DLV11-F

DLV11-F OFF LINE TEST

MD-11-DVDVC-A

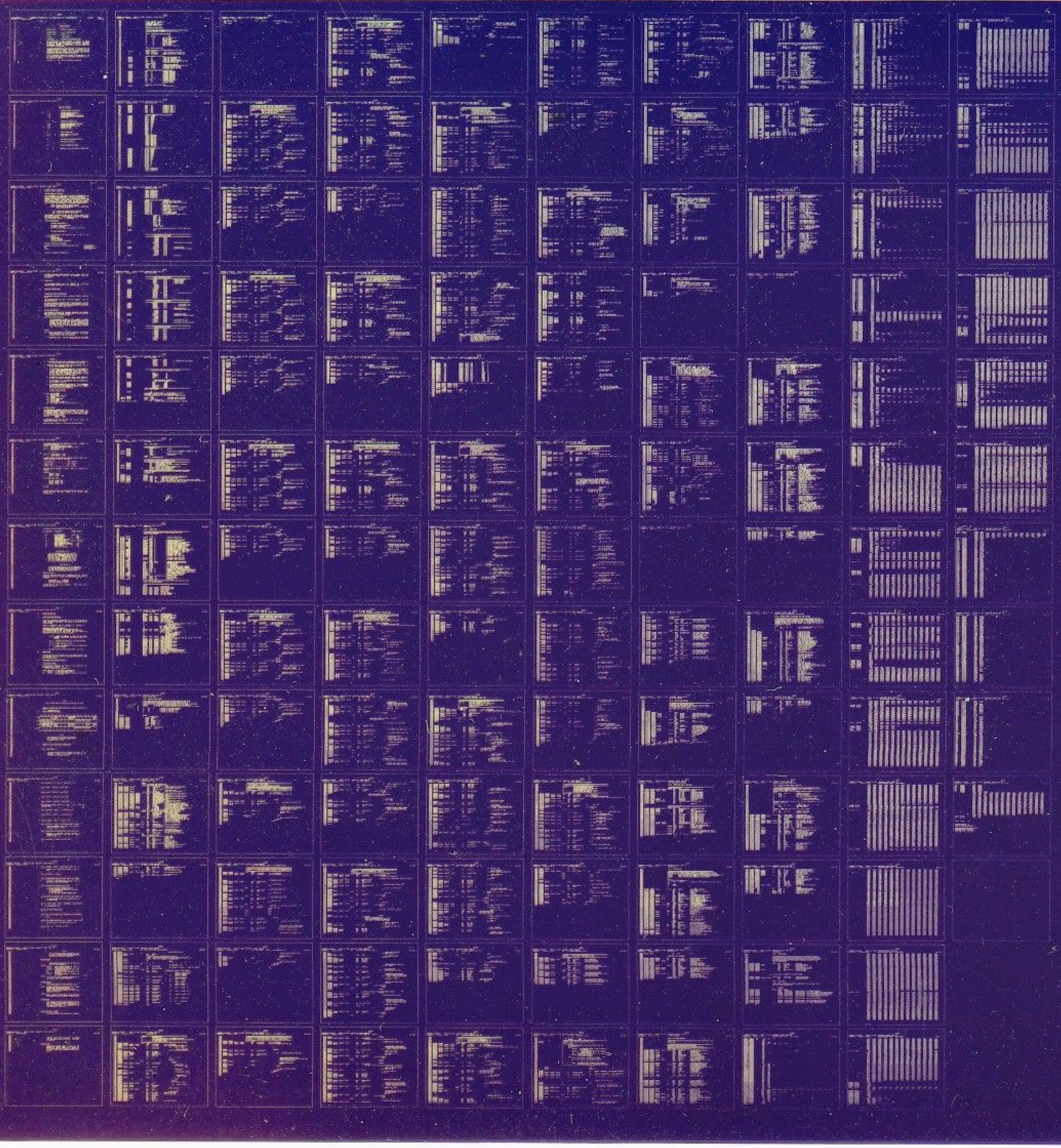
EP-DVDVC-A-DL-A

FICHE 1 OF 1

OCT 1977

digital

MADE IN USA



```
                        .REM    0
```

IDENTIFICATION

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

PRODUCT NAME: DLV11-F OFFLINE TEST

PRODUCT DATE: AUGUST, 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-F  SERIAL  LINE INTERFACE.  THE USER CAN SELECTIVELY
                                        ENABLE AND DISABLE TESTING OF  THE  OPTIONS  BY  ALTERING  THE
                                        CONTENTS  OF  'SUSER'.  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-F SERIAL LINE INTERFACES.  THE DEFAULT ADDRESSES ARE:

                                                177560 -CONSOLE INTERFACE DEVICE ADDRESS
                                                175610 -FIRST SERIAL LINE ADDRESS  OF  15  CONSECUTIVE
                                                        SERIAL LINE DEVICES.


                                                  60 - VECTOR FOR CONSOLE [illegible] INTERFACE.
                                                 300 - VECTOR FOR FIRST OF 15 DEVICES.

                                        THIS PROGRAM IS DESIGNED TO RUN  ON  ANY  PDP-11  WITH  4K  OF
                                        MEMORY AND A DLV11-F (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.

                                                HARDWARE REQUIREMENTS:

                                                ANY PDP-11 FAMILY PROCESSOR
                                                4K MEMORY - MINIMUM
                                               A SPECIAL WRAP CONNECTOR OR EQUIVALENT (OPTIONAL)


                                                SOFTWARE REQUIREMENTS:


                                                THIS DIAGNOSTIC IS DESIGNED TO  RUN  IN  ANY  OF  THE
                                         FOLLOWING [illegible]S:
                                                S[illegible] ALONE
                                                WITH APT MONITOR
                                                WITH ACT MONITOR
                                                WITH XXDP MONITOR (CHAINABLE)

                                1.3     RELATED DOCUMENTS AND STANDARDS.

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


                                1.4     DIAGNOSTIC HIERARCHY PREREQUISITES.
```

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

1.5 ASSUMPTIONS.

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

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-F'S. IT REQUIRES THE ADDRESS OF THE FIRST RCSR (STORED AT 'SBASE') AND ITS INTERRUPT VECTOR (STORED AT 'SVECT1'); AND WILL BE ABLE TO ADDRESS ANY DLV11-F STARTING AT THE SPECIFIED BASE ADDRESS UP TO 16 CONSECUTIVE DEVICES.

EXAMPLES: SBASE: 175610
           SVECT1: 300

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

SBASE AND SVECT1 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#14 -- BASE ADDRESS + (160)
                                   BASE VECTOR  + (160)

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

                         BIT 15 BIT 14  -    -    -       BIT 1  BIT 0
                        ---------------------------------------------
                        ! CON- ! UNIT !                  ! UNIT ! UNIT !
                        ! SOLE !  14  !                  ! #1   !  #0  !
                        ---------------------------------------------

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

                        EXAMPLE:
                                $BASE:  175610
                                $VECTOR:   300
                                $DEVM:  100013

                        THE PROGRAM WILL TEST-

                                UNIT#0  175610  300
                                UNIT#1  175620  310
                                UNIT#3  175640  330
                                CONSOLE 177560   60

                        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 SUSWR 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)	02(8) = 110 BAUD
12	BREAK GENERATION ENABLED	1 = YES
13	WRAP CONNECTOR INSTALLED	0 = NO
14	MAINT JUMPER	0 = NO
15	ERROR BITS ENABLED	0 = NO

NOTE ON BITS <7:11>

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.(TEST #16)

NOTE ON BIT 14

THIS SWITCH WHEN ON WILL ALLOW THE DIAGNOSTIC TO TEST IN MAINTAINCE MODE. IT IS ASSUMED THAT THE MAINTAINCE JUMPER IS INSTALLED ON ALL OF THE DLV11-F MODULES WHEN THIS BIT IS SET.

2.5 EXECUTION TIMES.

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

FIRST PASS-	2 MINUTES
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.
                                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.
```

```
383
384
385                                          NOTE
386
387                                THIS IS TYPED AFTER THE DEVICE HAS COMPLETED ITS PASS.
388
389
390
391                                AFTER ALL DEVICES HAVE BEEN EXERCISED AN END PASS STATEMENT IS
392                                TYPED:
393                                        "ENDPASS↵↵."
394                    5.0         DEVICE INFORMATION TABLES.
395                                ------ ----------- ------
396
397                                  15   14   13   12   11   10    9    8    7    6    5    4    3    2    1    0
398                                ********************************************************************************
399                    RCSR        !    !    !    !    !RCVR!    !    !    !RCVR!RCVR!    !    !    !    !    !RDR !
400                                !    !    !    !    !ACT !    !    !    !DONE!IE  !    !    !    !    !    !ENB !
401                                ********************************************************************************
402                    RBUF        !ERRO!OR  !FR  !P   !    !    !    !    !                RECEIVED DATA BUFFER            !
403                                !R  !ERR !ERR !ERR !    !    !    !    !                                                !
404                                ********************************************************************************
405                    TCSR        ! PROGRAMMABLE BAUD !PBR !    !    !    !XMIT!XMIT!    !    !    !    !MAIN!    !BREA!
406                                !   RATE  SELECT    !ENAB!    !    !    !RDY !IE  !    !    !    !    !T   !    !K   !
407                                ********************************************************************************
408                    TBUF        !    !    !    !    !    !    !    !    !                TRANSMITTER DATA BUFFER         !
409                                !    !    !    !    !    !    !    !    !                                                !
410                                ********************************************************************************
411
412
413
414                                          NOTE
415
416                                BLANK BOXES INDICATE UNUSED AND RESERVED
417                                BIT  POSITIONS.   SEE THE LISTING FOR AN
418                                EXPLANATION OF THE BITS.
419
420
421
422                    6.0         SUMMARY OF TESTS AND SPECIAL SUBROUTINES.
423                                ------- -- ----- --- ------- ------------
424
425
426                                TEST 1 ADDRESSABILITY
427                                ---- -
428
429                                        THIS TEST VERIFIES THAT THE ADDRESS AS PLACED  IN  THE
430                                        HARDWARE   P-TABLE  TO  BE  CORRECT  AND  THE  DLV11-F
431                                        RESPONDS TO THAT ADDRESS SPACE.
432
```

