADD + I
1219  002134  016701  177114                   MOV     DLADD,ADRS
1220  002140  066701  177126                   ADD     I,ADRS
1221  002144                                                       UNTIL I EQ #8.
1222  002144  026727  177122  000010           CMP     I,#8.
1223  002152  001353                           BNE     $1
1224  002154                                                           CLRVEC ILLMEM
1225  002154  010146                           MOV     R1,-(SP)        ;;PUSH R1 ON STACK
1226  002156  010246                           MOV     R2,-(SP)        ;;PUSH R2 ON STACK
1227  002160  012701  000004                   MOV     #ILLMEM,R1
1228  002164  010102                           MOV     R1,R2
1229  002166  062702  000002                   ADD     #2,R2
1230  002172  010221                           MOV     R2,(R1)+
1231  002174  005011                           CLR     (R1)
1232  002176  012602                           MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
  1233  002200  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
  1234                                                            ;END OF TEST
  1235  002202                                                    ENDTST
```

```
                        ;;*****************************************************************
                        ;* THE FOLLOWING 8 TESTS  TEST ALL 'READ WRITE' BITS
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*TEST 2        BREAK - TCSR0 SET, CLEAR, RESET
                        ;*               THIS BIT IS THE ONLY ONE IN THIS POSITION
                        ;*               THAT IS READ AND WRITE.
                        ;;*****************************************************************
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
002220  012767  002226  176662          MOV     #64$,$LPERR

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

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

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

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

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

                                                ; TRY TO CLEAR A SET BIT
                                                                BGNSUB
002266
002266  012767  002274  176614          MOV     #64$,$LPERR

                                                                    LET     @TCSR := @TCSR CLR.BY #BREAK
002274
002274  042777  000001  176762          BIC     #BREAK,@TCSR

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

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

```
1292  002314                          $5:
1293  002314                                                          ENDSUB
1294
1295                                                          ; NOW SEE IF RESET CLEARS IT
1296  002314                                                          BGNSUB
1297  002314  012767  002322  176566          MOV     #64$,$LPERR
1298
1299  002322                                                              LET     @TCSR := @TCSR SET.BY #BREAK
1300  002322  052777  000001  176734          BIS     #BREAK,@TCSR
1301                                                                  ; ISSUE BUS RESET
1302  002330                                                          BRESET
1303  002330  000005                          RESET                  IF      #BREAK SETIN @TCSR THEN
1304  002332
1305  002332  032777  000001  176724          BIT     #BREAK,@TCSR
1306  002340  001401                          BEQ     $6
1307                                                                  ; BREAK DID NOT RESET IN TCSR
1308  002342                                                          ERRHRD 5,,DIDNOT
1309  002342  104005                          ERROR   5
1310  002344                                                          ENDIF
1311  002344                          $6:
1312  002344                                                      ENDSUB
1313  002344                                                  ENDTST
1314
1315
```

```
1316                                    ;;*****************************************************************
1317                                    ;;*****************************************************************
1318                                    ;*TEST 3          MAINT - TCSR2 SET, CLEAR, RESET
1319                                    ;;*****************************************************************
1320  002344  000004                    $ST3:   SCOPE
1321  002346  012767  000010  176604            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1322  002354  012767  000003  176616            MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1323
1324                                                    ; SEE IF IT IS CLEAR
1325  002362                                                            BGNSUB
1326  002362  012767  002370  176520            MOV     #64$,$LPERR
1327
1328  002370                                                              IF      #MAINT SETIN @TCSR THEN
1329  002370  032777  000004  176666            BIT     #MAINT,@TCSR
1330  002376  001401                            BEQ     $7
1331
1332  002400                                                                  ; MAINT DID NOT RESET IN TCSR
1333  002400  104006                                                         ERRHRD 6,,DIDNOT
1334  002402                                    ERROR   6
1335  002402                                                              ENDIF
1336  002402                                $7:
1337
1338                                                    ; TRY TO SET MAINT BIT
1339  002402                                                            BGNSUB
1340  002402  012767  002410  176500            MOV     #64$,$LPERR
1341  002410                                                                LET     @TCSR := @TCSR SET.BY #MAINT
1342  002410  052777  000004  176646            BIS     #MAINT,@TCSR
1343                                                                    ; STUCK TO 0
1344  002416                                                            IF      #MAINT NOTSETIN @TCSR THEN
1345  002416  032777  000004  176640            BIT     #MAINT,@TCSR
1346  002424  001001                            BNE     $10
1347
1348  002426                                                                  ; MAINT DID NOT SET IN TCSR
1349  002426  104007                                                         ERRHRD 7,,DIDNOT
1350  002430                                                              ENDIF
1351  002430                                $10:
1352  002430                                                          ENDSUB
1353
1354                                                    ; TRY TO CLEAR A SET BIT
1355  002430                                                            BGNSUB
1356  002430  012767  002436  176452            MOV     #64$,$LPERR
1357
1358  002436                                                                LET     @TCSR := @TCSR CLR.BY #MAINT
1359  002436  042777  000004  176620            BIC     #MAINT,@TCSR
1360                                                                    ; SHOULD HAVE CLEARED
1361  002444                                                            IF      #MAINT SETIN @TCSR THEN
1362  002444  032777  000004  176612            BIT     #MAINT,@TCSR
1363  002452  001401                            BEQ     $11
1364
1365  002454                                                                  ; MAINT DID NOT CLEAR INTCSR
1366  002454  104010                                                         ERRHRD 8,,DIDNOT
1367  002456                                    ERROR   8
1368  002456                                                              ENDIF
1369  002456                                $11:
1370                                                                  ENDSUB
1371                                                    ; NOW SEE IF RESET CLEARS IT
```

```
1372  002456
1373  002456  012767  002464  176424          MOV     #64$,$LPERR
1374
1375  002464
1376  002464  052777  000004  176572          BIS     #MAINT,@TCSR
1377
1378  002472
1379  002472  000005                          RESET
1380  002474
1381  002474  032777  000004  176562          BIT     #MAINT,@TCSR
1382  002502  001401                          BEQ     $12
1383
1384  002504
1385  002504  104011                          ERROR   9
1386  002506
1387  002506                          $12:
1388  002506
1389  002506
1390
1391
1392
```

```
BGNSUB

    LET     @TCSR := @TCSR SET.SY #MAINT

        ; ISSUE BUS RESET
    BRESET

    IF      #MAINT SETIN @TCSR THEN

        ; MAINT DID NOT RESET IN TCSR
        ERRHRD 9,,DIDNOT

    ENDIF

ENDSUB
ENDTST
```

```
1393                                    ;;**********************************************************************
1394                                    ;;**********************************************************************
1395                                    ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
1396                                    ;;**********************************************************************
1397  002506  000004                   TST4:   SCOPE
1398  002510  012767  000010  176442           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1399  002516  012767  000004  176454           MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1400                                                                    ; USE PRIORITY OF 7
1401  002524  012746  000340                   MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
1402  002530  012746  002536                   MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1403  002534  000002                           RTI                     ;;POP NEW PC AND PS
1404  002536                            64$:
1405
1406                                                    ; SEE IF IT IS CLEAR
1407  002536                                                            BGNSUB
1408  002536  012767  002544  176344           MOV     #65$,$LPERR
1409
1410  002544                                                                IF      #XMITIE SETIN @TCSR THEN
1411  002544  032777  000100  176512           BIT     #XMITIE,@TCSR
1412  002552  001401                           BEQ     $13
1413                                                                        ; XMITIE DID NOT RESET IN TCSR
1414  002554                                                                ERRHRD 10,,DIDNOT
1415  002554  104012                           ERROR   10
1416  002556                                                                ENDIF
1417  002556                            $13:
1418  002556                                                            ENDSUB
1419
1420                                                    ; TRY TO SET XMITIE BIT
1421  002556                                                            BGNSUB
1422  002556  012767  002564  176324           MOV     #64$,$LPERR
1423  002564                                                                LET     @TCSR := @TCSR SET.BY #XMITIE
1424  002564  052777  000100  176472           BIS     #XMITIE,@TCSR
1425                                                                    ; STUCK TO 0
1426  002572                                                            IF      #XMITIE NOTSETIN @TCSR THEN
1427  002572  032777  000100  176464           BIT     #XMITIE,@TCSR
1428  002600  001001                           BNE     $14
1429                                                                        ; XMIT DID NOT RESET IN TCSR
1430  002602                                                                ERRHRD 11,,DIDNOT
1431  002602  104013                           ERROR   11
1432  002604                                                                ENDIF
1433  002604                            $14:
1434  002604                                                            ENDSUB
1435
1436                                                    ; TRY TO CLEAR A SET BIT
1437  002604                                                            BGNSUB
1438  002604  012767  002612  176276           MOV     #64$,$LPERR
1439
1440  002612                                                                LET     @TCSR := @TCSR CLR.BY #XMITIE
1441  002612  042777  000100  176444           BIC     #XMITIE,@TCSR
1442                                                                    ; SHOULD HAVE CLEARED
1443  002620                                                            IF      #XMITIE SETIN @TCSR THEN
1444  002620  032777  000100  176436           BIT     #XMITIE,@TCSR
1445  002626  001401                           BEQ     $15
1446                                                                        ;  XMIT DID NOT CLEAR IN TCSR
1447  002630                                                                ERRHRD 12,,DIDNOT
1448  002630  104014                           ERROR   12
```

```
002632                                                  ENDIF
002632                          $15:
002632                                               ENDSUB

                                            ; NOW SEE IF RESET CLEARS IT
002632                                               BGNSUB
002632  012767  002640  176250        MOV     #64$,$LPERR

002640                                                  LET     @TCSR := @TCSR SET.BY #XMITIE
002640  052777  000100  176416        BIS     #XMITIE,@TCSR

                                                           ; ISSUE BUS RESET
002646                                                  BRESET
002646  000005                        RESET
002650                                                  IF      #XMITIE SETIN @TCSR THEN
002650  032777  000100  176406        BIT     #XMITIE,@TCSR
002656  001401                        BEQ     $16

                                                           ; XMIT DID NOT RESET IN TCSR
002660                                                     ERRHRD 13,,DIDNOT
002660  104015                        ERROR   13
J02662                                                  ENDIF
002662                          $16:
002662                                               ENDSUB
J02662                                            ENDTST
```

```
                        ;;*****************************************************************
                        ;;*****************************************************************
                        ;*TEST 5         DTR - RCSR1    SET, CLEAR
                        ;*              NOTE:  RESET DOES NOT CLEAR THIS BIT
                        ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
                        ;*              THAT IS READ AND WRITE.
                        ;*              WE CANNOT TEST FOR AN INITIAL CONDITION
                        ;*              AS THIS BIT IS UNDEFINED UPON POWER UP AND
                        ;*              INIT DOESN'T AFFECT IT.
                        ;;*****************************************************************
002662  000004         TST5:   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
                                                ; TRY TO CLEAR DTR BIT
                                                                BGNSUB
002700
002700  012767  002706  176202          MOV     #64$,$LPERR
002706                                                              LET     @RCSR := @RCSR CLR.BY #DTR
002706  042777  000002  176344          BIC     #DTR,@RCSR
                                                                ; STUCK TO 0
002714                                                          IF      #DTR SETIN @RCSR THEN
002714  032777  000002  176336          BIT     #DTR,@RCSR
002722  001401                        BEQ     $17
                                                                ; DTR DID NOT CLEAR IN RCSR
002724                                                          ERRHRD 14,,DIDNOT
002724  104016                        ERROR   14
002726                                                          ENDIF
002726                         $17:
                                                                ENDSUB
                                                ; TRY TO SET  DTR
002726                                                          BGNSUB
002726  012767  002734  176154          MOV     #64$,$LPERR
                                                                LET     @RCSR := @RCSR SET.BY #DTR
002734
002734  052777  000002  176316          BIS     #DTR,@RCSR
002742                                                          IF      #DTR NOTSETIN @RCSR THEN
002742  032777  000002  176310          BIT     #DTR,@RCSR
002750  001001                        BNE     $20
                                                                ; DTR DID NOT SET IN RCSR
002752                                                          ERRHRD 15,,DIDNOT
002752  104017                        ERROR   15
002754                                                          ENDIF
002754                         $20:
                                                                ENDSUB
                                                ; TRY TO CLEAR IT AGAIN
002754                                                          BGNSUB
002754  012767  002762  176126          MOV     #64$,$LPERR
002762                                                              LET     @RCSR := @RCSR CLR.BY #DTR
002762  042777  000002  176270          BIC     #DTR,@RCSR
                                                                ; SHOULD HAVE CLEARED IT
002770                                                          IF      #DTR SETIN @RCSR THEN
002770  032777  000002  176262          BIT     #DTR,@RCSR
002776  001401                        BEQ     $21
                                                                ; DTR DID NOT CLEAR IN RCSR
003000                                                          ERRHRD 16,,DIDNOT
```

```
003000  104020              ERROR   16
003002                                              ENDIF
003002          $21:
003002                                          ENDSUB
003002                                      ENDTST
```

```
1539                                    ;;**********************************************************************
1540                                    ;;**********************************************************************
1541                                    ;*TEST 6          REQSEND - RCSR2          SET, CLEAR, RESET
1542                                    ;*               THIS TEST ASSUMES THAT ; MPER FR IS IN
1543                                    ;;**********************************************************************
1544  003002  000004                   TST6:   SCOPE
1545  003004  012767  000010  176146           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1546  003012  012767  000006  176160           MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1547  003020                                                                   IF #FRFD NOTSETIN $USWR THEN
1548  003020  032767  040000  176172           BIT     #FRFD,$USWR
1549  003026  001004                           BNE     $22
1550  003030                                                                       EXIT TST
1551  003030  012767  000001  176122           MOV     #1,$TIMES
1552  003036  000452                           BR      TST7            ;;;EXIT THIS TEST
1553  003040                                                                   ENDIF
1554  003040                           $22:
1555
1556                                                   ; SEE IF IT IS CLEAR
1557  003040                                                                   BGNSUB
1558  003040  012767  003046  176042           MOV     #64$,$LPERR
1559
1560  003046                                                                       IF      #REQSEND SETIN @RCSR THEN
1561  003046  032777  000004  176204           BIT     #REQSEND,@RCSR
1562  003054  001401                           BEQ     $23
1563                                                                                    ; REQSEND DID NOT RESET IN RCSR
1564  003056                                                                           ERRHRD 17,,DIDNOT
1565  003056  104021                           ERROR   17
1566  003060                                                                       ENDIF
1567  003060                           $23:
1568  003060                                                                   ENDSUB
1569
1570                                                   ; TRY TO SET REQSEND BIT
1571  003060                                                                   BGNSUB
1572  003060  012767  003066  176022           MOV     #64$,$LPERR
1573  003066                                                                       LET     @RCSR := @RCSR SET.BY #REQSEND
1574  003066  052777  000004  176164           BIS     #REQSEND,@RCSR
1575                                                                                ; STUCK TO 0
1576  003074                                                                       IF      #REQSEND NOTSETIN @RCSR THEN
1577  003074  032777  000004  176156           BIT     #REQSEND,@RCSR
1578  003102  001001                           BNE     $24
1579                                                                                    ; REQSEND DID NOT SET IN RCSR
1580  003104                                                                           ERRHRD 18,,DIDNOT
1581  003104  104022                           ERROR   18
1582  003106                                                                       ENDIF
1583  003106                           $24:
1584  003106                                                                   ENDSUB
1585
1586                                                   ; TRY TO CLEAR A SET BIT
1587  003106                                                                   BGNSUB
1588  003106  012767  003114  175774           MOV     #64$,$LPERR
1589
1590  003114                                                                       LET     @RCSR := @RCSR CLR.BY #REQSEND
1591  003114  042777  000004  176136           BIC     #REQSEND,@RCSR
1592                                                                                ; SHOULD HAVE CLEARED
1593  003122                                                                       IF      #REQSEND SETIN @RCSR THEN
1594  003122  032777  000004  176130           BIT     #REQSEND,@RCSR
```

```
1595  003130  001401                    BEQ     $25
1596                                                    ; REQSEND DID NOT CLEAR IN RCSR
1597  003132                                            ERRHRD 19,,DIDNOT
1598  003132  104023                    ERROR   19
1599  003134                                          ENDIF
1600  003134                          $25:
1601  003134                                        ENDSUB
1602
1603                                            ; NOW SEE IF RESET CLEARS IT
1604  003134                                        BGNSUB
1605  003134  012767  003142  175746    MOV     #64$,$LPERR
1606
1607  003142                                          LET     @RCSR := @RCSR SET.BY #REQSEND
1608  003142  052777  000004  176110    BIS     #REQSEND,@RCSR
1609                                                  ; ISSUE BUS RESET
1610  003150                                          BRESET
1611  003150  000005                    RESET
1612  003152                                          IF      #REQSEND SETIN @RCSR THEN
1613  003152  032777  000004  176100    BIT     #REQSEND,@RCSR
1614  003160  001401                    BEQ     $26
1615                                                    ; REQSEND DID NOT RESET IN RCSR
1616  003162                                            ERRHRD 20,,DIDNOT
1617  003162  104024                    ERROR   20
1618  003164                                          ENDIF
1619  003164                          $26:
1620  003164                                        ENDSUB
1621  003164                                        ENDTST
1622
1623
1624
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 7         SECXMIT - RCSR3        SET, CLEAR, RESET
                                ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
                                ;*              THAT IS READ AND WRITE.
                                ;;*****************************************************************
003164  000004                  TST7:   SCOPE
003166  012767  000010  175764          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003174  012767  000007  175776          MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                        ; SEE IF IT IS CLEAR
                                                                BGNSUB
003202
003202  012767  003210  175700          MOV     #64$,$LPERR
                                                                  IF    #SECXMIT SETIN @RCSR THEN
003210
003210  032777  000010  176042          BIT     #SECXMIT,@RCSR
003216  001401                          BEQ     $27
                                                                     ; SECXMIT DID NOT RESET IN RCSR
                                                                    ERRHRD 21,,DIDNOT
003220
003220  104025                          ERROR   21
                                                                  ENDIF
003222
003222                          $27:                                 ENDSUB
003222
                                                        ; TRY TO SET SECXMIT BIT
003222                                                          BGNSUB
003222  012767  003230  175660          MOV     #64$,$LPERR
                                                                  LET   @RCSR := @RCSR SET.BY #SECXMIT
003230
003230  052777  000010  176022          BIS     #SECXMIT,@RCSR
                                                                  ; STUCK TO 0
                                                                  IF    #SECXMIT NOTSETIN @RCSR THEN
003236
003236  032777  000010  176014          BIT     #SECXMIT,@RCSR
003244  001001                          BNE     $30
                                                                     ; SECXMIT DID NOT SET IN RCSR
                                                                    ERRHRD 22,,DIDNOT
003246
003246  104026                          ERROR   22
                                                                  ENDIF
003250
003250                          $30:                               ENDSUB
003250
                                                        ; TRY TO CLEAR A SET BIT
003250                                                          BGNSUB
003250  012767  003256  175632          MOV     #64$,$LPERR
                                                                  LET   @RCSR := @RCSR CLR.BY #SECXMIT
003256
003256  042777  000010  175774          BIC     #SECXMIT,@RCSR
                                                                  ; SHOULD HAVE CLEARED
                                                                  IF    #SECXMIT SETIN @RCSR THEN
003264
003264  032777  000010  175766          BIT     #SECXMIT,@RCSR
003272  001401                          BEQ     $31
                                                                     ; SECXMIT DID NOT CLEAR IN RCSR
                                                                    ERRHRD 23,,DIDNOT
003274
003274  104027                          ERROR   23
                                                                  ENDIF
003276
003276                          $31:                               ENDSUB
003276
```

```
1681  003276                                                          BGNSUB
1682  003276  012767  003304  175604          MOV     #64$,$LPERR
1683                                                    ; NOW SEE IF RESET CLEARS IT
1684
1685  003304                                                          LET     @RCSR := @RCSR SET.BY #SECXMIT
1686  003304  052777  000010  175746          BIS     #SECXMIT,@RCSR
1687                                                        ; ISSUE BUS RESET
1688  003312                                                          BRESET
1689  003312  000005                          RESET
1690  003314                                                          IF      #SECXMIT SETIN @RCSR THEN
1691  003314  032777  000010  175736          BIT     #SECXMIT,@RCSR
1692  003322  001401                          BEQ     $32
1693                                                        ; SECXMIT DID NOT RESET IN RCSR
1694  003324                                                          ERRHRD 24,,DIDNOT
1695  003324  104030                          ERROR   24
1696  003326                                                          ENDIF
1697  003326                          $32:
1698  003326                                                      ENDSUB
1699  003326                                                  ENDTST
1700
1701
1702
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 10        DATAIE - RCSR5 SET, CLEAR, RESET
                                ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
                                ;*              THAT IS READ AND WRITE.
                                ;;**********************************************************************
003326  000004                  TST10:  SCOPE
003330  012767  000010  175622          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003336  012767  000010  175634          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                ; SEE IF IT IS CLEAR
003344                                                          BGNSUB
003344  012767  003352  175536          MOV     #64$,$LPERR
                                                                IF      #DATAIE SETIN @RCSR THEN
003352
003352  032777  000040  175700          BIT     #DATAIE,@RCSR
003360  001401                          BEQ     $33
                                                                    ; DATAIE DID NOT RESET IN RCSR
                                                                    ERRHRD 25,,DIDNOT
003362
003362  104031                          ERROR   25
                                                                ENDIF
003364
003364                          $33:                            ENDSUB
003364
                                                ; TRY TO SET DATAIE BIT
003364                                                          BGNSUB
003364  012767  003372  175516          MOV     #64$,$LPERR
                                                                LET     @RCSR := @RCSR SET.BY #DATAIE
003372
003372  052777  000040  175660          BIS     #DATAIE,@RCSR
                                                            ; STUCK TO 0
003400                                                      IF      #DATAIE NOTSETIN @RCSR THEN
003400  032777  000040  175652          BIT     #DATAIE,@RCSR
003406  001001                          BNE     $34
                                                                    ; DATAIE DID NOT SET IN RCSR
                                                                    ERRHRD 26,,DIDNOT
003410
003410  104032                          ERROR   26
                                                                ENDIF
003412
003412                          $34:                            ENDSUB
003412
                                                ; TRY TO CLEAR A SET BIT
003412                                                          BGNSUB
003412  012767  003420  175470          MOV     #64$,$LPERR
                                                                LET     @RCSR := @RCSR CLR.BY #DATAIE
003420
003420  042777  000040  175632          BIC     #DATAIE,@RCSR
                                                            ; SHOULD HAVE CLEARED
003426                                                      IF      #DATAIE SETIN @RCSR THEN
003426  032777  000040  175624          BIT     #DATAIE,@RCSR
003434  001401                          BEQ     $35
                                                                    ; DATAIE DID NOT CLEAR IN RCSR
                                                                    ERRHRD 27,,DIDNOT
003436
003436  104033                          ERROR   27
                                                                ENDIF
003440
003440                          $35:                            ENDSUB
003440
```

```
1759                                                    ; NOW SEE IF RESET CLEARS IT
1760  003440                                                        BGNSUB
1761  003440  012767  003446  175442          MOV     #64$,$LPERR
1762
1763  003446                                                            LET     @RCSR := @RCSR SET.BY #DATAIE
1764  003446  052777  000040  175604          BIS     #DATAIE,@RCSR
1765                                                                ; ISSUE BUS RESET
1766  003454                                                        BRESET
1767  003454  000005                  RESET
1768  003456                                                        IF      #DATAIE SETIN @RCSR THEN
1769  003456  032777  000040  175574          BIT     #DATAIE,@RCSR
1770  003464  001401                  BEQ     $36
1771                                                                ; DATAIE DID NOT RESET IN RCSR
1772  003466                                                        ERRHRD 28,,DIDNOT
1773  003466  104034                  ERROR   28
1774  003470                                                        ENDIF
1775  003470                          $36:
1776  003470                                                    ENDSUB
1777  003470                                                ENDTST
1778
1779
1780
```

```
1781                                    ;;*****************************************************************
1782                                    ;;*****************************************************************
1783                                    ;*TEST 11        RCVRIE - RCSR6 SET, CLEAR, RESET
1784                                    ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
1785                                    ;*              THAT IS READ AND WRITE.
1786                                    ;;*****************************************************************
1787  003470  000004                    TST11:  SCOPE
1788  003472  012767  000010  175460            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1789  003500  012767  000011  175472            MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1790                                                            ; SEE IF IT IS CLEAR
1791  003506                                                            BGNSUB
1792  003506  012767  003514  175374            MOV     #64$,$LPERR
1793
1794  003514                                                                IF      #RCVRIE SETIN @RCSR THEN
1795  003514  032777  000100  175536            BIT     #RCVRIE,@RCSR
1796  003522  001401                            BEQ     $37
1797                                                                ; RCVRIE DID NOT RESET IN RCSR
1798  003524                                                            ERRHRD 29,,DIDNOT
1799  003524  104035                            ERROR   29
1800  003526                                                            ENDIF
1801  003526                                $37:
1802  003526                                                          ENDSUB
1803
1804                                                        ; TRY TO SET RCVRIE BIT
1805  003526                                                            BGNSUB
1806  003526  012767  003534  175354            MOV     #64$,$LPERR
1807  003534                                                                LET     @RCSR := @RCSR SET.BY #RCVRIE
1808  003534  052777  000100  175516            BIS     #RCVRIE,@RCSR
1809                                                                ; STUCK TO 0
1810  003542                                                            IF      #RCVRIE NOTSETIN @RCSR THEN
1811  003542  032777  000100  175510            BIT     #RCVRIE,@RCSR
1812  003550  001001                            BNE     $40
1813                                                                ; RCVRIE DID NOT SET IN RCSR
1814  003552                                                            ERRHRD 30,,DIDNOT
1815  003552  104036                            ERROR   30
1816  003554                                                            ENDIF
1817  003554                                $40:
1818  003554                                                          ENDSUB
1819
1820                                                        ; TRY TO CLEAR A SET BIT
1821  003554                                                            BGNSUB
1822  003554  012767  003562  175326            MOV     #64$,$LPERR
1823
1824  003562                                                                LET     @RCSR := @RCSR CLR.BY #RCVRIE
1825  003562  042777  000100  175470            BIC     #RCVRIE,@RCSR
1826                                                                ; SHOULD HAVE CLEARED
1827  003570                                                            IF      #RCVRIE SETIN @RCSR THEN
1828  003570  032777  000100  175462            BIT     #RCVRIE,@RCSR
1829  003576  001401                            BEQ     $41
1830                                                                ; RCVRIE DID NOT CLEAR IN RCSR
1831  003600                                                            ERRHRD 31,,DIDNOT
1832  003600  104037                            ERROR   31
1833  003602                                                            ENDIF
1834  003602                                $41:
1835  003602                                                          ENDSUB
1836
```

```
1837                                                    ; NOW SEE IF RESET CLEARS IT
1838  003602                                                        BGNSUB
1839  003602  012767  003610  175300          MOV     #64$,$LPERR
1840
1841  003610                                                            LET     @RCSR := @RCSR SET.BY #RCVRIE
1842  003610  052777  000100  175442          BIS     #RCVRIE,@RCSR
1843                                                                    ; ISSUE BUS RESET
1844  003616                                                            BRESET
1845  003616  000005                  RESET
1846  003620                                                            IF      #RCVRIE SETIN @RCSR THEN
1847  003620  032777  000100  175432          BIT     #RCVRIE,@RCSR
1848  003626  001401                  BEQ     $42
1849                                                                        ; RCVRIE DID NOT RESET IN RCSR
1850  003630                                                                ERRHRD 32,,DIDNOT
1851  003630  104040                  ERROR   32
1852  003632                                                            ENDIF
1853  003632                          $42:
1854  003632                                                            CKLOOP
1855  003632                                                        ENDSUB
1856  003632                                                ENDTST
1857
1858
1859
1860
```

```
                                ;;*****************************************************************
                                ;* THE FOLLOWING 4 TESTS VERIFY
                                ;* THAT RESET (INIT) INITIALIZES READ ONLY BITS.
                                ;*****************************************************************
                                ;*****************************************************************
                                ;*TEST 12        TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
                                ;*****************************************************************
003632  000004                  TST12:  SCOPE
003634  012767  000010  175316          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003642  012767  000012  175330          MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX



003650                                                          BGNSUB
003650  012767  003656  175232          MOV     #64$,$LPERR
003656                                      IF      #RCVRDONE SETIN @RCSR THEN
003656  032777  000200  175374          BIT     #RCVRDONE,@RCSR
003664  001402                          BEQ     $43

                                            ;RCVRDONE SHOULD HAVE CLEARED BY INIT
                                            ; RCVRDONE DID NOT CLEAR IN RCSR
003666                                      ERRHRD 33,HRESET, DIDNOT
003666  104041                  ERROR   33

                                            ;REISSUE RESET
003670                                      $RESET
003670  000005                  RESET
003672                                      ENDIF
003672                          $43:
                                                ;ALLOW LOOPING AFTER ERROR
003672                                      CKLOOP
003672                                  ENDSUB
003672                          ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 13        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
                                ;;**********************************************************************
                                ;;**********************************************************************
003672  000004                  $ST13:  SCOPE
003674  012767  000010  175256          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003702  012767  000013  175270          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


003710                                                                  BGNSUB
003710  012767  003716  175172          MOV     #64$,$LPERR

003716                                      IF      #XMITRDY NOTSETIN @TCSR THEN
003716  032777  000200  175340          BIT     #XMITRDY,@TCSR
003724  001002                          BNE     $44

                                            ;RESET SHOULD HAVE SET BIT.
                                            ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
003726                                      ERRHRD 34,HRESET,DIDNOT
003726  104042                          ERROR   34

                                            ;ISSUE ANOTHER RESET
003730                                      HRESET
003730  000005                          RESET

                                        ENDIF
003732  003732                  $44:
                                                    ;ALLOW LOOPING ON ERROR
003732                                          CKLOOP
003732                                      ENDSUB
003732                                  ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 14        TEST THAT DATAINT - RCSR 15 - IS CLEARED BY INIT.
                                ;;**********************************************************************
003732  000004                  TST14:  SCOPE
003734  012767  000010  175216          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003742  012767  000014  175230          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX


003750                                                  BGNSUB
003750  012767  003756  175132          MOV     #64$,$LPERR
003756                                      IF      #DATAINT SETIN @RCSR THEN
003756  032777  100000  175274          BIT     #DATAINT,@RCSR
003764  001402                          BEQ     $45

003766                                          ERRHRD 35, HRESET, DIDNOT
003766  104043                          ERROR   35

                                        ;TESTING EFFECT OF RESET ON BIT

                                        ;DATAINT DID NOT CLEAR IN RCSR
                                        ;ALLOW A FRESH START
003770                                  @RESET
003770  000005                          RESET
003772                                      ENDIF
003772                          $45:
003772                                      CKLOOP
003772                                  ENDSUB
003772                              ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;;*TEST 15        TEST THAT RCVRACT - RCSR 11 - 15 CLEARED BY INIT
                                ;;**********************************************************************
003772  000004                  TST15:  SCOPE
003774  012767  000010  175156          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004002  012767  000015  175170          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

004010                                                          IF      #CABLE NOTSETIN $USWR THEN
004010  032767  020000  175202          BIT     #CABLE,$USWR
004016  001004                          BNE     $46
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
004020                                                              EXIT TST
004020  012767  000001  175132          MOV     #1,$TIMES
004026  000411                          BR      TST16           ;;;EXIT THIS TEST
004030                                                          ENDIF
004030                          $46:

004030                                                          BGNSUB
004030  012767  004036  175052          MOV     #64$,$LPERR

004036                                                              IF      #RCVRACT SETIN @RCSR THEN
004036  032777  004000  175214          BIT     #RCVRACT,@RCSR
004044  001402                          BEQ     $47

                                                                        ;RESET SHOULD HAVE CLEARED RCVRACT
004046                                                                  ERRHRD 36, HRESET, DIDNOT
004046  104044                          ERROR   36

                                                                        ;TESTING EFFECT OF RESET ON BIT

                                                                        ;RCVRACT DID NOT CLEAR IN RCSR

                                                                        ;ALLOW ANOTHER TRY
004050                                                                  @RESET
004050  000005                          RESET
004052                                                              ENDIF
004052                          $47:

                                                                        ;ALLOW LOOPING ON ERROR
004052                                                              CKLOOP
004052                                                          ENDSUB
004052                                                      ENDTST
```

```
2012                                    ;;*****************************************************************
2013                                    ;* THE FOLLOWING 4 TESTS VERIFY
2014                                    ;* THAT THE EIA SIGNALS CAN BE TRANSMITTED
2015                                    ;* AND RECEIVED THROUGH THE CABLE
2016                                    ;;*****************************************************************
2017                                    ;;*****************************************************************
2018                                    ;*TEST 16        TEST THAT CARDET SETS AND CLEARS
2019                                    ;*              AS  DTR   SETS AND CLEARS
2020                                    ;;*****************************************************************
2021  004052  000004                    TST16:  SCOPE
2022  004054  012767  000010  175076            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2023  004062  012767  000016  175110            MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2024                                                                            ; CAN WE USE THE WRAPAROUND??
2025  004070                                                                IF      #CABLE NOTSETIN $USWR THEN
2026  004070  032767  020000  175122            BIT     #CABLE,$USWR
2027  004076  001004                            BNE     $50
2028                                                                            ; CAN'T TEST WITHOUT BERG OR H315.
2029  004100                                                                EXIT TST
2030  004100  012767  000001  175052            MOV     #1,$TIMES
2031  004106  000441                            BR      TST17           ;;;EXIT THIS TEST
2032  004110                                                                ENDIF
2033  004110                                $50:
2034
2035                                                                            ; DTR AND
2036                                                                            ; CARDET ARE CONNECTED
2037
2038                                                                        ; BY THE H315 OR EQUIV.
2039
2040                                                            ; CLEAR
2041  004110                                                            BGNSUB
2042  004110  012767  004116  174772            MOV     #64$,$LPERR
2043
2044                                                                        ; CLEAR  DTR
2045  004116                                                            LET     @RCSR := @RCSR CLR.BY #DTR
2046  004116  042777  000002  175134            BIC     #DTR,@RCSR
2047                                                                        ; CARDET SHOULD FOLLOW
2048  004124                                                            IF      #CARDET SETIN @RCSR THEN
2049  004124  032777  010000  175126            BIT     #CARDET,@RCSR
2050  004132  001401                            BEQ     $51
2051                                                                        ; CARDET DID NOT
2052  004134                                                            ERRHRD 37,,FORCE
2053  004134  104045                            ERROR   37
2054
2055                                                                        ;  CLEAR WITH DTR
2056  004136                                                            ENDIF
2057  004136                                $51:
2058  004136                                                        ENDSUB
2059
2060                                                            ; SET
2061  004136                                                        BGNSUB
2062  004136  012767  004144  174744            MOV     #64$,$LPERR
2063
2064                                                                        ; SET DTR
2065  004144                                                            LET     @RCSR := @RCSR SET.BY #DTR
2066  004144  052777  000002  175106            BIS     #DTR,@RCSR
2067                                                                        ; CARDET SHOULD FOLLOW
```

```
2068  004152
2069  004152  032777  010000  175100          BIT     #CARDET,@RCSR
2070  004160  001001                          BNE     $52
2071
2072  004162
2073  004162  104046                          ERROR   38
2074
2075
2076  004164
2077  004164                          $52:
2078  004164
2079
2080                                                    ; CLEAR
2081  004164
2082  004164  012767  004172  174716          MOV     #64$,$LPERR
2083
2084
2085  004172
2086  004172  042777  000002  175060          BIC     #DTR,@RCSR
2087
2088  004200
2089  004200  032777  010000  175052          BIT     #CARDET,@RCSR
2090  004206  001401                          BEQ     $53
2091
2092  004210
2093  004210  104047                          ERROR   39
2094
2095
2096  004212
2097  004212                          $53:
2098  004212                                                      ENDSUB
2099  004212                                                  ENDTST
2100
2101
2102
2103
```

```
    IF      #CARDET NOTSETIN @RCSR THEN

        ; CARDET DID NOT SET
        ERRHRD 38,,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 39,,FORCE

        ;  CLEAR WITH DTR
    ENDIF
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 17        TEST THAT CLRSEND SETS AND CLEARS
                                ;*              AS  DTR   SETS AND CLEARS
                                ;;**********************************************************************
004212  000004                  TST17:  SCOPE
004214  012767  000010  174736          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004222  012767  000017  174750          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
004230                                                          IF      @CABLE NOTSETIN $USWR THEN
004230  032767  020000  174762          BIT     #CABLE,$USWR
004236  001004                        BNE     $54
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
004240                                                              EXIT TST
004240  012767  000001  174712          MOV     #1,$TIMES
004246  000441                        BR      TST20           ;;;EXIT THIS TEST
004250                                                          ENDIF
004250                          $54:
                                                                        ; DTR AND
                                                                        ; CLRSEND ARE CONNECTED
                                                                ; BY THE H315 OR EQUIV.
                                                ; CLEAR
004250                                                          BGNSUB
004250  012767  004256  174632          MOV     #64$,$LPERR
                                                                ; CLEAR  DTR
                                                            LET   @RCSR := @RCSR CLR.BY #DTR
004256                                                      ; CLRSEND SHOULD FOLLOW
004256  042777  000002  174774          BIC     #DTR,@RCSR  IF    #CLRSEND SETIN @RCSR THEN
004264
004264  032777  020000  174766          BIT     #CLRSEND,@RCSR
004272  001401                        BEQ     $55
004274                                                          ; CLRSEND DID NOT
004274  104050                        ERROR   40              ERRHRD 40,,.FORCE
                                                                ;  CLEAR WITH DTR
                                                            ENDIF
004276
004276                          $55:                    ENDSUB
004276
                                                ; SET
004276                                                      BGNSUB
004276  012767  004304  174604          MOV     #64$,$LPERR
                                                                ; SET DTR
                                                            LET   @RCSR := @RCSR SET.BY #DTR
004304                                                      ; CLRSEND SHOULD FOLLOW
004304  052777  000002  174746          BIS     #DTR,@RCSR  IF    #CLRSEND NOTSETIN @RCSR THEN
004312
004312  032777  020000  174740          BIT     #CLRSEND,@RCSR
004320  001001                        BNE     $56
                                                                ; CLRSEND DID NOT SET
```

```
2160  004322
2161  004322  104051                    ERROR   41
2162
2163
2164  004324
2165  004324                        $56:
2166  004324
2167
2168                                                ; CLEAR
2169  004324
2170  004324  012767  004332  174556    MOV     #64$,$LPERR
2171
2172
2173  004332
2174  004332  042777  000002  174720    BIC     #DTR,@RCSR
2175
2176  004340
2177  004340  032777  020000  174712    BIT     #CLRSEND,@RCSR
2178  004346  001401                    BEQ     $57
2179
2180  004350
2181  004350  104052                    ERROR   42
2182
2183
2184  004352
2185  004352                        $57:
2186  004352
2187  004352                            ENDTST
2188
2189
2190
2191
```

```
        ERRHRD 41,,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 42,,FORCE

        ;  CLEAR WITH DTR
    ENDIF
ENDSUB
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 20        TEST THAT RING SETS AND CLEARS
                                ;*              AS  REQSEND    SETS AND CLEARS
                                ;;*****************************************************************
004352  000004                  TST20:  SCOPE
004354  012767  000010  174576          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004362  012767  000020  174610          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
                                                                IF      #CABLE NOTSETIN $USWR THEN
004370
004370  032767  020000  174622          BIT     #CABLE,$USWR
004376  001004                          BNE     $60
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
                                                                    EXIT TST
004400
004400  012767  000001  174552          MOV     #1,$TIMES
004406  000441                          BR      TST21           ;;;EXIT THIS TEST
                                                                ENDIF
004410
004410                          $60:
                                                                        ; REQSEND AND
                                                                        ; RING ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

                                ; CLEAR
004410                                                          BGNSUB
004410  012767  004416  174472          MOV     #64$,$LPERR

                                                                    ; CLEAR  REQSEND
                                                                LET     @RCSR := @RCSR CLR.BY #REQSEND
004416
004416  042777  000004  174634          BIC     #REQSEND,@RCSR
                                                                    ; RING SHOULD FOLLOW
004424                                                          IF      #RING SETIN @RCSR THEN
004424  032777  040000  174626          BIT     #RING,@RCSR
004432  001401                          BEQ     $61

                                                                    ; RING DID NOT
                                                                ERRHRD 43,,FORCE

004434
004434  104053                          ERROR   43
                                                                    ;  CLEAR WITH REQSEND
                                                                ENDIF
004436
004436                          $61:                        ENDSUB
004436

                                ; SET                       BGNSUB
004436
004436  012767  004444  174444          MOV     #64$,$LPERR

                                                                    ; SET REQSEND
                                                                LET     @RCSR := @RCSR SET.BY #REQSEND
004444
004444  052777  000004  174606          BIS     #REQSEND,@RCSR
                                                                    ; RING SHOULD FOLLOW
004452                                                          IF      #RING NOTSETIN @RCSR THEN
004452  032777  040000  174600          BIT     #RING,@RCSR
004460  001001                          BNE     $62

                                                                    ; RING DID NOT SET
```

```
2248  004462
2249  004462  104054                    ERROR   44
2250
2251
2252  004464
2253  004464                        $62:
2254  004464
2255
2256                                                ; CLEAR
2257  004464
2258  004464  012767  004472  174416    MOV     #64$,$LPERR
2259
2260
2261  004472
2262  004472  042777  000004  174560    BIC     #REQSEND.@RCSR
2263
2264  004500
2265  004500  032777  040000  174552    BIT     #RING,@RCSR
2266  004506  001401                    BEQ     $63
2267
2268  004510
2269  004510  104055                    ERROR   45
2270
2271
2272  004512
2273  004512                        $63:
2274  004512
2275  004512
2276
2277
2278
2279
```

```
        ERRHRD 44,,FORCE

    ;  WITH REQSEND
    ENDIF

ENDSUB

BGNSUB

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

    ; RING SHOULD FOLLOW
    IF     #RING SETIN @RCSR THEN

    ; RING DID NOT
    ERRHRD 45,,FORCE

    ;  CLEAR WITH REQSEND
    ENDIF

        ENDSUB
ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 21        TEST THAT SECREC SETS AND CLEARS
                                ;*              AS  SECXMIT    SETS AND CLEARS
                                ;;*****************************************************************
004512  000004                  TST21:  SCOPE
004514  012767  000010  174436          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004522  012767  000021  174450          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
                                                                IF      #CABLE NOTSETIN $USWR THEN
004530
004530  032767  020000  174462          BIT     #CABLE,$USWR
004536  001004                          BNE     $64
                                                                    ; CAN'T TEST WITHOUT BERG OR H315.
                                                                EXIT TST
004540
004540  012767  000001  174412          MOV     #1,$TIMES
004546  000441                          BR      TST22           ;;;EXIT THIS TEST
                                                                ENDIF
004550                          $64:

                                                                    ; SECXMIT AND
                                                                    ; SECREC ARE CONNECTED

                                                                ; BY THE H315 OR EQUIV.

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

004556
004556  042777  000010  174474          BIC     #SECXMIT,@RCSR
                                                                    ; CLEAR  SECXMIT
                                                                LET     @RCSR := @RCSR CLR.BY #SECXMIT
004564
004564  032777  002000  174466          BIT     #SECREC,@RCSR
004572  001401                          BEQ     $65
                                                                    ; SECREC SHOULD FOLLOW
                                                                IF      #SECREC SETIN @RCSR THEN
004574
004574  104056                          ERROR   46
                                                                    ; SECREC DID NOT
                                                                    ERRHRD 46,,FORCE
004576                                                              ;  CLEAR WITH SECXMIT
                                                                ENDIF
004576                          $65:
004576
                                                        ; SET
004576                                                          BGNSUB
004576  012767  004604  174304          MOV     #64$,$LPERR

004604
004604  052777  000010  174446          BIS     #SECXMIT,@RCSR
                                                                    ; SET SECXMIT
                                                                LET     @RCSR := @RCSR SET.BY #SECXMIT
004612
004612  032777  002000  174440          BIT     #SECREC,@RCSR
004620  001001                          BNE     $66
                                                                    ; SECREC SHOULD FOLLOW
                                                                IF      #SECREC NOTSETIN @RCSR THEN
                                                                    ; SECREC DID NOT SET
```

```
2336  004622
2337  004622  104057                  ERROR   47
2338
2339
2340  004624
2341  004624                  $66:
2342  004624
2343
2344                                            ; CLEAR
2345  004624
2346  004624  012767  004632  174256          MOV     #64$,$LPERR
2347
2348
2349  004632
2350  004632  042777  000010  174420          BIC     #SECXMIT,@RCSR
2351
2352  004640
2353  004640  032777  002000  174412          BIT     #SECREC,@RCSR
2354  004646  001401                          BEQ     $67
2355
2356  004650
2357  004650  104060                  ERROR   48
2358
2359
2360  004652
2361  004652                  $67:
2362  004652
2363  004652
2364
2365
2366
2367
```

```
        ERRHRD 47,,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 48,.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
                                ;;**********************************************************************
004652  000004                  TST22:  SCOPE
004654  012767  000010  174276          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004662  012767  000022  174310          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
004670                                                          IF      #CABLE NOTSETIN $USWR THEN
004670  032767  020000  174322          BIT     #CABLE,$USWR
004676  001004                        BNE     $70
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
004700                                                                  EXIT TST
004700  012767  000001  174252          MOV     #1,$TIMES
004706  000463                        BR      TST23           ;;;EXIT THIS TEST
004710                                                          ENDIF
004710                          $70:
                                                                        ;MAKE SURE NOTHING UNEXPECTED HAPPENS
004710  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
004714  012746  004722                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
004720  000002                        RTI                     ;;POP NEW PC AND PS
004722                          64$:
                                                                ;READ TWICE - CLEARS
                                                                BGNSUB
004722                                  
004722  012767  004730  174160          MOV     #65$,$LPERR
                                                                        ; CLEAR DTR
                                                                LET     @RCSR := @RCSR CLR.BY #DTR
004730                                  
004730  042777  000002  174322          BIC     #DTR,@RCSR
                                                                        ;WAIT 10 MICRO-SEC FOR CABLE
004736                                                          WAITUS  10
004736  010546                        MOV     R5, (SP)
004740  012745  000010                  MOV     #10,-(R5)
004744  004767  004544                  JSR     PC,WAIT
004750  012605                        MOV     (SP)+,R5
                                                                        ; READ RCSR - TO CLEAR DATAINT
004752                                                          LET     R3 := @RCSR
004752  017703  174302                  MOV     @RCSR,R3
                                                                        ; READ RCSR AGAIN
004756                                                          IF      #DATAINT SETIN @RCSR THEN
004756  032777  100000  174274          BIT     #DATAINT,@RCSR
004764  001401                        BEQ     $71
                                                                        ; READING RCSR DID NOT CLEAR DATAINT
004766                                                                  ERRHRD 49,EDATAINT
004766  104061                        ERROR   49
004770                                                          ENDIF
004770                          $71:
                                
004770                                                          ENDSUB
                                
                                                        ;  DTR SETTING SETS DATAINT
004770                                                          BGNSUB
```

```
2424  004770  012767  004776  174112          MOV     #64$,$LPERR
2425
2426
2427  004776
2428  004776  052777  000002  174254          BIS     #DTR,@RCSR
2429  005004
2430  005004  032777  100000  174246          BIT     #DATAINT,@RCSR
2431  005012  001001                          BNE     $72
2432
2433  005014
2434  005014  104062                          ERROR   50
2435  005016
2436  005016                          $72:
2437
2438  005016
2439  005016  032777  100000  174234          BIT     #DATAINT,@RCSR
2440  005024  001401                          BEQ     $73
2441
2442  005026
2443  005026  104063                          ERROR   51
2444  005030
2445  005030                          $73:
2446  005030
2447
2448                                                  ; DTR CLEARING SETS DATAINT
2449  005030
2450  005030  012767  005036  174052          MOV     #64$,$LPERR
2451
2452
2453  005036
2454  005036  042777  000002  174214          BIC     #DTR,@RCSR
2455  005044
2456  005044  032777  100000  174206          BIT     #DATAINT,@RCSR
2457  005052  001001                          BNE     $74
2458
2459  005054
2460  005054  104064                          ERROR   52
2461  005056
2462  005056                          $74:
2463  005056
2464  005056
2465
2466
2467
```

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

    IF      #DATAINT NOTSETIN @RCSR THEN

        ;SETTING DTR DID NOT SET DATAINT
        ERRHRD 50,, E2DATA

    ENDIF

    IF      #DATAINT SETIN @RCSR THEN

        ;READING RCSR DID NOT CLEAR DATAINT
        ERRHRD 51,EDATAINT

    ENDIF

ENDSUB

    BGNSUB

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

    IF      #DATAINT NOTSETIN @RCSR THEN

        ;CLEARING DTR DID NOT SET DATAINT
        ERRHRD 52,,E2DATA

    ENDIF

ENDSUB
ENDTST
```

```
2468                                    ;;**************************************************************
2469                                    ;;**************************************************************
2470                                    ;*TEST 23        TEST THAT DATAINT SETS WHEN RING SETS
2471                                    ;*              AND THAT DATAINT DOES NOT SET WHEN RING CLEARS
2472                                    ;;**************************************************************
2473  005056  000004                    TST23:  SCOPE
2474  005060  012767  000010  174072            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2475  005066  012767  000023  174104            MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2476                                                                            ; CAN WE USE THE WRAPAROUND??
2477  005074                                                                IF      #CABLE NOTSETIN $USWR THEN
2478  005074  032767  020000  174116            BIT     #CABLE,$USWR
2479  005102  001004                            BNE     $75
2480                                                                            ; CAN'T TEST WITHOUT BERG OR H315.
2481  005104                                                                EXIT TST
2482  005104  012767  000001  174046            MOV     #1,$TIMES
2483  005112  000473                            BR      TST24           ;;;EXIT THIS TEST
2484  005114                                                                ENDIF
2485  005114                            $75:
2486
2487                                                                            ;NO INTERRUPTS
2488  005114  012746  000340                    MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
2489  005120  012746  005126                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
2490  005124  000002                            RTI                     ;;POP NEW PC AND PS
2491  005126                            64$:
2492
2493                                                            ;START OFF WITH EVERYTHING CLEAR
2494  005126                                                                BGNSUB
2495  005126  012767  005134  173754            MOV     #65$,$LPERR
2496
2497                                                                        ;CLEAR RING
2498  005134                                                            LET     @RCSR := @RCSR CLR.BY #REQSEND
2499  005134  042777  000004  174116            BIC     #REQSEND,@RCSR
2500                                                                        ;WAIT 10 MICRO-SEC FOR CABLE
2501  005142                                                            WAITUS  10
2502  005142  010546                            MOV     R5,-(SP)
2503  005144  012745  000010                    MOV     #10,-(R5)
2504  005150  004767  004340                    JSR     PC,WAIT
2505  005154  012605                            MOV     (SP)+,R5
2506
2507  005156                                                                ;READ ONCE
2508  005156  017703  174076                                           LET     R3 := @RCSR
2509                                                                        ;READ TWICE
2510  005162                                                            IF      #DATAINT SETIN @RCSR THEN
2511  005162  032777  100000  174070            BIT     #DATAINT,@RCSR
2512  005170  001401                            BEQ     $76
2513                                                                        ;READING RCSR DID NOT CLEAR DATAINT
2514  005172                                                                ERRHRD 53, EDATAINT
2515  005172  104065                            ERROR   53
2516  005174                                                            ENDIF
2517  005174                            $76:
2518  005174                                                        ENDSUB
2519
2520                                                    ;       SET RING --> SET DATAINT
2521  005174                                                                BGNSUB
2522  005174  012767  005202  173706            MOV     #64$,$LPERR
2523
```

```
2524                                                                    ;WE ARE SETTING RING
2525                                                                    ; SET RING
2526  005202                                                        LET     @RCSR := @RCSR SET.BY #REQSEND
2527  005202  052777  000004  174050          BIS     #REQSEND,@RCSR
2528                                                                    ;WAIT 10 MICRO-SEC FOR CABLE
2529  005210                                                        WAITUS  10
2530  005210  010546                          MOV     R5,-(SP)
2531  005212  012745  000010                  MOV     #10,-(R5)
2532  005216  004767  004272                  JSR     PC,WAIT
2533  005222  012605                          MOV     (SP)+,R5
2534  005224                                                        IF      #DATAINT NOTSETIN @RCSR THEN
2535  005224  032777  100000  174026          BIT     #DATAINT,@RCSR
2536  005232  001001                          BNE     $77
2537                                                                    ;SETTING RING DID NOT SET DATAINT
2538  005234                                                            ERRHRD 82,, E2DATA
2539  005234  104122                          ERROR   82
2540  005236                                                        ENDIF
2541  005236                          $77:
2542  005236                                                    ENDSUB
2543
2544                                                        ;CLEAR RING CAUSES NO CHANGE IN DATAINT (CLEAR)
2545  005236                                                    BGNSUB
2546  005236  012767  005244  173644          MOV     #64$,$LPERR
2547
2548                                                                    ;CLEAR RING
2549  005244                                                        LET     @RCSR := @RCSR CLR.BY #REQSEND
2550  005244  042777  000004  174006          BIC     #REQSEND,@RCSR
2551                                                                    ;WAIT 10 MICRO-SEC FOR CABLE
2552  005252                                                        WAITUS  10
2553  005252  010546                          MOV     R5,-(SP)
2554  005254  012745  000010                  MOV     #10,-(R5)
2555  005260  004767  004230                  JSR     PC,WAIT
2556  005264  012605                          MOV     (SP)+,R5
2557  005266                                                        IF      #DATAINT SETIN @RCSR THEN
2558  005266  032777  100000  173764          BIT     #DATAINT,@RCSR
2559  005274  001401                          BEQ     $100
2560                                                                    ;CLEARING RING SET DATAINT
2561  005276                                                            ERRHRD 83, CLRRNG
2562  005276  104123                          ERROR   83
2563  005300                                                        ENDIF
2564  005300                          $100:
2565  005300                                                    ENDSUB
2566  005300                                                    EXIT    ;SKIP AROUND MESSAGE
2567  005300  000400                          BR      TST24           ;;;EXIT THIS TEST
2568  005302                                                        ENDTST
2569
2570
2571
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 24        TEST THAT DATAINT SETS WHEN SECREC CHANGES STATE
                                ;;*****************************************************************
005302  000004                  $TST24: SCOPE
005304  012767  000010  173646          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005312  012767  000024  173660          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ; CAN WE USE THE WRAPAROUND??
005320                                                                  IF      #CABLE NOTSETIN $USWR THEN
005320  032767  020000  173672          BIT     #CABLE,$USWR
005326  001004                        BNE     $101
                                                                        ; CAN'T TEST WITHOUT BERG OR H315.
                                                                      .IT'TST
005330
005330  012767  000001  173622          MOV     #1,$TIMES
005336  000454                        BR      TST25           ;;;EXIT THIS TEST
                                                                ENDIF
005340                          $101:
                                                                        ;NO INTERRUPTS
005340  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
005344  012746  005352                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
005350  000002                        RTI                     ;;POP NEW PC AND PS
005352                          64$:
                                                                ;START FRESH
                                                                        ;CLEAR SECREC
005352                                                          LET     @RCSR := @RCSR CLR.BY #SECXMIT
005352  042777  000010  173700          BIC     #SECXMIT,@RCSR
005360                                                          LET     R3 := @RCSR
005360  017703  173674                  MOV     @RCSR,R3
                                                        ;SET SECREC --> DATAINT SET
                                                        BGNSUB
005364
005364  012767  005372  173516          MOV     #65$,$LPERR
                                                                    ;SET SECREC
005372                                                          LET     @RCSR := @RCSR SET.BY #SECXMIT
005372  052777  000010  173660          BIS     #SECXMIT,@RCSR
                                                                    ;WAIT 10 MICRO-SEC FOR CABLE
005400                                                          WAITUS  10
005400  010546                        MOV     R5,-(SP)
005402  012745  000010                MOV     #10,-(R5)
005406  004767  004102                JSR     PC,WAIT
005412  012605                        MOV     (SP)+,R5
005414                                                          IF      #DATAINT NOTSETIN @RCSR THEN
005414  032777  100000  173636          BIT     #DATAINT,@RCSR
005422  001001                        BNE     $102
                                                                    ;SETTING SECREC DID NOT SET DATAINT
                                                                    ERRHRD 84,, E2DATA
005424
005424  104124                          ERROR   84
                                                                ENDIF
005426                          $102:
005426                                                  ENDSUB
                                                ;CLEAR SECREC --> DATAINT SET
```

```
2628  005426
2629  005426  012767  005434  173454          MOV     #64$,$LPERR
2630
2631  005434
2632  005434  042777  000010  173616          BIC     #SECXMIT,@RCSR
2633
2634  005442
2635  005442  010546                          MOV     R5,-(SP)
2636  005444  012745  000010                  MOV     #10,-(R5)
2637  005450  004767  004040                  JSR     PC,WAIT
2638  005454  012605                          MOV     (SP)+,R5
2639  005456
2640  005456  032777  100000  173574          BIT     #DATAINT,@RCSR
2641  005464  001001                          BNE     $103
2642
2643  005466
2644  005466  104125                          ERROR   85
2645  005470
2646  005470                          $103:
2647  005470
2648  005470
2649
2650
2651
```

```
BGNSUB

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

    ;WAIT 10 MICRO-SEC FOR CABLE
WAITUS  10



IF      #DATAINT NOTSETIN @RCSR THEN

    ;CLEARING SECREC DID NOT SET DATAINT
    ERRHRD 85,, E2DATA

ENDIF

ENDSUB
ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 25        TEST THAT XMIT RDY - TCSR 7 - CLEARS
                                ;*              WHEN TBUF IS LOADED WITH A CHARACTER
                                ;*              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
                                ;;**********************************************************************
005470  000004                  TST25:  SCOPE
005472  012767  000010  173460          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005500  012767  000025  173472          MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

005506                                                          BGNSUB
005506  012767  005514  173374          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
005514
005514  105077  173550                  CLRB    @TBUF

                                                                ;WAIT A MAXIMUM
                                                                ;OF 50 MSEC FOR
                                                                ;XMIT RDY TO SET IN TCSR
005520                                                      CALL TIMER IN (#5,#XMITRDY,TCSR,#SET)
005520  010546                          MOV     R5,-(SP)
005522  012745  177777                  MOV     #SET,-(R5)
005526  016745  173532                  MOV     TCSR,-(R5)
005532  012745  000200                  MOV     #XMITRDY,-(R5)
005536  012745  000005                  MOV     #5,-(R5)
005542  004767  003470                  JSR     PC,TIMER
005546  012605                          MOV     (SP)+,R5
                                                                ;TIMER RETURNS AN ERROR IF BIT DID
                                                                ;NOT MEET CONDITION WITHIN TIME LIMIT
005550                                                      IF.ERROR THEN
005550  103001                          BCC     $104
                                                                ;XMIT RDY DID NOT SET IN TCSR
005552                                                          ERRHRD 54,,DIDNOT
005552  104066                          ERROR   54
005554                                                      ENDIF
005554                          $104:
                                                        ENDSUB
                                                        BGNSUB
005554
005554  012767  005562  173326          MOV     #64$,$LPERR
                                            ; LOAD TBUF WITH A SECOND CHARACTER
                                            ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
                                            ; AND THEN WAIT FOR IT TO SET

                                                                ;SEND SECOND CHARACTER
005562                                                      LET @TBUF :B= #0
005562  105077  173502                  CLRB    @TBUF
005566  000240                                  NOP             ; GIVE IT TIME TO CLEAR
                                                                ; XMITRDY SHOULD HAVE CLEARED UPON
                                                                ; RECEIPT OF A CHARACTER
005570                                                      IF #XMITRDY SETIN @TCSR THEN
005570  032777  000200  173466          BIT     #XMITRDY,@TCSR
```

```
2708  005576  001401                    BEQ     $105
2709                                                          ; XMITRDY DID NOT CLEAR IN TCSR
2710  005600                                                  ERRHRD 55,,DIDNOT
2711  005600  104067                    ERROR   55
2712  005602                                                  ENDIF
2713  005602                          $105:
2714
2715                                                          ;WAIT A MAXIMUM
2716                                                          ;OF 50 MSEC FOR
2717                                                          ;XMIT RDY TO SET IN TCSR
2718  005602                                                  CALL TIMER IN <#5,#XMITRDY,TCSR,#SET>
2719  005602  010546                    MOV     R5,-(SP)
2720  005604  012745  177777            MOV     #SET,-(R5)
2721  005610  016745  173450            MOV     TCSR,-(R5)
2722  005614  012745  000200            MOV     #XMITRDY,-(R5)
2723  005620  012745  000005            MOV     #5,-(R5)
2724  005624  004767  003406            JSR     PC,TIMER
2725  005630  012605                    MOV     (SP)+,R5
2726  005632                                                  IF.ERROR THEN
2727  005632  103001                    BCC     $106
2728                                                          ;XMIT RDY DID NOT SET IN TCSR
2729  005634                                                  ERRHRD 56,,DIDNOT
2730  005634  104070                    ERROR   56
2731  005636                                                  ENDIF
2732  005636                          $106:
2733  005636                                              ENDSUB
2734  005636                                          ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 26        TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)
                                ;*              RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
                                ;*              AND THAT RESET CLEARS THE BIT.
                                ;;*****************************************************************
005636  000004                TST26:  SCOPE
005640  012767  000010  173312        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005646  012767  000026  173324        MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                                ; SET THE MAINTENANCE BIT
005654                                                      LET @TCSR := @TCSR SET.BY #MAINT
005654  052777  000004  173402        BIS     #MAINT,@TCSR

005662                                                      BGNSUB
005662  012767  005670  173220        MOV     #64$,$LPERR
                                        ; SEND A CHARACTER AND LET IT WRAP AROUND

005670                                                          LET @TBUF :B= #0
005670  105077  173374                CLRB    @TBUF

                                                                ; WAIT A MAXIMUM OF 50 MSEC
                                                                ; FOR RCVR DONE TO SET IN
                                                                ; RCSR
005674                                                          CALL TIMER IN <#5,#RCVRDONE,RCSR,#SET>
005674  010546                        MOV     R5,-(SP)
005676  012745  177777                MOV     #SET,-(R5)
005702  016745  173352                MOV     RCSR,-(R5)
005706  012745  000200                MOV     #RCVRDONE,-(R5)
005712  012745  000005                MOV     #5,-(R5)
005716  004767  003314                JSR     PC,TIMER
005722  012605                        MOV     (SP)+,R5

                                                            ;DIDN'T SET IN TIME
005724                                                      IF.ERROR THEN
005724  103001                        BCC     $107

                                                                ; RCVRDONE DID NOT SET IN RCSR
                                                                ERRHRD 57,,DIDNOT
005726
005726  104071                        ERROR   57
005730                                                          ENDIF
005730                          $107:

005730                                                      ENDSUB

005730                                                      BGNSUB
005730  012767  005736  173152        MOV     #64$,$LPERR
                                        ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
                                        ; THIS ALSO WILL CLEAR THE MAINT. BIT
005736                                                          BRESET
005736  000005                        RESET

005740                                                          IF #RCVRDONE SETIN @RCSR THEN
005740  032777  000200  173312        BIT     #RCVRDONE,@RCSR
005746  001401                        BEQ     $110

                                                                ; RCVRDONE DID NOT RESET IN RCSR.
                                                                ERRHRD 58,,DIDNOT
005750
```

```
2791  005750  104072                ERROR   58
2792  005752                                                        ENDIF
2793  005752                $110:
2794  005752                                                    ENDSUB
2795  005752                                                ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 27        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
                                ;;**********************************************************************
005752  000004                  TST27:  SCOPE
005754  012767  000010  173176          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005762  012767  000027  173210          MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                                        ; SET MAINT. BIT
005770                                                          LET @TCSR := @TCSR SET.BY #MAINT
005770  052777  000004  173266          BIS     #MAINT,@TCSR
005776                                                          BGNSUB
005776  012767  006004  173104          MOV     #6$,$LPERR
                                            ; OUTPUT A CHARACTER WITH MAINTENANCE
                                            ; SET, AND WAIT FOR XMITRDY TO SET.

                                                                        ; OUTPUT A CHARACTER
006004                                                              LET @TBUF :B= #0
006004  105077  173260                  CLRB    @TBUF

                                                                    ; WAIT MAXIMUM OF 500 MSEC
                                                                    ; FOR RCVRDONE TO SET IN
                                                                    ; RCSR
006010                                                              CALL TIMER IN <#50,#RCVRDONE,RCSR,#SET>
006010  010546                          MOV     R5,-(SP)
006012  012745  177777                  MOV     #SET,-(R5)
006016  016745  173236                  MOV     RCSR,-(R5)
006022  012745  000200                  MOV     #RCVRDONE,-(R5)
006026  012745  000050                  MOV     #50,-(R5)
006032  004767  003200                  JSR     PC,TIMER
006036  012605                          MOV     (SP)+,R5

                                                                        ; DID IT BECAME READY?
006040                                                              IF.ERROR  THEN
006040  103001                  BCC     $111

                                                                        ;RCVRDONE DID NOT SET IN RCSR
                                                                        ERRHRD 59,, DIDNOT
006042                                                              ENDIF
006042  104073                  ERROR   59
006044
006044                          $111:                                   ENDSUB

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

                                                                        ;READ BUFFER
006044                                                              LET R0 :B= @RBUF
006044  117700  173212                  MOVB    @RBUF,R0

006050                                                              IF #RCVRDONE SETIN @RCSR THEN
006050  032777  000200  173202          BIT     #RCVRDONE,@RCSR
006056  001401                  BEQ     $112

                                                                        ;RCVRDONE DID NOT CLEAR IN RCSR
                                                                        ERRHRD 60,,DIDNOT
006060
006060  104074                  ERROR   60
006062                                                              ENDIF
006062                          $112:                           ENDTST
006062
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 30        TEST THAT RCVRACT - RCSR 11 - SETS
                                ;*              WHEN A START BIT IS RECEIVED AND
                                ;*              CLEARS WHEN RCVRDONE - RCSR 7 - SETS
                                ;;**********************************************************************
006062  000004                  TST30:  SCOPE
006064  012767  000010  173066          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006072  012767  000030  173100          MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        LET @TCSR := @TCSR SET.BY #MAINT
006100
006100  052777  000004  173156          BIS     #MAINT,@TCSR
                                                                        LET FLAG :B= #CLR
006106
006106  112767  000000  003272          MOVB    #CLR,FLAG
                                                                        LET COUNT := #0
006114
006114  005067  000172                  CLR     COUNT
                                            ;LOAD A CHARACTER INTO TBUF
                                            ;WAIT FOR RCVRACT TO SET
                                ;
                                ;
                                                                    ;SEND A CHARACTER
006120                                                          LET @TBUF :B= #0
006120  105077  173144                  CLRB    @TBUF
                                                                REPEAT
006124
006124                          $113:
006124  032777  004000  173126          BIT     #RCVRACT,@RCSR
006132  001404                          BEQ     $114
                                                                    LET FLAG :B= #SET
006134
006134  112767  177777  003244          MOVB    #SET,FLAG
                                                                ELSE
006142
006142  000402                          BR      $115
006144                          $114:
                                                                    LET COUNT := COUNT + #1
006144
006144  005267  000142                  INC     COUNT
                                                                ENDIF
006150
006150                          $115:
                                                               UNTIL FLAG EQ #SET OR COUNT HI MAX
006150
006150  026727  003232  177777          CMP     FLAG,#SET
006156  001404                          BEQ     $116
006160  026767  000126  000122          CMP     COUNT,MAX
006166  101756                          BLOS    $113
006170                          $116:
                                                               IF COUNT HI MAX THEN
006170
006170  026767  000116  000112          CMP     COUNT,MAX
006176  101405                          BLOS    $117
                                                                    ;IT NEVER SET
                                                                    ;RCVRACT DID NOT SET IN RCSR.
                                                                    ERRHRD 61,, DIDNOT
006200
006200  104075                          ERROR   61
                                                                    EXIT TEST
006202
006202  012767  000001  172750          MOV     #1,$TIMES
006210  000441                          BR      TST31           ;;;EXIT THIS TEST
                                                                ENDIF
006212
006212                          $117:
```

```
                                                        ;CHECK FOR TIMING OF RCVRACT. CLEARING
                                                        ;VS RCVRDONE SETTING

                                                                                WHILE #RCVRACT SETIN @RCSR DO
006212                                  $120:
006212  032777  004000  173040          BIT     #RCVRACT,@RCSR
006220  001416                          BEQ     $121
                                                                                    IF #RCVRDONE SETIN @RCSR THEN
006222
006222  032777  000200  173030          BIT     #RCVRDONE,@RCSR
006230  001411                          BEQ     $122
006232                                                                                 IF #RCVRACT SETIN @RCSR THEN
006232  032777  004000  173020          BIT     #RCVRACT,@RCSR
006240  001405                          BEQ     $123

                                                                                ;RCVRDONE AND RCVRACT
                                                                                    ;BOTH SET
006242                                                                              ERRHRD 62, DONEACT
006242  104076                          ERROR   62
                                                                                ;NO USE CONTINUING
006244                                                                              EXIT TST
006244  012767  000001  172706          MOV     #1,$TIMES
006252  000420                          BR      TST31           ;;;EXIT THIS TEST
006254                                                                               ENDIF
006254                                  $123:                                       ENDIF
006254                                  $122:
006254                                                                          ENDDO
006254  000756                          BR      $120
006256                                  $121:

                                                                                ;RCVRACT = 0 NOW.
006256                                                                          IF #RCVRDONE NOTSETIN @RCSR THEN
006256  032777  000200  172774          BIT     #RCVRDONE,@RCSR
006264  001001                          BNE     $124

                                                                                    ;RCVRDONE DID NOT SET IN RCSR
006266                                                                              ERRHRD 63,,DIDNOT
006266  104077                          ERROR   63
                                                                                    ;SET IT BACK.
006270                                                                          ENDIF
006270                                  $124:
                                                        ;TEST THAT READING THE RECEIVER
                                                        ;BUFFER CLEARS RCVRDONE


                                                                                    ;READ CHAR.
006270                                                                          LET R0 := @RBUF
006270  017700  172766                  MOV     @RBUF,R0

006274                                                                          IF #RCVRDONE SETIN @RCSR THEN
006274  032777  000200  172756          BIT     #RCVRDONE,@RCSR
006302  001401                          BEQ     $125

                                                                                    ;RCVRDONE DID NOT CLEAR IN RCSR
006304                                                                              ERRHRD 64,,DIDNOT
006304  104100                          ERROR   64
006306                                                                          ENDIF
```

```
2964  006306                          $125:
2965
2966  006306                                                          EXIT
2967  006306  000402                          BR      TST31           ;;;EXIT THIS TEST
2968  006310  070000                          MAX:70000
2969  006312  000000                          COUNT:  0
2970
2971  006314                                                      ENDTST
2972
```

```
2973
2974                                    ;;*****************************************************************
2975                                    ;;*****************************************************************
2976                                    ;*TEST 31        TEST THE OVERRUN BIT - RBUF 14
2977                                    ;;*****************************************************************
2978  006314  000004                    TST31:  SCOPE
2979  006316  012767  000010  172634            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
2980  006324  012767  000031  172646            MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2981
2982  006332                                                            BGNSUB
2983  006332  012767  006340  172550            MOV     #64$,$LPERR
2984                                                        ;OUTPUT 2 CHARACTERS WITH
2985                                                        ;AMPLE DELAYS BETWEEN FOR RECEPTION.
2986                                                        ;THIS SHOULD AN CAUSE OVERRUN ERROR.
2987
2988                                                                        ;OUTPUT 1 CHARACTER
2989  006340                                                            LET @TBUF :B= #0
2990  006340  105077  172724                    CLRB    @TBUF
2991                                                                        ;GO AWAY FOR 500 M SEC
2992  006344                                                            WAITUS 50
2993  006344  010546                    MOV     R5,-(SP)
2994  006346  012745  000050            MOV     #50,-(R5)
2995  006352  004767  003136            JSR     PC,WAIT
2996  006356  012605                    MOV     (SP)+,R5
2997
2998                                                                        ;OUTPUT 2ND CHARACTER
2999  006360                                                            LET @TBUF :B= #0
3000  006360  105077  172704                    CLRB    @TBUF
3001                                                                        ;LET OVERRUN HAPPEN
3002  006364                                                            WAITUS 50
3003  006364  010546                    MOV     R5,-(SP)
3004  006366  012745  000050            MOV     #50,-(R5)
3005  006372  004767  003116            JSR     PC,WAIT
3006  006376  012605                    MOV     (SP)+,R5
3007
3008                                                                        ;READ BUFFER AND ERROR BITS
3009  006400                                                            LET R4 := @RBUF
3010  006400  017704  172656                    MOV     @RBUF,R4
3011
3012                                                                        ;IT DIDN'T SET
3013  006404                                                            IF #ORERR NOTSETIN R4 THEN
3014  006404  032704  040000                    BIT     #ORERR,R4
3015  006410  001005                    BNE     $126
3016                                                                        ;ORERR DID NOT SET IN RBUF
3017  006412                                                            ERRHRD 65,,DIDNOT
3018  006412  104101                    ERROR   65
3019
3020                                                                        ;NO USE COMPOUNDING ERRORS
3021  006414                                                            EXIT TST
3022  006414  012767  000001  172536            MOV     #1,$TIMES
3023  006422  000500                    BR      TST32           ;;;EXIT THIS TEST
3024  006424                                                                ENDIF
3025  006424                            $126:
3026  006424                                                        ENDSUB
3027
3028                                    ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
```

```
3029  006424                                                          BGNSUB
3030  006424  012767  006432  172456        MOV     #64$,$LPERR
3031  006432                                    IF #ERROR NOTSETIN R4 THEN
3032  006432  032704  100000                BIT     #ERROR,R4
3033  006436  001005                        BNE     $127
3034
3035                                                                      ;ERROR DID NOT SET IN RBUF
3036  006440                                                              ERRHRD 66,,DIDNOT
3037  006440  104102                        ERROR   66
3038
3039                                                                      ;-WHEN ORERR SET.
3040                                                                      ;GET OUT NOW.
3041  006442                                                              EXIT TST
3042  006442  012767  000001  172510        MOV     #1,$TIMES
3043  006450  000465                        BR      TST32           ;;;EXIT THIS TEST
3044  006452                                                          ENDIF
3045  006452                          $127:
3046  006452                                                          ENDSUB
3047
3048  006452                                                          BGNSUB
3049  006452  012767  006460  172430        MOV     #64$,$LPERR
3050                                            ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
3051
3052  006460                                                          IF #ORERR NOTSETIN @RBUF THEN
3053  006460  032777  040000  172574        BIT     #ORERR,@RBUF
3054  006466  001002                        BNE     $130
3055
3056                                                                      ;READING RBUF CLEARED ORERR.
3057  006470                                                              ERRHRD 67,ITCLRED
3058  006470  104103                        ERROR   67
3059                                                                      ;SKIP REST OF TEST
3060  006472                                                              EXIT
3061  006472  000454                        BR      TST32           ;;;EXIT THIS TEST
3062  006474                                                          ENDIF
3063  006474                          $130:
3064  006474                                                      ENDSUB
3065
3066  006474                                                      BGNSUB
3067  006474  012767  006502  172406        MOV     #64$,$LPERR
3068                                            ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
3069
3070                                                                  ;SEND A CHARACTER AROUND.
3071  006502                                                      LET @TBUF :B= #0
3072  006502  105077  172562                CLRB    @TBUF
3073                                                                  ;LET IT CIRCULATE
3074  006506                                                      WAITUS 50
3075  006506  010546                        MOV     R5,-(SP)
3076  006510  012745  000050                MOV     #50,-(R5)
3077  006514  004767  002774                JSR     PC,WAIT
3078  006520  012605                        MOV     (SP)+,R5
3079
3080  006522                                                      IF #ORERR SETIN @RBUF THEN
3081  006522  032777  040000  172532        BIT     #ORERR,@RBUF
3082  006530  001405                        BEQ     $131
3083                                                                  ;ORERR DID NOT CLEAR IN RBUF
3084  006532                                                          ERRHRD 68,,DIDNOT
```

```
                        ERROR   68

                                                        ;-AFTER RECEIVING ANOTHER CHAR
                                                        ;SKIP AROUND REST
                                                        EXIT TST
006534
006534  012767  000001  172416        MOV     #1,$TIMES
006542  000430                        BR      TST32           ;;;EXIT THIS TEST
                                                        ENDIF
006544
006544                          $131:
006544
                                                        IF #ERROR SETIN @RBUF THEN
006544  032777  100000  172510        BIT     #ERROR,@RBUF
006552  001401                        BEQ     $132
                                                        ;ERROR DID NOT CLEAR IN RBUF
                                                        ERRHRD 69,,DIDNOT
006554
006554  104105                        ERROR   69
006556                                                  ENDIF
006556                          $132:
006556                                                  ENDSUB
006556                                                  EXIT
006556  000422                        BR      TST32           ;;;EXIT THIS TEST
006560  042522  042101  047111                          ITCLRED:        .ASCIZ  /READING RBUF CLEARED OV
006566  020107  041122  043125
006574  041440  042514  051101
006602  042105  047440  042526
006610  051122  047125  042440
006616  051122  051117  000056
                                                                .EVEN
006624                                                  ENDTST
```