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

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


                                TEST  6 RCVRDONE - RCSR 7 - IS CLEARED BY INIT
                                ---- --


                                TEST 7 RCVRACT - RCSR 11 - 15 CLEARED BY INIT
                                ---- -


                                TEST 10 XMITRDY - TCSR 7 - IS SET BY INIT
                                ---- --

                                TEST 11 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF IS LOADED
                                ---- --
                                        WITH  A  CHARACTER  AND  THAT IT SETS WITHIN A
                                        REASONABLE AMOUNT OF TIME.


                                TEST 12 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 13 RCVRDONE IS CLEARED BY READING RBUF
                                ---- --
```

```
                        TEST 14 RCVRACT - RCSR 11 - SETS WHEN A START  BIT  IS
                        ---- --
                                RECEIVED  AND  CLEARS WHEN RCVRDONE - RCSR 7 -
                                SETS


                        TEST 15 OVERRUN BIT - RBUF 14
                        ---- --


                        TEST 16 PROGRAMMABLE BAUD RATE TEST TEST AT ALL SPEEDS
                        ---- --
                                AVAILABLE  A COMPARISON WILL BE MADE TO SEE IF
                                NEW TIME IS LESS THAN PREVIOUS.


                        TEST 17 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 20 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL
                        ---- --
                                OF  THE RECEIVER SIDE OF THE INTERRUPT LOGIC IN
                                 CHARACTER MODE.


                        TEST  21  TEST  ACTUAL  DATA  TRANSFERED  NON-INTERRUPT
                        ----  --
                                MAINTENANCE BIT SET


                        TEST 22 TEST DATA THROUGH WRAP
                        ---- --


                        TEST 23 FULL DATA TRANSFER WITH INTERRUPTS AND MAINTENANCE
                        ---- --
                                MODE.


                        TEST 24 TEST BREAK GENERATION LOGIC  TRANSMIT  KNOWN  CHAR
                        ---- --
                                WITH BREAK SET AND COMPARE RECEIVED WITH 0.


                        TEST 25 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
```

```
591
592
593                     'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
594                     THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
595                     TO LOOK FOR.
596
597
598             ROUTINE:CYCLE
599             ------------
600                     THIS ROUTINE  CAUSES  ADRS  TO  POINT  TO  THE
601                     ADDRESS  OF  DLV11-F  UNDER  TEST,  ADRS +2 TO
602                     POINT TO THE VECTOR OF THE DLV11-F UNDER TEST.
603                     IT  KEEPS  TRACK OF THE CURRENT DEVICE AND BIT
604                     MASKS.
```

```
                        .
                        .TITLE  MAINDEC-11-DVDVC-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 ***
001100                  STACK=  1100
                        .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                        .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

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

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

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

```
                        PR1=    40              ;;PRIORITY LEVEL 1
                        PR2=    100             ;;PRIORITY LEVEL 2
                        PR3=    140             ;;PRIORITY LEVEL 3
                        PR4=    200             ;;PRIORITY LEVEL 4
                        PR5=    240             ;;PRIORITY LEVEL 5
                        PR6=    300             ;;PRIORITY LEVEL 6
                        PR7=    340             ;;PRIORITY LEVEL 7

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
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
                        ;;*******************************************************************
                                ; UNUSED         BIT15
                                ; UNUSED         BIT14
                                ; UNUSED         BIT13
                                ; UNUSED         BIT12
004000                          RCVRACT=        BIT11           ; RECEIVER ACTIVE INDICATOR
                                ; UNUSED         BIT10
                                ; UNUSED         BIT09
                                ; UNUSED         BIT08
000200                          RCVRDONE=       BIT07           ; RECEIVER DONE
000100                          RCVRIE=         BIT06           ; RECEIVER INTERRUPT ENABLE
                                ; UNUSED         BIT05
                                ; UNUSED         BIT04
                                ; UNUSED         BIT03
                                ; UNUSED         BIT02
                                ; UNUSED         BIT01
000001                          RDRRUN=         BIT00           ; READER RUN
```

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

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

                                                ; BAUDE MUST BE ON THE UPPER
                                                ;  BYTE BOUNDRY OF SUSWR.--4 BITS
                007400                          BAUD    =             7400
                010000                          BRK     =            10000
                020000                          WRAP    =            20000
                040000                          MAINTJUMP =          40000
                100000                          ERRBITS =           100000
                                        ;;**************************************************
                                        .SBTTL  TRAP CATCHER

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

```
                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
000204                                  $SVPC=.                 ;SAVE PC
000046                                  .=46
000046  011552                          $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
000052                                  .=52
000052  000000                          .WORD   0               ;;2)SET LOC.52 TO ZERO
000204                                  .=$SVPC                 ;; RESTORE PC
001000                          .=1000
                                .SBTTL  APT PARAMETER BLOCK

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

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

```
                        .SBTTL  COMMON TAGS

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

                001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102  000            $TSTNM:  .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000            $ERFLG:  .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000         $ICNT:   .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000         $LPADR:  .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000         $LPERR:  .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000         $ERTTL:  .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000            $ITEMB:  .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001            $ERMAX:  .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000         $ERRPC:  .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000         $GDADR:  .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000         $BDADR:  .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000         $GDDAT:  .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000         $BDDAT:  .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000            $AUTOB:  .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000            $INTAG:  .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570         SWR:     .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570         DISPLAY: .WORD   DDISP           ;;ADDRESS OF DISPLAY REGISTER
                       $TKS:    177560                  ;;TTY KBD STATUS
                       $TKB:    177562                  ;;TTY KBD BUFFER
                       $TPS:    177564                  ;;TTY PRINTER STATUS REG. ADDRESS
                       $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
```

```
954  001212  000000                  $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
955  001214                          $ETABLE:                ;;APT ENVIRONMENT TABLE
956  001214      000                 $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
957  001215      000                 $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
958  001216  000000                  $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
959  001220  011110                  $USWR:  .WORD   AUSWR   ;;USER SWITCHES
960  001222  000000                  $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
961                                              BITS 15-11=CPU TYPE
962                                                   11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
963                                                   11/70=06,PDQ=07,Q=10
964                                              BIT 10=REAL TIME CLOCK
965                                              BIT  9=FLOATING POINT PROCESSOR
966                                              BIT  8=MEMORY MANAGEMENT
967  001224      000                 $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
968  001225      000                 $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
969                                              MEM.TYPE BYTE  --  (HIGH BYTE)
970                                                   900 NSEC CORE=001
971                                                   300 NSEC BIPOLAR=002
972                                                   500 NSEC MOS=003
973  001226  000000                  $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
974                                              MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
975  001230      000                 $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
976  001231      000                 $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
977  001232  000000                  $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
978  001234      000                 $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
979  001235      000                 $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
980  001236  000000                  $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
981  001240      000                 $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
982  001241      000                 $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
983  001242  000000                  $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
984  001244  000300                  $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
985  001246  000000                  $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
986  001250  175610                  $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
987  001252  100000                  $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
988  001254                          $ETEND:
989                                  .MEXIT
```