```
3116
3117                                    ;;*****************************************************************
3118                                    ;;*****************************************************************
3119                                    ;*TEST 32        PROGRAMMABLE BAUD RATE TEST
3120                                    ;*              TEST AT ALL SPEEDS AVAILABLE
3121                                    ;*              A COMPARISON WILL BE MADE TO SEE
3122                                    ;*              IF NEW TIME IS LESS THAN PREVIOUS.
3123                                    ;;*****************************************************************
3124  006624  000004                   TST32:  SCOPE
3125  006626  012767  000010  172324           MOV     #10,$TIMES      ;;DO 10 ITERATIONS
3126  006634  012767  000032  172336           MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                        IF #PBR NOTSETIN $USWR THEN
3127  006642
3128  006642  032767  000200  172350           BIT     #PBR,$USWR
3129  006650  001004                           BNE     $133
3130  006652                                                  EXIT TST
3131  006652  012767  000001  172300           MOV     #1,$TIMES
3132  006660  000542                           BR      TST33           ;;;EXIT THIS TEST
                                                        ENDIF
3133  006662
3134  006662                           $133:
3135
3136  006662                                                  LET OLD := #-1
3137  006662  012767  177777  000272           MOV     #-1,OLD
3138  006670                                                  LET OLD+2 := #-1
3139  006670  012767  177777  000266           MOV     #-1,OLD+2
3140  006676                                                  LET @TCSR := @TCSR SET.BY #MAINT
3141  006676  052777  000004  172360           BIS     #MAINT,@TCSR
3142                                                  ;EACH BAUD RATE
3143  006704                                  INCR R3 FROM #0 TO #15. BY #1
3144  006704  005003                           CLR     R3
3145  006706  000401                           BR      $134
3146  006710                           $135:
3147  006710  005203                           INC     R3
3148  006712                           $134:
3149  006712  020327  000017                   CMP     R3,#15.
3150  006716  003060                           BGT     $136
3151  006720                                  LET R0 := @RBUF
3152  006720  017700  172336                   MOV     @RBUF,R0
3153                                                  ;CHANGE BAUDE RATE
3154  006724                                      LET @TCSRHI :8= RATES(R3)
3155  006724  116377  007104  172334           MOVB    RATES(R3),@TCSRHI
3156                                                  ;FLAG
3157  006732                                      LET BIT := #0
3158  006732  005002                           CLR     BIT
3159                                                  ;OUTPUT THE CHARACTER
3160  006734                                      LET @TBUF := #0
3161  006734  005077  172330                   CLR     @TBUF
3162                                                  ;INITIALIZE COUNTER
3163  006740                                      LET NEW := #0
3164  006740  005067  000212                   CLR     NEW
3165  006744                                      LET NEW+2 := #0
3166  006744  005067  000210                   CLR     NEW+2
3167  006750                                      WHILE BIT EQ #0 DO
3168  006750                           $137:
3169  006750  005702                           TST     BIT
3170  006752  001014                           BNE     $140
3171  006754                                          IF #RCVRDONE SETIN @RCSR THEN
```

```
3172  006754  032777  000200  172276          BIT     #RCVRDONE,@RCSR
3173  006762  001403                          BEQ     $141
3174
3175  006764
3176  006764  012702  000001                  MOV     #1,BIT
3177  006770
3178  006770  000404                          BR      $142
3179  006772                          $141:
3180
3181  006772
3182  006772  005267  000160                  INC     NEW
3183  006776
3184  006776  005567  000156                  ADC     NEW+2
3185  007002
3186  007002                          $142:
3187
3188  007002
3189  007002  000762                          BR      $137
3190  007004                          $140:
3191
3192  007004
3193  007004  026767  000150  000152          CMP     NEW+2,OLD+2
3194  007012  103001                          BHIS    $143
3195
3196  007014
3197  007014  000412                          BR      $144
3198  007016                          $143:
3199
3200  007016
3201  007016  026767  000136  000140          CMP     NEW+2,OLD+2
3202  007024  001005                          BNE     $145
3203  007026  026767  000124  000126          CMP     NEW,OLD
3204  007034  103001                          BHIS    $145
3205
3206  007036
3207  007036  000401                          BR      $146
3208  007040                          $145:
3209
3210
3211
3212
3213  007040
3214  007040  104126                          ERROR   86
3215  007042
3216  007042                          $146:
3217  007042
3218  007042                          $144:
3219
3220  007042
3221  007042  016767  000110  000112          MOV     NEW,OLD
3222  007050
3223  007050  016767  000104  000106          MOV     NEW+2,OLD+2
3224
3225  007056
3226  007056  000714                          BR      $135
3227  007060                          $136:
```

```
        ;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 86, BAUDRATE

        ENDIF

    ENDIF

        ;UPDATE OLD TIME
    LET OLD := NEW

    LET OLD+2 := NEW+2


ENDINC  ;BAUD RATE
```

```
3228  007060                                                    LET R3 :B= $USWR+1 AND #17      ; PUT BAUD BACK
3229  007060  116703  172135                MOVB    $USWR+1,R3
3230  007064  110346                       MOVB    R3,-(SP)
3231  007066  142716  000017               BICB    #17,(SP)
3232  007072  142603                       BICB    (SP)+,R3
3233  007074                                                    LET @TCSRHI :B= RATES(R3)       ; LIKE HE WANTED IT
3234  007074  116377  007104  172164       MOVB    RATES(R3),@TCSRHI
3235
3236  007102                                                            EXIT    ;SKIP TABLE
3237  007102  000431                       BR      TST33           ;;;EXIT THIS TEST
3238
3239  007104                        RATES:  ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
3240                                ;UPPER BYTE OF XCSR FOR EACH BAUD RATE
3241                                ;** NOTE::      THE VALUE INDICATED IN THE COLUMN 'OFFSET
3242                                ;**             INTO TABLE' CAN BE PLACED INTO BITS<11:8>
3243                                ;**             OF LOCATION '$USWR' TO CAUSE THE CORROSPONDING
3244                                ;**             BAUD TO BE SELECTED IN THE DLV11-E UPON
3245                                ;**             COMPLETION OF THIS TEST.
3246
3247                                                            ;       BAUD    OFFSET INTO TABLE
3248  007104     010                R0050:  .BYTE   010        ;       50      0
3249  007105     030                R0070:  .BYTE   030        ;       70      1
3250  007106     050                R0110:  .BYTE   050        ;       110     2
3251  007107     070                R0135:  .BYTE   070        ;       135     3
3252  007110     110                R0150:  .BYTE   110        ;       150     4
3253  007111     130                R0300:  .BYTE   130        ;       300     5
3254  007112     150                R0600:  .BYTE   150        ;       600     6
3255  007113     170                R0200:  .BYTE   170        ;       1200    7
3256  007114     210                R1800:  .BYTE   210        ;       1800    10
3257  007115     230                R2000:  .BYTE   230        ;       2000    11
3258  007116     250                R2400:  .BYTE   250        ;       2400    12
3259  007117     270                R3600:  .BYTE   270        ;       3600    13
3260  007120     310                R4800:  .BYTE   310        ;       4800    14
3261  007121     330                R7200:  .BYTE   330        ;       7200    15
3262  007122     350                R9600:  .BYTE   350        ;       9600    16
3263  007123     370                R10000: .BYTE   370        ;       19200   17
3264
3265  007124  040502  042125  051040  BAUDRATE:       .ASCIZ  /BAUD RATE DIDN'T CHANGE./
3266  007132  052101  020105  044504
3267  007140  047104  052047  041440
3268  007146  040510  043516  027105
3269  007154     000
3270          007156                  .EVEN
3271  007156  000000  000000          NEW:    0,0
3272  007162  000000  000000          OLD:    0,0
3273  007166                                                                  ENDTST
3274
3275
3276
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*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
                                ;;**********************************************************************
007166  000004                  TST33:  SCOPE
007170  012767  000010  171762          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007176  012767  000033  171774          MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
                                                                LET INTFLAG := #0
007204  005067  002372                  CLR     INTFLAG

                                                                        ;GET VECTOR ADDRESS
                                                                LET R3 := DLVEC
007210  016703  172042                  MOV     DLVEC,R3
                                                                        ;FOR THE TRANSMITTER
                                                                LET R3 := R3 + #4
007214  062703  000004                  ADD     #4,R3
                                                                        ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
                                                                SETVEC R3, #INTSRV, #PR7
007220  010146                  MOV     R1,-(SP)
007222  010301                  MOV     R3,R1
007224  012721  011574          MOV     #INTSRV,(R1)+
007230  012711  000340          MOV     #PR7,(R1)
007234  012601                  MOV     (SP)+,R1
                                                                BGNSUB
007236  012767  007244  171644          MOV     #65$,$LPERR
                                                                        ;CLEAR INTERRUPT ENABLE
                                                                LET @TCSR := @TCSR CLR.BY #XMITIE
007244  042777  000100  172012          BIC     #XMITIE,@TCSR

                                                                        ;SET IT TO 0
007252  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007256  012746  007264                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
007262  000002                  RTI                             ;;POP NEW PC AND PS
007264                          65$:

                                                                        ;NOW SET I.E. BIT
                                                                LET @TCSR := @TCSR SET.BY #XMITIE
007264  052777  000100  171772          BIS     #XMITIE,@TCSR

                                                                        ;LET INTERRUPT HAVE TIME TO OCCUR
                                                                WAITUS 10
007272  010546                  MOV     R5,-(SP)
007274  012745  000010                  MOV     #10,-(R5)
007300  004767  002210                  JSR     PC,WAIT
007304  012605                  MOV     (SP)+,R5

                                                                        ;DID EXACTLY 1 INTERRUPT OCCUR
                                                                IF INTFLAG NE #1 THEN
007306  026727  002270  000001          CMP     INTFLAG,#1
007314  001406                  BEQ     $147
```

```
3333
3334  007316
3335  007316  005767  002260          TST     INTFLAG                         ;NO - WAS IT 0 OR MORE THAN ONCE
3336  007322  001002                  BNE     $150                            IF INTFLAG EQ #0 THEN
3337
3338  007324
3339  007324  104106                  ERROR   70                              ;TRANSMITTER DID NOT INTERRUPT IN TIME
3340  007326                                                                      ERRHRD 70,,DIDNOT
3341  007326  000401                  BR      $151                            ELSE
3342  007330                  $150:
3343
3344
3345  007330
3346  007330  104107                  ERROR   71                              ;TWICE
3347  007332                                                                  ;TRANSMITTER INTERRUPTED TWICE
3348  007332                  $151:                                               ERRHRD 71,,TWICE
3349  007332                                                                  ENDIF
3350  007332                  $147:                                       ENDIF
3351  007332                                                              ENDSUB
3352                                                          ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
3353  007332                                                              BGNSUB
3354  007332  012767  007340  171550  MOV     #64$,$LPERR
3355                                                                          ;CLEAR 'INTERRUPT OCCURED' FLAG
3356  007340                                                                  LET INTFLAG := #0
3357  007340  0C5067  002236          CLR     INTFLAG
3358                                                                          ;CLEAR INTERRUPT ENABLE
3359  007344                                                                  LET @TCSR := @TCSR CLR.BY #XMITIE
3360  007344  042777  000100  171712  BIC     #XMITIE,@TCSR
3361                                                                          ;NO INTERRUPTS SHOULD OCCUR.
3362  007352  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3363  007356  012746  007364          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
3364  007362  000002                  RTI                     ;;POP NEW PC AND PS
3365  007364                  65$:
3366                                                                          ;DARE IT TO HAPPEN
3367  007364                                                                  WAITUS 10
3368  007364  010546                  MOV     R5,-(SP)
3369  007366  012745  000010          MOV     #10,-(R5)
3370  007372  004767  002116          JSR     PC,WAIT
3371  007376  012605                  MOV     (SP)+,R5
3372  007400                                                                  IF INTFLAG NE #0 THEN
3373  007400  005767  002176          TST     INTFLAG
3374  007404  001401                  BEQ     $152
3375                                                                              ;INTERRUPT OCCURED WITH I E CLEARED
3376  007406                                                                      ERRHRD 72,NOTENAB
3377  007406  104110                  ERROR   72                              ENDIF
3378  007410
3379  007410                  $152:                                           BRESET
3380  007410
3381  007410  000005                  RESET                               ENDSUB
3382  007412                                                                  ;RESTORE VECTOR AREA
3383                                                                      CLRVEC R3
3384  007412
3385  007412  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3386  007414  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3387  007416  012701  000003          MOV     #R3,R1
3388  007422  010102                  MOV     R1,R2
```

```
3389  007424  062702  000002          ADD     #2,R2
3390  007430  010221                  MOV     R2,(R1)+
3391  007432  005011                  CLR     (R1)
3392  007434  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
3393  007436  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
3394
3395  007440                                                      ENDTST
3396
3397
3398
3399
3400
3401
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 34        RECEIVER INTERRUPT LOGIC TEST
                                ;*              THIS TEST COVERS ALL OF THE RECEIVER
                                ;*              SIDE OF THE INTERRUPT LOGIC, BOTH DATASET
                                ;*              AND CHARACTER MODES.
                                ;;**********************************************************************
007440  000004                  ↑ST34:  SCOPE
007442  012767  000010  171510          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
007450  012767  000034  171522          MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR INTERRUPT OCCURED FLAG
                                                                        ;SET UP RECEIVER INTER.VECTOR
007456                                                          SETVEC DLVEC,#INTSRV,#PR7
007456  010146                          MOV     R1,-(SP)
007460  016701  171572                  MOV     DLVEC,R1
007464  012721  011574                  MOV     #INTSRV,(R1)+
007470  012711  000340                  MOV     #PR7,(R1)
007474  012601                          MOV     (SP)+,R1
                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
007476                                                          BGNSUB
007476  012767  007504  171404          MOV     #64$,$LPERR
007504                                                          LET INTFLAG := #0
007504  005067  002072                  CLR     INTFLAG
                                                                        ;SET MAINT. BIT
007510                                                          LET @TCSR := @TCSR SET.BY #MAINT
007510  052777  000004  171546          BIS     #MAINT,@TCSR
                                                                        ;CLEAR INTERRUPTS
007516                                                          LET @RCSR := @RCSR CLR.BY #RCVRIE
007516  042777  000100  171534          BIC     #RCVRIE,@RCSR
                                                                        ;CHANGE PRIORITY
                                                                        ;..TO 0
007524  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007530  012746  007536                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
007534  000002                          RTI                     ;;POP NEW PC AND PS
007536                          65$:
                                                                        ;SEND A CHARACTER
007536                                                          LET @TBUF :B= #0
007536  105077  .71526                  CLRB    @TBUF
                                                                        ;WAIT A MAXIMUM
                                                                ;OF 50 MSEC FOR
                                                                ;XMIT RDY TO SET IN TCSR
007542                                                          CALL TIMER IN (#50,#XMITRDY,TCSR,#SET)
007542  010546                          MOV     R5,-(SP)
007544  012745  177777                  MOV     #SET,-(R5)
007550  016745  171510                  MOV     TCSR,-(R5)
007554  012745  000200                  MOV     #XMITRDY,-(R5)
007560  012745  000050                  MOV     #50,-(R5)
007564  004767  001446                  JSR     PC,↑TIMER
007570  012605                          MOV     (SP)+,R5
                                                                        ;SET INTERRUPT ENABLE
007572                                                          LET @RCSR := @RCSR SET.BY #RCVRIE
007572  052777  000100  171460          BIS     #RCVRIE,@RCSR
                                                                        ;LET IT COME IN.
007600                                                          WAITUS 10
007600  010546                          MOV     R5,-(SP)
```

```
3458  007602  012745  000010          MOV     #10,-(R5)
3459  007606  004767  001702          JSR     PC,WAIT
3460  007612  012605                  MOV     (SP)+,R5
3461
3462                                                    ;DID HE DO IT RIGHT?
3463  007614                                          IF INTFLAG NE #1 THEN
3464  007614  026727  001762  000001          CMP     INTFLAG,#1
3465  007622  001406                  BEQ     $153
3466
3467  007624                                              ;NONE OCCURED
3468  007624  005767  001752                  TST     INTFLAG     IF INTFLAG EQ #0 THEN
3469  007630  001002                  BNE     $154
3470
3471  007632                                              ;RECEIVER DID NOT INTERRUPT IN TIME
3472  007632  104111                  ERROR   73              ERRHRD 73,,DIDNOT
3473
3474  007634                                              ;TWICE OR MORE
3475  007634  000401                  BR      $155        ELSE
3476  007636                  $154:
3477
3478  007636                                              ;RECEIVER INTERRUPTED TWICE
3479  007636  104112                  ERROR   74              ERRHRD 74,,TWICE
3480  007640
3481  007640                  $155:                       ENDIF
3482  007640
3483  007640                  $153:                   ENDIF
3484
3485  007640                                                  ;RESET MAINT. BIT.
3486  007640  042777  000004  171416          BIC     #MAINT,@TCSR    LET @TCSR := @TCSR CLR.BY #MAINT
3487  007646                                          ENDSUB
3488
3489
3490
3491
3492
3493
3494                                          ;INTERRUPT WITHOUT I E SET.
3495  007646                                          BGNSUB
3496  007646  012767  007654  171234          MOV     #64$,$LPERR
3497  007654                                          LET @TCSR := @TCSR SET.BY #MAINT
3498  007654  052777  000004  171402          BIS     #MAINT,@TCSR
3499
3500                                                  ;CLEAR INTERRUPT FLAG
3501  007662                                      LET INTFLAG := #0
3502  007662  005067  001714                  CLR     INTFLAG
3503                                                  ;CLEAR INTERRUPT
3504  007666                                      LET @RCSR := @RCSR CLR.BY #RCVRIE
3505  007666  042777  000100  171364          BIC     #RCVRIE,@RCSR
3506                                                  ;DON'T LET THEM IN
3507  007674  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
3508  007700  012746  007706                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
3509  007704  000002                  RTI                             ;;POP NEW PC AND PS
3510  007706                  65$:
3511                                                  ;DARE IT
3512  007706                                      WAITUS 10
3513  007706  010546                  MOV     R5,-(SP)
```

```
3514  007710  012745  000010          MOV     #10,-(R5)
3515  007714  004767  001574          JSR     PC,WAIT
3516  007720  012605                  MOV     (SP)+,R5
3517                                                        ;DID IT HAPPEN?
3518  007722                                              IF INTFLAG NE #0 THEN
3519  007722  005767  001654          TST     INTFLAG
3520  007726  001401                  BEQ     $156
3521                                                            ;INTERRUPT OCCURED AT PRIORITY 7
3522  007730                                                    ERRHRD 77,NOTENAB
3523  007730  104115                  ERROR   77
3524  007732                                                  ENDIF
3525  007732                      $156:
3526                                                            ;CLEAR THE WORLD
3527  007732                                                  BRESET
3528  007732  000005                  RESET
3529
3530                                                        ;RESET MAINT. BIT.
3531  007734                                              LET @TCSR := @TCSR CLR.BY #MAINT
3532  007734  042777  000004  171322  BIC     #MAINT,@TCSR
3533  007742                                              ENDSUB
3534
3535                              ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-DATAIE
3536  007742                                              BGNSUB
3537  007742  012767  007750  171140  MOV     #64$,$LPERR
3538  007750                                              IF #CABLE NOTSETIN $USWR THEN
3539  007750  032767  020000  171242  BIT     #CABLE,$USWR
3540  007756  001004                  BNE     $157
3541                                                            ;CAN'T TEST WITHOUT A CABLE
3542  007760                                                  EXIT TST
3543  007760  012767  000001  171172  MOV     #1,$TIMES
3544  007766  000463                  BR      TST35           :::EXIT THIS TEST
3545  007770                                                  ENDIF
3546  007770                      $157:
3547                                                            ; CLEAR 'INTFLAG'
3548  007770                                                  LET INTFLAG := #0
3549  007770  005067  001606          CLR     INTFLAG
3550                                                            ;CLEAR INTERRUPTS
3551  007774                                                  LET @RCSR := @RCSR CLR.BY #DATAIE
3552  007774  042777  000040  171256  BIC     #DATAIE,@RCSR
3553                                                            ;CHANGE PRIORITY
3554                                                            ;..TO 5
3555  010002  012746  000000          MOV     #PRC,-(SP)      ;;PUT NEW PS ON STACK
3556  010006  012746  010014          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3557  010012  000002                  RTI                     ;;POP NEW PC AND PS
3558  010014                      64$:
3559  010014                                                  LET @RCSR := @RCSR CLR.BY #REQSEND
3560  010014  042777  000004  171236  BIC     #REQSEND,@RCSR
3561                                                            ;SET INTERRUPT ENABLE
3562  010022                                                  LET @RCSR := @RCSR SET.BY #DATAIE
3563  010022  052777  000040  171230  BIS     #DATAIE,@RCSR
3564  010030                                                  LET    @RCSR := @RCSR SET.BY #REQSEND
3565  010030  052777  000004  171222  BIS     #REQSEND,@RCSR
3566                                                            ;LET IT COME IN.
3567  010036                                                  WAITUS 10
3568  010036  010546                  MOV     R5,-(SP)
3569  010040  012745  000010          MOV     #10,-(R5)
```

```
3570  010044  004767  001444          JSR     PC,WAIT
3571  010050  012605                  MOV     (SP)+,R5
3572
3573
3574  010052
3575  010052  026727  001524  000001  CMP     INTFLAG,#1
3576  010060  001406                  BEQ     $160
3577
3578  010062
3579  010062  005767  001514          TST     INTFLAG
3580  010066  001002                  BNE     $161
3581
3582  010070
3583  010070  104113                  ERROR   75
3584
3585  010072
3586  010072  000401                  BR      $162
3587  010074                  $161:
3588
3589  010074
3590  010074  104114                  ERROR   76
3591  010076
3592  010076                  $162:
3593  010076
3594  010076                  $160:
3595  010076
3596  010076  042777  000040  171154  BIC     #DATAIE,@RCSR
3597  010104
3598
3599  010104
3600  010104  017704  171146          MOV     @DLVEC,R4
3601  010110
3602  010110  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
3603  010112  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
3604  010114  012701  000004          MOV     #R4,R1
3605  010120  010102                  MOV     R1,R2
3606  010122  062702  000002          ADD     #2,R2
3607  010126  010221                  MOV     R2,(R1)+
3608  010130  005011                  CLR     (R1)
3609  010132  012602                  MOV     (SP)+,R2        ;;POP STACK INTO R2
3610  010134  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
3611  010136
```

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


    ;NONE OCCURED
    IF INTFLAG EQ #0 THEN


    ;DATAINT  DID NOT INTERRUPT IN TIME
        ERRHRD 75,,DIDNOT

    ;TWICE OR MORE
    ELSE


    ; DATAINT INTERRUPTED TWICE
        ERRHRD 76,,TWICE

    ENDIF

ENDIF

LET @RCSR := @RCSR CLR.BY #DATAIE
                                ENDSUB

                                LET R4 := @DLVEC
                                CLRVEC R4
                                ENDTST
```

```
                        ;;**********************************************************************
                        ;;**********************************************************************
                        ;*TEST 35        TEST ACTUAL DATA TRANSFERED
                        ;*                NON-INTERRUPT MAINTENANCE BIT SET
                        ;;**********************************************************************
010136  000004          TST35:  SCOPE
010140  012767  000001  171012          MOV     #1,$TIMES       ;;DO 1 ITERATION
010146  012767  000035  171024          MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ;SET MAINT. BIT
                                                        LET @TCSR := @TCSR SET.BY #MAINT
010154
010154  052777  000004  171102          BIS     #MAINT,@TCSR
                                                                ;CHANGE PRIORITY
                                                                ;..TO 0
010162  012746  000000                  MOV     #PRO,-(SP)      ;;PUT NEW PS ON STACK
010166  012746  010174                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
010172  000002                          RTI                     ;;POP NEW PC AND PS
010174                          64$:
                                                                ;GET DATA MASK.
                                                        CALL DATLNG OUT <R1>
010174
010174  162705  000002                  SUB     #1*2,R5
010200  004767  001210                  JSR     PC,DATLNG
010204  012501                          MOV     (R5)+,R1
                                                                ;ALL BINARY CHAR.
                                                        INCR R2 FROM #0 TO #377 BY #1
010206
010206  005002                          CLR     R2
010210  000401                          BR      $163
010212                          $164:
010212  005202                          INC     R2
010214                          $163:
010214  020227  000377                  CMP     R2,#377
010220  003023                          BGT     $165
                        ;TRANSMIT CHAR IN R2
                        REPEAT
010222
010222                          $166:
                                    ;IS IT READY?
                        UNTIL #XMITRDY SETIN @TCSR
010222
010222  032777  000200  171034          BIT     #XMITRDY,@TCSR
010230  001774                          BEQ     $166
                                    ;TRANSMIT IT
                        LET @TBUF :B= R2
010232
010232  110277  171032                  MOVB    R2,@TBUF
                        REPEAT
010236
010236                          $167:
                                    ;WAIT TILL ITS HERE
                        UNTIL #RCVRDONE SETIN @RCSR
010236
010236  032777  000200  171014          BIT     #RCVRDONE,@RCSR
010244  001774                          BEQ     $167
                                    ;AND SAVE IT
                        LET R3 := @RBUF
010246
```

```
3668  010246  017703  171010          MOV     @RBUF,R3
3669
3670                                                    ;COMPARE TO SEE IF WE RECEIVED IT ALL
3671
3672                                                        ;CLEAN OFF NON-DATA BITS
3673                                                        ;ON BOTH TRANSMITTED AND
3674                                                    LET R4 := R2 CLR.BY R1
3675  010252
3676  010252  010204          MOV     R2,R4
3677  010254  040104          BIC     R1,R4
3678  010256                                            LET R3 := R3 CLR.BY R1
3679  010256  040103          BIC     R1,R3
3680
3681                                                    ;RECEIVED DATA
3682  010260                                            IF R4 NE R3 THEN
3683  010260  020403          CMP     R4,R3
3684  010262  001401          BEQ     $170
3685
3686  010264                                                ;DATA COMPARE ERROR
3687  010264  104116          ERROR   78                    ERRHRD 78,COMP,SBWAS
3688
3689  010266                                                ;<TRANSMITTED> <RECEIVED>
3690  010266              $170:                         ENDIF
3691  010266
3692  010266  000751          BR      $164        ENDINC  ; R2
3693  010270              $165:
3694
3695                                                        ;RESET MAINT. BIT.
3696  010270                                            LET @TCSR := @TCSR CLR.BY #MAINT
3697  010270  042777  000004  170766  BIC     #MAINT,@TCSR
3698  010276                                        ENDTST
3699
3700
3701
```

```
3702                                    ;;**********************************************************************
3703                                    ;;**********************************************************************
3704                                    ;*TEST 36        TEST DATA THROUGH CABLE
3705                                    ;;**********************************************************************
3706  010276  000004                    T$T36:  SCOPE
3707  010300  012767  000001  170652            MOV     #1,$TIMES       ;;DO 1 ITERATION
3708  010306  012767  000036  170664            MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3709  010314                                                                    IF #CABLE NOTSETIN $USWR THEN
3710  010314  032767  020000  170676            BIT     #CABLE,$USWR
3711  010322  001004                            BNE     $171
3712                                                                                ;CAN'T TEST WITHOUT A CABLE
3713  010324                                                                  EXIT TST
3714  010324  012767  000001  170626            MOV     #1,$TIMES
3715  010332  000456                            BR      T$T37           ;;;EXIT THIS TEST
3716  010334                                                                    ENDIF
3717  010334                                $171:
3718                                                                              ;DON'T USE MAINT.
3719  010334                                                                  LET @TCSR := @TCSR CLR.BY #MAINT
3720  010334  042777  000004  170722            BIC     #MAINT,@TCSR
3721                                                                              ;CHANGE PRIORITY
3722                                                                              ;..TO 0
3723  010342  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3724  010346  012746  010354                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3725  010352  000002                            RTI                     ;;POP NEW PC AND PS
3726  010354                                64$:
3727                                                                              ;GET DATA MASK
3728  010354                                                                  CALL DATLNG OUT <R1>
3729  010354  162705  000002                    SUB     #1*2,R5
3730  010360  004767  001030                    JSR     PC,DATLNG
3731  010364  012501                            MOV     (R5)+,R1
3732  010366                                                                LET R0 := @RBUF ; START CLEAN
3733  010366  017700  170670                    MOV     @RBUF,R0
3734                                                                              ;BINARY COUNT PATTERN
3735  010372                                                                INCR R2 FROM #0 TO #377 BY #1
3736  010372  005002                            CLR     R2
3737  010374  000401                            BR      $172
3738  010376                                $173:
3739  010376  005202                            INC     R2
3740  010400                                $172:
3741  010400  020227  000377                    CMP     R2,#377
3742  010404  003031                            BGT     $174
3743
3744
3745                                                                            ;TRANSMIT THE CHAR. IN R2.
3746
3747  010406                                                                  REPEAT
3748  010406                                $175:
3749                                                                              ;WAIT ON READY
3750  010406                                                                  UNTIL #XMITRDY SETIN @TCSR
3751  010406  032777  000200  170650            BIT     #XMITRDY,@TCSR
3752  010414  001774                            BEQ     $175
3753
3754                                                                              ;START IT ON ITS WAY
3755  010416                                                                  LET @TBUF :B= R2
3756  010416  110277  170646                    MOVB    R2,@TBUF
3757                                                                              ;GIVE IT A HEAD START.
```

```
3758  010422
3759  010422  010546                      MOV     R5,-(SP)
3760  010424  012745  000010              MOV     #10,-(R5)
3761  010430  004767  001060              JSR     PC,WAIT
3762  010434  012605                      MOV     (SP)+,R5
3763  010436
3764  010436                      $176:
3765
3766  010436
3767  010436  032777  000200  170614      BIT     #RCVRDONE,@RCSR
3768  010444  001774                      BEQ     $176
3769
3770
3771  010446
3772  010446  017703  170610              MOV     @RBUF,R3
3773
3774
3775  010452
3776  010452  010204                      MOV     R2,R4
3777  010454  040104                      BIC     R1,R4
3778  010456
3779  010456  040103                      BIC     R1,R3
3780
3781
3782  010460
3783  010460  020403                      CMP     R4,R3
3784  010462  001401                      BEQ     $177
3785
3786  010464
3787  010464  104117                      ERROR   79
3788
3789  010466
3790  010466                      $177:
3791
3792  010466
3793  010466  000743                      BR      $173
3794  010470                      $174:
3795
3796
3797
3798  010470
3799
3800
3801
3802
```

```
WAITUS 10


REPEAT

    ;IS IT HERE
UNTIL #RCVRDONE SETIN @RCSR


    ;RETRIEVE
LET R3 := @RB.


    ;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 79,COMP,SBWAS

                ;<R2> <R3>
            ENDIF

        ENDINC  ; R2


ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 37        FULL DATA TRANSFER WITH INTERRUPTS
                                ;*               AND MAINTENANCE MODE.
                                ;;**********************************************************************
010470  000004          TST37:  SCOPE
010472  012767  000001  170460        MOV     #1,$TIMES       ;;DO 1 ITERATION
010500  012767  000037  170472        MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                                                ;GET DATA MASK
                                                        CALL DATLNG OUT <R3>
010506
010506  162705  000002              SUB     #1*2,R5
010512  004767  000676              JSR     PC,DATLNG
010516  012503                      MOV     (R5)+,R3

                                ;SINCE THIS WILL BE A FULL SPEED INTERRUPT TEST
                                ;IF A DATA COMPARE ERROR OCCURS THE REPORT
                                ;WILL BE DEFERRED UNTIL ALL DATA PATTERNS HAVE
                                ;BEEN TRANSFERED.  IF THE NUMBER OF CHAR. TRANSMITTED
                                ;DOES NOT EQUAL THE NUMBER RECEIVED THIS ALSO WILL
                                ;BE REPORTED.

                                                                ;CHANGE PRIORITY
                                                                ;..TO 0
010520  012746  000000              MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
010524  012746  010532              MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
010530  000002                      RTI                     ;;POP NEW PC AND PS
010532                  64$:

                                                                ;GET VECTOR ADDRESS
                                                        LET R1 := DLVEC
010532
010532  016701  170520              MOV     DLVEC,R1

                                                                ;RCVR VECTOR
                                                        LET (R1)+ := #REC
010536
010536  012721  010710              MOV     #REC,(R1)+
                                                        LET (R1)+ := #PR7
010542
010542  012721  000340              MOV     #PR7,(R1)+

                                                                ;POINT TO TRANSMITTER VECTOR
                                                                ;AND SET IT UP ALSO
                                                        LET (R1)+ := #TRAN
010546
010546  012721  010646              MOV     #TRAN,(R1)+
                                                        LET (R1)  := #PR7
010552
010552  012711  000340              MOV     #PR7,(R1)

                                                                ;START COUNT AT 0
                                                        LET R1 := #0
010556
010556  005001                      CLR     R1

                                                                ;RECEIVER STORAGE
                                                        LET R2 := #0
010560
010560  005002                      CLR     R2

                                                                ;# OF RECEIVED CHAR. COUNT.
                                                        LET R4 := #0
010562
010562  005004                      CLR     R4

                                                        BRESET  ;SET UP ALL REGISTERS
010564
010564  000005                      RESET
```

```
3859
3860  010566
3861  010566  052777  000004  170470          BIS     #MAINT,@TCSR            ;SET UP MAINTENANCE
3862                                        LET @TCSR := @TCSR SET.BY #MAINT
3863
3864  010574                                                                    ;SET I.E. IN TRANSMITTER
3865  010574  052777  000100  170462          BIS     #XMITIE,@TCSR           LET @TCSR := @TCSR SET.BY #XMITIE
3866                                                                            ;AND RECEIVER
3867  010602                                                                LET @RCSR := @RCSR SET.BY #RCVRIE
3868  010602  052777  000100  170450          BIS     #RCVRIE,@RCSR
3869
3870
3871                                                                        ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
3872  010610                                                                REPEAT
3873  010610                          $200:
3874  010610                                                                UNTIL R4 HIS NUMBER
3875  010610  020467  000026                  CMP     R4,NUMBER
3876  010614  103775                          BLO     $200
3877
3878                                                                            ;DATA COMPARE ERRORS.
3879  010616                                                                IF ERRCNT NE #0 THEN
3880  010616  005767  000016                  TST     ERRCNT
3881  010622  001405                          BEQ     $201
3882
3883  010624                                                                    ; SET UP FOR ERROR PRINTING
3884  010624  016702  000014                  MOV     SB,R2                       LET R2 := SB
3885  010630
3886  010630  016703  000011                  MOV     WAS,R3                      LET R3 := WAS
3887                                                                                ;DATA COMPARE ERROR
3888  010634                                                                    ERRHRD 80,COMP,FIRST
3889  010634  104120                          ERROR   80
3890  010636                                                                ENDIF
3891  010636                          $201:
3892
3893                                                                        ;<R3> OCCURED, FIRST: SB <SB>, WAS <WAS>
3894  010636                                                                EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
3895  010636  000462                          BR      TST40           ;;;EXIT THIS TEST
3896
3897  010640  000000                                                        ERRCNT: 0
3898  010642  001000                                                        NUMBER: 1000
3899  010644     000                                                        SB:     .BYTE 0
3900  010645     000                                                        WAS:    .BYTE 0
3901
3902
3903
3904                                                                            ;TRANSMIT INTERRUPT HANDLER
3905  010646                                                                BGNSRV  TRAN
3906  010646                          TRAN:
3907
3908                                                                            ;SET UP FOR TRANSFER
3909  010646                                                                LET HOLD := R1      CLR.BY R3
3910  010646  010167  000032                  MOV     R1,HOLD
3911  010652  040367  000026                  BIC     R3,HOLD
3912                                                                            ;AND SEND.
3913  010656                                                                LET @TBUF := HOLD
3914  010656  016777  000022  170404          MOV     HOLD,@TBUF
```

```
3915
3916  010664                                                          ;INCREMENT CHAR COUNT
3917  010664  005201                        INC     R1              LET R1 := R1 + #1
3918
3919  010666                                                          ;ALL DONE
3920  010666  020167  177750                CMP     R1,NUMBER       IF R1 EQ NUMBER THEN
3921  010672  001003                        BNE     $202
3922
3923  010674                                                          ;STOP   INTERRUPT PROCESSING
3924  010674  042777  000100  170362        BIC     #XMITIE,@TCSR   LET @TCSR := @TCSR CLR.BY #XMITIE
3925  010702                                                          ENDIF
3926  010702                          $202:
3927
3928  010702  000401                        BR      ZZZ             ; EXIT SRV
3929
3930  010704  000000                                                  HOLD:0
3931
3932  010706                          ZZZ:                            ENDSRV
3933  010706  000002                        RTI
3934
3935
3936
3937                                                                  ;RECEIVER INTERRUPT HANDLER
3938  010710                                                          BGNSRV  REC
3939  010710                          REC:
3940
3941                                                                  ;GET CHAR IN + MASK IT
3942  010710                                                          LET R2 := @RBUF CLR.BY R3
3943  010710  017702  170346                MOV     @RBUF,R2
3944  010714  040302                        BIC     R3,R2
3945
3946  010716                                                          ;RHLD WILL CONTAIN EXPECTED INPUT
3947  010716  010467  000056                MOV     R4,RHLD         LET RHLD := R4 CLR.BY R3
3948  010722  040367  000052                BIC     R3,RHLD
3949
3950
3951  010726                                                          ;DO THEY COMPARE
3952  010726  020267  000046                CMP     R2,RHLD         IF R2 NE RHLD THEN
3953  010732  001412                        BEQ     $203
3954
3955  010734                                                          ;FIRST ERROR
3956  010734  005767  177700                TST     ERRCNT          IF ERRCNT EQ #0 THEN
3957  010740  001005                        BNE     $204
3958
3959  010742                                                          ;SAVE RECORD OF FIRST MISS
3960  010742  116767  000032  177674        MOVB    RHLD,SB             LET SB :B= RHLD
3961  010750                                                              LET WAS :B= R2
3962  010750  110267  177671                MOVB    R2,WAS          ENDIF
3963  010754
3964  010754                          $204:                           ;COUNT IT.
3965                                                                  LET ERRCNT := ERRCNT + #1
3966  010754
3967  010754  005267  177660                INC     ERRCNT          ENDIF
3968  010760
3969  010760                          $203:
3970
```

```
3971
3972  010760
3973  010760  005204                    INC     R4                          ;COUNT THIS CHAR.
3974                                                                        LET R4 := R4 + #1
3975  010762
3976  010762  020467  177654            CMP     R4,NUMBER                   ;ALL DONE?
3977  010766  001003                    BNE     $205                      \ IF R4 EQ NUMBER THEN
3978
3979  010770
3980  010770  042777  000100  170262    BIC     #RCVRIE,@RCSR               ;STOP RECEIVER INTERRUPTS
3981                                                                        LET @RCSR := @RCSR CLR.BY #RCVRIE
3982  010776
3983  010776                          $205:                                 ;MAIN REPEAT LOOP IS CHECKING
3984                                                                        ENDIF
3985
3986  010776  000401                    BR      ZZZZ                        ;FOR 'R4 = NUMBER' ALSO
3987                                                                      ; EXIT SRV
3988  011000  000000
3989  011002                           ZZZZ:                                RHLD:0
3990  011002
3991  011002  000002                    PTI                             ENDSRV
3992
3993  011004                                                          ENDTST
3994
3995
3996
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 40        TEST BREAK GENERATION LOGIC
                                ;*               TRANSMIT KNOWN CHAR WITH BREAK SET
                                ;*               AND COMPARE RECEIVED WITH 0.
                                ;;**********************************************************************
011004  000004                  TST40:  SCOPE
011006  012767  000010  170144          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
011014  012767  000040  170156          MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;SET MAINTENANCE BIT
011022                                                          LET @TCSR := @TCSR SET.BY #MAINT
011022  052777  000004  170234          BIS     #MAINT,@TCSR
                                                                        ;SET BREAK BIT
011030                                                          LET @TCSR := @TCSR SET BY #BREAK
011030  052777  000001  170226          BIS     #BREAK,@TCSR
                                                                        ;NON-ZERO CHAR. '*'
011036                                                          LET @TBUF := #252
011036  012777  000252  170224          MOV     #252,@TBUF
011044                                                          REPEAT
011044                                  $206:
                                                                        ;WAIT FOR DONE
011044                                                          UNTIL  #RCVRDONE SETIN @RCSR
011044  032777  000200  170206          BIT     #RCVRDONE,@RCSR
011052  001774                          BEQ     $206
                                                                IFB @RBUF NE #0 THEN
011054
011054  105777  170202                  TSTB    @RBUF
011060  001401                          BEQ     $207
                                                                        ; BREAK DID NOT EQUAL 0
                                                                    ERRHRD 81 ,BADBRK
011062
011062  104121                          ERROR   81
                                                                ENDIF
011064
011064                                  $207:
                                                                BRESET  ;CLEAN UP
011064
011064  000005                          RESET
                                                                EXIT
011066
011066  000413                          BR      TST41           ;;;EXIT THIS TEST
011070  051102  040505  020113                                  BADBRK: .ASCIZ  /BREAK DID NOT EQUAL 0/
011076  044504  020104  047516
011104  020124  050505  040525
011112  020114  000060
                                ENDTST
011116
```

```
 4040
 4041                                   ;;**********************************************************************
 4042                                   ;*TEST 41        NOT A TEST - SEND BACK TO LOOP
 4043                                   ;;**********************************************************************
 4044  011116  000004                  TST41:  SCOPE
 4045  011120  012767  000001  170032          MOV     #1,$TIMES       ;;DO 1 ITERATION
 4046  011126  104401  011134                  TYPE    ,65$            ;;TYPE ASCIZ STRING
 4047  011132  000404                          BR      64$             ;;GET OVER THE ASCIZ
 4048                                   ;;65$:  .ASCIZ  <CRLF>*CSR: *
 4049  011144                          64$:
 4050  011144  016746  170104                  MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
 4051  011150  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4052  011152  104401  011160                  TYPE    ,67$            ;;TYPE ASCIZ STRING
 4053  011156  000405                          BR      66$             ;;GET OVER THE ASCIZ
 4054                                   ;;67$:  .ASCIZ  *,VECTOR: *
 4055  011172                          66$:
 4056  011172  016746  170060                  MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
 4057  011176  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4058  011200  104401  011206                  TYPE    ,69$            ;;TYPE ASCIZ STRING
 4059  011204  000405                          BR      68$             ;;GET OVER THE ASCIZ
 4060                                   ;;69$:  .ASCIZ  *,ERRORS: *
 4061  011220                          68$:
 4062  011220  016746  167666                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
 4063  011224  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 4064  011226  005067  167660                  CLR     $ERTTL          ;RESET FOR NEXT DEVICE/PASS
 4065  011232  000167  170462                  JMP     LOOP            ;BACK UP TO THE BEGINNING
```

```
4066
4067                                    ;;BGNMOD        SUBS
4068                                ;;**************************************************************
4069  011236                                                ROUTINE TIMER (HOWLONG,WHICHBIT,REG,SETCLR)
4070  011236                        TIMER:
4071                                ;* ROUTINE:TIMER
4072                                ;*      THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
4073                                ;*      IN ANY REGISTER.
4074                                ;*      INPUTS:
4075                                ;*            HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
4076                                ;*                       THIS ROUTINE.
4077                                ;*            WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
4078                                ;*                         TO BE CHECKED.
4079                                ;*            REG      A POINTER TO THE REGISTER TO BE CHECKED
4080                                ;*            SETCLR   THE DESIRED RESULTS
4081                                ;*                     EITHER #SET OR #CLEAR
4082                                ;*      OUTPUT:
4083                                ;*            THE 'C' BIT IS SET TO INDICATE AN ERROR
4084                                ;*            BUT IT IS TESTED BY THE IF.ERROR STATEMENT
4085                                ;*
4086                                ;*
4087                                ;*            NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
4088                                ;*                    MECHANISM BETWEEN THE CALLER AND THE CALLED
4089                                ;;**************************************************************
4090
4091          000001                                        TRUE=   1
4092          000000                                        FALSE=  0
4093
4094  011236                        LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
4095  011236  016567  000004  000136        MOV     REG(R5),REGSAV
4096  011244                        LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
4097  011244  016567  000000  000132        MOV     HOWLONG(R5),TIMSAV
4098  011252                        LET     FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA
4099  011252  112767  000000  000126        MOVB    #FALSE,FLAG
4100
4101
4102                                ;       START OF AN INFINITE LOOP
4103
4104  011260                        LOOP
4105  011260                    $212:
4106                                    ; TEST TO SEE IF WHICHBIT IS SET
4107  011260                        IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
4108  011260  035577  000002  000114        BIT     WHICHBIT(R5),@REGSAV
4109  011266  001004                        BNE     $214
4110  011270                            LET     HOLDSC :B= #CLR
4111  011270  112767  000000  000111        MOVB    #CLR,HOLDSC
4112  011276                        ELSE
4113  011276  000403                        BR      $215
4114  011300                    $214:
4115  011300                            LET     HOLDSC :B= #SET ; REMEMBER THIS
4116  011300  112767  177777  000101        MOVB    #SET,HOLDSC
4117  011306                        ENDIF
4118  011306                    $215:
4119
4120                                ; NOW SEE IF THAT WAS WHAT WE WANTED
4121  011306                        IFB     HOLDSC EQ SETCLR(R5) THEN
```

```
4122  011306  126765  000075  000006          CMPB    HOLDSC,SETCLR(R5)
4123  011314  001003                          BNE     $216
4124
4125  011316
4126  011316  112767  000001  000062          MOVB    #TRUE,FLAG
4127  011324
4128  011324                          $216:
4129
4130  011324
4131  011324  126727  000056  000001          CMPB    FLAG,#TRUE
4132  011332  001414                          BEQ     $213
4133  011334  005767  000044                  TST     TIMSAV
4134  011340  003411                          BLE     $213
4135
4136
4137
4138  011342
4139  011342  010546                          MOV     R5,-(SP)
4140  011344  012745  000010                  MOV     #10,-(R5)
4141  011350  004767  000140                  JSR     PC,WAIT
4142  011354  012605                          MOV     (SP)+,R5
4143  011356
4144  011356  005367  000022                  DEC     TIMSAV
4145  011362
4146  011362  000736                          BR      $212
4147  011364                          $213:
4148
4149                                          ; ONLY 2 WAYS TO GET HERE
4150                                          ;       1). WE RAN OUT OF TIME---ERROR ! !
4151                                          ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
4152
4153  011364                                                          IFB     FLAG EQ #TRUE THEN
4154  011364  126727  000016  000001          CMPB    FLAG,#TRUE
4155  011372  001001                          BNE     $217
4156  011374                                                              RETURN  NO.ERROR    ; GOOD
4157  011374  000405                          BR      $210
4158  011376                                                          ENDIF
4159  011376                          $217:
4160  011376                                                          RETURN  ERROR           ; BAD
4161  011376  000261                          SEC
4162  011400  000404                          BR      $211
4163
4164  011402  000000                                                  REGSAV: .WORD 0
4165  011404  000000                                                  TIMSAV: .WORD 0
4166  011406     000                                                  FLAG:   .BYTE 0
4167  011407     000                                                  HOLDSC: .BYTE 0
4168                                                          ; WE ARE DONE  GO BACK HOME
4169  011410                                                              ENDRTN
4170  011410                          $210:
4171  011410  000241                          CLC
4172  011412                          $211:
4173  011412  000207                          RTS     PC
```

```
        ; JUST THE THING WE NEEDED
            LET     FLAG :B= #TRUE

        ENDIF


    EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0


        ; ONE WAY OR THE OTHER, WE ARE DONE
; IF WE ARE STILL HERE THEN HANG AROUND A WHILE

        WAITUS  10          ;WAIT FOR 10 MILLI-SECONDS


        LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN

    ENDLOOP                     ; CONTINUED AT THE TOP
```

```
                                ;;*****************************************************************
                                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.
                                ;;*****************************************************************
                                                                        LET MASK(R5) := #0              ; START
011414  005065  000000                  CLR     MASK(R5)
                                                                        LET NUMBR := $USWR AND #DATA
011420  016767  167574  000062          MOV     $USWR,NUMBR
011426  016746  000056                  MOV     NUMBR,-(SP)
011432  042716  000017                  BIC     #DATA,(SP)
011436  042667  000046                  BIC     (SP)+,NUMBR

011442                                                                  INCR I FROM #1 TO NUMBR BY #1
011442  012767  000001  167622          MOV     #1,I
011450  000402                          BR      $222
011452                          $223:
011452  005267  167614                  INC     I
011456                          $222:
011456  026767  167610  000024          CMP     I,NUMBR
011464  003006                          BGT     $224
011466                                                                      LET MASK(R5) := MASK(R5) SHIFT #1
011466  006365  000000                  ASL     MASK(R5)
011472                                                                      LET MASK(R5) := MASK(R5) SET.BY #1
011472  052765  000001  000000          BIS     #1,MASK(R5)
011500                                                                  ENDINC
011500  000764                          BR      $223
011502                          $224:
011502                                                                  LET MASK(R5) := COMP MASK(R5)
011502  005165  000000                  COM     MASK(R5)
011506                                                          RETURN
011506  000401                          BR      $220
011510  000000                          NUMBR:0
011512                                                  ENDRTN
011512                          $220:
011512                          $221:
011512  000207                          RTS     PC
```

```
                                ;;*****************************************************************
                                ROUTINE WAIT    (TIME)
                                WAIT:
                                ;* ROUTINE:WAIT
                                ;*      THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
                                ;*      MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
                                ;*      THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
                                ;*      REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
                                ;*      TO APPROXIMATE 1 MICRO SEC.
                                ;;*****************************************************************
011514  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
011516  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
011520  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
011522                                  LET R1 := TIME(R5)
011522  016501  000000                  MOV     TIME(R5),R1
011526                                  INCRU R2 FROM #1 TO R1 BY #1
011526  012702  000001                  MOV     #1,R2
011532  000402                          BR      $227
011534                          $230:
011534  062702  000001                  ADD     #01,R2
011540                          $227:
011540  020201                          CMP     R2,R1
011542  101010                          BHI     $231
011544                                  INCR R3 FROM #0 TO #700 BY #1
011544  005003                          CLR     R3
011546  000401                          BR      $232
011550                          $233:
011550  005203                          INC     R3
011552                          $232:
011552  020327  000700                  CMP     R3,#700
011556  003001                          BGT     $234
011560                                  ENDINC
011560  000773                          BR      $233
011562                          $234:
011562                                  ENDINC
011562  000764                          BR      $230
011564                          $231:
011564  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
011566  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
011570  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
011572                          ENDRTN
011572                          $225:
011572                          $226:
011572  000207                          RTS     PC
```

```
                                .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
                                ;;*****************************************************************
011574                          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
011574  005267  000002                INC     INTFLAG
011600                                  ENDSRV                  ;THAT'S ALL
011600  000002                          RTI
011602  000000                INTFLAG:  0
```

```
                                ROUTINE MYTYPE
                                MYTYPE:
                                ;;*****************************************************************
011604  104401  011612                TYPE    ,65$            ;;TYPE ASCIZ STRING
011610  000405                      BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>*TEST # *
                                64$:
011624                                  MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
011624  016746  167350                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011630  104402                        TYPE    ,67$            ;;TYPE ASCIZ STRING
011632  104401  011640                BR      66$             ;;GET OVER THE ASCIZ
011636  000405                  ;;67$:  .ASCIZ  *,ERROR # *
                                66$:
011652                                  MOVB    $ITEMB,$FATAL   ; APT FATAL ERROR NUMBER
011652  116767  167236  167316        MOV     $FATAL,-(SP)    ;;SAVE $FATAL FOR TYPEOUT
011660  016746  167312                TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011664  104405                        TYPE    ,69$            ;;TYPE ASCIZ STRING
011666  104401  011674                BR      68$             ;;GET OVER THE ASCIZ
011672  000404                  ;;69$:  .ASCIZ  *,PC = *
                                68$:
011704                                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
011704  016746  167206                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011710  104402                        TYPE    ,71$            ;;TYPE ASCIZ STRING
011712  104401  011720                BR      70$             ;;GET OVER THE ASCIZ
011716  000404                  ;;71$:  .ASCIZ  *,CSR: *
                                70$:
011730                                  MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
011730  016746  167320                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011734  104402                        TYPE    ,73$            ;;TYPE ASCIZ STRING
011736  104401  011744                BR      72$             ;;GET OVER THE ASCIZ
011742  000405                  ;;73$:  .ASCIZ  *,VECTOR: *
                                72$:
011756                                  MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
011756  016746  167274                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
011762  104402                  ENDRTN
011764                          $235:
011764                          $236:
011764  000207                        RTS     PC
```

```
4324  011766                          ROUTINE CYCLE
4325  011766                          CYCLE:
4326                                  ;;*****************************************************************
4327                                  ;* ROUTINE:     CYCLE
4328                                  ;*       THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
4329                                  ;*       ADDRESS  OF  DLV11-E  UNDER  TEST,  ADRS +2 TO
4330                                  ;*       POINT TO THE VECTOR OF THE DLV11-E UNDER TEST.
4331                                  ;*      IT  KEEPS  TRACK  OF THE CURRENT DEVICE AND BIT
4332                                  ;*      MASKS.
4333                                  ;;*****************************************************************
4334  011766                                                  REPEAT
4335  011766                          $241:
4336  011766                                                      IF BITMASK EQ #0 THEN
4337  011766  005767  000122                  TST     BITMASK
4338  011772  001027                          BNE     $242
4339  011774                                                          IF INITFLAG EQ #1 THEN
4340  011774  026727  000116  000001          CMP     INITFLAG,#1
4341  012002  001003                          BNE     $243
4342  012004                                                              LET INITFLAG := #0
4343  012004  005067  000106                  CLR     INITFLAG
4344  012010                                                          ELSE
4345  012010  000403                          BR      $244
4346  012012                          $243:
4347  012012                                                              CALL $EOP   ; AS A SUBROUTINE
4348  012012  004767  000110                  JSR     PC,$EOP
4349                                  ; BECAUSE $EOP RETURNS AS A JUMP
4350  012016                          SPECIALADDRESS:                         LET R0 := POP
4351  012016                                                          ENDIF
4352  012016  012600                          MOV     (SP)+,R0
4353  012020                                                          LET BITMASK := #1
4354  012020                          $244:
4355  012020                                                          LET $DEVCT := #1
4356  012020  012767  000001  000066          MOV     #1,BITMASK
4357  012026                                                          LET ADDRESS := $BASE
4358  012026  012767  000001  167150          MOV     #1,$DEVCT
4359  012034                                                          LET VECTOR := $VECT1
4360  012034  016767  167210  000056          MOV     $BASE,ADDRESS
4361  012042                                                      ELSE
4362  012042  016767  167176  000052          MOV     $VECT1,VECTOR
4363  012050                                                          LET R4 := #10
4364  012050  000410                          BR      $245
4365  012052                          $242:                           LET BITMASK := BITMASK ROTATE 1
4366  012052                                                          LET ADDRESS  := ADDRESS + R4
4367  012052  012704  000010                  MOV     #10,R4
4368  012056                                                          LET VECTOR := VECTOR + R4
4369  012056  006167  000032                  ROL     BITMASK
4370  012062                                                      ENDIF
4371  012062  060467  000032                  ADD     R4,ADDRESS
4372  012066                          $245:                   UNTIL   BITMASK SETIN $DEVM
4373  012066  060467  000030                  ADD     R4,VECTOR
4374  012072
4375  012072
4376  012072
4377  012072  036767  000016  167152          BIT     BITMASK,$DEVM
4378  012100  001732                          BEQ     $241
4379
```

```
012102                                                 LET ADRS := #ADDRESS
012102  012701  012120            MOV     #ADDRESS,ADRS
012106                                            LET $DEVCT := $DEVCT + #1
012106  005267  167072            INC     $DEVCT
012112                                            RETURN
012112  000404                    BR      $237
012114  000000           BITMASK:         0
012116  000001           INITFLAG:        1
012120  000000           ADDRESS:         0
012122  000000           VECTOR:          0

012124                                            ENDRTN
012124                   $237:
012124                   $240:
012124  000207                    RTS     PC