```
                                .SBTTL  ERROR POINTER TABLE

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

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


001254                          $ERRTB:
                                ;;              GLOBAL DATA
001254  175610                      DLADD:  DLADDR
001256  000300                      DLVEC:  300
001260  175610                      RCSR:   DLADDR  +  0
001262  175612                      RBUF:   DLADDR  +  2
001264  175614                      TCSR:   DLADDR  +  4
001266  175615                      TCSRHI: DLADDR  +  5
001270  175616                      TBUF:   DLADDR  +  6
001272  000000                      I:      0
001274  000020                      .BLKW 20    ;FOR R5 STACK
001334  000000                      R5STACK:    .WORD 0
```

```
1016  001336                          START:
1017                                  .SBTTL  INITIALIZE THE COMMON TAGS
1018                                  ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1019  001336  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1020  001342  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1021  001344  022706  001140          CMP     #SWR,R6 ;;DONE?
1022  001350  001374                  BNE     .-6             ;;LOOP BACK IF NO
1023  001352  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1024                                  ;;INITIALIZE A FEW VECTORS
1025  001356  012737  013474  000020  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
1026  001364  012737  000340  000022  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
1027  001372  012737  013274  000030  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1028  001400  012737  000340  000032  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1029  001406  012737  014426  000034  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1030  001414  012737  000340  000036  MOV     #340,@#TRAPVEC+2;LEVEL 7
1031  001422  012737  011606  000024  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
1032  001430  012737  000340  000026  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
1033  001436  016767  010056  010046  MOV     $ENDCT,$ENDPCT  ;;SETUP END-OF-PROGRAM COUNTER
1034  001444  005067  177510          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
1035  001450  005067  177506          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
1036  001454  112767  000001  177433  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
1037  001462  012767  001462  177416  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
1038  001470  012767  001470  177412  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
1039                                  ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1040                                  ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
1041  001476  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1042  001502  012737  001536  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1043  001510  012767  177570  177422  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1044  001516  012767  177570  177416  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1045  001524  022777  177777  177406  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1046  001532  001012                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1047                                                          ;;AND  THE HARDWARE SWR IS NOT = -1
1048  001534  000403                  BR      65$             ;;BRANCH IF NO TIMEOUT
1049  001536  012716  001544  64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1050  001542  000002                  RTI
1051  001544  012767  000176  177366  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1052  001552  012767  000174  177362          MOV     #DISPREG,DISPLAY
1053  001560  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1054
1055  001564  005067  177412          CLR     $PASS           ;;CLEAR PASS COUNT
1056  001570  132767  000200  177417  BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1057  001576  001403                  BEQ     67$             ;;YES,USE NON-APT SWITCH
1058  001600  012767  001216  177332  MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1059  001606                          67$:
1060                                  .SBTTL  TYPE PROGRAM NAME
1061                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1062  001606  005227  177777          INC     #-1             ;;FIRST TIME?
1063  001612  001037                  BNE     68$             ;;BRANCH IF NO
1064  001614  022737  011552  000042  CMP     #$ENDAD,@#42    ;;ACT-11?
1065  001622  001433                  BEQ     68$             ;;BRANCH IF YES
1066  001624  104401  001672          TYPE    ,69$            ;;TYPE ASCIZ STRING
1067                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1068  001630  005737  000042          TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1069  001634  001012                  BNE     70$             ;;BRANCH IF YES
1070  001636  126727  177352  000001  CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1071  001644  001406                  BEQ     70$             ;;BRANCH IF YES
```

```
MAINDEC-11-DVDVC-A      MACY11 27(1006)  08-AUG-77  09:20  PAGE 25
DVDVCA.P11    08-AUG-77 09:16            GET VALUE FOR SOFTWARE SWITCH REGISTER                    SEQ 0024

  1072  001546  026727  177266  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
  1073  001654  001005                          BNE     71$             ;;BRANCH IF NO
  1074  001656  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
  1075  001660  000403                          BR      71$
  1076  001662  112767  000001  177244  70$:     MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
  1077  001670                          71$:
  1078  001670  000410                          BR      68$             ;;GET OVER THE ASCIZ
  1079                                  ;;69$:   .ASCIZ  <CRLF>*MD-11-DVDVC-A*<CRLF>
  1080  001712                          68$:
```

```
                                                        WHILE $DEVM EQ #0 DO
                        $1:
001712                          TST     $DEVM
001712  005767  177334        BNE     $2
001716  001101                  TYPTXT <<CRLF>!I HAVE NO DEVICE TO TEST.!>
001720                          TYPTXT  <<CRLF>!SET UP $DEVM TO INDICATE ACTUAL CONFIGURATION.!>
001762                          TYPTXT  <<CRLF>!TYPE PROCEED (P) TO CONTINUE.!>
002050                          HALT
002116  000000                                                  ENDDO
002120
002120  000674                  BR      $1
                        $2:
002122                                                          LET INITFLAG := #1
002122  012767  000001  007122  MOV     #1,INITFLAG
002130                                                          LET BITMASK := #BIT15   ; START AT CONSOLE
002130  012767  100000  007112  MOV     #BIT15,BITMASK
                        LOOP:
002136                          CALL    CYCLE           ; NO ARGUMENTS--ADDRS -> NEXT ADDRESS
002136  004767  006666          JSR     PC,CYCLE
                                ;               ADDRS+2 -> NEXT VECTOR
                                                        ;GET UNIT ADDRESS
002142                                                  LET     DLADD := (ADRS)+
002142  012167  177106  ~       MOV     (ADRS)+,DLADD
                                                        ;GET UNIT VECTOR
002146                                                  LET     DLVEC := (ADRS)
002146  011167  177104          MOV     (ADRS),DLVEC
002152                                                  LET     ADRS  := DLADD
002152  016701  177076          MOV     DLADD,ADRS
                                                        ;RCSR = DLADD  +  0
002156                                                  LET     RCSR  := DLADD
002156  016767  177072  177074  MOV     DLADD,RCSR
002164                                                  LET     RBUF  := DLADD + #2
002164  016767  177064  177070  MOV     DLADD,RBUF
002172  062767  000002  177062  ADD     #2,RBUF
002200                                                  LET     TCSR  := DLADD + #4
002200  016767  177050  177056  MOV     DLADD,TCSR
002206  062767  000004  177050  ADD     #4,TCSR
002214                                                  LET     TCSRHI := DLADD + #5
002214  016767  177034  177044  MOV     DLADD,TCSRHI
002222  062767  000005  177036  ADD     #5,TCSRHI
002230                                                  LET     TBUF  := DLADD + #6
002230  016767  177020  177032  MOV     DLADD,TBUF
002236  062767  000006  177024  ADD     #6,TBUF
002244                                                  LET     R5    := #R5STACK
002244  012705  001334          MOV     #R5STACK,R5
                                                        ;;BRESET
002250  000005                  RESET
```

```
                                ;;*****************************************************************
                                ;*TEST 1         ADDRESSABILITY
                                ;*              THIS TEST VERIFIES THAT THE ADDRESS AS PLACED IN
                                ;*              THE HARDWARE P-TABLE TO BE CORRECT AND THE DLV11-F RESPONDS
                                ;*              TO THAT ADDRESS SPACE
                                ;;*****************************************************************
002252  000004                TST1:   SCOPE
002254  012767  000002  176676        MOV     #2,$TIMES       ;;DO 2 ITERATIONS
002262  012767  000001  176710        MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002270                                                                  LET     ADRS := DLADD
002270  016701  176760                MOV     DLADD,ADRS
                                                                        ; SET UP INTERRUPT
002274                                                          SETVEC  #ILLMEM,#INTSRV,#PR7
002274  010146                        MOV     R1,-(SP)
002276  012701  000004                MOV     #ILLMEM,R1
002302  012721  011020                MOV     #INTSRV,(R1)+
002306  012711  000340                MOV     #PR7,(R1)
002312  012601                        MOV     (SP)+,R1
002314                                                          LET     I := #0
002314  005067  176752                CLR     I
002320                                                          REPEAT
002320                        $3:
002320                                                              BGNSUB
002320  012767  002326  176562        MOV     #64$,$LPERR
                                                                        ;CLEAR FLAG
                                                                        LET INTFLAG := #0
002326
002326  005067  006474                CLR     INTFLAG
                                                                        ;READ FLAG
002332  005711                        TST @ADRS
002334                                                                  IF INTFLAG NE #0 THEN
002334  005767  006466                TST     INTFLAG
002340  001401                        BEQ     $4
                                                                        ; FATAL ERROR
002342                                                                      ERROF 1,,NODL
002342  104001                        ERROR   1
002344                                                                  ENDIF
002344                        $4:
                                                                    ENDSUB
                                                                    LET     I := I + #2
002344  062767  000002  176720        ADD     #2,I
                                                                    LET     ADRS := DLADD + I
002352
002352  016701  176676                MOV     DLADD,ADRS
002356  066701  176710                ADD     I,ADRS
002362                                                          UNTIL I EQ #8.
002362  026727  176704  000010        CMP     I,#8.
002370  001353                        BNE     $3
002372                                                          CLRVEC ILLMEM
002372  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
002374  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
002376  012701  000004                MOV     #ILLMEM,R1
003002  010102                        MOV     R1,R2
003004  062702  000002                ADD     #2,R2
003010  012221                        MOV     R2,(R1)+
003012  005011                        CLR     (R1)
003014  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
1183  002416  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
1184                                                          ;END OF TEST
1185  002420                              ENDTST
```

```
1186                                    ;;*****************************************************************
1187                                    ;* THE FOLLOWING 8 TESTS  TEST ALL 'READ WRITE' BITS
1188                                    ;;*****************************************************************
1189                                    ;;*****************************************************************
1190                                    ;*TEST 2        BREAK - TCSR0 SET, CLEAR, RESET
1191                                    ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
1192                                    ;*              THAT IS READ AND WRITE.
1193                                    ;;*****************************************************************
1194  002420  000004                    TST2:   SCOPE
1195  002422  012767  000010  176530            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1196  002430  012767  000002  176542            MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1197
1198  002436                                                            IF #BRK NOTSETIN $USWR THEN
1199  002436  032767  010000  176554            BIT     #BRK,$USWR
1200  002444  001004                            BNE     $5
1201  002446                                                                EXIT TEST   ; BREAK NOT INSTALLED
1202  002446  012767  000001  176504            MOV     #1,$TIMES
1203  002454  000452                            BR      TST3            ;;;EXIT THIS TEST
1204  002456                                                            ENDIF
1205  002456                            $5:
1206
1207                                                        ; SEE IF IT IS CLEAR
1208  002456                                                            BGNSUB
1209  002456  012767  002464  176424            MOV     #64$,$LPERR
1210
1211  002464                                                                IF      #BREAK SETIN @TCSR THEN
1212  002464  032777  000001  176572            BIT     #BREAK,@TCSR
1213  002472  001401                            BEQ     $6
1214                                                                          ; BREAK DID NOT RESET IN TCSR
1215  002474                                                                  ERRHRD 2,,DIDNOT
1216  002474  104002                            ERROR   2
1217  002476                                                                ENDIF
1218  002476                            $6:
1219  002476                                                            ENDSUB
1220
1221                                                        ; TRY TO SET BREAK BIT
1222  002476                                                            BGNSUB
1223  002476  012767  002504  176404            MOV     #64$,$LPERR
1224  002504                                                                LET     @TCSR := @TCSR SET.BY #BREAK
1225  002504  052777  000001  176552            BIS     #BREAK,@TCSR
1226                                                                      ; STUCK TO 0
1227  002512                                                                IF      #BREAK NOTSETIN @TCSR THEN
1228  002512  032777  000001  176544            BIT     #BREAK,@TCSR
1229  002520  001001                            BNE     $7
1230                                                                          ; BREAK DID NOT SET IN TCSR
1231  002522                                                                  ERRHRD 3,,DIDNOT
1232  002522  104003                            ERROR   3
1233  002524                                                                ENDIF
1234  002524                            $7:
1235  002524                                                            ENDSUB
1236
1237                                                        ; TRY TO CLEAR A SET BIT
1238  002524                                                            BGNSUB
1239  002524  012767  002532  176356            MOV     #64$,$LPERR
1240
1241  002532                                                                LET     @TCSR := @TCSR CLR.BY #BREAK
```

```
1242  002532  042777  000001  176524          BIC     #BREAK,@TCSR
1243                                                          ; SHOULD HAVE CLEARED
1244  002540                                                IF      #BREAK SETIN @TCSR THEN
1245  002540  032777  000001  176516          BIT     #BREAK,@TCSR
1246  002546  001401                          BEQ     $10
1247                                                          ; BREAK  DID NOT CLEAR IN TCSR
1248  002550                                                ERRHRD 4,,DIDNOT
1249  002550  104004                          ERROR   4
1250  002552                                                ENDIF
1251  002552                          $10:
1252  002552                                              ENDSUB
1253
1254                                                  ; NOW SEE IF RESET CLEARS IT
1255  002552                                              BGNSUB
1256  002552  012767  002560  176330          MOV     #64$,$LPERR
1257
1258  002560                                                LET     @TCSR := @TCSR SET.BY #BREAK
1259  002560  052777  000001  176476          BIS     #BREAK,@TCSR
1260                                                          ; ISSUE BUS RESET
1261  002566                                                BRESET
1262  002566  000005                          RESET
1263  002570                                                IF      #BREAK SETIN @TCSR THEN
1264  002570  032777  000001  176466          BIT     #BREAK,@TCSR
1265  002576  001401                          BEQ     $11
1266                                                          ; BREAK DID NOT RESET IN TCSR
1267  002600                                                ERRHRD 5,,DIDNOT
1268  002600  104005                          ERROR   5
1269  002602                                                ENDIF
1270  002602                          $11:
1271  002602                                              ENDSUB
1272  002602                                            ENDTST
1273
1274
```

E03


```
1275                                    ;;**********************************************************************
1276                                    ;;**********************************************************************
1277                                    ;*TEST 3          MAINT - TCSR2 SET, CLEAR, RESET
1278                                    ;;**********************************************************************
1279  002602  000004                    TST3:   SCOPE
1280  002604  012767  000010  176346            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1281  002612  012767  000003  176360            MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1282
1283  002620                                                              IF #MAINTJUMP NOTSETIN $USWR THEN
1284  002620  032767  040000  176372            BIT     #MAINTJUMP,$USWR
1285  002626  001004                            BNE     $12
1286  002630                                                                  EXIT TEST
1287  002630  012767  000001  176322            MOV     #1,$TIMES
1288  002636  000452                            BR      TST4            ;;;EXIT THIS TEST
1289  002640                                                              ENDIF
1290  002640                                $12:
1291
1292                                                        ; SEE IF IT IS CLEAR
1293  002640                                                              BGNSUB
1294  002640  012767  002646  176242            MOV     #64$,$LPERR
1295
1296  002646                                                                  IF      #MAINT SETIN @TCSR THEN
1297  002646  032777  000004  176410            BIT     #MAINT,@TCSR
1298  002654  001401                            BEQ     $13
1299
1300  002656                                                                      ; MAINT DID NOT RESET IN TCSR
1301  002656  104006                                                             ERRHRD 6,,DIDNOT
1302  002660                                                                  ENDIF
1303  002660                                $13:
1304  002660                                                              ENDSUB
1305
1306                                                        ; TRY TO SET MAINT BIT
1307  002660                                                              BGNSUB
1308  002660  012767  002666  176222            MOV     #64$,$LPERR
1309  002666                                                                  LET     @TCSR := @TCSR SET.BY #MAINT
1310  002666  052777  000004  176370            BIS     #MAINT,@TCSR
1311                                                                          ; STUCK TO 0
1312  002674                                                                  IF      #MAINT NOTSETIN @TCSR THEN
1313  002674  032777  000004  176362            BIT     #MAINT,@TCSR
1314  002702  001001                            BNE     $14
1315
1316  002704                                                                      ; MAINT DID NOT SET IN TCSR
1317  002704  104007                                                             ERRHRD 7,,DIDNOT
1318  002706                                                                  ENDIF
1319  002706                                $14:
1320  002706                                                              ENDSUB
1321
1322                                                        ; TRY TO CLEAR A SET BIT
1323  002706                                                              BGNSUB
1324  002706  012767  002714  176174            MOV     #64$,$LPERR
1325
1326  002714                                                                  LET     @TCSR := @TCSR CLR.BY #MAINT
1327  002714  042777  000004  176342            BIC     #MAINT,@TCSR
1328                                                                          ; SHOULD HAVE CLEARED
1329  002722                                                                  IF      #MAINT SETIN @TCSR THEN
1330  002722  032777  000004  176334            BIT     #MAINT,@TCSR
```