```

```
                                .SBTTL  END OF PASS ROUTINE

                                ;;**************************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
                                ;*TYPE "END PASS #XXXXX" (WHERE XXXXX IS A DECIMAL NUMBER)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN T JUMP TO SPECIALADDRESS

012126                          $EOP:
                                        SCOPE
012126  000004
012130  005067  166746                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
012134  005067  167020                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
012140  005267  167036                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
012144  042767  100000  167030          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
012152  005327                          DEC     (PC)+           ;;LOOP?
012154  000001                  $EOPCT: .WORD   1
012156  003022                          BGT     $DOAGN          ;;YES
012160  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
012162  000001                  $ENDCT: .WORD   1
012164  012154                          $EOPCT
012166  104401  012233                  TYPE    ,$ENDMG         ;;TYPE "END PASS #"
012172  016746  167004                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
012176  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
012200  104401  012230                  TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
012204  013700  000042          $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
012210  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
012212  000005                          RESET                   ;;CLEAR THE WORLD
012214  004710                  $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
012216  000240                          NOP                     ;;SAVE ROOM
012220  000240                          NOP                     ;;FOR
012222  000240                          NOP                     ;;ACT11
012224                          $DOAGN:
012224  000137                          JMP     @(PC)+          ;;RETURN
012226  012016                  $RTNAD: .WORD   SPECIALADDRESS
012230     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
012233     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
012240  050040  051501  020123
012246  000043
```

```
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;*****************************************************************
                                ;POWER DOWN ROUTINE
012250  012737  012410  000024 $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
012256  012737  000340  000026         MOV     #340,@#PWRVEC+2  ;;PRIO:7
012264  010046                        MOV     R0,-(SP)         ;;PUSH R0 ON STACK
012266  010146                        MOV     R1,-(SP)         ;;PUSH R1 ON STACK
012270  010246                        MOV     R2,-(SP)         ;;PUSH R2 ON STACK
012272  010346                        MOV     R3,-(SP)         ;;PUSH R3 ON STACK
012274  010446                        MOV     R4,-(SP)         ;;PUSH R4 ON STACK
012276  010546                        MOV     R5,-(SP)         ;;PUSH R5 ON STACK
012300  017746  166634                MOV     @SWR,-(SP)       ;;PUSH @SWR ON STACK
012304  010667  000104                MOV     SP,$SAVR6        ;;SAVE SP
012310  012737  012322  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
012316  000000                        HALT
012320  000776                        BR      -2               ;;HANG UP

                                ;;*****************************************************************
                                ;POWER UP ROUTINE
012322  012737  012410  000024 $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
012330  016706  000060                MOV     $SAVR6,SP        ;;GET SP
012334  005067  000054                CLR     $SAVR6           ;;WAIT LOOP FOR THE TTY
012340  005267  000050         1$:    INC     $SAVR6           ;;WAIT FOR THE INC
012344  001375                        BNE     1$               ;;OF  WORD
012346  012677  166566                MOV     (SP)+,@SWR       ;;POP STACK INTO @SWR
012352  012605                        MOV     (SP)+,R5         ;;POP STACK INTO R5
012354  012604                        MOV     (SP)+,R4         ;;POP STACK INTO R4
012356  012603                        MOV     (SP)+,R3         ;;POP STACK INTO R3
012360  012602                        MOV     (SP)+,R2         ;;POP STACK INTO R2
012362  012601                        MOV     (SP)+,R1         ;;POP STACK INTO R1
012364  012600                        MOV     (SP)+,R0         ;;POP STACK INTO R0
012366  012737  012250  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
012374  012737  000340  000026        MOV     #340,@#PWRVEC+2  ;;PRIO:7
012402  104401                        TYPE                     ;REPORT THE POWER FAILURE
012404  012416               $PWRMG:  .WORD   $POWER           ;;POWER FAIL MESSAGE POINTER
012406  000002                        RTI
012410  000000               $ILLUP:  HALT                     ;;THE POWER UP SEQUENCE WAS STARTED
012412  000776                        BR      .-2              ;;  BEFORE THE POWER DOWN WAS COMPLETE
012414  000000               $SAVR6:  0                        ;;PUT THE SP HERE
012416  005015  047520  042527 $POWER: .ASCIZ  <15><12>"POWER"
012424  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
                                ;*

012426  105767  166525          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
012432  100002                         BPL     1$              ;;BR IF YES
012434  000000                         HALT                    ;;HALT HERE IF NO TERMINAL
012436  000430                         BR      3$              ;;LEAVE
012440  010046                  1$:    MOV     R0,-(SP)        ;;SAVE R0
012442  017600  000002                 MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
012446  122767  000001  166540         CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
012454  001011                         BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
012456  132767  000100  166531         BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
012464  001405                         BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
012466  010067  000004                 MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
012472  004767  000774                 JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
012476  000000                  61$:   .WORD   0               ;;MESSAGE ADDRESS
012500  132767  000040  166507  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
012506  001003                         BNE     60$             ;;YES,SKIP TYPE OUT
012510  112046                  2$:    MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
012512  001005                         BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
012514  005726                         TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
012516  012600                  60$:   MOV     (SP)+,R0        ;;RESTORE R0
012520  062716  000002          3$:    ADD     #2,(SP)         ;;ADJUST RETURN PC
012524  000002                         RTI                     ;;RETURN
012526  122716  000011          4$:    CMPB    #HT,(SP)        ;;BRANCH IF <HT>
012532  001430                         BEQ     8$
012534  122716  000200                 CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
012540  001006                         BNE     5$
012542  005726                         TST     (SP)+           ;;POP  <CR><LF> EQUIV
012544  104401                         TYPE                    ;;TYPE A CR AND LF
012546  001171                         $CRLF
012550  105067  000130                 CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
012554  000755                         BR      2$              ;;GET NEXT CHARACTER
012556  004767  000056          5$:    JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
012562  126726  166370          6$:    CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
012566  001350                         BNE     2$              ;;IF NO GO GET NEXT CHAR.
012570  016746  166360                 MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                               ;;AND THE NULL CHAR.
012574  105366  000001          7$:    DECB    1(SF)           ;;DOES A NULL NEED TO BE TYPED?
012600  002770                         BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
012602  004767  000032                 JSR     PC,$TYPEC       ;;GO TYPE A NULL
012606  105367  000072                 DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
4535  012612  000770                        BR      7$              ;;LOOP
4536
4537                                  ;HORIZONTAL TAB PROCESSOR
4538
4539  012614  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
4540  012620  004767  000014          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
4541  012624  132767  000007  000052        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
4542  012632  001372                        BNE     9$              ;;TAB STOP
4543  012634  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
4544  012636  000724                        BR      2$              ;;GET NEXT CHARACTER
4545  012640  105777  166304        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
4546  012644  100375                        BPL     $TYPEC
4547  012646  116677  000002  166276        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
4548  012654  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
4549  012662  001003                        BNE     1$              ;;BRANCH IF NO
4550  012664  105067  000014                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
4551  012670  000406                        BR      $TYPEX          ;;EXIT
4552  012672  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
4553  012700  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
4554  012702  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
4555  012704  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
4556  012706  000207          $TYPEX: RTS     PC
4557
```

```
                        .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.
012710  022767  000176  166222  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
012716  001074                        BNE     15$             ;;BRANCH IF NO
012720  105777  166220                TSTB    @$TKS           ;;CHAR THERE?
012724  100071                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
012726  117746  166214                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
012732  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
012736  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
012742  001062                        BNE     15$             ;;NO, RETURN TO USER
012744  126727  166164  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
012752  001456                        BEQ     15$             ;;BRANCH IF YES

012754  104401  013435                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
012760  104401  013442        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
012764  016746  165206                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
012770  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012772  104401  013453                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
012776  005046          19$:    CLR     -(SP)           ;;CLEAR COUNTER
013000  005046                  CLR     -(SP)           ;;THE NEW SWR
013002  105777  166136          7$:     TSTB    @$TKS           ;;CHAR THERE?
013006  100375                  BPL     7$              ;;IF NOT TRY AGAIN

013010  117746  166132                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
013014  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII


013020  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
013024  001005                  BNE     10$             ;;BRANCH IF NOT
013026  104401  013430                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
013032  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
013036  000757                  BR      19$             ;;LET'S TRY IT AGAIN

013040  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
013044  001022                  BNE     16$             ;;BRANCH IF NO
013046  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
013052  001403                        BEQ     11$             ;;BRANCH IF YES
013054  016677  000002  166056        MOV     2(SP),@SWR      ;;SAVE NEW SWR
013062  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
013066  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
013072  126727  166037  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
013100  001003                        BNE     15$             ;;BRANCH IF NOT
013102  012777  000100  166034        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
013110  000002          15$:    RTI                     ;;RETURN
013112  004767  177522          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
013116  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+80.,R3  ;;BUFFER FULL?
```

```
4670  013324  101405                          BLOS    4$              ;;BR IF YES
4671  013326  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
4672  013330  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
4673  013332  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
4674  013336  001003                          BNE     3$              ;;SKIP IF NOT
4675  013340  104401  001170          4$:     TYPE    $QUES           ;;TYPE A '?'
4676  013344  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
4677  013346  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
4678  013352  104401  013416                  TYPE    9$
4679  013356  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
4680  013362  001356                          BNE     2$              ;;LOOP IF NOT RETURN
4681  013364  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
4682  013370  104401  001172                  TYPE    $LF             ;;TYPE A LINE FEED
4683  013374  012603                          MOV     (SP)+,R3        ;;RESTORE R3
4684  013376  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
4685  013400  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
4686  013406  012766  013420  000004          MOV     #$TTYIN,4(SP)
4687  013414  000002                          RTI                     ;;RETURN
4688  013416     000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
4689  013417     000                  .BYTE   0                       ;;TERMINATOR
4690  013420  000010          $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
4691  013430  052536  005015     000  $CNTLU: .ASCIZ  /↑U/<15><12>    ;;CONTROL "U"
4692  013435     136  006507  000012  $CNTLG: .ASCIZ  /↑G/<15><12>    ;;CONTROL "G"
4693  013442  005015  053523  020123  $MSWR:  .ASCIZ  <15><12>/SWR = /
4694  013450  020075     000
4695  013453     040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
4696  013460  036440  000040
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE

                                        ;;**********************************************************************
013464  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
013472  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
013500  000403                          BR      $ATYC
013502  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
013510                          $ATYC:
013510  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013512  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013514  105767  000206                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
013520  001450                          BEQ     5$              ;;IF NOT:  BR
013522  122767  000001  165464          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
013530  001031                          BNE     3$              ;;IF NOT:  BR
013532  132767  000100  165455          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
013540  001425                          BEQ     3$              ;;IF NOT:  BR
013542  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
013546  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
013554  005767  165414          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
013560  001375                          BNE     1$              ;;IF NOT:  WAIT
013562  010067  165422                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
013566  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
013570  001376                          BNE     2$
013572  166700  165412                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
013576  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
013600  010067  165406                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
013604  012767  000004  165362          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
013612  000413                          BR      5$
013614  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
013622  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
013630  016746  164142                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
013634  004767  176566                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
013640  000000                  4$:     .WORD   0
013642                          5$:
013642  105767  000062          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
013646  001416                          BEQ     12$             ;;IF NOT:  BR
013650  005767  165340                  TST     $ENV            ;;RUNNING UNDER APT?
013654  001413                          BEQ     12$             ;;IF NOT:  BR
013656  005767  165312          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
013662  001375                          BNE     11$             ;;IF NOT:  WAIT
013664  017667  000004  165304          MOV     @4(SP),$FATAL   ;;GET ERROR #
013672  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
013700  005267  165270                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
013704  105067  000020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
013710  105067  000013                  CLRB    $LFLG           ;;CLEAR LOG FLAG
013714  105067  000006                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
013720  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
013722  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
013724  000207                          RTS     PC              ;;RETURN
013726     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
013727     000          $LFLG:  .BYTE   0               ;;LOG FLAG
013730     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
        013732                          .EVEN
        000200          APTSIZE=200
        000001          APTENV=001
        000100          APTSPOOL=100
```

```
  4753          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

013732                          $ERROR:
013732  104407                          CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
013734  105267  165143        7$:     INCB    $ERFLG                  ;;SET THE ERROR FLAG
013740  001775                        BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
013742  016777  165134  165172        MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER AND ERROR FLAG
013750  032777  002000  165162        BIT     #BIT10,@SWR             ;;BELL ON ERROR?
013756  001402                        BEQ     1$                      ;;NO - SKIP
013760  104401  001164                TYPE    ,$BELL                  ;;RING BELL
013764  005267  165122        1$:     INC     $ERTTL                  ;;COUNT THE NUMBER OF ERRORS
013770  011667  165122                MOV     (SP),$ERRPC             ;;GET ADDRESS OF ERROR INSTRUCTION
013774  162767  000002  165114        SUB     #2,$ERRPC
014002  117767  165110  165104        MOVB    @$ERRPC,$ITEMB          ;;STRIP AND SAVE THE ERROR ITEM CODE
014010  032777  020000  165122        BIT     #BIT13,@SWR             ;;SKIP TYPEOUT IF SET
014016  001004                        BNE     20$                     ;;SKIP TYPEOUTS
014020  004767  175560                JSR     PC,MYTYPE               ;;GO TO USER ERROR ROUTINE
014024  104401  001171                TYPE    ,$CRLF
014030                          20$:
014030  122767  000001  165156        CMPB    #APTENV,$ENV            ;;RUNNING IN APT MODE
014036  001007                        BNE     2$                      ;;NO,SKIP APT ERROR REPORT
014040  116767  165050  000004        MOVB    $ITEMB,21$              ;;SET ITEM NUMBER AS ERROR NUMBER
014046  004767  177430                JSR     PC,$ATY4                ;;REPORT FATAL ERROR TO APT
014052     000            21$:     .BYTE   0
014053     000                     .BYTE   0
014054  000777            22$:     BR      22$                     ;;APT ERROR LOOP
014056  005777  165056        2$:     TST     @SWR                    ;;HALT ON ERROR
014062  100002                        BPL     3$                      ;;SKIP IF CONTINUE
014064  000000                        HALT                            ;;HALT ON ERROR!
014066  104407                        CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
014070  032777  001000  165042  3$:     BIT     #BIT09,@SWR             ;;LOOP ON ERROR SWITCH SET?
014076  001402                        BEQ     4$                      ;;BR IF NO
014100  016716  165004                MOV     $LPERR,(SP)             ;;FUDGE RETURN FOR LOOPING
014104  005767  165052        4$:     TST     $ESCAPE                 ;;CHECK FOR AN ESCAPE ADDRESS
014110  001402                        BEQ     5$                      ;;BR IF NONE
014112  016716  165044                MOV     $ESCAPE,(SP)            ;;FUDGE RETURN ADDRESS FOR ESCAPE
014116                          5$:
014116  022737  012214  000042        CMP     #$ENDAD,@#42            ;;ACT-11 AUTO-ACCEPT?
014124  001001                        BNE     6$                      ;;BRANCH IF NO
014126  000000                        HALT                            ;;YES
014130                          6$:
014130  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

014132                          $SCOPE:
014132  104407                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
014134  032777  040000  164776  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
014142  001114                          BNE     $OVER           ;;YES IF SW14=1
                                ;*****START OF CODE FOR THE XOR TESTER*****
014144  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
014146  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
014152  012737  014172  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
014160  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
014164  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
014170  000463                          BR      $SVLAD          ;;GO TO THE NEXT TEST
014172  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
014174  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
014200  000423                          BR      7$              ;;LOOP ON THE PRESENT TEST
014202                          6$:;*****END OF CODE FOR THE XOR TESTER*****
014202  032777  000400  164730          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
014210  001404                          BEQ     2$              ;;BR IF NO
014212  127767  164722  164662          CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
014220  001465                          BEQ     $OVER           ;;BR IF YES
014222  105767  164655          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
014226  001421                          BEQ     3$              ;;BR IF NO
014230  126767  164661  164645          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
014236  101015                          BHI     3$              ;;BR IF NO
014240  032777  001000  164672          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
014246  001404                          BEQ     4$              ;;BR IF NO
014250  016767  164634  164630  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
014256  000446                          BR      $OVER
014260  105067  164617          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
014264  005067  164670                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
014270  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
014272  032777  004000  164640  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
014300  001011                          BNE     1$              ;;BR IF YES
014302  005767  164674                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
014306  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
014310  005267  164570                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
014314  026767  164640  164562          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
014322  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
014324  012767  000001  164552  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
014332  016767  000052  164620          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
014340  105267  164536          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
014344  116767  164532  164626          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 4864  014352  011667  164530                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 4865  014356  011667  164526                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 4866  014362  005067  164574                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 4867  014366  112767  000001  164521         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 4868  014374  016777  164502  164540 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 4869  014402  016716  164500                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 4870  014406  000002                         RTI                     ;;FIXES PS
 4871  014410  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

014412                          $TYPDS:
014412  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
014414  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
014416  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
014420  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
014422  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
014424  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
014430  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
014434  100004                          BPL     1$              ;;BR IF INPUT IS POS.
014436  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
014440  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
014446  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
014450  012703  014626                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
014454  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
014460  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
014462  016001  014616                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
014466  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
014470  002402                          BLT     4$              ;;BR IF DONE
014472  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
014474  000774                          BR      3$
014476  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
014500  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
014502  001002                          BNE     5$              ;;FALL THROUGH IF 0
014504  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
014506  100407                          BMI     7$              ;;BR IF YES
014510  106316                  5$:     ASLB    (SP)            ;;MSD?
014512  103003                          BCC     6$              ;;BR IF NO
014514  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
014522  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
014526  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
014532  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
014534  005720                          TST     (R0)+           ;;JUST INCREMENTING
014536  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
014542  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
014544  003002                          BGT     8$              ;;GO TO EXIT
014546  010502                          MOV     R5,R2           ;;GET THE LSD
014550  000764                          BR      6$              ;;GO CHANGE TO ASCII
014552  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
014554  100003                          BPL     9$              ;;BR IF NO
014556  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
014564  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
014566  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
014570  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
014572  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 4928  014574  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 4929  014576  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 4930  014600  104401  014626                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4931  014604  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4932  014612  012616                        MOV     (SP)+,(SP)
 4933  014614  000002                        RTI                     ;;RETURN TO USER
 4934  014616  023420                $DTBL:  10000.
 4935  014620  001750                        1000.
 4936  014622  000144                        100.
 4937  014624  000012                        10.
 4938  014626  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

014636  017646  000000  $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
014642  116667  000001  000211  MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
014650  112667  000207          MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
014654  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
014660  000406                  BR      $TYPON
014662  112767  000001  000171 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
014670  112767  000006  000165        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
014676  112767  000005  000154 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
014704  010346                  MOV     R3,-(SP)        ;;SAVE R3
014706  010446                  MOV     R4,-(SP)        ;;SAVE R4
014710  010546                  MOV     R5,-(SP)        ;;SAVE R5
014712  116704  000145          MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
014716  005404                  NEG     R4
014720  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
014724  110467  000132          MOVB    R4,$OMODE       ;;SAVE IT FOR USE
014730  116704  000125          MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
014734  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
014740  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
014742  006105          1$:     ROL     R5              ;;ROTATE MSB INTO "C"
014744  000404                  BR      3$              ;;GO DO MSB
014746  006105          2$:     ROL     R5              ;;FORM THIS DIGIT
014750  006105                  ROL     R5
014752  006105                  ROL     R5
014754  010503                  MOV     R5,R3
014756  006103          3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
014760  105367  000076          DECB    $OMODE          ;;TYPE THIS DIGIT?
014764  100016                  BPL     7$              ;;BR IF NO
014766  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
014772  001002                  BNE     4$              ;;TEST FOR 0
014774  005704                  TST     R4              ;;SUPPRESS THIS 0?
014776  001403                  BEQ     5$              ;;BR IF YES
```

```
 4995  015000  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 4996  015002  052703  000060                   BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 4997  015006  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 4998  015012  110367  000040                   MOVB    R3,8$           ;;SAVE FOR TYPING
 4999  015016  104401  015056                   TYPE    ,8$             ;;GO TYPE THIS DIGIT
 5000  015022  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
 5001  015026  003347                          BGT     2$              ;;BR IF MORE TO DO
 5002  015030  002402                          BLT     6$              ;;BR IF DONE
 5003  015032  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 5004  015034  000744                          BR      2$              ;;GO DO THE LAST DIGIT
 5005  015036  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
 5006  015040  012604                          MOV     (SP)+,R4        ;;RESTORE R4
 5007  015042  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 5008  015044  016666  000002  000004         MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 5009  015052  012616                          MOV     (SP)+,(SP)
 5010  015054  000002                          RTI                     ;;RETURN
 5011  015056     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 5012  015057     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 5013  01506C     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 5014  015061     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 5015  015062  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.

015064  010046          $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
015066  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
015072  005740                  TST     -(R0)           ;;BACKUP BY 2
015074  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
015076  006300                  ASL     R0              ;;POSITION FOR INDEXING
015100  016000  015120          MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
015104  000200                  RTS     R0              ;;GO TO ROUTINE


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

015106  011646          $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
015110  016666  000004  000002  MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
015116  000002                  RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
015120  015106          $TRPAD: .WORD   $TRAP2
015122  012426                  $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
015124  014662                  $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
015126  014636                  $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
015130  014676                  $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
015132  014412                  $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

015134  012760                  $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

015136  012710                  $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT SWR
015140  013172                  $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
015142  013312                  $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
        000001          .END
```

```
ABASE = 175610            3#   1008    1049
ACOM1 = 000000         1008
ACOM2 = 000000         1008
ACPUDP= 000000         1008    1023
ADDRES  012120         4360*   4371*   4381    4388#
ADDW0 = 000000         1008
ADDW1 = 000000         1008
ADDW10= 000000         1008
ADDW11= 000000         1008
ADDW12= 000000         1008
ADDW13= 000000         1008
ADDW14= 000000         1008
ADDW15= 000000         1008
ADDW2 = 000000         1008
ADDW3 = 000000         1008
ADDW4 = 000000         1008
ADDW5 = 000000         1008
ADDW6 = 000000         1008
ADDW7 = 000000         1008
ADDW8 = 000000         1008
ADDW9 = 000000         1008
ADEVCT= 000000         1008    1014
ADEVM = 000001            3#   1008    1050
AENV  = 000000         1008    1019
AENVM = 000000         1008    1020
AFATAL= 000000         1008    1011
AMADR1= 000000         1008    1036
AMADR2= 000000         1008    1040
AMADR3= 000000         1008    1043
AMADR4= 000000         1008    1046
AMAMS1= 000000         1008    1030
AMAMS2= 000000         1008    1038
AMAMS3= 000000         1008    1041
AMAMS4= 000000         1008    1044
AMSGAD= 000000         1008    1016
AMSGLG= 000000         1008    1017
AMSGTY= 000000         1008    1010
AMTYP1= 000000         1008    1031
AMTYP2= 000000         1008    1039
AMTYP3= 000000         1008    1042
AMTYP4= 000000         1008    1045
APASS = 000000         1008    1013
APRIOR= 000000         1008
APTCSU= 000040         4509    4753#
APTENV= 000001         4502    4709    4751#   4785
APTSIZ= 000200         1119    4750#
APTSPO= 000100         4504    4711    4752#
ASWREG= 000000         1008    1021
ATESTM= 000000         1008    1012
AUNIT = 000000         1008    1015
AUSWR = 071110            3#   1008    1022
AVECT1= 000300            3#   1008    1047
AVECT2= 000000         1008    1048
BADBRK  011070         4034#
BAUD  = 007400          912#
BAUDRA  007124         3265#
```

```
BITMAS  012114          4337    4356*   4369*   4377    4386#
BIT0   = 000001          787#
BIT00  = 000001          777#    787     856     877     898
BIT01  = 000002          776#    786     835     855     897
BIT02  = 000004          775#    785     834     854     875     896
BIT03  = 000010          774#    784     833     853     895
BIT04  = 000020          773#    783     852     894
BIT05  = 000040          772#    782     831     851     893
BIT06  = 000100          771#    781     830     850     871     892
BIT07  = 000200          770#    780     829     849     870     891
BIT08  = 000400          769#    779     4838
BIT09  = 001000          768#    778     4796    4846
BIT1   = 000002          786#
BIT10  = 002000          767#    826     4773
BIT11  = 004000          766#    825     865     4853
BIT12  = 010000          765#    824     844     864
BIT13  = 020000          764#    823     843     863     4780
BIT14  = 040000          763#    822     842     862     4824
BIT15  = 100000          762#    821     841     861
BIT2   = 000004          785#
BIT3   = 000010          784#
BIT4   = 000020          783#
BIT5   = 000040          782#
BIT6   = 000100          781#
BIT7   = 000200          780#
BIT8   = 000400          779#
BIT9   = 001000          778#
BPTVEC= 000014           794#
BREAK  = 000001          877#    1253    1266    1269    1283    1286    1300    1305    1011
BRK    = 010000          913#
CABLE  = 020000          914#    1976    2026    2114    2202    2290    2380    2478    2581    3539    3710
CARDET= 010000           824#    2049    2069    2089
CKSWR  = 104407         4769    4795    4823    5055#
CLR    = 000000          815#    2865    4111
CLRSEN= 020000           823#    2137    2157    2177
COMSPD= 000100           907#
COUNT   006312          2867*   2885*   2891    2895    2969#
CR     = 000015          702#    4548    4558
CRLF   = 000200          703#    1143    4049    4292    4519    4558
CYCLE   011766          1148    4325#
DATA   = 000017          904#    4195    4196
DATAIE= 000040           831#    1717    1730    1733    1747    1750    1764    1769    3552    3563    3596
DATAIN= 100000           821#    1946    2412    2430    2439    2456    2511    2535    2558    2618    2640
DATLNG  011414          3633    3730    3815    4177#
DDISP  = 177570          709#     989    1107
DISPLA  001142           989#    1107*   1115*   4772*   4868*
DISPRE  000174           924#    1115
DLADD   001254          1069#   1152*   1157    1160    1162    1165    1168    1171    1187    1219    4050    4312
DLADDR= 175610           811#    1069    1071    1072    1073    1074    1075
DLVEC   001256          1070#   1155*   3295    3416    3600    3834    4056    4318
DSWR   = 177570          708#     988    1106
DTR    = 000002          835#    1492    1495    1509    1511    1524    1527    2046    2066    2086    2134    2154    2174
                        2400    2428    2454
EMTVEC= 000030           797#    1090*   1091*
ERRCNT  010640          3880    3897#   3956    3967*
ERROR  = 100000C         841#    3032    3096
```