```
1331  002730  001401                    BEQ     $15
1332                                                        ; MAINT DID NOT CLEAR INTCSR
1333  002732                                                ERRHRD 10,,DIDNOT
1334  002732  104010                    ERROR   10
1335  002734                                                ENDIF
1336  002734                          $15:
1337  002734                                              ENDSUB
1338
1339                                                  ; NOW SEE IF RESET CLEARS IT
1340  002734                                              BGNSUB
1341  002734  012767  002742  176146    MOV     #64$,$LPERR
1342
1343  002742                                                LET   @TCSR := @TCSR SET.BY #MAINT
1344  002742  052777  000004  176314    BIS     #MAINT,@TCSR
1345                                                      ; ISSUE BUS RESET
1346  002750                                              BRESET
1347  002750  000005                    RESET
1348  002752                                              IF    #MAINT SETIN @TCSR THEN
1349  002752  032777  000004  176304    BIT     #MAINT,@TCSR
1350  002760  001401                    BEQ     $16
1351                                                        ; MAINT DID NOT RESET IN TCSR
1352  002762                                                ERRHRD 11,,DIDNOT
1353  002762  104011                    ERROR   11
1354  002764                                                ENDIF
1355  002764                          $16:
1356  002764                                            ENDSUB
1357  002764                                          ENDTST
1358
1359
1360
```

```
1361                                    ;;*****************************************************************
1362                                    ;;*****************************************************************
1363                                    ;*TEST 4          XMITIE - TCSR6 SET, CLEAR, RESET
1364                                    ;;*****************************************************************
1365  002764  000004                    TST4:   SCOPE
1366  002766  012767  000010  176164            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1367  002774  012767  000004  176176            MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1368                                                                    ; USE PRIORITY OF 7
1369  003002  012746  000340                    MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
1370  003006  012746  003014                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
1371  003012  000002                            RTI                     ;;POP NEW PC AND PS
1372  003014                            64$:
1373
1374                                                    ; SEE IF IT IS CLEAR
1375  003014                                                          BGNSUB
1376  003014  012767  003022  176066            MOV     #65$,$LPERR
1377
1378  003022                                                              IF      #XMITIE SETIN @TCSR THEN
1379  003022  032777  000100  176234            BIT     #XMITIE,@TCSR
1380  003030  001401                            BEQ     $17
1381
1382  003032                                                                  ; XMITIE DID NOT RESET IN TCSR
1383  003032  104012                                                         ERRHRD 12,,DIDNOT
1384  003034                                                              ENDIF
1385  003034                            $17:
1386  003034                                                          ENDSUB
1387
1388                                                    ; TRY TO SET XMITIE BIT
1389  003034                                                          BGNSUB
1390  003034  012767  003042  176046            MOV     #64$,$LPERR
1391  003042                                                              LET     @TCSR := @TCSR SET.BY #XMITIE
1392  003042  052777  000100  176214            BIS     #XMITIE,@TCSR
1393                                                                  ; STUCK TO 0
1394  003050                                                              IF      #XMITIE NOTSETIN @TCSR THEN
1395  003050  032777  000100  176206            BIT     #XMITIE,@TCSR
1396  003056  001001                            BNE     $20
1397                                                                      ; XMIT DID NOT RESET IN TCSR
1398  003060                                                              ERRHRD 13,,DIDNOT
1399  003060  104013                            ERROR   13
1400  003062                                                              ENDIF
1401  003062                            $20:
1402  003062                                                          ENDSUB
1403
1404                                                    ; TRY TO CLEAR A SET BIT
1405  003062                                                          BGNSUB
1406  003062  012767  003070  176020            MOV     #64$,$LPERR
1407
1408  003070                                                              LET     @TCSR := @TCSR CLR.BY #XMITIE
1409  003070  042777  000100  176166            BIC     #XMITIE,@TCSR
1410                                                                  ; SHOULD HAVE CLEARED
1411  003076                                                              IF      #XMITIE SETIN @TCSR THEN
1412  003076  032777  000100  176160            BIT     #XMITIE,@TCSR
1413  003104  001401                            BEQ     $21
1414                                                                      ;  XMIT DID NOT CLEAR IN TCSR
1415  003106                                                              ERRHRD 14,,DIDNOT
1416  003106  104014                            ERROR   14
```

```
1417  003110                                                              ENDIF
1418  003110                          $21:
1419  003110                                                          ENDSUB
1420
1421                                                    ; NOW SEE IF RESET CLEARS IT
1422  003110                                                          BGNSUB
1423  003110  012767  003116  175772          MOV     #64$,$LPERR
1424
1425  003116                                                          LET     @TCSR := @TCSR SET.BY #XMITIE
1426  003116  052777  000100  176140          BIS     #XMITIE,@TCSR
1427                                                                      ; ISSUE BUS RESET
1428  003124                                                          BRESET
1429  003124  000005                  RESET
1430  003126                                                          IF      #XMITIE SETIN @TCSR THEN
1431  003126  032777  000100  176130          BIT     #XMITIE,@TCSR
1432  003134  001401                  BEQ     $22
1433                                                                        ; XMIT DID NOT RESET IN TCSR
1434  003136                                                                ERRHRD 15,,DIDNOT
1435  003136  104015                  ERROR   15
1436  003140                                                          ENDIF
1437  003140                          $22:
1438  003140                                                      ENDSUB
1439  003140                                                  ENDTST
1440
1441
1442
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 5         RCVRIE - RCSR6 SET, CLEAR, RESET
                                ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
                                ;*              THAT IS READ AND WRITE.
                                ;;*****************************************************************
003140  000004                  TST5:   SCOPE
003142  012767  000010  176010          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
003150  012767  000005  176022          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; SEE IF IT IS CLEAR
003156                                                                  BGNSUB
003156  012767  003164  175724          MOV     #64$,$LPERR
003164                                                                      IF      #RCVRIE SETIN @RCSR THEN
003164  032777  000.30  176066          BIT     #RCVRIE,@RCSR
003172  001401                          BEQ     $23
                                                                            ; RCVRIE DID NOT RESET IN RCSR
                                                                            ERRHRD 35,,DIDNOT
003174
003174  104035                          ERROR   35
003176                                                                      ENDIF
003176                          $23:                                    ENDSUB
003176
                                                        ; TRY TO SET RCVRIE BIT
003176                                                                  BGNSUB
003176  012767  003204  175704          MOV     #64$,$LPERR
003204                                                                      LET     @RCSR := @RCSR SET.BY #RCVRIE
003204  052777  000100  176046          BIS     #RCVRIE,@RCSR
                                                                            ; STUCK TO 0
003212                                                                      IF      #RCVRIE NOTSETIN @RCSR THEN
003212  032777  000100  176040          BIT     #RCVRIE,@RCSR
003220  001001                          BNE     $24
                                                                            ; RCVRIE DID NOT SET IN RCSR
003222                                                                      ERRHRD 36,,DIDNOT
003222  104036                          ERROR   36
003224                                                                      ENDIF
003224                          $24:                                    ENDSUB

                                                        ; TRY TO CLEAR A SET BIT
003224                                                                  BGNSUB
003224  012767  003232  175656          MOV     #64$,$LPERR
                                                                            LET     @RCSR := @RCSR CLR.BY #RCVRIE
003232  042777  000100  176020          BIC     #RCVRIE,@RCSR
                                                                            ; SHOULD HAVE CLEARED
003240                                                                      IF      #RCVRIE SETIN @RCSR THEN
003240  032777  000100  176012          BIT     #RCVRIE,@RCSR
003246  001401                          BEQ     $25
                                                                            ; RCVRIE DID NOT CLEAR IN RCSR
003250                                                                      ERRHRD 37,,DIDNOT
003250  104037                          ERROR   37
003252                                                                      ENDIF
003252                          $25:                                    ENDSUB
003252
```

```
MAINDEC-11-DVDVC-A     MACY11 27(1006)  08-AUG-77  09:20  PAGE 36
DVDVCA.P11    08-AUG-77 09:16          T5       RCVRIE - RCSR6 SET, CLEAR, RESET                                    SEQ 0035

                                                                ; NOW SEE IF RESET CLEARS IT
                                                                    BGNSUB
003252
003252  012767  003260  175630          MOV     #64$,$LPERR
003260
003260  052777  000100  175772          BIS     #RCVRIE,@RCSR
                                                                    LET     @RCSR := @RCSR SET.BY #RCVRIE
                                                                        ; ISSUE BUS RESET
                                                                    BRESET
003266
003266  000005                          RESET
003270
003270  032777  000100  175762          BIT     #RCVRIE,@RCSR
003276  001401                          BEQ     $26
                                                                    IF      #RCVRIE SETIN @RCSR THEN
                                                                        ; RCVRIE DID NOT RESET IN RCSR
                                                                        ERRHRD 40,,DIDNOT
003300
003300  104040                          ERROR   40
                                                                    ENDIF
003302
003302                          $26:
                                                                    CKLOOP
003302                                                          ENDSUB
003302                                                      ENDTST
003302
```