F10


```
ERRVEC= 000004            790* 1104  1105* 1116* 4829  4830* 4832* 4835*
EVENOD= 000040            906*
FALSE = 000000           4092* 4099
FLAG    011406           2865* 2880* 2889  4099* 4126* 4131  4154  4166*
FRERR = 020000            843*
FRFD  = 040000            915* 1548
GNS   = ****** U          923  1142  4048  4054  4060  4291  4297  4304  4310  4316  5047  5048  5049
                         5050  5051  5053  5055  5056  5057
GTSWR = 104406           1137  5053*
HOLD    010704           3910* 3911* 3914  3930*
HOLDSC  011407           4111* 4116* 4122  4167*
HOWLON= 000000           4071* 4097
HT    = 000011            700* 4517  4558
I       001272           1076* 1196* 1217* 1220  1222  4200* 4203* 4205
ILLMEM= 000004            802* 1191  1227
INITFL  012116           1145* 4340  4343* 4387*
INTFLA  011602           1203* 1208  3291* 3331  3335  3357* 3373  3424* 3464  3468  3502* 3519  3549*
                         3575  3579  4281* 4284*
INTSRV  011574           1192  3303  3417  4271*
IOTVEC= 000020            795* 1088* 1089*
TTCLRE  006560           3107*
_F    = 000012            701* 4552  4558
LOOP    001720           1146* 4065
MAINT = 000004            875* 1329  1342  1345  1359  1362  1376  1381  2747  2806  2863  3141  3427
                         3486  3498  3532  3622  3697  3720  3861  4008
MASK  = 000000           4178* 4192* 4208* 4210* 4215*
MAX     006310           2891  2895  2968*
MYTYPE  011604           4287* 4782
NEW     007156           3164* 3166* 3182* 3184* 3193  3201  3203  3221  3223  3271*
NUMBER  010642           3875  3898* 3920  3976
NUMBR   011510           4194* 4195  4197* 4205  4218*
OLD     007162           3137* 3139* 3193  3201  3203  3221* 3223* 3272*
ORERR = 040000            842* 3014  3053  3081
PARITY= 000020            905*
PBAUDS= 004000            865*
PBAUD0= 010000            864*
PBAUD1= 020000            863*
PBAUD2= 040000            862*
PBAUD3= 100000            861*
PBR   = 000200            908* 3128
PERR  = 010000            844*
PIRQ  = 177772            707*
PIRQVE= 000240            801*
PR0   = 000000            724* 3313  3362  3433  3555  3626  3723  3828
PR1   = 000040            725*
PR2   = 000100            726*
PR3   = 000140            727*
PR4   = 000200            728*
PR5   = 000240            729*
PR6   = 000300            730*
PR7   = 000340            731* 1193  1401  2390  2488  2591  3304  3418  3507  3839  3845
PS    = 177776            704*  705
PSW   = 177776            705*
PWRVEC= 000024            796* 1094* 1095* 4440* 4441* 4450* 4456* 4468* 4469*
RATES   007104           3153  3234  3239*
RBUF    001262           1072* 1162* 1163* 2841  2955  3010  3053  3081  3096  3152  3668  3733  3772
```

```
                    3943    4023
RCSR    001260      1071#   1160*   1492*   1495    1509*   1511    1524*   1527    1561    1574*   1577    1591*   1594
                    1608*   1613    1639    1652*   1655    1669*   1672    1686*   1691    1717    1730*   1733    1747*
                    1750    1764*   1769    1795    1808*   1811    1825*   1828    1842*   1847    1879    1946    1991
                    2046*   2049    2066*   2069    2086*   2089    2134*   2137    2154*   2157    2174*   2177    2222*
                    2225    2242*   2245    2262*   2265    2310*   2313    2330*   2333    2350*   2353    2400*   2409
                    2412    2428*   2430    2439    2454*   2456    2499*   2508    2511    2527*   2535    2550*   2558
                    2600*   2602    2610*   2618    2632*   2640    2762    2787    2821    2844    2877    2914    2918
                    2921    2941    2958    3172    3430*   3454*   3505*   3552*   3560*   3563*   3565*   3596*   3664
                    3767    3868*   3980*   4019
RCVRAC= 004000       825#   1991    2877    2914    2921
RCVRDO= 000200       829#   1879    2763    2787    2822    2844    2918    2941    2958    3172    3664    3767    4019
RCVRIE= 000100       830#   1795    1808    1811    1825    1828    1842    1847    3430    3454    3505    3868    3980
RDATA0= 000001       856#
RDATA1= 000002       855#
RDATA2= 000004       854#
RDATA3= 000010       853#
RDATA4= 000020       852#
RDATA5= 000040       851#
RDATA6= 000100       850#
RDATA7= 000200       849#
RDCHR = 104410      4671    5056#
RDLIN = 104411      5057#
REC     010710      3837    3939#
REG   = 000004      4071#   4095
REGSAV  011402      4095*   4108    4164#
REQSEN= 000004       834#   1561    1574    1577    1591    1594    1608    1613    2222    2242    2262    2499    2527
                    2550    3560    3565
RESVEC= 000010       791#
RHLD    011000      3947*   3948*   3952    3960    3988#
RING  = 040000       822#   2225    2245    2265
R0050   007104      3248#
R0070   007105      3249#
R0110   007106      3250#
R0135   007107      3251#
R0150   007110      3252#
R0200   007113      3255#
R0300   007111      3253#
R0600   007112      3254#
R100000 007123      3263#
R1800   007114      3256#
R2000   007115      3257#
R2400   007116      3258#
R3600   007117      3259#
R4800   007120      3260#
RSSTAC  001334      1078#   1174
R7200   007121      3261#
R9600   007122      3262#
SB      010644      3884    3899#   3960*
SECREC= 002000       826#   2313    2333    2353
SECXMI= 000010       833#   1639    1652    1655    1669    1672    1686    1691    2310    2330    2350    2600    2610
                    2632
SET   = 177777       814#   2677    2720    2761    2820    2880    2889    3446    4116
SETCLR= 000006      4071#   4122
SPECIA  012016      4350#   4431
STACK = 001100       695#   1086
```

```
START   001336            927    1079#
STKLMT= 177774            706#
SWR     001140            988#   1084    1106*   1108    1114*   1121*   1135    4448    4461*   4568    4605*   4773    4780
                         4792    4796    4824    4838    4840    4846    4853
SWREG   000176            925#   1114    1135    4568    4581
SW0   = 000001            759#
SW00  = 000001            749#    759
SW01  = 000002            748#    758
SW02  = 000004            747#    757
SW03  = 000010            746#    756
SW04  = 000020            745#    755
SW05  = 000040            744#    754
SW06  = 000100            743#    753
SW07  = 000200            742#    752
SW08  = 000400            741#    751
SW09  = 001000            740#    750
SW1   = 000002            758#
SW10  = 002000            739#
SW11  = 004000            738#
SW12  = 010000            737#
SW13  = 020000            736#
SW14  = 040000            735#
SW15  = 100000            734#
SW2   = 000004            757#
SW3   = 000010            756#
SW4   = 000020            755#
SW5   = 000040            754#
SW6   = 000100            753#
SW7   = 000200            752#
SW8   = 000400            751#
SW9   = 001000            750#
TBITVE= 000014            792#
TBUF    001270           1075#   1171*   1172*   2671*   2702*   2754*   2814*   2873*   2990*   3000*   3072*   3161*   3440*
                         3658*   3756*   3914*   4014*
TCSR    001264           1073#   1165*   1166*   1253    1266*   1269    1283*   1286    1300*   1305    1329    1342*   1345
                         1359*   1362    1376*   1381    1411    1424*   1427    1441*   1444    1458*   1463    1913    2678
                         2707    2721    2747*   2806*   2863*   3141*   3310*   3320*   3360*   3427*   3447    3486*   3498*
                         3532*   3622*   3653    3697*   3720*   3751    3861*   3865*   3924*   4008*   4011*
TCSRHI  001266           1074#   1168*   1169*   3155*   3234*
TDATA0= 000001            898#
TDATA1= 000002            897#
TDATA2= 000004            896#
TDATA3= 000010            895#
TDATA4= 000020            894#
TDATA5= 000040            893#
TDATA6= 000100            892#
TDATA7= 000200            891#
TIME  = 000000           4227#   4238
TIMER   011236           2681    2724    2765    2824    3450    4070#
TIMSAV  011404           4097*   4133    4144*   4165#
TKVEC = 000060            799#
TPVEC = 000064            800#
TRAN    010646           3843    3906#
TRAPVE= 000034            798#   1092*   1093*
TRTVEC= 000014            793#
TRUE  = 000001           4091#   4126    4131    4154
```

```
TST1    002034          1183#
TST10   003326          1709#
TST11   003470          1787#
TST12   003632          1868#
TST13   003672          1902#
TST14   003732          1936#
TST15   003772          1970#
TST16   004052          1981    2021#
TST17   004212          2031    2109#
TST2    002202          1244#
TST20   004352          2119    2197#
TST21   004512          2207    2285#
TST22   004652          2295    2375#
TST23   005056          2385    2473#
TST24   005302          2483    2567    2576#
TST25   005470          2586    2658#
TST26   005636          2741#
TST27   005752          2800#
TST3    002344          1320#
TST30   006062          2858#
TST31   006314          2903    2930    2967    2978#
TST32   006624          3023    3043    3061    3091    3106    3124#
TST33   007166          3132    3237    3286#
TST34   007440          3409#
TST35   010136          3544    3617#
TST36   010276          3706#
TST37   010470          3715    3808#
TST4    002506          1397#
TST40   011004          3895    4003#
TST41   011116          4033    4044#
TST5    002662          1485#
TST6    003002          1544#
TST7    003164          1552    1631#
TYPDS = 104405          4063    4301    4420    5051#
TYPE  = 104401          1129    4046    4052    4058    4289    4295    4302    4308    4314    4418    4421    4470    4522
                        4579    4580    4583    4596    4607    4626    4675    4678    4682    4775    4783    4930    4999
                        5047#
TYPOC = 104402          4051    4057    4294    4307    4313    4319    4582    5048#
TYPON = 104404          5050#
TYPOS = 104403          5049#
VECTOR  012122          4362*   4373*   4389#
WAIT    011514          2405    2504    2532    2555    2615    2637    2995    3005    3077    3326    3370    3459    3515
                        3570    3761    4141    4226#
WAS     010645          3886    3900#   3962#
WHICHB= 000002          4071#   4108
XMITIE= 000100           871#   1411    1424    1427    1441    1444    1458    1463    3310    3320    3360    3865    3924
XMITRD= 000200           870#   1913    2679    2707    2722    3448    3653    3751
ZZZ     010706          3928    3932#
ZZZZ    011002          3986    3989#
$APTHD  001000           948     954#
$ASTAT= ****** U        4731    4746
$ATYC   013510          4702    4704#
$ATY1   013464          4700#
$ATY3   013472          4507    4701#
$ATY4   013502          4703#   4788
$AUTOB  001134           985#   1139*   4576    4697
```

```
$BASE   001250          1049#   4360
$BOADR  001122           980#
$BODAT  001126           982#
$BELL   001164          1000#   4775    4808
$BGNLE= 177777             3#
$CHARC  012704          4524#   4534#   4541    4550#   4555#
$CKSWR  012710          4568#   5055
$CMTAG  001100           968#   1081    1082    1090    1096    1097    1098
$CM3  = 000000           998#
$CNTLG  013435          4579    4692#
$CNTLU  013430          4596    4691#
$CPUOP  001222          1023#
$CRLF   001171          1002#   4523    4558    4607    4691    4783    4808
$DBLK   014626          4896    4930    4938#
$DEVCT  001204          1014#   4358*   4383*
$DEVM   001252          1050#   4377
$DOAGN  012224          4414    4423    4429#
$DTBL   014616          4899    4934#
$ENDAD  012214           934    1127    4425#   4803
$ENDCT  012162          1096    4416#
$ENDMG  012233          4418    4433#
$ENULL  012230          4421    4432#
$ENV    001214          1019#   1133    4502    4709    4733    4785
$ENVM   001215          1020#   1119    4504    4509    4711
$EOP    012126          4348    4406#
$EOPCT  012154          1096*   4413#   4417
$ERFLG  001103           971#   4770*   4808    4813    4842    4844    4850*   4872
$ERMAX  001115           977#   1099*   4844    4867*   4872
$ERRFL= 000000          1145#   1146    1148#   1152#   1153    1155#   1156    1157#   1158    1160#   1161    1162#   1164
                        1165#   1167    1168#   1170    1171#   1173    1174#   1175    1187#   1188    1190#   1191#   1192#
                        1193#   1194#   1195    1196#   1197    1200#   1201    1203#   1204    1217#   1218    1219#   1221
                        1227#   1228#   1230#   1231#   1232    1250#   1251    1264#   1265    1266#   1267    1280#   1281
                        1283#   1284    1297#   1298    1300#   1301    1326#   1327    1340#   1341    1342#   1343    1356#
                        1357    1359#   1360    1373#   1374    1376#   1377    1408#   1409    1422#   1423    1424#   1425
                        1438#   1439    1441#   1442    1455#   1456    1458#   1459    1490#   1491    1492#   1493    1506#
                        1507    1509#   1510    1522#   1523    1524#   1525    1551#   1552    1558#   1559    1572#   1573
                        1574#   1575    1588#   1589    1591#   1592    1605#   1606    1608#   1609    1636#   1637    1650#
                        1651    1652#   1653    1666#   1667    1669#   1670    1682#   1683    1686#   1687    1714#   1715
                        1728#   1729    1730#   1731    1744#   1745    1747#   1748    1761#   1762    1764#   1765    1792#
                        1793    1806#   1807    1808#   1809    1822#   1823    1825#   1826    1839#   1840    1842#   1843
                        1877#   1878    1910#   1911    1944#   1945    1980#   1981    1988#   1989    2030#   2031    2042#
                        2043    2046#   2047    2062#   2063    2066#   2067    2082#   2083    2086#   2087    2118#   2119
                        2130#   2131    2134#   2135    2150#   2151    2154#   2155    2170#   2171    2174#   2175    2206#
                        2207    2218#   2219    2222#   2223    2238#   2239    2242#   2243    2258#   2259    2262#   2263
                        2294#   2295    2306#   2307    2310#   2311    2326#   2327    2330#   2331    2346#   2347    2350#
                        2351    2384#   2385    2397#   2398    2400#   2401    2403#   2409#   2410    2424#   2425    2428#
                        2429    2450#   2451    2454#   2455    2482#   2483    2495#   2496    2499#   2500    2502#   2508#
                        2509    2522#   2523    2527#   2528    2530#   2546#   2547    2550#   2551    2553#   2585#   2586
                        2600#   2601    2602#   2603    2606#   2607    2610#   2611    2613#   2629#   2630    2632#   2633
                        2635#   2663#   2664    2671#   2672    2676#   2695#   2696    2702#   2703    2719#   2747#   2748
                        2750#   2751    2754#   2755    2760#   2780#   2781    2806#   2807    2808#   2809    2814#   2815
                        2819#   2841#   2842    2863#   2864    2865#   2866    2867#   2868    2873#   2874    2880#   2881
                        2885#   2886    2902#   2903    2929#   2930    2955#   2956    2993#   2984    2990#   2991    2993#
                        3000#   3001    3003#   3010#   3011    3022#   3023    3030#   3031    3042#   3043    3049#   3050
                        3067#   3068    3072#   3073    3075#   3090#   3091    3131#   3132    3137#   3138    3139#   3140
                        3141#   3142    3144#   3145    3147    3151    3152#   3153    3155#   3156    3158#   3159    3161#
```

```
                    3162    3164#   3165    3166#   3167    3176#   3177    3182#   3183    3184#   3185    3221#   3222
                    3223#   3224    3229#   3233    3234#   3235    3291#   3292    3295#   3296    3298#   3299    3301#
                    3302#   3303#   3304#   3305#   3306    3307#   3308    3310#   3311    3320#   3321    3324#   3354#
                    3355    3357#   3358    3360#   3361    3368#   3387#   3388#   3390#   3391#   3392    3415#   3416#
                    3417#   3418#   3419#   3420    3422#   3423    3424#   3425    3427#   3428    3430#   3431    3440#
                    3441    3445#   3454#   3455    3457#   3486#   3487    3496#   3497    3498#   3499    3502#   3503
                    3505#   3506    3513#   3532#   3533    3537#   3538    3543#   3544    3549#   3550    3552#   3553
                    3560#   3561    3563#   3564    3565#   3566    3568#   3596#   3597    3600#   3601    3604#   3605#
                    3607#   3608#   3609    3622#   3623    3632#   3638#   3639    3641    3645    3658#   3659    3668#
                    3669    3676#   3678    3679#   3680    3697#   3698    3714#   3715    3720#   3721    3729#   3733#
                    3734    3736#   3737    3739    3743    3756#   3757    3759#   3772#   3773    3776#   3778    3779#
                    3780    3814#   3834#   3835    3837#   3838    3839#   3840    3843#   3844    3845#   3846    3849#
                    3850    3852#   3853    3855#   3856    3861#   3862    3865#   3866    3868#   3869    3884#   3885
                    3886#   3887    3910#   3912    3914#   3915    3917#   3918    3924#   3925    3943#   3945    3947#
                    3949    3960#   3961    3962#   3963    3967#   3968    3973#   3974    3980#   3981    4008#   4009
                    4011#   4012    4014#   4015    4095#   4096    4097#   4098    4099#   4100    4111#   4112    4116#
                    4117    4126#   4127    4139#   4144#   4145    4157#   4158    4161#   4163    4192#   4193    4194#
                    4198    4200#   4201    4203    4207    4208#   4209    4210#   4211    4215#   4216    4238#   4239
                    4240#   4241    4243    4247    4248#   4249    4251    4255    4281#   4282    4343#   4344    4348#
                    4352#   4353    4356#   4357    4358#   4359    4360#   4361    4362#   4363    4367#   4368    4369#
                    4370    4371#   4372    4373#   4374    4381#   4382    4383#   4384

$ERROR  013732      1090    4768#
$ERRPC  001116       978#   4306    4777*   4778*   4779    4808
$ERRTB  001254      1067#
$ERTTL  001112       975#   4062    4064*   4776*   4808
$ESCAP  001162       999#   1098*   4799    4801    4808    4866*
$ETABL  001214      1018#
$ETEND  001254       960    1051#
$FATAL  001176      1011#   4299*   4300    4737*
$FFLG   013730      4700*   4703*   4731    4740*   4748#
$FILLC  001156       996#   4527    4558
$FILLS  001155       995#   4558
$FSAND= 000310         3#   1210    1255    1271    1288    1307    1331    1347    1364    1383    1413    1429    1446
                    1465    1497    1513    1529    1550    1563    1579    1596    1615    1641    1657    1674    1693
                    1719    1735    1752    1771    1797    1813    1830    1849    1881    1915    1948     978    1993
                    2028    2051    2071    2091    2116    2139    2159    2179    2204    2227    2247    2267    2292
                    2315    2335    2355    2382    2414    2432    2441    2458    2480    2513    2537    2560    2563
                    2620    2642    2709    2789    2846    2879    2897    2916    2920    2923    2943    2960    3016
                    3034    3055    3083    3098    3130    3171    3174    3195    3203    3333    3337    3375    3466
                    3470    3521    3541    3577    3581    3685    3712    3785    3882    3922    3954    3958    3978
                    4025    4110    4124    4156    4339    4342
$FSBAD= 000401         3#   1210    1255    1271    1288    1307    1331    1347    1364    1383    1413    1429    1446
                    1465    1497    1513    1529    1550    1563    1579    1596    1615    1641    1657    1674    1693
                    1719    1735    1752    1771    1797    1813    1830    1849    1881    1915    1948    1978    1993
                    2028    2051    2071    2091    2116    2139    2159    2179    2204    2227    2247    2267    2292
                    2315    2335    2355    2382    2414    2432    2441    2458    2480    2513    2537    2560    2583
                    2620    2642    2709    2789    2846    2879    2897    2916    2920    2923    2943    2960    3016
                    3034    3055    3083    3098    3130    3171    3174    3195    3205    3333    3337    3375    3466
                    3470    3521    3541    3577    3581    3685    3712    3785    3882    3922    3954    3958    3978
                    4025    4110    4124    4156    4339    4342
$FSBLA= 000170         3#
$FSCAS= 000150         3#
$FSDEC= 000220         3#
$FSGOO= 000400         3#   1208    1253    1269    1286    1305    1329    1345    1362    1381    1411    1427    1444
                    1463    1495    1511    1527    1548    1561    1577    1594    1613    1639    1655    1672    1691
                    1717    1733    1750    1769    1795    1811    1828    1847    1879    1913    1946    1976    1991
```

```
                    2026    2049    2069    2089    2114    2137    2157    2177    2202    2225    2245    2265    2290
                    2313    2333    2353    2380    2412    2430    2439    2456    2478    2511    2535    2558    2581
                    2618    2640    2686    2707    2727    2770    2787    2828    2844    2877    2895    2913    2914
                    2918    2921    2941    2958    3014    3032    3053    3081    3096    3128    3168    3169    3172
                    3193    3201    3331    3335    3373    3464    3468    3519    3539    3575    3579    3683    3710
                    3783    3880    3920    3952    3956    3976    4023    4108    4122    4154    4337    4340
$FSIF = 000110        3#    1208    1214    1253    1259    1269    1275    1286    1292    1305    1311    1329    1335
                    1345    1351    1362    1368    1381    1387    1411    1417    1427    1433    1444    1450    1463
                    1469    1495    1501    1511    1517    1527    1533    1548    1554    1561    1567    1577    1583
                    1594    1600    1613    1619    1639    1645    1655    1661    1672    1678    1691    1697    1717
                    1723    1733    1739    1750    1756    1769    1775    1795    1801    1811    1817    1828    1834
                    1847    1853    1879    1890    1913    1924    1946    1959    1976    1983    1991    2006    2026
                    2033    2049    2057    2069    2077    2089    2097    2114    2121    2137    2145    2157    2165
                    2177    2185    2202    2209    2225    2233    2245    2253    2265    2273    2290    2297    2313
                    2321    2333    2341    2353    2361    2380    2387    2412    2418    2420    2436    2439    2445
                    2456    2462    2478    2485    2511    2517    2535    2541    2558    2564    2581    2588    2618
                    2624    2640    2646    2686    2691    2707    2713    2727    2732    2770    2775    2787    2793
                    2828    2833    2844    2850    2877    2882    2887    2895    2905    2918    2921    2932    2934
                    2941    2948    2958    2964    3014    3025    3032    3045    3053    3063    3081    3093    3096
                    3103    3128    3134    3172    3178    3186    3193    3197    3201    3203    3207    3216    3218
                    3331    3335    3341    3348    3350    3373    3379    3464    3468    3475    3481    3483    3519
                    3525    3539    3546    3575    3579    3586    3592    3594    3683    3690    3710    3717    3783
                    3790    3880    3891    3920    3926    3952    3956    3964    3969    3976    3983    4023    4029
                    4108    4113    4118    4122    4128    4154    4159    4337    4340    4345    4354    4364    4375
$FSINC= 000210        3#    3144    3226    3638    3692    3736    3793    4200    4212    4240    4248    4256    4259
$FSLOO= 000200        3#    4105    4131    4146
$FSNAM= 000160        3#
$FSNO = 000403        3#    1208    1253    1269    1286    1305    1329    1345    1362    1381    1411    1427    1444
                    1463    1495    1511    1527    1548    1561    1577    1594    1613    1639    1655    1672    1691
                    1717    1733    1750    1769    1795    1811    1828    1847    1879    1913    1946    1976    1991
                    2026    2049    2069    2089    2114    2137    2157    2177    2202    2225    2245    2265    2290
                    2313    2333    2353    2380    2412    2430    2439    2456    2478    2511    2535    2558    2581
                    2618    2640    2707    2787    2844    2877    2895    2914    2918    2921    2941    2958    3014
                    3032    3053    3081    3096    3128    3169    3172    3193    3201    3203    3331    3335    3373
                    3464    3468    3519    3539    3575    3579    3683    3710    3783    3880    3920    3952    3956
                    3976    4108    4337    4340
$FSOR = 000320        3#    1210    1255    1271    1288    1307    1331    1347    1364    1383    1413    1429    1446
                    1465    1497    1513    1529    1550    1563    1579    1596    1615    1641    1657    1674    1693
                    1719    1735    1752    1771    1797    1813    1830    1849    1881    1915    1948    1978    1993
                    2028    2051    2071    2091    2116    2139    2159    2179    2204    2227    2247    2267    2292
                    2315    2335    2355    2382    2414    2432    2441    2458    2480    2513    2537    2560    2583
                    2620    2642    2709    2789    2846    2879    2897    2916    2920    2923    2943    2960    3016
                    3034    3055    3083    3098    3130    3171    3174    3195    3203    3205    3333    3337    3375
                    3466    3470    3521    3541    3577    3581    3685    3712    3785    3882    3922    3954    3958
                    3978    4025    4110    4124    4156    4339    4342
$FSRTN= 000300        3#    4071    4170    4178    4220    4227    4265    4288    4321    4326    4392
$FSSEL= 000140        3#
$FSUNT= 000130        3#    1198    1222    2875    2889    3650    3653    3661    3664    3748    3751    3764    3767
                    3873    3875    4016    4019    4335    4377
$FSWHI= 000120        3#    2913    2914    2936    3168    3169    3189    3203
$FSYES= 000402        3#    1208    1253    1269    1286    1305    1329    1345    1362    1381    1411    1427    1444
                    1463    1495    1511    1527    1548    1561    1577    1594    1613    1639    1655    1672    1691
                    1717    1733    1750    1769    1795    1811    1828    1847    1879    1913    1946    1976    1991
                    2026    2049    2069    2089    2114    2137    2157    2177    2202    2225    2245    2265    2290
                    2313    2333    2353    2380    2412    2430    2439    2456    2478    2511    2535    2558    2581
                    2618    2640    2707    2787    2844    2877    2895    2914    2918    2921    2941    2958    3014
```

```
                    3032    3053    3081    3096    3128    3169    3172    3193    3201    3203    3331    3335    3373
                    3464    3468    3519    3539    3575    3579    3683    3710    3783    3880    3920    3952    3956
                    3976    4023    4108    4122    4154    4337    4340
$GDADR  001120       979#
$GDDAT  001124       981#
$GET42  012204      4422#
$GTSWR  012760      4580#   5053
$HD   = 000000       680
$HIBTS  001000       955#
$ICNT   001104       972#   4857*   4858    4860*   4871
$IFLEV= 177777         3#   1208#   1214#   1253#   1259#   1269#   1275#   1286#   1292#   1305#   1311#   1329#   1335#
                    1345#   1351#   1362#   1368#   1381#   1387#   1411#   1417#   1427#   1433#   1444#   1450#   1463#
                    1469#   1495#   1501#   1511#   1517#   1527#   1533#   1548#   1554#   1561#   1567#   1577#   1583#
                    1594#   1600#   1613#   1619#   1639#   1645#   1655#   1661#   1672#   1678#   1691#   1697#   1717#
                    1723#   1733#   1739#   1750#   1756#   1769#   1775#   1795#   1801#   1811#   1817#   1828#   1834#
                    1847#   1853#   1879#   1890#   1913#   1924#   1946#   1959#   1976#   1983#   1991#   2006#   2026#
                    2033#   2049#   2057#   2069#   2077#   2089#   2097#   2114#   2121#   2137#   2145#   2157#   2165#
                    2177#   2185#   2202#   2209#   2225#   2233#   2245#   2253#   2265#   2273#   2290#   2297#   2313#
                    2321#   2333#   2341#   2353#   2361#   2380#   2387#   2412#   2418#   2430#   2436#   2439#   2445#
                    2456#   2462#   2478#   2485#   2511#   2517#   2535#   2541#   2558#   2564#   2581#   2588#   2618#
                    2624#   2640#   2646#   2686#   2691#   2707#   2713#   2727#   2732#   2770#   2775#   2787#   2793#
                    2828#   2833#   2844#   2850#   2877#   2887#   2895#   2905#   2916#   2934#   2941#   2948#   2958#
                    2964#   3014#   3025#   3032#   3045#   3053#   3063#   3081#   3093#   3096#   3103#   3128#   3134#
                    3172#   3186#   3193#   3218#   3331#   3350#   3373#   3379#   3464#   3483#   3519#   3525#   3539#
                    3546#   3575#   3594#   3683#   3690#   3710#   3717#   3783#   3790#   3880#   3891#   3920#   3926#
                    3952#   3969#   3976#   3983#   4023#   4029#   4108#   4118#   4122#   4128#   4154#   4159#   4337#
                    4340#   4354#   4375#
$ILLUP  012410      4440    4456    4473#
$INTAG  001135       986#   4608    4697
$ISKO = 000001      1208#   1214    1253#   1259    1269#   1275    1286#   1292    1305#   1311    1329#   1335    1345#
                    1351    1362#   1368    1381#   1387    1411#   1417    1427#   1433    1444#   1450    1463#   1469
                    1495#   1501    1511#   1517    1527#   1533    1548#   1554    1561#   1567    1577#   1583    1594#
                    1600    1613#   1619    1639#   1645    1655#   1661    1672#   1678    1691#   1697    1717#   1723
                    1733#   1739    1750#   1756    1769#   1775    1795#   1801    1811#   1817    1828#   1834    1847#
                    1853    1879#   1890    1913#   1924    1946#   1959    1976#   1983    1991#   2006    2026#   2033
                    2049#   2057    2069#   2077    2089#   2097    2114#   2121    2137#   2145    2157#   2165    2177#
                    2185    2202#   2209    2225#   2233    2245#   2253    2265#   2273    2290#   2297    2313#   2321
                    2333#   2341    2353#   2361    2380#   2387    2412#   2418    2430#   2436    2439#   2445    2456#
                    2462    2478#   2485    2511#   2517    2535#   2541    2558#   2564    2581#   2588    2618#   2624
                    2640#   2646    2686#   2691    2707#   2713    2727#   2732    2770#   2775    2787#   2793    2828#
                    2833    2844#   2850    2877#   2887    2895#   2905    2916#   2934    2941#   2948    2958#   2964
                    3014#   3025    3032#   3045    3053#   3063    3081#   3093    3096#   3103    3128#   3134    3172#
                    3186    3193#   3218    3331#   3350    3373#   3379    3464#   3483    3519#   3525    3539#   3546
                    3575#   3594    3683#   3690    3710#   3717    3783#   3790    3880#   3891    3920#   3926    3952#
                    3969    3976#   3983    4023#   4029    4108#   4118    4122#   4128    4154#   4159    4337#   4375
$ISK1 = 000001      2921#   2932    3201#   3216    3335#   3348    3468#   3481    3579#   3592    3956#   3964    4340#
                    4354
$ITEMB  001114       976#   4299    4779*   4787    4808
$LF     001172      1003#   4558    4682    4691    4808
$LFLG   013727      4741*   4747#
$LOCTA= 177777         3#   1198    1199    1209    1210    1214    1215    1223    1224    1254    1255    1259    1260
                    1270    1271    1275    1276    1287    1288    1292    1293    1306    1307    1311    1329    1330
                    1331    1335    1336    1346    1347    1351    1352    1363    1364    1368    1369    1382    1383
                    1387    1388    1412    1413    1417    1418    1428    1429    1433    1434    1445    1446    1450
                    1451    1464    1465    1469    1470    1496    1497    1501    1502    1512    1513    1517    1518
```

		1528	1529	1533	1534	1549	1550	1554	1555	1562	1563	1567	1568	1578
		1579	1583	1584	1595	1596	1600	1601	1614	1615	1619	1620	1640	1641
		1645	1646	1656	1657	1661	1662	1673	1674	1678	1679	1692	1693	1697
		1698	1718	1719	1723	1724	1734	1735	1739	1740	1751	1752	1756	1757
		1770	1771	1775	1776	1796	1797	1801	1802	1812	1813	1817	1818	1829
		1830	1834	1835	1848	1849	1853	1854	1880	1881	1890	1891	1914	1915
		1924	1925	1947	1948	1959	1960	1977	1978	1983	1984	1992	1993	2006
		2007	2027	2028	2033	2034	2050	2051	2057	2058	2070	2071	2077	2078
		2090	2091	2097	2098	2115	2116	2121	2122	2138	2139	2145	2146	2158
		2159	2165	2166	2178	2179	2185	2186	2203	2204	2209	2210	2226	2227
		2233	2234	2246	2247	2253	2254	2266	2267	2273	2274	2291	2292	2297
		2298	2314	2315	2321	2322	2334	2335	2341	2342	2354	2355	2361	2362
		2381	2382	2387	2388	2413	2414	2418	2419	2431	2432	2436	2437	2440
		2441	2445	2446	2457	2458	2462	2463	2479	2480	2495	2486	2512	2513
		2517	2518	2536	2537	2541	2542	2559	2560	2564	2565	2582	2583	2588
		2589	2619	2620	2624	2625	2641	2642	2646	2647	2686	2687	2691	2692
		2708	2709	2713	2714	2727	2728	2732	2733	2770	2771	2775	2776	2788
		2789	2793	2794	2828	2829	2833	2834	2845	2846	2850	2851	2875	2876
		2878	2879	2882	2883	2884	2887	2888	2890	2891	2892	2893	2894	2896
		2897	2905	2906	2913	2914	2915	2916	2919	2920	2922	2923	2932	2933
		2934	2935	2936	2937	2938	2942	2943	2948	2949	2959	2960	2964	2965
		3015	3016	3025	3026	3033	3034	3045	3046	3054	3055	3063	3064	3082
		3083	3093	3094	3097	3098	3103	3104	3129	3130	3134	3135	3145	3146
		3147	3148	3149	3150	3151	3168	3169	3170	3171	3173	3174	3178	3179
		3180	3186	3187	3189	3190	3191	3194	3195	3197	3198	3199	3202	3203
		3204	3205	3207	3208	3209	3216	3217	3218	3219	3226	3227	3228	3332
		3333	3336	3337	3341	3342	3343	3348	3349	3350	3351	3374	3375	3379
		3380	3465	3466	3469	3470	3475	3476	3477	3481	3482	3483	3484	3520
		3521	3525	3526	3540	3541	3546	3547	3576	3577	3580	3581	3586	3587
		3588	3592	3593	3594	3595	3639	3640	3641	3642	3643	3644	3645	3650
		3651	3654	3655	3661	3662	3665	3666	3684	3685	3690	3691	3692	3693
		3694	3711	3712	3717	3718	3737	3738	3739	3740	3741	3742	3743	3748
		3749	3752	3753	3764	3765	3768	3769	3784	3785	3790	3791	3793	3794
		3795	3873	3874	3876	3877	3881	3882	3891	3892	3921	3922	3926	3927
		3953	3954	3957	3958	3964	3965	3969	3970	3977	3978	3983	3984	4016
		4017	4020	4021	4024	4025	4029	4030	4070	4105	4106	4109	4110	4113
		4114	4115	4118	4119	4123	4124	4128	4129	4132	4133	4134	4135	4146
		4147	4148	4155	4156	4157	4158	4159	4160	4162	4163	4170	4171	4172
		4173	4177	4201	4202	4203	4204	4205	4206	4207	4212	4213	4214	4217
		4218	4220	4221	4222	4226	4241	4242	4243	4244	4245	4246	4247	4249
		4250	4251	4252	4253	4254	4255	4256	4257	4258	4259	4260	4261	4265
		4266	4267	4287	4321	4322	4323	4325	4335	4336	4338	4339	4341	4342
		4345	4346	4347	4354	4355	4364	4365	4366	4375	4376	4378	4379	4385
		4386	4392	4393	4394									
$LPADR	001106	973#	1100*	4848*	4864*	4869	4871							
$LPERR	001110	974#	1101*	1200*	1250*	1264*	1280*	1297*	1326*	1340*	1356*	1373*	1408*	1422*
		1438*	1455*	1490*	1506*	1522*	1558*	1572*	1588*	1605*	1636*	1650*	1666*	1682*
		1714*	1728*	1744*	1761*	1792*	1806*	1822*	1839*	1877*	1910*	1944*	1988*	2042*
		2062*	2082*	2130*	2150*	2170*	2218*	2238*	2258*	2306*	2326*	2346*	2397*	2424*
		2450*	2495*	2522*	2546*	2606*	2629*	2663*	2695*	2750*	2780*	2808*	2983*	3030*
		3049*	3067*	3307*	3354*	3422*	3496*	3537*	4798	4848	4865*	4871		
$LSTCN=	177777	3#	1198	1199	1208	1210	1214	1222	1253	1255	1259	1269	1271	1275
		1286	1288	1292	1305	1307	1311	1329	1331	1335	1345	1347	1351	1362
		1364	1368	1381	1383	1387	1411	1413	1417	1427	1429	1433	1444	1446
		1450	1463	1465	1469	1495	1497	1501	1511	1513	1517	1527	1529	1533
		1548	1550	1554	1561	1563	1567	1577	1579	1583	1594	1596	1600	1613