```
1523                                ;;**************************************************************
1524                                ;* THE FOLLOWING 4 TESTS VERIFY
1525                                ;* THAT RESET (INIT) INITIALIZES READ ONLY BITS.
1526                                ;**************************************************************
1527                                ;**************************************************************
1528                                ;TEST 6         TEST THAT RCVRDONE - RCSR 7 - IS CLEARED BY INIT
1529                                ;**************************************************************
1530  003302  000004                TST6:   SCOPE
1531  003304  012767  000010  175646        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1532  003312  012767  000006  175660        MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1533
1534
1535
1536                                                                BGNSUB
1537  003320
1538  003320  012767  003326  175562        MOV     #64$,$LPERR         IF      #RCVRDONE SETIN @RCSR THEN
1539  003326
1540  003326  032777  000200  175724        BIT     #RCVRDONE,@RCSR
1541  003334  001402                        BEQ     $27
1542
1543                                                                    ;RCVRDONE SHOULD HAVE CLEARED BY INIT
1544                                                                    ; RCVRDONE DID NOT CLEAR IN RCSR
1545  003336                                                            ERRHRD 41,HRESET, DIDNOT
1546  003336  104041                ERROR   41
1547                                                                    ;REISSUE RESET
1548  003340                                                            BRESET
1549  003340  000005                RESET
1550  003342                                                        ENDIF
1551  003342                        $27:
1552                                                                    ;ALLOW LOOPING AFTER ERROR
1553  003342                                                        CKLOOP
1554  003342                                                    ENDSUB
1555  003342                                                ENDTST
1556
1557
1558
```

```
1559                                    ;;**************************************************************
1560                                    ;;**************************************************************
1561                                    ;;*TEST 7          TEST THAT RCVRACT - RCSR 11 - IS CLEARED BY INIT
1562                                    ;;**************************************************************
1563  003342  000004                    TST7:   SCOPE
1564  003344  012767  000010  175606            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1565  003352  012767  000007  175620            MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1566
1567
1568
1569                                                                    IF CONSOLE EQ #TRUE THEN
1570  003360
1571  003360  026727  005703  000001            CMP     CONSOLE,#TRUE
1572  003366  001001                            BNE     $30
1573                                                                        ;; EXECUTE TEST
1574  003370                                                            ELSE
1575  003370  000416                            BR      $31
1576  003372                            $30:
1577  003372                                                            IF #WRAP SETIN $USWR THEN
1578  003372  032767  020000  175620            BIT     #WRAP,$USWR
1579  003400  001401                            BEQ     $32
1580                                                                        ;; EXECUTE TEST
1581  003402                                                            ELSE
1582  003402  000411                            BR      $33
1583  003404                            $32:
1584  003404                                                            IF #MAINT SETIN $USWR THEN
1585  003404  032767  000004  175606            BIT     #MAINT,$USWR
1586  003412  001401                            BEQ     $34
1587                                                                        ;;EXECUTE TEST
1588  003414                                                            ELSE
1589  003414  000404                            BR      $35
1590  003416                            $34:
1591  003416                                                               EXIT TEST    ; LINE MUST BE TERMINATED
1592  003416  012767  000001  175534            MOV     #1,$TIMES
1593  003424  000414                            BR      TST10           ;;;EXIT THIS TEST
1594  003426                                                            ENDIF
1595  003426                            $35:                            ENDIF
1596  003426
1597  003426                            $33:                            ENDIF
1598  003426
1599  003426                            $31:
1600
1601  003426                                          BGNSUB
1602  003426  012767  003434  175454            MOV     #64$,$LPERR
1603
1604  003434                                              IF      #RCVRACT SETIN @RCSR THEN
1605  003434  032777  004000  175616            BIT     #RCVRACT,@RCSR
1606  003442  001405                            BEQ     $36
1607
1608                                                                    ;RESET SHOULD HAVE CLEARED RCVRACT
1609  003444                                                            LET @TCSR := @TCSR CLR.BY #MAINT
1610  003444  042777  000004  175612            BIC     #MAINT,@TCSR
1611  003452                                                            ERRHRD 44, HRESET, DIDNOT
1612  003452  104044                            ERROR   44
1613
1614                                                                    ;TESTING EFFECT OF RESET ON BIT
```

M03


```
1615
1616                                                                    ;RCVRACT DID NOT CLEAR IN RCSR
1617
1618                                                                    ;ALLOW ANOTHER TRY
1619  003454                                                            BRESET
1620  003454  000005                    RESET
1621  003456                                                          ENDIF
1622  003456                    $36:
1623                                                                    ;ALLOW LOOPING ON ERROR
1624  003456                                                          CKLOOP
1625  003456                                                        ENDSUB
1626  003456                                                    ENDTST
1627
```

```
1628                                    ;;**********************************************************
1629                                    ;;**********************************************************
1630                                    ;*TEST 10        TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
1631                                    ;;**********************************************************
1632  003456  000004                    TST10:  SCOPE
1633  003460  012767  000010  175472            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1634  003466  012767  000010  175504            MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1635
1636
1637
1638
1639  003474                                                            BGNSUB
1640  003474  012767  003502  175406            MOV     #64$,$LPERR
1641
1642  003502                                                                IF      #XMITRDY NOTSETIN @TCSR THEN
1643  003502  032777  000200  175554            BIT     #XMITRDY,@TCSR
1644  003510  001002                            BNE     $37
1645
1646                                                                          ;RESET SHOULD HAVE SET BIT.
1647                                                                          ;XMITRDY DID NOT SET IN TCSR (AFTER RESE
1648  003512                                                                  ERRHRD 42,HRESET,DIDNOT
1649  003512  104042                            ERROR   42
1650                                                                          ;ISSUE ANOTHER RESET
1651  003514                                                                  BRESET
1652  003514  000005                            RESET
1653  003516                                                                ENDIF
1654  003516                            $37:
1655                                                                          ;ALLOW LOOPING ON ERROR
1656  003516                                                                CKLOOP
1657  003516                                                            ENDSUB
1658  003516                                                        ENDTST
1659
1660
1661
```

```
1662                                    ;;*****************************************************************
1663                                    ;;*****************************************************************
1664                                    ;*TEST 11        TEST THAT XMIT RDY - TCSR 7 - CLEARS
1665                                    ;*              WHEN TBUF IS LOADED WITH A CHARACTER
1666                                    ;*              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
1667                                    ;;*****************************************************************
1668  003516  000004                    TST11:  SCOPE
1669  003520  012767  000010  175432            MOV     #10,$TIMES      ;;DO 10 ITERATIONS
1670  003526  012767  000011  175444            MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1671                                                                            IFB APTCON EQ #TRUE THEN
1672  003534
1673  003534  126727  005526  000001            CMPB    APTCON,#TRUE
1674  003542  001004                            BNE     $40
1675  003544                                                                        EXIT TEST
1676  003544  012767  000001  175406            MOV     #1,$TIMES
1677  003552  000513                            BR      TST12           ;;;EXIT THIS TEST
1678  003554                                                                        ENDIF
1679  003554                            $40:
1680
1681  003554                                                                    LET PASS := #1  ;INIT COUNT OF TIMES THRU
1682  003554  012767  000001  000212            MOV     #1,PASS
1683  003562                                                                    LOOP            ; START OF LOOP
1684  003562                            $41:
1685                                                                                        ;  MAX OF 2 TIMES THRU
1686  003562                                                                       LET ERRORFLAG := #CLR
1687  003562  012767  000000  000206            MOV     #CLR,ERRORFLAG
1688  003570                                                                        LET EXITFLAG := #CLR
1689  003570  012767  000000  000202            MOV     #CLR,EXITFLAG
1690                                                        ; LOAD TBUF WITH ONE CHARACTER
1691                                                        ; WAIT FOR READY TO SET
1692                                                        ; (SHOULD BE VERY SHORT WAIT
1693                                                        ; SINCE UART DOUBLE BUFFERS ITS INPUT)
1694
1695                                                                            ;SEND A CHARACTER
1696  003576                                                                    LET @TBUF :B= #0
1697  003576  105077  175466                    CLRB    @TBUF
1698                                                                            ;WAIT A MAXIMUM
1699                                                                            ;OF 50 MSEC FOR
1700                                                                            ;XMIT RDY TO SET IN TCSR
1701  003602                                                                    CALL TIMER IN <#5,#XMITRDY,TCSR,#SET>
1702  003602  010546                            MOV     R5,-(SP)
1703  003604  012745  177777                    MOV     #SET,-(R5)
1704  003610  016745  175450                    MOV     TCSR,-(R5)
1705  003614  012745  000200                    MOV     #XMITRDY,-(R5)
1706  003620  012745  000005                    MOV     #5,-(R5)
1707  003624  004767  004632                    JSR     PC,TIMER
1708  003630  012605                            MOV     (SP)+,R5
1709                                                                            ;TIMER RETURNS AN ERROR IF BIT DID
1710                                                                            ;NOT MEET CONDITION WITHIN TIME LIMIT
1711  003632                                                                    IF.ERROR THEN
1712  003632  103001                            BCC     $43
1713                                                                            ;XMIT RDY DID NOT SET IN TCSR
1714  003634                                                                    ERRHRD 66,,DIDNOT
1715  003634  104066                            ERROR   66
1716  003636                                                                    ENDIF
1717  003636                            $43:
```

```
1718
1719                                                    ; LOAD TBUF WITH A SECOND CHARACTER
1720                                                    ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
1721                                                    ; AND THEN WAIT FOR IT TO SET
1722
1723                                                                    ;SEND SECOND CHARACTER
1724  003636                                                        LET @TBUF :B= #0
1725  003636  105077  175426            CLRB    @TBUF
1726  003642  000240                            NOP                ; GIVE IT TIME TO CLEAR
1727                                                                ; XMITRDY SHOULD HAVE CLEARED UPON
1728                                                                ; RECEIPT OF A CHARACTER
1729  003644                                                        IF #XMITRDY SETIN @TCSR THEN
1730  003644  032777  000200  175412    BIT     #XMITRDY,@TCSR
1731  003652  001404                    BEQ     $44
1732
1733  003654                                                            ; XMITRDY DID NOT CLEAR IN TCSR
1734  003654  012767  177777  000114    MOV     #SET,ERRORFLAG          LET ERRORFLAG := #SET
1735                                                                            ; DEFER ERROR TYPEOUT
1736
1737  003662                                                        ELSE
1738  003662  000416                    BR      $45
1739  003664                    $44:
1740                                                                ;WAIT A MAXIMUM
1741                                                                ;OF 50 MSEC FOR
1742                                                                ;XMIT RDY TO SET IN TCSR
1743  003664                                                        CALL TIMER IN <#5,#XMITRDY,TCSR,#SET>
1744  003664  010546                    MOV     R5,-(SP)
1745  003666  012745  177777            MOV     #SET,-(R5)
1746  003672  016745  175366            MOV     TCSR,-(R5)
1747  003676  012745  000200            MOV     #XMITRDY,-(R5)
1748  003702  012745  000005            MOV     #5,-(R5)
1749  003706  004767  004550            JSR     PC,TIMER
1750  003712  012605                    MOV     (SP)+,R5
1751  003714
1752  003714  103001                    BCC     $46                     IF.ERROR THEN
1753
1754  003716                                                                ;XMIT RDY DID NOT SET IN TCSR
1755  003716  104070                    ERROR   70                          ERRHRD 70,,DIDNOT
1756  003720                                                            ENDIF
1757  003720                    $46:
1758  003720                                                        ENDIF ; OF DEFERED ERROR CALL
1759  003720                    $45:
1760  003720                                                    IF ERRORFLAG EQ #SET THEN
1761  003720  026727  000052  177777    CMP     ERRORFLAG,#SET
1762  003726  001011                    BNE     $47
1763  003730                                                        IF PASS GT #1 THEN
1764  003730  026727  000040  000001    CMP     PASS,#1
1765  003736  003404                    BLE     $50
1766                                                                    ; CALL ERROR IF 2ND TRY
1767  003740                                                            ERRHRD 67,,DIDNOT
1768  003740  104067                    ERROR   67
1769  003742                                                            LET EXITFLAG := #SET
1770  003742  012767  177777  000030    MOV     #SET,EXITFLAG           ENDIF
1771  003750
1772  003750                    $50:                                ELSE            ; NO ERROR
1773  003750
```

```
1774  003750  000403                          BR      $51
1775  003752                          $47:
1776  003752                                                          LET EXITFLAG := #SET
1777  003752  012767  177777  000020          MOV     #SET,EXITFLAG
1778  003760                                                      ENDIF
1779  003760                          $51:
1780  003760                                                  EXIF        EXITFLAG EQ #SET
1781  003760  026727  000014  177777          CMP     EXITFLAG,#SET
1782  003766  001401                          BEQ     $42
1783  003770                                                  ENDLOOP
1784  003770  000674                          BR      $41
1785  003772                          $42:
1786  003772                                                      EXIT ; SKIP AROUND FLAG WORDS
1787  003772  000403                          BR      TST12          ;;;EXIT THIS TEST
1788  003774  000000                  PASS:   0
1789  003776  000000                  ERRORFLAG:  0
1790  004000  000000                  EXITFLAG:   0
1791  004002                                                  ENDTST
```

```
                                ;;**************************************************************
                                ;;**************************************************************
                                ;*TEST 12        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.
                                ;;**************************************************************
004002  000004                  TST12:  SCOPE
004004  012767  000010  175146          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004012  012767  000012  175160          MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                IF #MAINTJUMP NOTSETIN $USWR ORB APTCON EQ #TRUE
004020  032767  040000  175172          BIT     #MAINTJUMP,$USWR
004026  001404                          BEQ     $52
004030  126727  005232  000001          CMPB    APTCON,#TRUE
004036  001004                          BNE     $53
004040                          $52:                                    EXIT TEST
004040  012767  000001  175112          MOV     #1,$TIMES
004046  000442                          BR      TST13           ;;;EXIT THIS TEST
                                                                ENDIF
004050                          $53:
                                                                        ; SET THE MAINTENANCE BIT
                                                                LET @TCSR := @TCSR SET.BY #MAINT
004050  052777  000004  175206          BIS     #MAINT,@TCSR
                                                                BGNSUB
004056  012767  004064  175024          MOV     #64$,$LPERR
                                                ; SEND A CHARACTER AND LET IT WRAP AROUND
                                                                LET @TBUF :B= #0
004064  105077  175200                  CLRB    @TBUF
                                                ; WAIT A MAXIMUM OF 50 MSEC
                                                ; FOR RCVR DONE TO SET IN
                                                ; RCSR
004070                                          CALL TIMER IN <#5,#RCVRDONE,RCSR,#SET>
004070  010546                          MOV     R5,-(SP)
004072  012745  177777                  MOV     #SET,-(R5)
004076  016745  175156                  MOV     RCSR,-(R5)
004102  012745  000200                  MOV     #RCVRDONE,-(R5)
004106  012745  000005                  MOV     #5,-(R5)
004112  004767  004344                  JSR     PC,TIMER
004116  012605                          MOV     (SP)+,R5
                                                    ;DIDN'T SET IN TIME
004120                                          IF.ERROR THEN
004120  103004                          BCC     $54
                                                    ; RCVRDONE DID NOT SET IN RCSR
                                                    ; CAN NOT LEAVE WITH MAINT SET
004122                                          LET    @TCSR := @TCSR CLR.BY #MAINT
004122  042777  000004  175134          BIC     #MAINT,@TCSR
                                                    ERRHRD 71,,DIDNOT
004130  104071                          ERROR   71
                                                ENDIF
004132                          $54:
```

```
MAINDEC-11-DVDVC-A      MACY11 27(1006)  08-AUG-77  09:20  PAGE 45
DVDVCA.P11    08-AUG-77 09:16           T12      TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH MAINT SET)                    SEQ 0044

 1848
 1849  004132                                                                ENDSUB
 1850
 1851  004132                                                                BGNSUB
 1852  004132  012767  004140  174750          MOV     #64$,$LPERR
 1853                                                    ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
 1854                                                    ; THIS ALSO WILL CLEAR THE MAINT. BIT
 1855  004140                                                                BRESET
 1856  004140  000005                          RESET
 1857
 1858  004142                                                                IF #RCVRDONE SETIN @RCSR THEN
 1859  004142  032777  000200  175110          BIT     #RCVRDONE,@RCSR
 1860  004150  001401                          BEQ     $55
 1861                                                        ; RCVRDONE DID NOT RESET IN RCSR.
 1862  004152                                                                ERRHRD 72,,DIDNOT
 1863  004152  104072                          ERROR   72
 1864  004154                                                                ENDIF
 1865  004154                          $55:
 1866  004154                                                                ENDSUB
 1867  004154                                                        ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;;*TEST 13        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
                                ;;*****************************************************************
004154  000004                  TST13:  SCOPE
004156  012767  000010  174774          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004164  012767  000013  175006          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
004172                                                          IF #MAINTJUMP NOTSETIN $USWR ORB APTCON EQ #TRUE
004172  032767  040000  175020          BIT     #MAINTJUMP,$USWR
004200  001404                          BEQ     $56
004202  126727  005060  000001          CMPB    APTCON,#TRUE
004210  001004                          BNE     $57
004212                  $56:                                    EXIT TEST
004212  012767  000001  174740          MOV     #1,$TIMES
004220  000451                          BR      TST14           ;;;EXIT THIS TEST
004222                                                          ENDIF
004222                  $57:
004222                                                                  ; SET MAINT. BIT
004222                                                          LET @TCSR := @TCSR SET.BY #MAINT
004222  052777  000004  175034          BIS     #MAINT,@TCSR
004230                                                          BGNSUB
004230  012767  004236  174652          MOV     #64$,$LPERR
                                        ; OUTPUT A CHARACTER WITH MAINTENANCE
                                        ; SET, AND WAIT FOR XMITRDY TO SET.