1615 1619 1639 1641 1645 1655 1657 1661 1672 1674 1678 1691 1693
1697 1717 1719 1723 1733 173? 1739 1750 1752 1756 1769 1771 1775
1795 1797 1801 1811 1813 18' 18?8 1830 1834 1847 1849 1853 1879
1881 18?0 1913 1915 1924 1946 1948 1959 1976 1978 1983 1991 1993
2006 2026 2028 2033 2049 2051 2057 2069 2071 2077 2089 2091 2097
2114 2116 2121 2137 2139 2145 2157 2159 2165 2177 2179 2185 2202
2204 2209 2225 2227 2233 2245 2247 2253 2265 2267 2273 2290 2292
2297 2313 2315 2321 2333 2335 2341 2353 2355 2361 2380 2382 2387
2412 2414 2418 2430 2432 2436 2439 2441 2445 2456 2458 2462 2478
2480 2485 2511 2513 2517 2535 2537 2541 2558 2560 2564 2581 2583
2588 2618 2620 2624 2640 2642 2646 2686 2687 2691 2707 2709 2713
2727 2728 2732 2770 2771 2775 2787 2789 2793 2828 2829 2833 2844
2846 2850 2875 2876 2877 2879 2883 2884 2887 2889 2894 2895 2897
2905 291? 2914 2916 2918 2920 2921 2923 2932 2934 2936 2937 2941
2943 2948 2956 2960 2964 3014 3016 3025 3032 3034 3045 3053 3055
3063 3081 3083 3093 3096 3098 3103 3128 3130 3134 3144 3146 3147
3148 3151 3168 3169 3171 3172 3174 3179 3180 3186 3169 3190 3193
3195 3198 3199 3201 3205 3208 3209 3216 3218 3226 3227 3331 3333
3335 3337 3342 3343 3348 3350 3373 3375 3379 3464 3466 3468 3470
3476 3477 3481 3483 3519 3521 3525 3539 3541 3546 3575 3577 3579
3581 3587 3588 3592 3594 3638 3640 3641 3642 3645 3650 3651 3653
3661 3662 3?64 3683 3685 3690 3692 3693 3710 3712 3717 3736 3738
3739 3740 3?43 3748 3749 3751 3764 3765 3767 3783 3785 3790 3793
3794 3873 3874 3875 3880 3882 3891 3920 3922 3926 3952 3954 3956
3958 3964 3969 3976 3978 3983 4016 4017 4019 4023 4025 4029 4071
4105 4106 ?108 4110 4114 4115 4118 4122 4124 4128 4146 4147 4154
4156 4159 4170 4178 4200 4202 4203 4204 4207 4212 4213 4220 4227
4240 4242 4243 4244 4247 4248 4250 4251 4252 4255 4256 4257 4259
4260 4265 4288 4321 4326 4335 4336 4337 4339 4340 4342 4346 4347
4354 4365 4366 4375 4377 4392

$LSTIN= 000000 3# 1145 1146 1148 1149 1152 1153 1155 1156 1157 1158 1160 1161
1162 1163 1164 1165 1166 1167 1168 1169 1170 1171 1172 1173 1174
1175 1187 1188 1190 1191 1192 1193 1194 1195 1196 1197 1200 1201
1203 1204 1208 1209 1210 1217 1218 1219 1220 1221 1222 1223 1224
1227 1228 1229 1230 1231 1232 1250 1251 1253 1254 1255 ?264 1265
1266 1267 1269 1270 1271 1280 1281 1283 1284 1286 1287 1288 1297
1298 1300 1301 1305 1306 1307 132? 1327 1329 1330 1331 1340 1341
1342 1343 1345 1346 1347 1356 1357 1359 1360 1362 1363 1364 1373
1374 1376 1377 1381 1382 1383 1408 1409 1411 1412 1413 1422 1423
1424 1425 1427 1428 1429 1438 1439 1441 1442 1444 1445 1446 1455
?456 1458 1459 1463 1464 1465 1490 1491 1492 1493 1495 1496 1497
?506 1507 1509 1510 1511 1512 1513 1522 1523 1524 1525 1527 1528
1529 1548 1549 1550 1551 1552 1558 1559 1561 1562 1563 1572 1573
1574 1575 1577 1578 1579 1588 1589 1591 1592 1594 1595 1596 1605
1606 1608 1609 1613 1614 1615 1636 1637 1639 1640 1641 1650 1651
1552 1653 1655 1656 1657 1666 1667 1669 1670 1672 1673 1674 1682
1683 1686 1687 1691 1692 1693 1714 1715 1717 1718 1719 1728 1729
1730 1731 1733 1734 1735 1744 1745 1747 1748 1750 1751 1752 1761
1762 1764 1765 1769 1770 1771 1792 1793 1795 1796 1797 1806 1807
1808 1809 1811 1812 1813 1822 1823 1825 1826 1828 1829 1830 1839
1840 1842 1843 1847 1848 1849 1877 1878 1879 1880 1881 1910 1911
1913 1914 1915 1944 1945 1946 1947 1948 1976 1977 1978 1980 1981
1988 1989 1991 1992 1993 2026 2027 2028 2030 2031 2042 2043 2046
2047 2049 2050 2051 2062 2063 2066 2067 2069 2070 2071 2082 2083
2086 2087 2089 2090 2091 2114 2115 2116 2118 2119 2130 2131 2134
2135 2137 2138 2139 2150 2151 2154 2155 2157 2158 2159 2170 2171

2174	2175	2177	2178	2179	2202	2203	2204	2206	2207	2218	2219	2222
2223	2225	2226	2227	2238	2239	2242	2243	2245	2246	2247	2258	2259
2262	2263	2265	2266	2267	2290	2291	2292	2294	2295	2306	2307	2310
2311	2313	2314	2315	2326	2327	2330	2331	2333	2334	2335	2346	2347
2350	2351	2353	2354	2355	2380	2381	2382	2384	2385	2397	2398	2400
2401	2403	2404	2405	2406	2407	2409	2410	2412	2413	2414	2424	2425
2428	2429	2430	2431	2432	2439	2440	2441	2450	2451	2454	2455	2456
2457	2458	2478	2479	2480	2482	2483	2495	2496	2499	2500	2502	2503
2504	2505	2506	2508	2509	2511	2512	2513	2522	2523	2527	528	2530
2531	2532	2533	2534	2535	2536	2537	2546	2547	2550	2551	2553	2554
2555	2556	2557	2558	2559	2560	2581	2582	2583	2585	2586	2600	2601
2602	2603	2606	2607	2610	2611	2613	2614	2615	2616	2617	2618	2619
2620	2629	2630	2632	2633	2635	2636	2637	2638	2639	2640	2641	2642
2663	2664	2671	2672	2676	2677	2678	2679	2680	2681	2682	2683	2686
2687	2695	2696	2702	2703	2707	2708	2709	2719	2720	2721	2722	2723
2724	2725	272*	2727	2729	2747	2748	2750	2751	2754	2755	2760	2761
2762	2763	27	2765	2766	2767	2770	2771	2780	2781	2787	2788	2789
2806	2807	2808	2809	2814	2815	2819	2820	2821	2822	2823	2824	2825
2826	2828	2829	2841	2842	2844	2845	2846	2863	2864	2865	2866	2867
2868	2873	2874	2877	2878	2879	2880	2881	2882	2883	2885	2886	2889
2890	2891	2892	2893	2895	2896	2897	2902	2903	2914	2915	2916	2918
2919	2920	2921	2922	2923	2929	2930	2936	2937	2941	2942	2943	2955
2956	2958	2959	2960	2983	2984	2990	2991	2993	2994	2995	2996	2997
3000	3001	3003	3004	3005	3006	3007	3010	3011	3014	3015	3016	3022
3023	3030	3031	3032	3033	3034	3042	3043	3049	3050	3053	3054	3055
3067	3068	3072	3073	3075	3076	3077	3078	3079	3081	3082	3083	3090
3091	3096	3097	3098	3128	3129	3130	3131	3132	3137	3138	3139	3140
3141	3142	3144	3145	3146	3147	3148	3149	3150	3151	3152	3153	3155
3156	3158	3159	3161	3162	3164	3165	3166	3167	3169	3170	3171	3172
3173	3174	3176	3177	3178	3179	3182	3183	3184	3185	3189	3190	3193
3194	3195	3197	3198	3201	3202	3203	3204	3205	3207	3208	3221	3222
3223	3224	3226	3227	3229	3230	3231	3232	3233	3234	3235	3291	3292
3295	3296	3298	3299	3301	3302	3303	3304	3305	3306	3307	3308	3310
3311	3320	3321	3324	3325	3326	3327	3328	3331	3332	3333	3335	3336
3337	3341	3342	3354	3355	3357	3358	3360	3361	3368	3369	3370	3371
3372	3373	3374	3375	3387	3388	3389	3390	3391	3392	3415	3416	3417
3418	3419	3420	3422	3423	3424	3425	3427	3428	3430	3431	3440	3441
3445	3446	3447	3448	3449	3450	3451	3452	3454	3455	3457	3458	3459
3460	3461	3464	3465	3466	3468	3469	3470	3475	3476	3486	3487	3496
3497	3498	3499	3502	3503	3505	3506	3513	3514	3515	3516	3517	3519
3520	3521	3532	3533	3537	3538	3539	3540	3541	3543	3544	3549	3550
3552	3553	3560	3561	3563	3564	3565	3566	3568	3569	3570	3571	3572
3575	3576	3577	3579	3580	3581	3586	3587	3596	3597	3600	3601	3604
3605	3606	3607	3608	3609	3622	3623	3632	3633	3634	3635	3638	3639
3640	3641	3642	3643	3644	3645	3653	3654	3655	3658	3659	3664	3665
3666	3668	3669	3676	3677	3678	3679	3680	3683	3684	3685	3692	3693
3697	3698	3710	3711	3712	3714	3715	3720	3721	3729	3730	3731	3732
3733	3734	3736	3737	3738	3739	3740	3741	3742	3743	3751	3752	3753
3756	3757	3759	3760	3761	3762	3763	3767	3768	3769	3772	3773	3776
3777	3778	3779	3780	3783	3784	3785	3793	3794	3814	3815	3816	3817
3834	3835	3837	3838	3839	3840	3843	3844	3845	3846	3849	3850	3852
3853	3855	3856	3861	3862	3865	3866	3868	3869	3875	3876	3877	3880
3881	3882	3884	3885	3886	3887	3910	3911	3912	3914	3915	3917	3918
3920	3921	3922	3924	3925	3943	3944	3945	3947	3948	3949	3952	3953
3954	3956	3957	3958	3960	3961	3962	3963	3967	3968	3973	3974	3976
3977	3978	3980	3981	4008	4009	4011	4012	4014	4015	4019	4020	4021

	4023	4024	4025	4095	4096	4097	4098	4099	4100	4108	4109	4110	4111
	4112	4113	4114	4116	4117	4122	4123	4124	4126	4127	4131	4132	4133
	4134	4135	4139	4140	4141	4142	4143	4144	4145	4146	4147	4154	4155
	4156	4157	4158	4161	4162	4163	4171	4172	4173	4174	4192	4193	4194
	4195	-196	4197	4198	4200	4201	4202	4203	4204	4205	4206	4207	4208
	4209	4210	4211	4212	4213	4215	4216	4217	4218	4222	4223	4238	4239
	4240	4241	4242	4243	4244	4245	4246	4247	4248	4249	4250	4251	4252
	4253	4254	4255	4256	4257	4259	4260	4267	4268	4281	4282	4323	4324
	4337	4338	4339	4340	4341	4342	4343	4344	4345	4346	4348	4349	4352
	4353	4356	4357	4358	4359	4360	4361	4362	4363	4364	4365	4367	4368
	4369	4370	4371	4372	4373	4374	4377	4378	4379	4381	4382	4383	4384
	4385	4386	4394	4395									
$LSTST= 177777	3#	1198	1199	1208	1210	1214	1222	1253	1255	1259	1269	1271	1275
	1286	1288	1292	1305	1307	1311	1329	1331	1335	1345	1347	1351	1362
	1364	1368	1381	1383	1387	1411	1413	1417	1427	1429	1433	1444	1446
	1450	1463	1465	1469	1495	1497	1501	1511	1513	1517	1527	1529	1533
	1548	1550	1554	1561	1563	1567	1577	1579	1583	1594	1596	1600	1613
	1615	1619	1639	1641	1645	1655	1657	1661	1672	1674	1678	1691	1693
	1697	1717	1719	1723	1733	1735	1739	1750	1752	1756	1769	1771	1775
	1795	1797	1801	1811	1813	1817	1828	1830	1834	1847	1849	1853	1879
	1881	1890	1913	1915	1924	1946	1948	1959	1976	1978	1983	1991	1993
	2006	2026	2028	2033	2049	2051	2057	2069	2071	2077	2089	2091	2097
	2114	2116	2121	2137	2139	2145	2157	2159	2165	2177	2179	2185	2202
	2204	2209	2225	2227	2233	2245	2247	2253	2265	2267	2273	2290	2292
	2297	2313	2315	2321	2333	2335	2341	2353	2355	2361	2380	2382	2387
	2412	2414	2418	2430	2432	2436	2439	2441	2445	2456	2458	2462	2478
	2480	2485	2511	2513	2517	2535	2537	2541	2558	2560	2564	2581	2583
	2588	2618	2620	2624	2640	2642	2646	2686	2687	2691	2707	2709	2713
	2727	2728	2732	2770	2771	2775	2787	2789	2793	2828	2829	2833	2844
	2846	2850	2875	2876	2877	2879	2882	2883	2884	2887	2889	2895	2897
	2905	2913	2914	2916	2918	2920	2921	2923	2932	2934	2936	2937	2941
	2943	2948	2958	2960	2964	3014	3016	3025	3032	3034	3045	3053	3055
	3063	3081	3083	3093	3096	3098	3103	3128	3130	3134	3144	3146	3147
	3148	3151	3168	3169	3171	3172	3174	3178	3179	3180	3186	3189	3190
	3193	3195	3197	3198	3199	3201	3205	3207	3208	3209	3216	3218	3226
	3227	3331	3333	3335	3337	3341	3342	3343	3348	3350	3373	3375	3379
	3464	3466	3468	3470	3475	3476	3477	3481	3483	3519	3521	3525	3539
	3541	3546	3575	3577	3579	3581	3586	3587	3588	3592	3594	3638	3640
	3641	3642	3645	3650	3651	3653	3661	3662	3664	3683	3685	3690	3692
	3693	3710	3712	3717	3736	3738	3739	3740	3743	3748	3749	3751	3764
	3765	3767	3783	3785	3790	3793	3794	3873	3874	3875	3880	3882	3891
	3920	3922	3926	3952	3954	3956	3958	3964	3969	3976	3978	3983	4016
	4017	4019	4023	4025	4029	4071	4105	4106	4108	4110	4113	4114	4115
	4118	4122	4124	4128	4131	4146	4147	4154	4156	4159	4170	4178	4200
	4202	4203	4204	4207	4212	4213	4220	4227	4240	4242	4243	4244	4247
	4248	4250	4251	4252	4255	4256	4257	4259	4260	4265	4288	4321	4326
	4335	4336	4337	4339	4340	4342	4345	4346	4347	4354	4364	4365	4366
	4375	4377	4392										
$LSTTA= 000000	3#	1198	1199	1214	1215	1259	1260	1275	1276	1292	1293	1311	1312
	1335	1336	1351	1352	1368	1369	1387	1388	1417	1418	1433	1434	1450
	1451	1469	1470	1501	1502	1517	1518	1533	1534	1554	1555	1567	1568
	1583	1584	1600	1601	1619	1620	1645	1646	1661	1662	1678	1679	1697
	1698	1723	1724	1739	1740	1756	1757	1775	1776	1801	1802	1817	1818
	1834	1835	1853	1854	1890	1891	1924	1925	1959	1960	1983	1984	2006
	2007	2033	2034	2057	2058	2077	2078	2097	2098	2121	2122	2145	2146
	2165	2166	2185	2186	2209	2210	2233	2234	2253	2254	2273	2274	2297

```
                    2298  2321  2322  2341  2342  2361  2362  2387  2388  2418  2419  2436  2437
                    2445  2446  2462  2463  2485  2486  2517  2518  2541  2542  2564  2565  2588
                    2589  2624  2625  2646  2647  2691  2692  2713  2714  2732  2733  2775  2776
                    2793  2794  2833  2834  2850  2851  2875  2876  2883  2884  2887  2888  2893
                    2894  2905  2906  2913  2914  2932  2933  2934  2935  2937  2938  2948  2949
                    2964  2965  3025  3026  3045  3046  3063  3064  3093  3094  3103  3104  3134
                    3135  3146  3147  3148  3149  3168  3169  3179  3180  3186  3187  3190  3191
                    3198  3199  3208  3209  3216  3217  3218  3219  3227  3228  3342  3343  3348
                    3349  3350  3351  3379  3380  3476  3477  3481  3482  3483  3484  3525  3526
                    3546  3547  3587  3588  3592  3593  3594  3595  3640  3641  3642  3643  3650
                    3651  3661  3662  3690  3691  3693  3694  3717  3718  3738  3739  3740  3741
                    3748  3749  3764  3765  3790  3791  3794  3795  3873  3874  3891  3892  3926
                    3927  3964  3965  3969  3970  3983  3984  4016  4017  4029  4030  4070  4071
                    4105  4106  4114  4115  4118  4119  4128  4129  4147  4148  4159  4160  4170
                    4171  4172  4173  4177  4178  4202  4203  4204  4205  4213  4214  4220  4221
                    4222  4226  4227  4242  4243  4244  4245  4250  4251  4252  4253  4257  4258
                    4260  4261  4265  4266  4267  4287  4288  4321  4322  4323  4325  4326  4335
                    4336  4346  4347  4354  4355  4365  4366  4375  4376  4392  4393  4394
$MADR1  001226      1036#
$MADR2  001232      1040#
$MADR3  001236      1043#
$MADR4  001242      1046#
$MAIL   001174       956   960  1009#  1118  1133  1185  1246  1322  1399  1487  1546  1633  1711
                    1789  1870  1904  1938  1972  2023  2111  2199  2287  2377  2475  2578  2660
                    2743  2802  2860  2980  3126  3288  3411  3619  3708  3810  4005  4502  4785
                    4863
$MAMS1  001224      1030#
$MAMS2  001230      1038#
$MAMS3  001234      1041#
$MAMS4  001240      1044#
$MBADR  001002       956#
$MCALL= 000000         1#     3
$MFLG   013726      4701#  4707   4742#  4746#
$MNEW   013453      4583   4695#
$MSGAD  001210      1016#  4717#  4720
$MSGLG  001212      1017#  4722#
$MSGTY  001174      1010#  4715   4723#  4735   4739#
$MSWR   013442      4580   4693#
$MTYP1  001225      1031#
$MTYP2  001231      1039#
$MTYP3  001235      1042#
$MTYP4  001241      1045#
$MXCNT  014410      4861   4871#
$NESTL= 177777         3#  1198#  1208#  1214#  1222#  1253#  1259#  1269#  1275#  1286#  1292#  1305#  1311#
                    1329#  1335#  1345#  1351#  1362#  1368#  1381#  1387#  1411#  1417#  1427#  1433#  1444#
                    1450#  1463#  1469#  1495#  1501#  1511#  1517#  1527#  1533#  1548#  1554#  1561#  1567#
                    1577#  1583#  1594#  1600#  1613#  1619#  1639#  1645#  1655#  1661#  1672#  1678#  1691#
                    1697#  1717#  1723#  1733#  1739#  1750#  1756#  1769#  1775#  1795#  1801#  1811#  1817#
                    1828#  1834#  1847#  1853#  1879#  1890#  1913#  1924#  1946#  1959#  1976#  1983#  1991#
                    2006#  2026#  2033#  2049#  2057#  2069#  2077#  2089#  2097#  2114#  2121#  2137#  2145#
                    2157#  2165#  2177#  2185#  2202#  2209#  2225#  2233#  2245#  2253#  2265#  2273#  2290#
                    2297#  2313#  2321#  2333#  2341#  2353#  2361#  2380#  2387#  2412#  2418#  2430#  2436#
                    2439#  2445#  2456#  2462#  2478#  2485#  2511#  2517#  2535#  2541#  2558#  2564#  2581#
                    2588#  2618#  2624#  2640#  2646#  2686#  2691#  2707#  2713#  2727#  2732#  2770#  2775#
                    2787#  2793#  2828#  2833#  2844#  2850#  2875#  2877#  2882   2887#  2889#  2895#  2905#
                    2913#  2918#  2921#  2932#  2934#  2936#  2941#  2948#  2958#  2964#  3014#  3025#  3032#
```

```
                    3045#  3053#  3063#  3081#  3093#  3096#  3103#  3128#  3134#  3144#  3168#  3172#  3178
                    3186#  3189#  3193#  3197   3201#  3207   3216#  3218#  3226#  3331#  3335#  3341   3348#
                    3350#  3373#  3379#  3464#  3468#  3475   3481#  3483#  3519#  3525#  3539#  3546#  3575#
                    3579#  3586   3592#  3594#  3638#  3650#  3653#  3661#  3664#  3683#  3690#  3692#  3710#
                    3717#  3736#  3748#  3751#  3764#  3767#  3783#  3790#  3793#  3873#  3875#  3880#  3891#
                    3920#  3926#  3952#  3956#  3964#  3969#  3976#  3983#  4016#  4019#  4023#  4029#  4071#
                    4105#  4108#  4113   4118#  4122#  4129#  4131   4146#  4154#  4159#  4170#  4178#  4200#
                    4212#  4220#  4227#  4240#  4248#  4256#  4259#  4265#  4288#  4321#  4326#  4335#  4337#
                    4340#  4345   4354#  4364   4375#  4377#  4392#
$NSKO = 000300       1198#  1222   1253#  1259   1269#  1275   1286#  1292   1305#  1311   1329#  1335   1345#
                     1351   1362#  1368   1381#  1387   1411#  1417   1427#  1433   1444#  1450   1463#  1469
                     1495#  1501   1511#  1517   1527#  1533   1548#  1554   1561#  1567   1577#  1583   1594#
                     1600   1613#  1619   1639#  1645   1655#  1661   1672#  1678   1691#  1697   1717#  1723
                     1733#  1739   1750#  1756   1769#  1775   1795#  1801   1811#  1817   1828#  1834   1847#
                     1853   1879#  1890   1913#  1924   1946#  1959   1976#  1983   1991#  2006   2026#  2033
                     2049#  2057   2069#  2077   2089#  2097   2114#  2121   2137#  2145   2157#  2165   2177#
                     2185   2202#  2209   2225#  2233   2245#  2253   2265#  2273   2290#  2297   2313#  2321
                     2333#  2341   2353#  2361   2380#  2387   2412#  2418   2430#  2436   2439#  2445   2456#
                     2462   2478#  2485   2511#  2517   2535#  2541   2558#  2564   2581#  2588   2618#  2624
                     2640#  2646   2686#  2691   2707#  2713   2727#  2732   2770#  2775   2787#  2793   2828#
                     2833   2844#  2850   2875#  2889   2895#  2905   2913#  2936   2941#  2948   2958#  2964
                     3014#  3025   3032#  3045   3053#  3063   3081#  3093   3096#  3103   3128#  3134   3144#
                     3226   3331#  3350   3373#  3379   3464#  3483   3519#  3525   3539#  3546   3575#  3594
                     3638#  3692   3710#  3717   3736#  3793   3873#  3875   3880#  3891   3920#  3926   3952#
                     3969   3976#  3983   4016#  4019   4023#  4029   4071#  4170   4178#  4220   4227#  4265
                     4288#  4321   4326#  4392
$NSK1 = 000130       1208#  1214   2877#  2882   2887   2918#  2934   3168#  3189   3193#  3197   3218   3335#
                     3341   3348   3468#  3475   3481   3579#  3586   3592   3650#  3653   3661#  3664   3683#
                     3690   3748#  3751   3764#  3767   3783#  3790   3956#  3964   4105#  4131   4146   4154#
                     4159   4200#  4212   4240#  4259   4335#  4377
$NSK2 = 000110       2921#  2932   3172#  3178   3186   3201#  3207   3216   4108#  4113   4118   4122#  4128
                     4248#  4256   4337#  4364   4375
$NSK3 = 000110       4340#  4345   4354
$NULL   001154        994#  4529   4558
$NWTST= 000001       1177#  1179   1239#  1241   1317#  1394#  1476#  1478   1540#  1542   1626#  1628   1704#
                     1706   1782#  1784   1865#  1899#  1933#  1967#  2017#  2019   2105#  2107   2193#  2195
                     2281#  2283   2369#  2371   2469#  2471   2573#  2653#  2655   2736#  2738   2797#  2853#
                     2855   2975#  3118#  3120   3278#  3280   3403#  3405   3613#  3615   3703#  3804#  3806
                     3998#  4000   4041#
$OCNT   015060       4971*  5000*  5013#
$OMODE  015062       4966*  4970*  4975   4978*  4989*  5015#
$OVER   014374       4825   4841   4849   4859   4868#
$PASS   001202       1013#  1118*  4410*  4411*  4419   4432   4855   4872
$PASTM  001006        958#
$POWER  012416       4471   4476#
$PWRDN  012250       1094   4440#  4468
$PWRMG  012404       4471#
$PWRUP  012322       4450   4456#
$QUES   001170       1001#  4558   4626   4675   4691   4808
$RDCHR  013172       4639#  5056
$RDDEC= ****** U     5058
$RDLIN  013312       4667#  5057
$RDOCT= ****** U     5058
$RDSZ = 000010       4660#
$RTNAD  012226       4431#
$R2A  = ****** L     5058
```

```
$SAVLE= 177777          3#  2936#  2937#  3147#  3151#  3189#  3190#  3641#  3645#  3739#  3743#  4146#  4147#
                     4203#  4207#  4243#  4247#  4251#  4255#
$SAVRE= ****** U     5058
$SAVR6  012414       4449*  4457   4458*  4459*  4475#
$SCOPE  014132       1088   4822#
$SETUP= 000137       1079#  1087   1088   1090   1092   1094   1096   1097   1098   1100   1127   1130   4408
                     4563   4697   4769   4795   4803   4823
$SSKO = 000233       2936#  2937   3147#  3151   3189#  3190   3641#  3645   3739#  3743   4146#  4147   4203#
                     4207   4243#  4247   4251#  4255
$STUP = 177777       1079#
$SVLAD  014340       4833   4862#
$SVPC = 000204        932#   937
$SWR  = 167400          3#   680    684    685    686    687    688    689    690    998    999   1000   1097
                     1098   1100   1101   1184   1245   1321   1398   1486   1545   1632   1710   1788   1869
                     1903   1937   1971   2022   2110   2198   2286   2376   2474   2577   2659   2742   2801
                     2859   2979   3125   3287   3410   3618   3707   3809   4004   4045   4403   4409   4424
                     4430   4432   4472   4760   4761   4762   4763   4764   4773   4780   4792   4796   4808
                     4814   4815   4816   4817   4818   4824   4836   4838   4839   4842   4843   4844   4851
                     4852   4853   4865   4868   4871
$SWREG  001216       1021#  1121
$SWRMK= 000000        690    691   4818   4819   4840
$TAGLE= 177777          3#  1199#  1210#  1214#  1222#  1255#  1259#  1271#  1275#  1288#  1292#  1307#  1311#
                     1331#  1335#  1347#  1351#  1364#  1368#  1383#  1387#  1413#  1417#  1429#  1433#  1446#
                     1450#  1465#  1469#  1497#  1501#  1513#  1517#  1529#  1533#  1550#  1554#  1563#  1567#
                     1579#  1583#  1596#  1600#  1615#  1619#  1641#  1645#  1657#  1661#  1674#  1678#  1693#
                     1697#  1719#  1723#  1735#  1739#  1752#  1756#  1771#  1775#  1797#  1801#  1813#  1817#
                     1830#  1834#  1849#  1853#  1881#  1890#  1915#  1924#  1948#  1959#  1978#  1983#  1993#
                     2006#  2028#  2033#  2051#  2057#  2071#  2077#  2091#  2097#  2116#  2121#  2139#  2145#
                     2159#  2165#  2179#  2185#  2204#  2209#  2227#  2233#  2247#  2253#  2267#  2273#  2292#
                     2297#  2315#  2321#  2335#  2341#  2355#  2361#  2382#  2387#  2414#  2418#  2432#  2436#
                     2441#  2445#  2458#  2462#  2480#  2485#  2513#  2517#  2537#  2541#  2560#  2564#  2583#
                     2588#  2620#  2624#  2642#  2646#  2687#  2691#  2709#  2713#  2728#  2732#  2771#  2775#
                     2789#  2793#  2829#  2833#  2846#  2850#  2876#  2879#  2883#  2884#  2887#  2889#  2897#
                     2905#  2914#  2916#  2920#  2923#  2932#  2934#  2936#  2943#  2948#  2960#  2964#  3016#
                     3025#  3034#  3045#  3055#  3063#  3083#  3093#  3098#  3103#  3130#  3134#  3146#  3148#
                     3151#  3169#  3171#  3174#  3179#  3180#  3186#  3189#  3195#  3198#  3199#  3205#  3208#
                     3209#  3216#  3218#  3226#  3227#  3333#  3337#  3342#  3343#  3348#  3350#  3375#  3379#
                     3466#  3470#  3476#  3477#  3481#  3483#  3521#  3525#  3541#  3546#  3577#  3581#  3587#
                     3588#  3592#  3594#  3640#  3642#  3645#  3651#  3653#  3662#  3664#  3685#  3690#  3692#
                     3693#  3712#  3717#  3738#  3740#  3743#  3749#  3751#  3765#  3767#  3785#  3790#  3793#
                     3794#  3874#  3875#  3882#  3891#  3922#  3926#  3954#  3958#  3964#  3969#  3978#  3983#
                     4017#  4019#  4025#  4029#  4106#  4110#  4114#  4115#  4118#  4124#  4128#  4131   4146#
                     4156#  4159#  4202#  4204#  4207#  4212#  4213#  4242#  4244#  4247#  4250#  4252#  4255#
                     4256#  4257#  4259#  4260#  4336#  4339#  4342#  4346#  4347#  4354#  4365#  4366#  4375#
                     4377#
STAGNU= 000246          3#  1198   1199#  1209   1210#  1254   1255#  1270   1271#  1287   1288#  1306   1307#
                     1330   1331#  1346   1347#  1363   1364#  1382   1383#  1412   1413#  1428   1429#  1445
                     1446#  1464   1465#  1496   1497#  1512   1513#  1528   1529#  1549   1550#  1562   1563#
                     1578   1579#  1595   1596#  1614   1615#  1640   1641#  1656   1657#  1673   1674#  1692
                     1693#  1718   1719#  1734   1735#  1751   1752#  1770   1771#  1796   1797#  1812   1813#
                     1829   1830#  1848   1849#  1880   1881#  1914   1915#  1947   1948#  1977   1978#  1992
                     1993#  2027   2028#  2050   2051#  2070   2071#  2090   2091#  2115   2116#  2138   2139#
                     2158   2159#  2178   2179#  2203   2204#  2226   2227#  2246   2247#  2266   2267#  2291
                     2292#  2314   2315#  2334   2335#  2354   2355#  2381   2382#  2413   2414#  2431   2432#
                     2440   2441#  2457   2458#  2479   2480#  2512   2513#  2536   2537#  2559   2560#  2582
                     2583#  2619   2620#  2641   2642#  2686   2687#  2708   2709#  2727   2728#  2770   2771#
```

	2788	2789#	2828	2829#	2845	2846#	2875	2876#	2878	2879#	2882	2884#	2890
	2893	2894#	2896	2897#	2913	2914#	2915	2916#	2919	2920#	2922	2923#	2942
	2943#	2959	2960#	3015	3016#	3033	3034#	3054	3055#	3082	3083#	3097	3098#
	3129	3130#	3145	3146#	3147#	3150	3151#	3168	3169#	3170	3171#	3173	3174#
	3178	3180#	3194	3195#	3197	3199#	3202	3204	3205#	3207	3209#	3332	3333#
	3336	3337#	3341	3343#	3374	3375#	3465	3466#	3469	3470#	3475	3477#	3520
	3521#	3540	3541#	3576	3577#	3580	3581#	3586	3588#	3639	3640#	3641#	3644
	3645#	3650	3651#	3661	3662#	3684	3685#	3711	3712#	3737	3738#	3739#	3742
	3743#	3748	3749#	3764	3765#	3784	3785#	3873	3874#	3881	3882#	3921	3922#
	3953	3954#	3957	3958#	3977	3978#	4016	4017#	4024	4025#	4071#	4105	4106#
	4109	4110#	4113	4115#	4123	4124#	4155	4156#	4178#	4201	4202#	4203#	4206
	4207#	4227#	4241	4242#	4243#	4246	4247#	4249	4250#	4251#	4254	4255#	4288#
	4326#	4335	4336#	4338	4339#	4341	4342#	4345	4347#	4364	4366#		
$TEMP = 000300	1145#	1146#	1152#	1153#	1155#	1156#	1157#	1158#	1160#	1161#	1162#	1164#	1165#
	1167#	1168#	1170#	1171#	1173#	1174#	1175#	1187#	1188#	1190#	1191#	1192#	1193#
	1194#	1195#	1196#	1197#	1200#	1201#	1203#	1204#	1214#	1217#	1218#	1219#	1221#
	1222#	1223	1227#	1228#	1230#	1231#	1232#	1250#	1251#	1259#	1264#	1265#	1266#
	1267#	1275#	1280#	1281#	1283#	1284#	1292#	1297#	1298#	1300#	1301#	1311#	1326#
	1327#	13__#	1340#	1341#	1342#	1343#	1351#	1356#	1357#	1359#	1360#	1368#	1373#
	1374#	1376#	1377#	1387#	1408#	1409#	1417#	1422#	1423#	1424#	1425#	1433#	1438#
	1439#	1441#	1442#	1450#	1455#	1456#	1458#	1459#	1469#	1490#	1491#	1492#	1493#
	1501#	1506#	1507#	1509#	1510#	1517#	1522#	1523#	1524#	1525#	1533#	1551#	1552#
	1554#	1558#	1559#	1567#	1572#	1573#	1574#	1575#	1583#	1588#	1589#	1591#	1592#
	1600#	1605#	1606#	1608#	1609#	1619#	1636#	1637#	1645#	1650#	1651#	1652#	1653#
	1661#	1666#	1667#	1669#	1670#	1678#	1682#	1683#	1686#	1687#	1697#	1714#	1715#
	1723#	1728#	1729#	1730#	1731#	1739#	1744#	1745#	1747#	1748#	1756#	1761#	1762#
	1764#	1765#	1775#	1792#	1793#	1801#	1806#	1807#	1808#	1809#	1817#	1822#	1823#
	1825#	1826#	1834#	1839#	1840#	1842#	1843#	1853#	1877#	1878#	1890#	1910#	1911#
	1924#	1944#	1945#	1959#	1980#	1981#	1983#	1988#	1989#	2006#	2030#	2031#	2033#
	2042#	2043#	2046#	2047#	2057#	2062#	2063#	2066#	2067#	2077#	2082#	2083#	2086#
	2087#	2097#	2118#	2119#	2121#	2130#	2131#	2134#	2135#	2145#	2150#	2151#	2154#
	2155#	2165#	2170#	2171#	2174#	2175#	2185#	2206#	2207#	2209#	2218#	2219#	2222#
	2223#	2233#	2238#	2239#	2242#	2243#	2253#	2258#	2259#	2262#	2263#	2273#	2294#
	2295#	2297#	2306#	2307#	2310#	2311#	2321#	2326#	2327#	2330#	2331#	2341#	2346#
	2347#	2350#	2351#	2361#	2384#	2385#	2387#	2397#	2398#	2400#	2401#	2409#	2410#
	2418#	2424#	2425#	2428#	2429#	2436#	2445#	2450#	2451#	2454#	2455#	2462#	2482#
	2483#	2485#	2495#	2496#	2499#	2500#	2508#	2509#	2517#	2522#	2523#	2527#	2528#
	2541#	2546#	2547#	2550#	2551#	2564#	2585#	2586#	2588#	2600#	2601#	2602#	2603#
	2606#	2607#	2610#	2611#	2624#	2629#	2630#	2632#	2633#	2646#	2663#	2664#	2671#
	2672#	2691#	2695#	2696#	2702#	2703#	2713#	2732#	2747#	2748#	2750#	2751#	2754#
	2755#	2775#	2780#	2781#	2793#	2806#	2807#	2808#	2809#	2814#	2815#	2833#	2841#
	2842#	2850#	2863#	2864#	2865#	2866#	2867#	2868#	2873#	2874#	2880#	2881#	2882#
	2883#	2885#	2886#	2887#	2889#	2892	2902#	2903#	2905#	2929#	2930#	2932#	2934#
	2936#	2937#	2948#	2955#	2956#	2964#	2983#	2984#	2990#	2991#	3000#	3001#	3010#
	3011#	3022#	3023#	3025#	3030#	3031#	3042#	3043#	3045#	3049#	3050#	3063#	3067#
	3068#	3072#	3073#	3090#	3091#	3093#	3103#	3131#	3132#	3134#	3137#	3138#	3139#
	3140#	3141#	3142#	3148#	3151#	3152#	3153#	3155#	3156#	3158#	3159#	3161#	3162#
	3164#	3165#	3166#	3167#	3176#	3177#	3178#	3179#	3182#	3183#	3184#	3185#	3186#
	3189#	3190#	3197#	3198#	3207#	3208#	3216#	3218#	3221#	3222#	3223#	3224#	3226#
	3227#	3229#	3233#	3234#	3235#	3291#	3292#	3295#	3296#	3298#	3299#	3301#	3302#
	3303#	3304#	3305#	3306#	3307#	3308#	3310#	3311#	3320#	3321#	3341#	3342#	3348#
	3350#	3354#	3355#	3357#	3358#	3360#	3361#	3379#	3387#	3388#	3390#	3391#	3392#
	3415#	3416#	3417#	3418#	3419#	3420#	3422#	3423#	3424#	3425#	3427#	3428#	3430#
	3431#	3440#	3441#	3454#	3455#	3475#	3476#	3481#	3483#	3486#	3487#	3496#	3497#
	3498#	3499#	3502#	3503#	3505#	3506#	3525#	3532#	3533#	3537#	3538#	3543#	3544#
	3546#	3549#	3550#	3552#	3553#	3560#	3561#	3563#	3564#	3565#	3566#	3586#	3587#

```
                    3592#  3594#  3596#  3597#  3600#  3601#  3604#  3605#  3607#  3608#  3609#  3622#  3623#
                    3642#  3645#  3653#  3654   3658#  3659#  3664#  3665   3668#  3669#  3676#  3678#  3679#
                    3680#  3690#  3692#  3693#  3697#  3698#  3714#  3715#  3717#  3720#  3721#  3733#  3734#
                    3740#  3743#  3751#  3752   3756#  3757#  3767#  3768   3772#  3773#  3776#  3778#  3779#
                    3780#  3790#  3793#  3794#  3834#  3835#  3837#  3838#  3839#  3840#  3843#  3844#  3845#
                    3846#  3849#  3850#  3852#  3853#  3855#  3856#  3861#  3862#  3865#  3866#  3868#  3869#
                    3875#  3876   3884#  3885#  3886#  3887#  3891#  3910#  3912#  3914#  3915#  3917#  3918#
                    3924#  3925#  3926#  3943#  3945#  3947#  3949#  3960#  3961#  3962#  3963#  3964#  3967#
                    3968#  3969#  3973#  3974#  3980#  3981#  3983#  4008#  4009#  4011#  4012#  4014#  4015#
                    4019#  4020   4029#  4095#  4096#  4097#  4098#  4099#  4100#  4111#  4112#  4113#  4114#
                    4116#  4117#  4118#  4126#  4127#  4128#  4131#  4132   4134   4144#  4145#  4146#  4147#
                    4159#  4170#  4192#  4193#  4194#  4198#  4204#  4207#  4208#  4209#  4210#  4211#  4212#
                    4213#  4215#  4216#  4220#  4238#  4239#  4244#  4247#  4252#  4255#  4256#  4257#  4259#
                    4260#  4265#  4281#  4282#  4321#  4343#  4344#  4345#  4346#  4352#  4353#  4354#  4356#
                    4357#  4358#  4359#  4360#  4361#  4362#  4363#  4364#  4365#  4367#  4368#  4369#  4370#
                    4371#  4372#  4373#  4374#  4375#  4377#  4378   4381#  4382#  4383#  4384#  4392#
$TESTN  001200      1012#  1185*  1246*  1322*  1399*  1487*  1546*  1633*  1711*  1789*  1870*  1904*  1938*
                    1972*  2023*  2111*  2199*  2287*  2377*  2475*  2578*  2660*  2743*  2802*  2860*  2980*
                    3126*  3288*  3411*  3619*  3708*  3810*  4005*  4293   4863*
$TIMES  001160       998#  1097*  1184*  1245*  1321*  1398*  1486*  1545*  1551*  1632*  1710*  1788*  1869*
                    1903*  1937*  1971*  1980*  2022*  2030*  2110*  2118*  2198*  2206*  2286*  2294*  2376*
                    2384*  2474*  2482*  2577*  2585*  2659*  2742*  2801*  2859*  2902*  2929*  2979*  3022*
                    3042*  3090*  3125*  3131*  3287*  3410*  3543*  3618*  3707*  3714*  3809*  4004*  4045*
                    4409*  4851*  4858   4861*  4871
$TKB    001146       991#  4561   4572   4589   4643   4649
$TKS    001144       990#  4561   4570   4586   4610*  4641   4647
$TN   = 000042         3#   680   1177   1184#  1185   1239   1245#  1246   1317   1321#  1322   1394   1398#
                    1399   1476   1486#  1487   1540   1545#  1546   1552   1626   1632#  1633   1704   1710#
                    1711   1782   1788#  1789   1865   1869#  1870   1899   1903#  1904   1933   1937#  1938
                    1967   1971#  1972   1981   2017   2022#  2023   2031   2105   2110#  2111   2119   2193
                    2198#  2199   2207   2281   2286#  2287   2295   2369   2376#  2377   2385   2469   2474#
                    2475   2483   2567   2573   2577#  2578   2586   2653   2659#  2660   2736   2742#  2743
                    2797   2801#  2802   2853   2859#  2860   2903   2930   2967   2975   2979#  2980   3023
                    3043   3061   3091   3106   3118   3125#  3126   3132   3237   3278   3287#  3288   3403
                    3410#  3411   3544   3613   3618#  3619   3703   3707#  3708   3715   3804   3809#  3810
                    3895   3998   4004#  4005   4033   4041   4045#
$TPB    001152       993#  4547*  4558
$TPFLG  001157       997#  4496   4558
$TPS    001150       992#  4545   4558
$TRAP   015064      1092   5024#
$TRAP2  015106      5035#  5046
$TRP  = 000012      5039#  5048#  5049#  5050#  5051#  5052#  5053   5054#  5055   5056#  5057#  5058#
$TRPAD  015120      5029   5046#
$TSKD = 000241      1199#  1222   1255#  1259   1271#  1275   1288#  1292   1307#  1311   1331#  1335   1347#
                    1351   1364#  1368   1383#  1387   1413#  1417   1429#  1433   1446#  1450   1465#  1469
                    1497#  1501   1513#  1517   1529#  1533   1550#  1554   1563#  1567   1579#  1583   1596#
                    1600   1615#  1619   1641#  1645   1657#  1661   1674#  1678   1693#  1697   1719#  1723
                    1735#  1739   1752#  1756   1771#  1775   1797#  1801   1813#  1817   1830#  1834   1849#
                    1853   1881#  1890   1915#  1924   1948#  1959   1978#  1983   1993#  2006   2028#  2033
                    2051#  2057   2071#  2077   2091#  2097   2116#  2121   2139#  2145   2159#  2165   2179#
                    2185   2204#  2209   2227#  2233   2247#  2253   2267#  2273   2292#  2297   2315#  2321
                    2335#  2341   2355#  2361   2382#  2387   2414#  2418   2432#  2436   2441#  2445   2458#
                    2462   2480#  2485   2513#  2517   2537#  2541   2560#  2564   2583#  2588   2620#  2624
                    2642#  2646   2687#  2691   2709#  2713   2728#  2732   2771#  2775   2789#  2793   2829#
                    2833   2846#  2850   2876#  2889   2897#  2905   2914#  2936   2943#  2948   2960#  2964
                    3016#  3025   3034#  3045   3055#  3063   3083#  3093   3098#  3103   3130#  3134   3146#
```

```
                    3148    3151#   3227    3333#   3350    3375#   3379    3466#   3483    3521#   3525    3541#   3546
                    3577#   3594    3640#   3642    3645#   3693    3712#   3717    3738#   3740    3743#   3794    3874#
                    3875    3882#   3891    3922#   3926    3954#   3969    3978#   3983    4017#   4019    4025#   4029
                    4106#   4146    4156#   4159    4202#   4204    4207#   4213    4242#   4244    4247#   4260    4336#
                    4377
$TSK1 = 000245      1210#   1214    2879#   2883    2884#   2887    2916#   2936    3151#   3226    3337#   3342    3343#
                    3348    3470#   3476    3477#   3481    3581    3587    3588#   3592    3645#   3692    3743#   3793
                    3958#   3964    4106#   4131    4146    4207    4212    4247#   4259    4339#   4365    4366#   4375
$TSK2 = 000244      2920#   2934    3169#   3189    3195#   3198    3199#   3218    3651#   3653    3662#   3664    3685#
                    3690    3749#   3751    3765#   3767    3785#   3790    4110#   4114    4115#   4118    4124#   4128
                    4250#   4252    4255#   4257    4342#   4346    4347#   4354
$TSK3 = 000233      2923#   2932    3171#   3189    3205#   3208    3209#   3216    4255#   4256
$TSK4 = 000142      3174#   3179    3180#   3186
$TSTM   001004       957#
$TSTNM  001102       970#   4408#   4772    4808    4813    4840    4862#   4863    4868    4872
$TTYIN  013420      4668    4669    4686    4690#
$TYPBN= ****** U    5052
$TYPOS  014412      4884#   5051
$TYPE   012426      4496#   4728    5039    5047
$TYPEC  012640      4526    4533    4540    4545#   4546    4612
$TYPEX  012706      4551    4553    4556#
$TYPOC  014662      4969#   5048
$TYPON  014676      4968    4971#   5050
$TYPOS  014636      4964#   5049
$UNIT   001206      1015#
$UNITM  001010       959#
$USWR   001220      1022#   1548    1976    2026    2114    2202    2290    2380    2478    2581    3128    3229    3539
                    3710    4194
$VECT1  001244      1047#   4362
$VECT2  001246      1048#
$XTSTR  014144      4827#
$YESNO= 000001      1163#   1164#   1166#   1167#   1169#   1170#   1172#   1173#   1217#   1218#   1220#   1221#   1229#
                    1230#   1266#   1267#   1283#   1284#   1300#   1301#   1342#   1343#   1359#   1360#   1376#   1377#
                    1424#   1425#   1441#   1442#   1458#   1459#   1492#   1493#   1509#   1510#   1524#   1525#   1574#
                    1575#   1591#   1592#   1608#   1609#   1652#   1653#   1669#   1670#   1686#   1687#   1730#   1731#
                    1747#   1748#   1764#   1765#   1808#   1809#   1825#   1826#   1842#   1843#   2046#   2047#   2066#
                    2067#   2086#   2087#   2134#   2135#   2154#   2155#   2174#   2175#   2222#   2223#   2242#   2243#
                    2262#   2263#   2310#   2311#   2330#   2331#   2350#   2351#   2400#   2401#   2428#   2429#   2454#
                    2455#   2499#   2500#   2527#   2528#   2550#   2551#   2600#   2601#   2610#   2611#   2632#   2633#
                    2747#   2748#   2806#   2807#   2863#   2864#   2885#   2886#   3141#   3142#   3147#   3148#   3182#
                    3183#   3184#   3185#   3230#   3233#   3298#   3299#   3310#   3311#   3320#   3321#   3360#   3361#
                    3389#   3390#   3427#   3428#   3430#   3431#   3454#   3455#   3486#   3487#   3498#   3499#   3505#
                    3506#   3532#   3533#   3552#   3553#   3560#   3561#   3563#   3564#   3565#   3566#   3596#   3597#
                    3606#   3607#   3622#   3623#   3641#   3642#   3677#   3678#   3679#   3680#   3697#   3698#   3720#
                    3721#   3739#   3740#   3777#   3778#   3779#   3780#   3861#   3862#   3865#   3866#   3868#   3869#
                    3911#   3912#   3917#   3918#   3924#   3925#   3944#   3945#   3948#   3949#   3967#   3968#   3973#
                    3974#   3980#   3981#   4008#   4009#   4011#   4012#   4144#   4145#   4195#   4198#   4203#   4204#
                    4208#   4209#   4210#   4211#   4215#   4216#   4243#   4244#   4251#   4252#   4281#   4282#   4369#
                    4370#   4371#   4372#   4373#   4374#   4383#   4384#
$$ARGC= 000000      4071#   4178#   4227#   4288#   4326#
$$BYTE= 000403      1208#   1253#   1269#   1286#   1305#   1329#   1345#   1362#   1381#   1411#   1427#   1444#   1463#
                    1495#   1511#   1527#   1548#   1561#   1577#   1594#   1613#   1639#   1655#   1672#   1691#   1717#
                    1733#   1750#   1769#   1795#   1811#   1828#   1847#   1879#   1913#   1946#   1976#   1991#   2026#
                    2049#   2069#   2089#   2114#   2137#   2157#   2177#   2202#   2225#   2245#   2265#   2290#   2313#
                    2333#   2353#   2380#   2412#   2430#   2439#   2456#   2478#   2511#   2535#   2558#   2581#   2618#
                    2640#   2707#   2726#   2844#   2877#   2895#   2914#   2918#   2921#   2941#   2958#   3014#   3032#
```

$$DST = 000067
$$FLAG= 000001

3053# 3081# 3096# 3128# 3169# 3172# 3193# 3201# 3203# 3331# 3335# 3373# 3464#
3468# 3519# 3539# 3575# 3579# 3683# 3710# 3783# 3880# 3920# 3952# 3956# 3976#
4023# 4108# 4122# 4154# 4337# 4340#
3230# 4195#
1208# 1210 1214# 1253# 1255 1259# 1269# 1271 1275# 1286# 1288 1292# 1305#
1307 1311# 1329# 1331 1335# 1345# 1347 1351# 1362# 1364 1368# 1381# 1383
1387# 1411# 1413 1417# 1427# 1429 1433# 1444# 1446 1450# 1463# 1465 1469#
1495# 1497 1501# 1511# 1513 1517# 1527# 1529 1533# 1548# 1550 1554# 1561#
1563 1567# 1577# 1579 1583# 1594# 1596 1600# 1613# 1615 1619# 1639# 1641
1645# 1655# 1657 1661# 1672# 1674 1678# 1691# 1693 1697# 1717# 1719 1723#
1733# 1735 1739# 1750# 1752 1756# 1769# 1771 1775# 1795# 1797 1801# 1811#
1813 1817# 1828# 1830 1834# 1847# 1849 1853# 1879# 1881 1890# 1913# 1915
1924# 1946# 1948 1959# 1976# 1978 1983# 1991# 1993 2006# 2026# 2028 2033#
2049# 2051 2057# 2069# 2071 2077# 2089# 2091 2097# 2114# 2116 2121# 2137#
2139 2145# 2157# 2159 2165# 2177# 2179 2185# 2202# 2204 2209# 2225# 2227
2233# 2245# 2247 2253# 2265# 2267 2273# 2290# 2292 2297# 2313# 2315 2321#
2333# 2335 2341# 2353# 2355 2361# 2380# 2382 2387# 2412# 2414 2418# 2430#
2432 2436# 2439# 2441 2445# 2456# 2458 2462# 2478# 2480 2485# 2511# 2513
2517# 2535# 2537 2541# 2558# 2560 2564# 2581# 2583 2588# 2618# 2620 2624#
2640# 2642 2646# 2686# 2691# 2707# 2709 2713# 2727# 2732# 2770# 2775# 2787#
2789 2793# 2828# 2833# 2844# 2846 2860# 2877# 2879 2887# 2895# 2897 2905#
2913# 2914# 2916 2918# 2920 2921# 2923 2932# 2934# 2941# 2943 2948# 2958#
2960 2964# 3014# 3016 3025# 3032# 3034 3045# 3053# 3055 3063# 3081# 3083
3093# 3096# 3098 3103# 3128# 3130 3134# 3168# 3169# 3171 3172# 3174 3186#
3193# 3195 3201# 3203# 3205 3216# 3218# 3331# 3333 3335# 3337 3348# 3350#
3373# 3375 3379# 3464# 3466 3468# 3470 3481# 3483# 3519# 3521 3525# 3539#
3541 3546# 3575# 3577 3579# 3581 3592# 3594# 3683# 3685 3690# 3710# 3712
3717# 3783# 3785 3790# 3880# 3882 3891# 3920# 3922 3926# 3952# 3954 3956#
3958 3964# 3969# 3976# 3978 3983# 4023# 4025 4029# 4108# 4110 4118# 4122#
4124 4128# 4154 4156 4159# 4337# 4339 4340# 4342 4354# 4375#

$$FROM= 000000
1148# 2403# 2502# 2530# 2553# 2613# 2635# 2676# 2719# 2760# 2819# 2993# 3003#
3075# 3324# 3368# 3445# 3457# 3513# 3568# 3632# 3729# 3759# 3814# 4139# 4348#

$$GET4= 000000
4424#

$$LOC = 012100
1209# 1210 1223# 1224 1254# 1255 1270# 1271 1287# 1288 1306# 1307 1330#
1331 1346# 1347 1363# 1364 1382# 1383 1412# 1413 1428# 1429 1445# 1446
1464# 1465 1496# 1497 1512# 1513 1528# 1529 1549# 1550 1562# 1563 1578#
1579 1595# 1596 1614# 1615 1640# 1641 1656# 1657 1673# 1674 1692# 1693
1718# 1719 1734# 1735 1751# 1752 1770# 1771 1796# 1797 1812# 1813 1829#
1830 1848# 1849 1880# 1881 1914# 1915 1947# 1948 1977# 1978 1992# 1993
2027# 2028 2050# 2051 2070# 2071 2090# 2091 2115# 2116 2138# 2139 2158#
2159 2178# 2179 2203# 2204 2226# 2227 2246# 2247 2266# 2267 2291# 2292
2314# 2315 2334# 2335 2354# 2355 2381# 2382 2413# 2414 2431# 2432 2440#
2441 2457# 2458 2479# 2480 2512# 2513 2536# 2537 2559# 2560 2582# 2583
2619# 2620 2641# 2642 2686# 2687 2708# 2709 2727# 2728 2770# 2771 2788#
2789 2828# 2829 2845# 2846 2878# 2879 2892# 2893 2896# 2897 2915# 2916
2919# 2920 2922# 2923 2942# 2943 2959# 2960 3015# 3016 3033# 3034 3054#
3055 3082# 3083 3097# 3098 3129# 3130 3170# 3171 3173# 3174 3194# 3195
3202# 3203 3204# 3205 3332# 3333 3336# 3337 3374# 3375 3465# 3466 3469#
3470 3520# 3521 3540# 3541 3576# 3577 3580# 3581 3654# 3655 3665# 3666
3684# 3685 3711# 3712 3752# 3753 3768# 3769 3784# 3785 3876# 3877 3881#
3882 3921# 3922 3953# 3954 3957# 3958 3977# 3978 4020# 4021 4024# 4025
4109# 4110 4123# 4124 4155# 4156 4338# 4339 4341# 4342 4378# 4379

$$LOCN= 000000
4071# 4173 4178# 4222 4227# 4267 4288# 4323 4326# 4394

$$RETU= 000000
4071# 4157# 4161# 4171 4178# 4221 4227# 4266 4288# 4322 4326# 4393

$$RTN1= 000237
4071# 4157 4170 4178# 4217 4220 4227# 4265 4289# 4321 4326# 4385 4392

$$RTN2= 000240
4071# 4162 4172 4178# 4221 4227# 4266 4288# 4322 4326# 4393

```
$$SRC = 000027          3230#  4195#
$$TO  = 000000          1148#  1149    2403#  2406    2502#  2505    2530#  2533    2553#  2556    2613#  2616    2635#
                        2638   2676#   2682   2719#   2725   2760#   2766   2819#   2825   2993#   2996   3003#   3006
                        3075#  3078    3324#  3327    3368#  3371    3445#  3451    3457#  3460    3513#  3516    3568#
                        3571   3632#   3633   3634    3729#  3730    3731   3759#   3762   3814#   3815   3816    4139#
                        4142   4348#   4349
$OFILL  015061          4965*  4969*   4979   5014#
$1      002102          1198#  1223
$1U     002430          1346   1351#
$100    005300          2559   2564#
$101    005340          2582   2588#
$102    005426          2619   2624#
$103    005470          2641   2646#
$104    005554          2686   2691#
$105    005602          2708   2713#
$106    005636          2727   2732#
$107    005730          2770   2775#
$11     002456          1363   1368#
$110    005752          2788   2793#
$111    006044          2828   2833#
$112    006062          2845   2850#
$113    006124          2875#  2892
$114    006144          2878   2883#
$115    006150          2882   2887#
$116    006170          2890   2893#
$117    006212          2896   2905#
$12     002506          1382   1387#
$120    006212          2913#  2936
$121    006256          2915   2937#
$122    006254          2919   2934#
$123    006254          2922   2932#
$124    006270          2942   2948#
$125    006306          2959   2964#
$126    006424          3015   3025#
$127    006452          3033   3045#
$13     002556          1412   1417#
$130    006474          3054   3063#
$131    006544          3082   3093#
$132    006556          3097   3103#
$133    006662          3129   3134#
$134    006712          3145   3148#
$135    006710          3146#  3226
$136    007060          3150   3227#
$137    006750          3168#  3189
$14     002604          1428   1433#
$140    007004          3170   3190#
$141    006772          3173   3179#
$142    007002          3178   3186#
$143    007016          3194   3198#
$144    007042          3197   3218#
$145    007040          3202   3204    3208#
$146    007042          3207   3216#
$147    007332          3332   3350#
$15     002632          1445   1450#
$150    007330          3336   3342#
$151    007332          3341   3348#
```

$152	007410	3374	3379#	
$153	007640	3465	3483#	
$154	007636	3469	3476#	
$155	007640	3475	3481#	
$156	007732	3520	3525#	
$157	007770	3540	3546#	
$16	002662	1464	1469#	
$160	010076	3576	3594#	
$161	010074	3580	3587#	
$162	010076	3586	3592#	
$163	010214	3639	3642#	
$164	010212	3640#	3692	
$165	010270	3644	3693#	
$166	010222	3650#	3654	
$167	010236	3661#	3665	
$17	002726	1496	1501#	
$170	010266	3684	3690#	
$171	010334	3711	3717#	
$172	010400	3737	3740#	
$173	010376	3738#	3793	
$174	010470	3742	3794#	
$175	010406	3748#	3752	
$176	010436	3764#	3768	
$177	010466	3784	3790#	
$2	002126	1209	1214#	
$20	002754	1512	1517#	
$200	010610	3873#	3876	
$201	010636	3881	3891#	
$202	010702	3921	3926#	
$203	010760	3953	3969#	
$204	010754	3957	3964#	
$205	010776	3977	3983#	
$206	011044	4016#	4020	
$207	011064	4024	4029#	
$21	003002	1528	1533#	
$210	011410	4157	4170#	
$211	011412	4162	4172#	
$212	011260	4105#	4146	
$213	011364	4132	4134	4147#
$214	011300	4109	4114#	
$215	011306	4113	4118#	
$216	011324	4123	4128#	
$217	011376	4155	4159#	
$22	003040	1549	1554#	
$220	011512	4217	4220#	
$221	011512	4221#		
$222	011456	4201	4204#	
$223	011452	4202#	4212	
$224	011502	4206	4213#	
$225	011572	4265#		
$226	011572	4266#		
$227	011540	4241	4244#	
$23	003060	1562	1567#	
$230	011534	4242#	4259	
$231	011564	4246	4260#	
$232	011552	4249	4252#	

```
$233     011550            4250#   4256
$234     011562            4254    4257#
$235     011764            4321#
$236     011764            4322#
$237     012124            4385    4392#
$24      003106            1578    1583#
$240     012124            4393#
$241     011766            4335#   4378
$242     012052            4338    4365#
$243     012012            4341    4346#
$244     012020            4345    4354#
$245     012072            4364    4375#
$25      003134            1595    1600#
$26      003164            1614    1619#
$27      003222            1640    1645#
$3       002240            1254    1259#
$30      003250            1656    1661#
$31      003276            1673    1678#
$32      003326            1692    1697#
$33      003364            1718    1723#
$34      003412            1734    1739#
$35      003440            1751    1756#
$36      003470            1770    1775#
$37      003526            1796    1801#
$4       002266            1270    1275#
$40      003554            1812    1817#
$40CAT=  ****** U          4782    4824
$41      003602            1829    1834#
$42      003632            1848    1853#
$43      003672            1880    1890#
$44      003732            1914    1924#
$45      003772            1947    1959#
$46      004030            1977    1983#
$47      004052            1992    2006#
$5       002314            1287    1292#
$50      004110            2027    2033#
$51      004136            2050    2057#
$52      004164            2070    2077#
$53      004212            2090    2097#
$54      004250            2115    2121#
$55      004276            2138    2145#
$56      004324            2158    2165#
$57      004352            2178    2185#
$6       002344            1306    1311#
$60      004410            2203    2209#
$61      004436            2226    2233#
$62      004464            2246    2253#
$63      004512            2266    2273#
$64      004550            2291    2297#
$65      004576            2314    2321#
$66      004624            2334    2341#
$67      004652            2354    2361#
$7       002402            1330    1335#
$70      004710            2381    2387#
$71      004770            2413    2418#
$72      005016            2431    2436#
```

```
$73     005030          2440    2445#
$74     005056          2457    2462#
$75     005114          2479    2485#
$76     005174          2512    2517#
$77     005236          2536    2541#
.     = 015144           919#    923#    932     933#    935#    937#    938#    944     945#    947#    949#    967#    1004
                        1077#   1085    1100    1101    1209    1210    1223    1224    1254    1255    1270    1271    1287
                        1288    1306    1307    1330    1331    1346    1347    1363    1364    1382    1383    1412    1413
                        1428    1429    1445    1446    1464    1465    1496    1497    1512    1513    1528    1529    1549
                        1550    1562    1563    1578    1579    1595    1596    1614    1615    1640    1641    1656    1657
                        1673    1674    1692      93    1718    1719    1734    1735    1751    1752    1770    1771    1796
                        1797    1812    1813    1829    1830    1848    1849    1860    1881    1914    1915    1947    1948
                        1977    1978    1992    1993    2027    2028    2050    2051    2070    2071    2090    2091    2115
                        2116    2138    2139    2158    2159    2178    2179    2203    2204    2226    2227    2246    2247
                        2266    2267    2291    2292    2314    2315    2334    2335    2354    2355    2381    2382    2413
                        2414    2431    2432    2440    2441    2457    2458    2479    2480    2512    2513    2536    2537
                        2559    2560    2582    2583    2619    2620    2641    2642    2686    2687    2708    2709    2727
                        2728    2770    2771    2788    2789    2828    2829    2845    2846    2878    2879    2892    2893
                        2896    2897    2915    2916    2919    2920    2922    2923    2942    2943    2959    2960    3015
                        3016    3033    3034    3054    3055    3082    3083    3097    3098    3129    3130    3170    3171
                        3173    3174    3194    3195    3202    3203    3204    3205    3270#   3332    3333    3336    3337
                        3374    3375    3465    3466    3469    3470    3520    3521    3540    3541    3576    3577    3580
                        3581    3654    3655    3665    3666    3684    3685    3711    3712    3752    3753    3768    3769
                        3784    3785    3876    3877    3881    3882    3921    3922    3953    3954    3957    3958    3977
                        3978    4020    4021    4024    4025    4049#   4109    4110    4123    4124    4155    4156    4292#
                        4305#   4311#   4338    4339    4341    4342    4378    4379    4432    4436    4452    4474    4558
                        4561    4690#   4691    4697    4749#   4808    4871    4872    4938#
.$ASTA= ****** U        4701    4704
.$X   = 001000           944#    949
```

```
BEGIN       1#
BGNHRD      3#
BGNHW       3#
BGNINI      3#
BGNMOD      3#  4067
BGNMSG      3#
BGNSFT      3#
BGNSRV      3#  3905  3938  4271
BGNSUB      3#  1199  1249  1263  1279  1296  1325  1339  1355  1372  1407  1421  1437  1454  1489
          1505  1521  1557  1571  1587  1604  1635  1649  1665  1681  1713  1727  1743  1760  1791
          1805  1821  1838  1876  1909  1943  1987  2041  2061  2081  2129  2149  2169  2217  2237
          2257  2305  2325  2345  2396  2423  2449  2494  2521  2545  2605  2628  2662  2694  2749
          2779  2807  2982  3029  3048  3066  3306  3353  3421  3495  3536
BGNSW       3#
BRESET      3#  1175  1302  1378  1460  1610  1688  1766  1844  1887  1921  1956  2003  2783  3380
          3527  3857  4030
CALL        1#  1147  2403  2502  2530  2553  2613  2635  2675  2718  2759  2818  2953  3003  3075
          3324  3368  3444  3457  3513  3568  3631  3728  3759  3813  4139  4347
CASE        1#
CKLOOP      3#  1854  1892  1926  1960  2008
CLRVEC      3#  1224  3384  3601
COMMEN      1#   802#
DECR        1#
DECRU       1#
DEFAUL      1#
DEVREG      3#
DEVTYP      3#
DISPAT      3#
ELSE        1#  2881  3177  3196  3206  3340  3474  3585  4112  4344  4363
END         1#
ENDCLN      3#
ENDCOM      1#   802#
ENDDEC      1#
ENDDO       1#  2935  3188
ENDHRD      3#
ENDHW       3#
ENDIF       1#  1213  1258  1274  1291  1310  1334  1350  1367  1386  1416  1432  1449  1468  1500
          1516  1532  1553  1566  1582  1599  1618  1644  1660  1677  1696  1722  1738  1755  1774
          1800  1816  1833  1852  1889  1923  1958  1982  2005  2032  2056  2076  2096  2120  2144
          2164  2184  2208  2232  2252  2272  2296  2320  2340  2360  2386  2417  2435  2444  2461
          2484  2516  2540  2563  2587  2623  2645  2690  2712  2731  2774  2792  2832  2849  2886
          2904  2931  2933  2947  2963  3024  3044  3062  3092  3102  3133  3185  3215  3217  3347
          3349  3378  3480  3482  3524  3545  3591  3593  3689  3716  3789  3890  3925  3963  3968
          3982  4028  4117  4127  4158  4353  4374
ENDINC      1#  3225  3691  3792  4211  4255  4258
ENDINI      3#
ENDLOO      1#  4145
ENDMOD      3#
ENDMSG      3#
ENDRTN      1#  4169  4219  4264  4320  4391
ENDSEL      1#
ENDSFT      3#
ENDSRV      3#  3932  3990  4282
ENDSUB      3#  1215  1260  1276  1293  1312  1336  1352  1369  1388  1418  1434  1451  1470  1502
          1518  1534  1568  1584  1601  1620  1646  1662  1679  1698  1724  1740  1757  1776  1802
          1818  1835  1855  1893  1927  1961  2009  2058  2078  2098  2146  2166  2186  2234  2254
```

```
                2274    2322    2342    2362    2420    2446    2463    2518    2542    2565    2625    2647    2692    2733    2777
                2794    2834    3026    3046    3064    3104    3351    3382    3487    3533    3597
ENDSW              3#
ENDTST             3#   1235    1313    1389    1471    1535    1621    1699    1777    1856    1894    1928    1962    2010    2099
                2187    2275    2363    2464    2568    2648    2734    2795    2851    2971    3114    3273    3395    3611    3698
                3798    3993    4039
EQUALS             3#
ERRDF              3#   1211
ERRHRD             3#   1256    1272    1289    1308    1332    1348    1365    1384    1414    1430    1447    1466    1498    1514
                1530    1564    1580    1597    1616    1642    1658    1675    1694    1720    1736    1753    1772    1798    1814
                1831    1850    1884    1918    1949    1995    2052    2072    2092    2140    2160    2180    2228    2248    2268
                2316    2336    2356    2415    2433    2442    2459    2514    2538    2561    2621    2643    2688    2710    2729
                2772    2790    2830    2847    2899    2925    2944    2961    3017    3036    3057    3084    3099    3213    3338
                3345    3376    3471    3478    3522    3582    3589    3686    3786    3888    4026
ERROR            696#   1212    1257    1273    1290    1309    1333    1349    1366    1385    1415    1431    1448    1467    1499
                1515    1531    1565    1581    1598    1617    1643    1659    1676    1695    1721    1737    1754    1773    1799
                1815    1832    1851    1885    1919    1950    1996    2053    2073    2093    2141    2161    2161    2229    2249
                2269    2317    2337    2357    2416    2434    2443    2460    2515    2539    2562    2622    2644    2689    2711
                2730    2773    2791    2831    2848    2900    2926    2945    2962    3018    3037    3058    3085    3100    3214
                3339    3346    3377    3472    3479    3523    3583    3590    3687    3787    3889    4027
ESCAPE             1#    802#
EXIF               1#
EXIFB              1#   4130
EXIT               3#   1550    1979    2029    2117    2205    2293    2383    2481    2566    2584    2901    2928    2966    3021
                3041    3060    3089    3105    3130    3236    3542    3713    3894    4032
GETPRI             1#    802#
GETSWR             1#    802#   1130#
GPHARD             3#
GPRMA              3#
GPRMD              3#
GPRML              3#
HEADER             3#
IF                 1#   1207    1252    1268    1285    1304    1328    1344    1361    1380    1410    1426    1443    1462    1494
             1510    1526    1547    1560    1576    1593    1612    1638    1654    1671    1690    1716    1732    1749    1768
             1794    1810    1827    1846    1878    1912    1945    1975    1990    2025    2048    2068    2088    2113    2136
             2156    2176    2201    2224    2244    2264    2289    2312    2332    2352    2379    2411    2429    2438    2455
             2477    2510    2534    2557    2580    2617    2639    2706    2786    2843    2876    2894    2917    2920    2940
             2957    3013    3031    3052    3080    3095    3127    3171    3192    3200    3330    3334    3372    3463    3467
             3518    3538    3574    3578    3682    3709    3782    3879    3919    3951    3955    3975    4107    4336    4339
IFB                1#   4022    4121    4153
IFCOND             1#
IF.ERR             1#   2685    2726    2769    2827
IF.NO.             1#
INCR               1#   3143    3637    3735    4199    4247
INCRU              1#   4239
INLINE             1#
LASTAD             3#
LEAVE              1#
LET                1#   1144    1151    1154    1156    1159    1161    1164    1167    1170    1173    1186    1190    1191    1192
             1193    1194    1195    1200    1202    1216    1218    1227    1228    1230    1231    1250    1264    1265    1280
             1282    1297    1299    1326    1340    1341    1356    1358    1373    1375    1408    1422    1423    1438    1440
             1455    1457    1490    1491    1506    1508    1522    1523    1551    1558    1572    1573    1588    1590    1605
             1607    1636    1650    1651    1666    1668    1682    1685    1714    1726    1729    1744    1746    1761    1763
             1792    1806    1807    1822    1824    1839    1841    1877    1910    1944    1980    1988    2030    2042    2045
             2062    2065    2082    2085    2118    2130    2133    2150    2153    2170    2173    2206    2218    2221    2238
             2241    2258    2261    2294    2306    2309    2326    2329    2346    2349    2384    2397    2399    2408    2424
```

```
                2427    2450    2453    2482    2495    2498    2507    2522    2526    2546    2549    2585    2599    2601    2606
                2609    2629    2631    2663    2670    2695    2701    2746    2750    2753    2780    2805    2808    2813    2840
                2862    2864    2866    2872    2879    2884    2902    2929    2954    2983    2989    2999    3009    3022    3030
                3042    3049    3067    3071    3090    3131    3136    3138    3140    3151    3154    3157    3160    3163    3165
                3175    3181    3183    3220    3222    3228    3233    3290    3294    3297    3301    3302    3303    3304    3305
                3307    3309    3319    3354    3356    3359    3387    3388    3390    3391    3415    3416    3417    3418    3419
                3422    3423    3426    3429    3439    3453    3485    3496    3497    3501    3504    3531    3537    3543    3548
                3551    3559    3562    3564    3595    3599    3604    3605    3607    3608    3621    3657    3667    3675    3678
                3696    3714    3719    3732    3755    3771    3775    3778    3833    3836    3838    3842    3844    3848    3851
                3854    3860    3864    3867    3883    3885    3909    3913    3916    3923    3942    3946    3959    3961    3966
                3972    3979    4007    4010    4013    4094    4096    4098    4110    4115    4125    4143    4191    4193    4207
                4209    4214    4237    4280    4342    4351    4355    4357    4359    4361    4366    4368    4370    4372    4380
                4382
LOCAL           1#
LOOP            1#      4104
MSG         1177#   1179    1239#   1241    1476#   1478    1540#   1542    1626#   1628    1704#   1706    1782#   1784    2017#
            2019    2105#   2107    2193#   2195    2281#   2283    2369#   2371    2469#   2471    2653#   2655    2736#   2738
            2853#   2855    3118#   3120    3278#   3280    3403#   3405    3613#   3615    3804#   3806    3998#   4000
MULT            1#    802#
NEWTST          1#    802#  1177    1239    1317    1394    1476    1540    1626    1704    1782    1865    1899    1933    1967
            2017    2105    2193    2281    2369    2469    2573    2653    2736    2797    2853    2975    3118    3278    3403
            3613    3703    3804    3998    4041
NOLOCA          1#
POINTE          3#
POP             1#    802#  1232    3392    3609    4261    4461    4462    4743    4744    4925
PRINTB          3#
PUSH            1#    802#  1225    3385    3602    4234    4442    4448    4704    4706    4727    4884
REPEAT          1#   1197   2874    3649    3660    3747    3763    3872    4015    4334
REPORT          1#      3#    802#
RETURN          1#   4156   4160    4216    4384
ROUTIN          1#   4069   4176    4225    4286    4324
SAVR14          1#
SCOPE         697#   1183   1244    1320    1397    1485    1544    1631    1709    1787    1868    1902    1936    1970    2021
            2109    2197    2285    2375    2473    2576    2658    2741    2800    2858    2978    3124    3286    3409    3617
            3706    3808    4003    4044    4407
SELECT          1#
SETPRI          1#    802#  1401    2390    2488    2591    3313    3362    3433    3507    3555    3626    3723    3828
SETTRA       5039#   5048   5049    5050    5051    5053    5055    5056    5057
SETUP           1#    802#  1079
SETVEC          3#   1189   3300    3414
SKIP            1#    802#  1552    1981    2031    2119    2207    2295    2385    2483    2567    2586    2903    2930    2967
            3023    3043    3061    3091    3106    3132    3237    3544    3715    3895    4033
SLASH           1#    802#
SPACE         802#
STARS           1#    802#   818     820     838     840     858     860     880     882     901     916     930     941     943
             950    963    1004    1007    1177    1192    1236    1238    1239    1243    1316    1317    1319    1393    1394
            1396    1475    1476    1484    1539    1540    1543    1625    1626    1630    1703    1704    1708    1781    1782
            1786    1861    1864    1865    1867    1898    1899    1901    1932    1933    1935    1966    1967    1969    2012
            2016    2017    2020    2104    2105    2108    2192    2193    2196    2280    2281    2284    2368    2369    2374
            2468    2469    2472    2572    2573    2575    2652    2653    2657    2735    2736    2740    2796    2797    2799
            2852    2853    2857    2974    2975    2977    3117    3118    3123    3277    3278    3285    3402    3403    3408
            3612    3613    3616    3702    3703    3705    3803    3804    3807    3997    3998    4002    4041    4043    4068
            4089    4175    4189    4224    4233    4270    4277    4288    4326    4333    4399    4438    4454    4481    4560
            4563    4631    4660    4699    4756    4810    4874    4941    5018
STRUCT          1#      3#
SWPSU           1#    802#  1102#
```

```
TRMTRP   5039#
TYPBIN      1#    802#
TYPDEC      1#    802#  4062  4300  4419
TYPNAM      1#    802#  1123
TYPNUM      1#    802#
TYPOCS      1#    802#
TYPOCT      1#    802#  4050  4056  4293  4306  4312  4318  4581
TYPTXT      1#    802#  4046  4052  4058  4289  4295  4302  4308  4314
UNTIL       1#   1221  2888  3652  3663  3750  3766  3874  4018  4376
UNTILB      1#
WAITMS      3#
WAITUS      3#   2402  2501  2529  2552  2612  2634  2992  3002  3074  3323  3367  3456  3512  3567
         3758    4138
WHILE       1#   2912  3167
WHILEB      1#
$ADDON      1#   1198  1199  1208  1210  1253  1255  1269  1271  1286  1288  1305  1307  1329  1331
         1345    1347  1362  1364  1381  1383  1411  1413  1427  1429  1444  1446  1463  1465  1495
         1497    1511  1513  1527  1529  1548  1550  1561  1563  1577  1579  1594  1596  1613  1615
         1639    1641  1655  1657  1672  1674  1691  1693  1717  1719  1733  1735  1750  1752  1769
         1771    1795  1797  1811  1813  1828  1830  1847  1849  1879  1881  1913  1915  1946  1948
         1976    1978  1991  1993  2026  2028  2049  2051  2069  2071  2089  2091  2114  2116  2137
         2139    2157  2159  2177  2179  2202  2204  2225  2227  2245  2247  2265  2267  2290  2292
         2313    2315  2333  2335  2353  2355  2380  2382  2412  2414  2430  2432  2439  2441  2456
         2458    2478  2480  2511  2513  2535  2537  2558  2560  2581  2583  2618  2620  2640  2642
         2686    2687  2707  2709  2727  2728  2770  2771  2787  2789  2828  2829  2844  2846  2875
         2876    2877  2879  2884  2894  2895  2897  2913  2914  2916  2918  2920  2921  2923  2936
         2941    2943  2958  2960  3014  3016  3032  3034  3053  3055  3081  3083  3096  3098  3128
         3130    3144  3146  3147  3151  3168  3169  3171  3172  3174  3180  3189  3193  3195  3199
         3201    3205  3209  3331  3333  3335  3337  3343  3373  3375  3464  3466  3468  3470  3477
         3519    3521  3539  3541  3575  3577  3579  3581  3588  3638  3640  3641  3645  3650  3651
         3661    3662  3683  3685  3710  3712  3736  3738  3739  3743  3748  3749  3764  3765  3783
         3785    3873  3874  3880  3882  3920  3922  3952  3954  3956  3958  3976  3978  4016  4017
         4023    4025  4071  4105  4106  4108  4110  4115  4122  4124  4146  4154  4156  4178  4200
         4202    4203  4207  4227  4240  4242  4243  4247  4248  4250  4251  4255  4288  4326  4335
         4336    4337  4339  4340  4342  4347  4366
$AND        1#   3201
$BRANC      1#   1209  1223  1254  1270  1287  1306  1330  1346  1363  1382  1412  1428  1445  1464
         1496    1512  1528  1549  1562  1578  1595  1614  1640  1656  1673  1692  1718  1734  1751
         1770    1796  1812  1829  1848  1880  1914  1947  1977  1992  2027  2050  2070  2090  2115
         2138    2158  2178  2203  2226  2246  2266  2291  2314  2334  2354  2381  2413  2431  2440
         2457    2479  2512  2536  2559  2582  2619  2641  2686  2708  2727  2770  2788  2828  2845
         2878    2882  2890  2892  2896  2915  2919  2922  2936  2942  2959  3015  3033  3054  3082
         3097    3129  3145  3150  3170  3173  3178  3189  3194  3197  3202  3204  3207  3226  3332
         3336    3341  3374  3465  3469  3475  3520  3540  3576  3580  3586  3639  3644  3654  3665
         3684    3692  3711  3737  3742  3752  3768  3784  3793  3876  3881  3921  3953  3957  3977
         4020    4024  4109  4113  4123  4132  4134  4146  4155  4157  4162  4201  4206  4212  4217
         4241    4246  4249  4254  4256  4259  4338  4341  4345  4364  4378  4385
$BRCOD      1#   2889  3149  3643  3741  4131  4133  4205  4245  4253
$CALL       1#   1148  2403  2502  2530  2553  2613  2635  2676  2719  2760  2819  2993  3003  3075
         3324    3368  3445  3457  3513  3568  3632  3729  3759  3814  4139  4348
$CHECK      1#   1208  1253  1269  1286  1305  1329  1345  1362  1381  1411  1427  1444  1463  1495
         1511    1527  1548  1561  1577  1594  1613  1639  1655  1672  1691  1717  1733  1750  1769
         1795    1811  1828  1847  1879  1913  1946  1976  1991  2026  2049  2069  2089  2114  2137
         2157    2177  2202  2225  2245  2265  2290  2313  2333  2353  2380  2412  2430  2439  2456
         2478    2511  2535  2558  2581  2618  2640  2707  2787  2844  2877  2895  2914  2918  2921
         2941    2958  3014  3032  3053  3081  3096  3128  3169  3172  3193  3201  3331  3335  3373
```

```
                3464    3468    3519    3539    3575    3579    3683    3710    3783    3880    3920    3952    3956    3976    4023
                4108    4122    4154    4337    4340
$CHK1             1#    1145    1152    1155    1157    1160    1174    1187    1190    1191    1192    1193    1194    1196    1200
                1203    1227    1230    1231    1250    1264    1280    1297    1326    1340    1356    1373    1408    1422    1438
                1455    1490    1506    1522    1551    1558    1572    1588    1605    1636    1650    1666    1682    1714    1728
                1744    1761    1792    1806    1822    1839    1877    1910    1944    1980    1988    2030    2042    2062    2082
                2118    2130    2150    2170    2206    2218    2238    2258    2294    2306    2326    2346    2384    2397    2409
                2424    2450    2482    2495    2508    2522    2546    2585    2602    2606    2629    2663    2671    2695    2702
                2750    2754    2780    2808    2814    2841    2865    2867    2873    2880    2902    2929    2955    2983    2990
                3000    3010    3022    3030    3042    3049    3067    3072    3090    3131    3137    3139    3144    3147    3152
                3155    3158    3161    3164    3166    3176    3221    3223    3234    3291    3295    3301    3302    3303    3304
                3305    3307    3354    3357    3387    3390    3391    3415    3416    3417    3418    3419    3422    3424    3440
                3496    3502    3537    3543    3549    3600    3604    3607    3608    3638    3641    3658    3668    3714    3733
                3736    3739    3756    3772    3834    3837    3839    3843    3845    3849    3852    3855    3884    3886    3914
                3960    3962    4014    4095    4097    4099    4111    4116    4126    4192    4200    4203    4215    4238    4240
                4243    4248    4251    4343    4352    4356    4358    4360    4362    4367    4381
$CKOP2            1#    1162    1165    1168    1171    1217    1219    1228    1266    1283    1300    1342    1359    1376    1424
                1441    1458    1492    1509    1524    1574    1591    1608    1652    1669    1686    1730    1747    1764    1808
                1825    1842    2046    2066    2086    2134    2154    2174    2222    2242    2262    2310    2330    2350    2400
                2428    2454    2499    2527    2550    2600    2610    2632    2747    2806    2863    2885    3141    3182    3184
                3229    3298    3310    3320    3360    3388    3427    3430    3454    3486    3498    3505    3532    3552    3560
                3563    3565    3596    3605    3622    3676    3679    3697    3720    3776    3779    3861    3865    3868    3910
                3917    3924    3943    3947    3967    3973    3980    4008    4011    4144    4194    4208    4210    4281    4369
                4371    4373    4383
$CKR6             1#    3230    4195
$CMND             1#    1208    1253    1269    1286    1305    1329    1345    1362    1381    1411    1427    1444    1463    1495
                1511    1527    1548    1561    1577    1594    1613    1639    1655    1672    1691    1717    1733    1750    1769
                1795    1811    1828    1847    1879    1913    1946    1976    1991    2026    2049    2069    2089    2114    2137
                2157    2177    2202    2225    2245    2265    2290    2313    2333    2353    2380    2412    2430    2439    2456
                2478    2511    2535    2558    2581    2618    2640    2707    2787    2844    2877    2895    2914    2918    2921
                2941    2958    3014    3032    3053    3081    3096    3128    3169    3172    3193    3201    3203    3331    3335
                3373    3464    3468    3519    3539    3575    3579    3683    3710    3783    3880    3920    3952    3956    3976
                4023    4108    4122    4154    4337    4340
$COMPA            1#    1208    1253    1269    1286    1305    1329    1345    1362    1381    1411    1427    1444    1463    1495
                1511    1527    1548    1561    1577    1594    1613    1639    1655    1672    1691    1717    1733    1750    1769
                1795    1811    1828    1847    1879    1913    1946    1976    1991    2026    2049    2069    2089    2114    2137
                2157    2177    2202    2225    2245    2265    2290    2313    2333    2353    2380    2412    2430    2439    2456
                2478    2511    2535    2558    2581    2618    2640    2686    2707    2727    2770    2787    2828    2844    2877
                2895    2914    2918    2921    2941    2958    3014    3032    3053    3081    3096    3128    3144    3169    3172
                3193    3201    3331    3335    3373    3464    3468    3519    3539    3575    3579    3638    3683    3710    3736
                3783    3880    3920    3952    3956    3976    4023    4108    4122    4154    4200    4240    4248    4337    4340
$COUNT            1#    1148    2403    2502    2530    2553    2613    2635    2676    2719    2760    2819    2993    3003    3075
                3324    3368    3445    3457    3513    3568    3632    3729    3759    3814    4139    4348
$DO               1#    2914    3169
$ELSE             1#
$ERRMS            1#
$EXIFA            1#
$EXIF0            1#
$EXIF2            1#    4131
$EXIF3            1#
$GENBR            1#    1209    1223    1254    1270    1287    1306    1330    1346    1363    1382    1412    1428    1445    1464
                1496    1512    1528    1549    1562    1578    1595    1614    1640    1656    1673    1692    1718    1734    1751
                1770    1796    1812    1829    1848    1880    1914    1947    1977    1992    2027    2050    2070    2090    2115
                2138    2158    2178    2203    2226    2246    2266    2291    2314    2334    2354    2381    2413    2431    2440
                2457    2479    2512    2536    2559    2582    2619    2641    2686    2708    2727    2770    2788    2828    2845
                2878    2882    2890    2892    2896    2915    2919    2922    2936    2942    2959    3015    3033    3054    3082
```

```
            3097  3129  3145  3150  3170  3173  3178  3189  3194  3197  3202  3204  3207  3226  3332
            3336  3341  3374  3465  3469  3475  3520  3540  3576  3580  3586  3639  3644  3654  3665
            3684  3692  3711  3737  3742  3752  3768  3784  3793  3876  3881  3921  3953  3957  3977
            4020  4024  4109  4113  4123  4132  4134  4146  4155  4157  4162  4201  4206  4212  4217
            4241  4246  4249  4254  4256  4259  4338  4341  4345  4364  4378  4385
$GENTA        1#  1198  1214  1259  1275  1292  1311  1335  1351  1368  1387  1417  1433  1450  1469
            1501  1517  1533  1554  1567  1583  1600  1619  1645  1661  1678  1697  1723  1739  1756
            1775  1801  1817  1834  1853  1890  1924  1959  1983  2006  2033  2057  2077  2097  2121
            2145  2165  2185  2209  2233  2253  2273  2297  2321  2341  2351  2387  2418  2436  2445
            2462  2485  2517  2541  2564  2588  2624  2646  2691  2713  2732  2775  2793  2833  2850
            2875  2883  2887  2893  2905  2913  2932  2934  2937  2948  2964  3025  3045  3063  3093
            3103  3134  3146  3148  3168  3179  3186  3190  3198  3208  3216  3218  3227  3342  3348
            3350  3379  3476  3481  3483  3525  3546  3587  3592  3594  3640  3642  3650  3661  3690
            3693  3717  3738  3740  3748  3764  3790  3794  3873  3891  3926  3964  3969  3983  4016
            4029  4105  4114  4118  4128  4147  4159  4170  4172  4202  4204  4213  4220  4221  4242
            4244  4250  4252  4257  4260  4265  4266  4321  4322  4335  4346  4354  4365  4375  4352
            4393
$IF           1#  1208  1253  1269  1286  1305  1329  1345  1362  1381  1411  1427  1444  1463  1495
            1511  1527  1548  1561  1577  1594  1613  1639  1655  1672  1691  1717  1733  1750  1769
            1795  1811  1828  1847  1879  1913  1946  1976  1991  2026  2049  2069  2089  2114  2137
            2157  2177  2202  2225  2245  2265  2290  2313  2333  2353  2380  2412  2430  2439  2456
            2478  2511  2535  2558  2581  2618  2640  2707  2787  2844  2877  2895  2918  2921  2941
            2958  3014  3032  3053  3081  3096  3128  3172  3193  3201  3331  3335  3373  3464  3468
            3519  3539  3575  3579  3683  3710  3783  3880  3920  3952  3956  3976  4023  4108  4122
            4154  4337  4340
$IFCOD        1#  1208  1222  1253  1269  1286  1305  1329  1345  1362  1381  1411  1427  1444  1463
            1495  1511  1527  1548  1561  1577  1594  1613  1639  1655  1672  1691  1717  1733  1750
            1769  1795  1811  1828  1847  1879  1913  1946  1976  1991  2026  2049  2069  2089  2114
            2137  2157  2177  2202  2225  2245  2265  2290  2313  2333  2353  2380  2412  2430  2439
            2456  2478  2511  2535  2558  2581  2618  2640  2707  2787  2844  2877  2891  2895  2914
            2918  2921  2941  2958  3014  3032  3053  3081  3096  3128  3169  3172  3193  3201  3203
            3331  3335  3373  3464  3468  3519  3539  3575  3579  3653  3664  3683  3710  3751  3767
            3783  3875  3880  3920  3952  3956  3976  4019  4023  4108  4122  4154  4337  4340  4377
$IFCON        1#  2686  2727  2770  2828
$IFOPR        1#  1209  1223  1254  1270  1287  1306  1330  1346  1363  1382  1412  1428  1445  1464
            1496  1512  1528  1549  1562  1578  1595  1614  1640  1656  1673  1692  1718  1734  1751
            1770  1796  1812  1829  1848  1880  1914  1947  1977  1992  2027  2050  2070  2090  2115
            2138  2158  2178  2203  2226  2246  2266  2291  2314  2334  2354  2381  2413  2431  2440
            2457  2479  2512  2536  2559  2582  2619  2641  2686  2708  2727  2770  2788  2828  2845
            2878  2892  2896  2915  2919  2922  2942  2959  3015  3033  3054  3082  3097  3129  3170
            3173  3194  3202  3204  3332  3336  3374  3465  3469  3520  3540  3576  3580  3654  3665
            3684  3711  3752  3768  3784  3876  3881  3921  3953  3957  3977  4020  4024  4109  4123
            4155  4338  4341  4378
$LET          1#  1145  1152  1155  1157  1160  1162  1165  1168  1171  1174  1187  1190  1191  1192
            1193  1194  1196  1200  1203  1217  1219  1227  1228  1230  1231  1250  1264  1266  1280
            1283  1297  1300  1326  1340  1342  1356  1359  1373  1376  1408  1422  1424  1438  1441
            1455  1458  1490  1492  1506  1509  1522  1524  1551  1558  1572  1574  1588  1591  1605
            1608  1636  1650  1652  1666  1669  1682  1686  1714  1728  1730  1744  1747  1761  1764
            1792  1806  1808  1822  1825  1839  1942  1877  1910  1944  1980  1988  2030  2042  2046
            2062  2066  2082  2086  2118  2130  2134  2150  2154  2170  2174  2206  2218  2222  2238
            2242  2258  2262  2294  2306  2310  2326  2330  2346  2350  2384  2397  2400  2409  2424
            2428  2450  2454  2482  2495  2499  2508  2522  2527  2546  2550  2585  2600  2602  2606
            2610  2629  2632  2663  2671  2695  2702  2747  2750  2754  2780  2806  2808  2814  2841
            2863  2865  2867  2873  2880  2885  2902  2929  2955  2983  2990  3000  3010  3022  3030
            3042  3049  3067  3072  3090  3131  3137  3139  3141  3152  3155  3158  3161  3164  3166
            3176  3182  3184  3221  3223  3229  3234  3291  3295  3298  3301  3302  3303  3304  3305
```

```
            3307    3310    3320    3354    3357    3360    3387    3388    3390    3391    3415    3416    3417    3418    3419
            3422    3424    3427    3430    3440    3454    3486    3496    3498    3502    3505    3532    3537    3543    3549
            3552    3560    3563    3565    3596    3600    3604    3605    3607    3608    3622    3658    3668    3676    3679
            3697    3714    3720    3733    3756    3772    3776    3779    3834    3837    3839    3843    3845    3849    3852
            3855    3861    3865    3868    3884    3886    3910    3914    3917    3924    3943    3947    3960    3962    3967
            3973    3980    4008    4011    4014    4095    4097    4099    4111    4116    4126    4144    4192    4194    4208
            4210    4215    4238    4281    4343    4352    4356    4358    4360    4362    4367    4369    4371    4373    4381
            4383
$LPCNT        1#  3144    3638    3736    4200    4240    4248
$OPADD        1#  1163    1166    1169    1172    1217    1220    1229    2885    3147    3182    3184    3298    3389    3606
            3641    3739    3917    3967    3973    4203    4243    4251    4281    4371    4373    4383
$OPAND        1#  3230    4195
$OPCD1        1#  1163    1166    1169    1172    1217    1220    1229    1266    1283    1300    1342    1359    1376    1424
            1441    1458    1492    1509    1524    1574    1591    1608    1652    1669    1686    1730    1747    1764    1808
            1825    1842    2046    2066    2086    2134    2154    2174    2222    2242    2262    2310    2330    2350    2400
            2428    2454    2499    2527    2550    2600    2610    2632    2747    2806    2863    2885    3141    3147    3182
            3184    3230    3298    3310    3320    3360    3389    3427    3430    3454    3486    3498    3505    3532    3552
            3560    3563    3565    3596    3606    3622    3641    3677    3679    3697    3720    3739    3777    3779    3861
            3865    3868    3911    3917    3924    3944    3948    3967    3973    3980    4008    4011    4144    4195    4203
            4208    4210    4215    4243    4251    4281    4369    4371    4373    4383
$OPCD2        1#  4208    4215    4369
$OPCOD        1#  1163    1166    1169    1172    1217    1220    1229    1266    1283    1300    1342    1359    1376    1424
            1441    1458    1492    1509    1524    1574    1591    1608    1652    1669    1686    1730    1747    1764    1808
            1825    1842    2046    2066    2086    2134    2154    2174    2222    2242    2262    2310    2330    2350    2400
            2428    2454    2499    2527    2550    2600    2610    2632    2747    2806    2863    2885    3141    3147    3182
            3184    3230    3298    3310    3320    3360    3389    3427    3430    3454    3486    3498    3505    3532    3552
            3560    3563    3565    3596    3606    3622    3641    3677    3679    3697    3720    3739    3777    3779    3861
            3865    3868    3911    3917    3924    3944    3948    3967    3973    3980    4008    4011    4144    4195    4203
            4208    4210    4215    4243    4251    4281    4369    4371    4373    4383
$OPCOM        1#  4215
$OPDEF        1#  1145    1148    1152    1155    1157    1160    1162    1163    1165    1166    1168    1169    1171    1172
            1174    1187    1190    1191    1192    1193    1194    1196    1200    1203    1208    1209    1217    1219    1220
            1222    1223    1227    1228    1229    1230    1231    1250    1253    1254    1264    1266    1269    1270    1280
            1283    1286    1287    1297    1300    1305    1306    1326    1329    1330    1340    1342    1345    1346    1356
            1359    1362    1363    1373    1376    1381    1382    1408    1411    1412    1422    1424    1427    1428    1438
            1441    1444    1445    1455    1458    1463    1464    1490    1492    1495    1496    1506    1509    1511    1512
            1522    1524    1527    1528    1548    1549    1551    1558    1561    1562    1572    1574    1577    1578    1588
            1591    1594    1595    1605    1608    1613    1614    1636    1639    1640    1650    1652    1655    1656    1666
            1669    1672    1673    1682    1686    1691    1692    1714    1717    1718    1728    1730    1733    1734    1744
            1747    1750    1751    1761    1764    1769    1770    1792    1795    1796    1806    1808    1811    1812    1922
            1825    1828    1829    1839    1842    1847    1848    1877    1879    1880    1910    1913    1914    1944    1946
            1947    1976    1977    1980    1988    1991    1992    2026    2027    2030    2042    2046    2049    2050    2062
            2066    2069    2070    2082    2086    2089    2090    2114    2115    2118    2130    2134    2137    2138    2150
            2154    2157    2158    2170    2174    2177    2178    2202    2203    2206    2218    2222    2225    2226    2238
            2242    2245    2246    2258    2262    2265    2266    2290    2291    2294    2306    2310    2313    2314    2326
            2330    2333    2334    2346    2350    2353    2354    2380    2381    2384    2397    2400    2403    2404    2405
            2406    2409    2412    2413    2424    2428    2430    2431    2439    2440    2450    2454    2456    2457    2478
            2479    2482    2495    2499    2502    2503    2504    2505    2508    2511    2512    2522    2527    2530    2531
            2532    2533    2535    2536    2546    2550    2553    2554    2555    2556    2558    2559    2581    2582    2585
            2600    2602    2606    2610    2613    2614    2615    2616    2618    2619    2629    2632    2635    2636    2637
            2638    2640    2641    2663    2671    2676    2677    2678    2679    2680    2681    2682    2686    2695    2702
            2707    2708    2719    2720    2721    2722    2723    2724    2725    2727    2747    2750    2754    2760    2761
            2762    2763    2764    2765    2766    2770    2780    2787    2788    2806    2808    2814    2819    2820    2821
            2822    2823    2824    2825    2828    2841    2844    2845    2863    2865    2867    2873    2877    2878    2880
            2882    2885    2889    2890    2891    2892    2895    2896    2902    2914    2915    2918    2919    2921    2922
            2929    2936    2941    2942    2955    2958    2959    2983    2990    2993    2994    2995    2996    3000    3003
```

	3004	3005	3006	3010	3014	3015	3022	3030	3032	3033	3042	3049	3053	3054	3067
	3072	3075	3076	3077	3078	3081	3082	3090	3096	3097	3128	3129	3131	3137	3139
	3141	3144	3145	3147	3149	3150	3152	3155	3158	3161	3164	3166	3169	3170	3172
	3173	3176	3178	3182	3184	3189	3193	3194	3197	3201	3202	3203	3204	3207	3221
	3223	3226	3229	3230	3231	3232	3234	3291	3295	3298	3301	3302	3303	3304	3305
	3307	3310	3320	3324	3325	3326	3327	3331	3332	3335	3336	3341	3354	3357	3360
	3368	3369	3370	3371	3373	3374	3387	3388	3389	3390	3391	3415	3416	3417	3418
	3419	3422	3424	3427	3430	3440	3445	3446	3447	3448	3449	3450	3451	3454	3457
	3458	3459	3460	3464	3465	3468	3469	3475	3486	3496	3498	3502	3505	3513	3514
	3515	3516	3519	3520	3532	3537	3539	3540	3543	3549	3552	3560	3563	3565	3568
	3569	3570	3571	3575	3576	3579	3580	3586	3596	3600	3604	3605	3606	3607	3608
	3622	3632	3633	3634	3638	3639	3641	3643	3644	3653	3654	3658	3664	3665	3668
	3676	3677	3679	3683	3684	3692	3697	3710	3711	3714	3720	3729	3730	3731	3733
	3736	3737	3739	3741	3742	3751	3752	3756	3759	3760	3761	3762	3767	3768	3772
	3776	3777	3779	3783	3784	3793	3814	3815	3816	3834	3837	3839	3843	3845	3849
	3852	3855	3861	3865	3868	3875	3876	3880	3881	3884	3886	3910	3911	3914	3917
	3920	3921	3924	3943	3944	3947	3948	3952	3953	3956	3957	3960	3962	3967	3973
	3976	3977	3980	4008	4011	4014	4019	4020	4023	4024	4095	4097	4099	4108	4109
	4111	4113	4116	4122	4123	4126	4131	4132	4133	4134	4139	4140	4141	4142	4144
	4146	4154	4155	4167	4161	4162	4171	4173	4192	4194	4195	4196	4197	4200	4201
	4203	4205	4206	4208	4210	4212	4215	4217	4222	4238	4240	4241	4243	4245	4246
	4248	4249	4251	4253	4254	4256	4259	4267	4281	4323	4337	4338	4340	4341	4343
	4345	4348	4352	4356	4358	4360	4362	4364	4367	4369	4371	4373	4377	4378	4381
	4383	4385	4394												
$OPEQU	1#														
$OPNAN	1#														
$OPNEG	1#														
$OPNOR	1#														
$OPNOT	1#	1283	1359	1441	1492	1524	1591	1669	1747	1825	2046	2086	2134	2174	2222
	2262	2310	2350	2400	2454	2499	2550	2600	2632	3310	3360	3430	3486	3505	3532
	3552	3560	3596	3677	3679	3697	3720	3777	3779	3911	3924	3944	3948	3980	
$OPOR	1#	1266	1300	1342	1376	1424	1458	1509	1574	1608	1652	1686	1730	1764	1808
	1842	2066	2154	2242	2330	2428	2527	2610	2747	2806	2863	3141	3320	3427	3454
	3498	3563	3565	3622	3861	3865	3868	4008	4011	4210					
$OPROT	1#	4369													
$OPRO	1#	1145	1152	1155	1157	1160	1174	1187	1190	1191	1192	1193	1194	1196	1200
	1203	1227	1230	1231	1250	1264	1280	1297	1326	1340	1356	1373	1408	1422	1438
	1455	1490	1506	1522	1551	1558	1572	1588	1605	1636	1650	1666	1682	1714	1728
	1744	1761	1792	1806	1822	1839	1877	1910	1944	1980	1988	2030	2042	2062	2082
	2118	2130	2150	2170	2206	2218	2238	2258	2294	2306	2326	2346	2384	2397	2409
	2424	2450	2482	2495	2508	2522	2546	2585	2602	2606	2629	2663	2671	2695	2702
	2750	2754	2780	2808	2814	2841	2865	2867	2873	2880	2902	2929	2955	2983	2990
	3000	3010	3022	3030	3042	3049	3067	3072	3090	3131	3137	3139	3144	3152	3155
	3158	3161	3164	3166	3176	3221	3223	3234	3291	3295	3301	3302	3303	3304	3305
	3307	3354	3357	3387	3390	3391	3415	3416	3417	3418	3419	3422	3424	3440	3496
	3502	3537	3543	3549	3600	3604	3607	3608	3638	3658	3668	3714	3733	3736	3756
	3772	3834	3837	3839	3843	3845	3849	3852	3855	3884	3886	3914	3960	3962	4014
	4095	4097	4099	4111	4116	4126	4192	4200	4238	4240	4248	4343	4352	4356	4358
	4360	4362	4367	4381											
$OPR1	1#	3147	3641	3739	4203	4215	4243	4251							
$OPR2	1#	1162	1165	1168	1171	1217	1219	1228	1266	1283	1300	1342	1359	1376	1424
	1441	1458	1492	1509	1524	1574	1591	1608	1652	1669	1686	1730	1747	1764	1808
	1825	1842	2046	2066	2086	2134	2154	2174	2222	2242	2262	2310	2330	2350	2400
	2428	2454	2499	2527	2550	2600	2610	2632	2747	2806	2863	2885	3141	3182	3184
	3229	3298	3310	3320	3360	3388	3427	3430	3454	3486	3498	3505	3532	3552	3560
	3563	3565	3596	3605	3622	3676	3679	3697	3720	3776	3779	3861	3865	3868	3910

```
                3917    3924    3943    3947    3967    3973    3980    4008    4011    4144    4194    4208    4210    4281    4369
                4371    4373    4383
$OPSHF             1#   4208
$OPSUB             1#   4144
$OPSWB             1#
$OPXOR             1#
$OR                1#
$PUT               1#   2404    2503    2531    2554    2514    2636    2677    2720    2761    2820    2994    3004    3076    3325
                3369    3446    3458    3514    3569    3760    4140
$STRUC             1#
$SUBON             1#   1214    1222    1259    1275    1292    1311    1335    1351    1368    1387    1417    1433    1450    1469
                1501    1517    1533    1554    1567    1583    1600    1619    1645    1661    1678    1697    1723    1739    1756
                1775    1801    1817    1834    1853    1890    1924    1959    1983    2006    2033    2057    2077    2097    2121
                2145    2165    2185    2209    2233    2253    2273    2297    2321    2341    2361    2387    2418    2436    2445
                2462    2485    2517    2541    2564    2588    2624    2646    2691    2713    2732    2775    2793    2833    2850
                2883    2887    2889    2905    2932    2934    2936    2937    2948    2964    3025    3045    3063    3093    3103
                3134    3148    3151    3179    3186    3189    3190    3198    3208    3216    3218    3226    3227    3342    3348
                3350    3379    3476    3481    3483    3525    3546    3587    3592    3594    3642    3645    3653    3664    3690
                3692    3693    3717    3740    3743    3751    3767    3790    3793    3794    3875    3891    3926    3964    3969
                3983    4019    4029    4114    4118    4128    4146    4147    4159    4170    4204    4207    4212    4213    4220
                4244    4247    4252    4255    4256    4257    4259    4260    4265    4321    4346    4354    4365    4375    4377
                4392
$THEN              1#   1208    1253    1269    1286    1305    1329    1345    1362    1381    1411    1427    1444    1463    1495
                1511    1527    1548    1561    1577    1594    1613    1639    1655    1672    1691    1717    1733    1750    1769
                1795    1811    1828    1847    1879    1913    1946    1976    1991    2026    2049    2069    2089    2114    2137
                2157    2177    2202    2225    2245    2265    2290    2313    2333    2353    2380    2412    2430    2439    2456
                2478    2511    2535    2558    2581    2618    2640    2707    2787    2844    2877    2895    2918    2921    2941
                2958    3014    3032    3053    3081    3096    3128    3172    3193    3203    3331    3335    3373    3464    3468
                3519    3539    3575    3579    3683    3710    3783    3880    3920    3952    3956    3976    4023    4108    4122
                4154    4337    4340
$TILA              1#
$TILO              1#
$UNTL2             1#   2889
$UNTL3             1#
$WHILE             1#   2913    3168
$SCMRE           961#
$SCMTM           961#
$SDEFA             1#
$SENDS             1#
$SERRO             1#
$SESCA             1#    802#
$SGEN              1#   1198    1214    1259    1275    1292    1311    1335    1351    1368    1387    1417    1433    1450    1469
                1501    1517    1533    1554    1567    1583    1600    1619    1645    1661    1678    1697    1723    1739    1756
                1775    1801    1817    1834    1853    1890    1924    1959    1983    2006    2033    2057    2077    2097    2121
                2145    2165    2185    2209    2233    2253    2273    2297    2321    2341    2361    2387    2418    2436    2445
                2462    2485    2517    2541    2564    2588    2624    2646    2691    2713    2732    2775    2793    2833    2850
                2875    2883    2887    2893    2905    2913    2932    2934    2937    2948    2964    3025    3045    3063    3093
                3103    3134    3146    3148    3168    3179    3186    3190    3198    3208    3216    3218    3227    3342    3348
                3350    3379    3476    3481    3483    3525    3546    3587    3592    3594    3640    3642    3650    3661    3690
                3693    3717    3738    3740    3748    3764    3790    3794    3873    3891    3926    3964    3969    3983    4016
                4029    4070    4105    4114    4118    4128    4147    4159    4170    4172    4177    4202    4204    4213    4220
                4221    4226    4242    4244    4250    4252    4257    4260    4265    4266    4287    4321    4322    4325    4335
                4346    4354    4365    4375    4392    4393
$GETS              1#   1214    1222    1259    1275    1292    1311    1335    1351    1368    1387    1417    1433    1450    1469
                1501    1517    1533    1554    1567    1583    1600    1619    1645    1661    1678    1697    1723    1739    1756
                1775    1801    1817    1834    1853    1890    1924    1959    1983    2006    2033    2057    2077    2097    2121
```

```
                2145    2165    2185    2209    2233    2253    2273    2297    2321    2341    2361    2387    2418    2436    2445
                2462    2485    2517    2541    2564    2588    2624    2646    2691    2713    2732    2775    2793    2833    2850
                2882    2883    2887    2889    2905    2932    2934    2936    2937    2948    2964    3025    3045    3063    3093
                3103    3134    3148    3151    3178    3179    3186    3189    3190    3197    3198    3207    3208    3216    3218
                3226    3227    3341    3342    3348    3350    3379    3475    3476    3481    3483    3525    3546    3586    3587
                3592    3594    3642    3645    3653    3664    3690    3692    3693    3717    3740    3743    3751    3767    3790
                3793    3794    3875    3891    3926    3964    3969    3983    4019    4029    4113    4114    4118    4128    4131
                4146    4147    4159    4170    4204    4207    4212    4213    4220    4244    4247    4252    4255    4256    4257
                4259    4260    4265    4321    4345    4346    4354    4364    4365    4375    4377    4392
SSGETT             1#   2882    3178    3197    3207    3341    3475    3586    4113    4131    4345    4364
SSLPCN             1#   3147    3641    3739    4203    4243    4251
SSNEWT             1#    802#   1177    1239    1317    1394    1476    1540    1626    1704    1782    1865    1899    1933    1967
                2017    2105    2193    2281    2369    2469    2573    2653    2736    2797    2853    2975    3118    3278    3403
                3613    3703    3804    3998    4041
SSPOP              1#   1214    1222    1259    1275    1292    1311    1335    1351    1368    1387    1417    1433    1450    1469
                1501    1517    1533    1554    1567    1583    1600    1619    1645    1661    1678    1697    1723    1739    1756
                1775    1801    1817    1834    1853    1890    1924    1959    1983    2006    2033    2057    2077    2097    2121
                2145    2165    2185    2209    2233    2253    2273    2297    2321    2341    2361    2387    2418    2436    2445
                2462    2485    2517    2541    2564    2588    2624    2646    2691    2713    2732    2775    2793    2833    2850
                2883    2887    2889    2905    2932    2934    2936    2937    2948    2964    3025    3045    3063    3093    3103
                3134    3148    3151    3179    3186    3189    3190    3198    3208    3216    3218    3226    3227    3342    3348
                3350    3379    3476    3481    3483    3525    3546    3587    3592    3594    3642    3645    3653    3664    3690
                3692    3693    3717    3740    3743    3751    3767    3790    3793    3794    3875    3891    3926    3964    3969
                3983    4019    4029    4114    4118    4128    4146    4147    4159    4170    4204    4207    4212    4213    4220
                4244    4247    4252    4255    4256    4257    4259    4260    4265    4321    4346    4354    4365    4375    4377
                4392
SSPUSH             1#   1198    1199    1208    1210    1253    1255    1269    1271    1286    1288    1305    1307    1329    1331
                1345    1347    1362    1364    1381    1383    1411    1413    1427    1429    1444    1446    1463    1465    1495
                1497    1511    1513    1527    1529    1548    1550    1561    1563    1577    1579    1594    1596    1613    1615
                1639    1641    1655    1657    1672    1674    1691    1693    1717    1719    1733    1735    1750    1752    1769
                1771    1795    1797    1811    1813    1828    1830    1847    1849    1879    1881    1913    1915    1946    1948
                1976    1978    1991    1993    2026    2028    2049    2051    2069    2071    2089    2091    2114    2116    2137
                2139    2157    2159    2177    2179    2202    2204    2225    2227    2245    2247    2265    2267    2290    2292
                2313    2315    2333    2335    2353    2355    2380    2382    2412    2414    2430    2432    2439    2441    2456
                2452    2478    2480    2511    2513    2535    2537    2558    2560    2581    2583    2618    2620    2640    2642
                2686    2687    2707    2709    2727    2728    2770    2771    2787    2789    2828    2829    2844    2846    2875
                2876    2877    2879    2884    2895    2897    2913    2914    2916    2918    2920    2921    2923    2936    2941
                2943    2958    2960    3014    3016    3032    3034    3053    3055    3081    3083    3096    3098    3128    3130
                3144    3146    3147    3151    3168    3169    3171    3172    3174    3180    3189    3193    3195    3199    3201
                3205    3209    3331    3333    3335    3337    3343    3373    3375    3464    3466    3468    3470    3477    3519
                3521    3539    3541    3575    3577    3579    3581    3588    3638    3640    3641    3645    3650    3651    3661
                3662    3683    3685    3710    3712    3736    3738    3739    3743    3748    3749    3764    3765    3783    3785
                3873    3874    3880    3882    3920    3922    3952    3954    3956    3958    3976    3978    4016    4017    4023
                4025    4071    4105    4106    4108    4110    4115    4122    4124    4146    4154    4156    4178    4200    4202
                4203    4207    4227    4240    4242    4243    4247    4248    4250    4251    4255    4288    4326    4335    4336
                4337    4339    4340    4342    4347    4366
SSSELE             1#
SSSET           5039#   5048    5049    5050    5051    5052    5055    5056    5057
SSSETM          1118#
SSSETS             1#   1198    1199    1208    1210    1253    1255    1269    1271    1286    1288    1305    1307    1329    1331
                1345    1347    1362    1364    1381    1383    1411    1413    1427    1429    1444    1446    1463    1465    1495
                1497    1511    1513    1527    1529    1548    1550    1561    1563    1577    1579    1594    1596    1613    1615
                1639    1641    1655    1657    1672    1674    1691    1693    1717    1719    1733    1735    1750    1752    1769
                1771    1795    1797    1811    1813    1828    1830    1847    1849    1879    1881    1913    1915    1946    1948
                1976    1978    1991    1993    2026    2028    2049    2051    2069    2071    2089    2091    2114    2116    2137
                2139    2157    2159    2177    2179    2202    2204    2225    2227    2245    2247    2265    2267    2290    2292
```

```
            2313    2315    2333    2335    2353    2355    2380    2382    2412    2414    2430    2432    2439    2441    2456
            2458    2478    2480    2511    2513    2535    2537    2558    2560    2581    2583    2618    2620    2640    2642
            2686    2687    2707    2709    2727    2728    2770    2771    2787    2789    2828    2829    2844    2846    2875
            2876    2877    2879    2884    2895    2897    2913    2914    2916    2918    2920    2921    2923    2936    2941
            2943    2958    2960    3014    3016    3032    3034    3053    3055    3081    3083    3096    3098    3128    3130
            3144    3146    3147    3151    3168    3169    3171    3172    3174    3180    3189    3193    3195    3199    3201
            3205    3209    3331    3333    3335    3337    3343    3373    3375    3464    3466    3468    3470    3477    3519
            3521    3539    3541    3575    3577    3579    3581    3588    3638    3640    3641    3645    3650    3651    3661
            3662    3683    3685    3710    3712    3736    3738    3739    3743    3748    3749    3764    3765    3783    3785
            3873    3874    3880    3882    3920    3922    3952    3954    3956    3958    3976    3978    4016    4017    4023
            4025    4071    4105    4106    4108    4110    4115    4122    4124    4146    4154    4156    4178    4200    4202
            4203    4207    4227    4240    4242    4243    4247    4248    4250    4251    4255    4288    4326    4335    4336
            4337    4339    4340    4342    4347    4366
$$SETT       1#
$$SKIP       1#     802#    1552    1981    2031    2119    2207    2295    2385    2483    2567    2586    2903    2930    2967
            3023    3043    3061    3091    3106    3132    3237    3544    3715    3895    4033
.EQUAT       1#       3#     692
.HEADE       1#       3#     670
.KT11        1#
.SETUP       1#       3#    1079
.SWRHI       1#       3#     680
.SWRLO     691#
.$ACTI       1#       3#     928
.$APTB       1#       3#    1005#
.$APTH       1#       3#     939
.$APTY       1#       3#    4697
.$ASTA       1#
.$CATC       1#       3#     917
.$CMTA       1#       3#     961
.$D82D       1#
.$D820       1#
.$DIV        1#
.$EOP        1#       3#    4397
.$ERRO       1#       3#    4754
.$ERRT       1#
.$MULT       1#
.$POWE       1#       3#    4436
.$RAND       1#
.$RDDE       1#
.$RDOC       1#
.$READ       1#       3#    4558
.$R2AZ       1#
.$SAVE       1#
.$SB2D       1#
.$SB20       1#
.$SCOP       1#       3#    4808
.$SIZE       1#
.$SUPR       1#
.$TRAP       1#       3#    5016
.$TYPB       1#
.$TYPD       1#       3#    4872
.$TYPE       1#       3#    4479
.$TYPO       1#       3#    4939
.$40CA       1#
.1170        1#
```

. ABS. 015144 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

.DVDVAA/CRF/NL:TOC/SOL=DVDVAA.SML,SPMAC.SML,DVDVAA.P11
RUN-TIME: 91 97 6 SECONDS
RUN-TIME RATIO: 4641/195=23.7
CORE USED: 43K (85 PAGES)