                                                                ; OUTPUT A CHARACTER
004236                                                  LET @TBUF :B= #0
004236  105077  175026          CLRB    @TBUF
                                                        ; WAIT MAXIMUM OF 500 MSEC
                                                        ; FOR RCVRDONE TO SET IN
                                                        ; RCSR
004242                                                  CALL TIMER IN <#50,#RCVRDONE,RCSR,#SET>
004242  010546                  MOV     R5,-(SP)
004244  012745  177777          MOV     #SET,-(R5)
004250  016745  175004          MOV     RCSR,-(R5)
004254  012745  000200          MOV     #RCVRDONE,-(R5)
004260  012745  000050          MOV     #50,-(R5)
004264  004767  004172          JSR     PC,TIMER
004270  012605                  MOV     (SP)+,R5
                                                                ; DID IT BECAME READY?
004272                                                  IF.ERROR  THEN
004272  103007                  BCC     $60
                                                                ;RCVRDONE DID NOT SET IN RCSR
                                                                ; CAN NOT LEAVE WITH MAINT SET
004274                                                  LET     @TCSR := @TCSR CLR.BY #MAINT
004274  042777  000004  174762          BIC     #MAINT,@TCSR
004302                                                          ERRHRD 73,, DIDNOT
004302  104073                  ERROR   73
                                                                ; SET IT BACK TO CONTINUE
004304                                                  LET @TCSR := @TCSR SET.BY #MAINT
004304  052777  000004  174752          BIS     #MAINT,@TCSR
004312                                                          ENDIF
004312                  $60:
```

```
                                                                                    ENDSUB

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

                                                                    ;READ BUFFER
                                                                LET R0 :B= @RBUF
004312  117700  174744          MOVB    @RBUF,R0

                                                                IF #RCVRDONE SETIN @RCSR THEN

004316  032777  000200  174734  BIT     #RCVRDONE,@RCSR
004324  001407                  BEQ     $61

                                                                    ;RCVRDONE DID NOT CLEAR IN RCSR
                                                                    ; CAN NOT LEAVE WITH MAINT SET
004326                                                          LET    @TCSR := @TCSR CLR.BY #MAINT
004326  042777  000004  174730  BIC     #MAINT,@TCSR

                                                                    ERRHRD 74,,DIDNOT
004334  104074                  ERROR   74

                                                                    ; SET IT BACK TO CONTINUE
004336                                                          LET @TCSR := @TCSR SET.BY #MAINT
004336  052777  000004  174720  BIS     #MAINT,@TCSR

                                                                    ENDIF
004344
                        $61:
004344                                                      ENDTST
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 14        TEST THAT RCVRACT - RCSR 11 - SETS
                                ;*              WHEN A START BIT IS RECEIVED AND
                                ;*              CLEARS WHEN RCVRDONE - RCSR 7 - SETS
                                ;;**********************************************************************
004344  000004                  TST14:  SCOPE
004346  012767  000010  174604          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004354  012767  000014  174616          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
004362                                                                  IFB APTCON EQ #FALSE AND #MAINTJUMP SETIN $USWR
004362  126727  004700  000000          CMPB    APTCON,#FALSE
004370  001005                          BNE     $62
004372  032767  040000  174620          BIT     #MAINTJUMP,$USWR
004400  001401                          BEQ     $62
004402                                                                      ;; EXECUTE TEST
                                                                        ELSE
004402  000404                          BR      $63
004404                          $62:
004404                                                                      EXIT TEST
004404  012767  000001  174546          MOV     #1,$TIMES
004412  000526                          BR      TST15           ;;;EXIT THIS TEST
004414                                                                  ENDIF
004414                          $63:
004414                                                                  LET @TCSR := @TCSR SET.BY #MAINT
004414  052777  000004  174642          BIS     #MAINT,@TCSR
004422                                                                      LET FLAG :B= #CLR
004422  112767  000000  004202          MOVB    #CLR,FLAG
004430                                                                      LET COUNT := #0
004430  005067  000232                  CLR     COUNT
                                                    ;LOAD A CHARACTER INTO TBUF
                                                    ;WAIT FOR RCVRACT TO SET
                                                                      ;SEND A CHARACTER
004434                                                                LET @TBUF :B= #0
004434  105077  174630                  CLRB    @TBUF
004440                                                          REPEAT
004440                          $64:
004440                                                              IF #RCVRACT SETIN @RCSR THEN
004440  032777  004000  174612          BIT     #RCVRACT,@RCSR
004446  001404                          BEQ     $65
004450                                                                  LET FLAG :B= #SET
004450  112767  177777  004154          MOVB    #SET,FLAG
004456                                                              ELSE
004456  000402                          BR      $66
004460                          $65:
004460                                                                  LET COUNT := COUNT + #1
004460  005267  000202                  INC     COUNT
004464                                                              ENDIF
004464                          $66:
004464                                                          UNTILB FLAG EQ #SET OR COUNT HI MAX
004464  126727  004142  177777          CMPB    FLAG,#SET
004472  001404                          BEQ     $67
004474  026767  000166  000162          CMP     COUNT,MAX
004502  101756                          BLOS    $64
004504                          $67:
004504                                                          IF COUNT HI MAX THEN
```

```
2004  004504  026767  000156  000152          CMP     COUNT,MAX
2005  004512  101410                          BLOS    $70
2006                                                                    ;IT NEVER SET
2007                                                                    ;RCVRACT DID NOT SET IN RCSR.
2008                                                                    ; CAN NOT LEAVE WITH MAINT SET
2009  004514                                                          LET   @TCSR := @TCSR CLR.BY #MAINT
2010  004514  042777  000004  174542          BIC     #MAINT,@TCSR
2011  004522                                                            ERRHRD 75,, DIDNOT
2012  004522  104075                          ERROR   75
2013  004524                                                            EXIT TEST
2014  004524  012767  000001  174426          MOV     #1,$TIMES
2015  004532  000456                          BR      TST15           ;;;EXIT THIS TEST
2016  004534                                                          ENDIF
2017  004534                          $70:
2018
2019
2020                                                    ;CHECK FOR TIMING OF RCVRACT. CLEARING
2021                                                    ;VS RCVRDONE SETTING
2022
2023
2024  004534                                                          WHILE #RCVRACT SETIN @RCSR DO
2025  004534                          $71:
2026  004534  032777  004000  174516          BIT     #RCVRACT,@RCSR
2027  004542  001421                          BEQ     $72
2028
2029  004544                                                            IF #RCVRDONE SETIN @RCSR THEN
2030  004544  032777  000200  174506          BIT     #RCVRDONE,@RCSR
2031  004552  001414                          BEQ     $73
2032  004554                                                              IF #RCVRACT SETIN @RCSR THEN
2033  004554  032777  004000  174476          BIT     #RCVRACT,@RCSR
2034  004562  001410                          BEQ     $74
2035                                                                    ;RCVRDONE AND RCVRACT
2036                                                                      ;BOTH SET
2037                                                                    ; CAN NOT LEAVE WITH MAINT SET
2038  004564                                                          LET   @TCSR := @TCSR CLR.BY #MAINT
2039  004564  042777  000004  174472          BIC     #MAINT,@TCSR
2040  004572                                                              ERRHRD 76, DONEACT
2041  004572  104076                          ERROR   76
2042                                                                    ;NO USE CONTINUING
2043  004574                                                              EXIT TST
2044  004574  012767  000001  174356          MOV     #1,$TIMES
2045  004602  000432                          BR      TST15           ;;;EXIT THIS TEST
2046  004604                                                            ENDIF
2047  004604                          $74:
2048  004604                                                          ENDIF
2049  004604                          $73:
2050  004604                                                        ENDDO
2051  004604  000753                          BR      $71
2052  004606                          $72:
2053
2054                                                                  ;RCVRACT = 0 NOW.
2055  004606                                                          IF #RCVRDONE NOTSETIN @RCSR THEN
2056  004606  032777  000200  174444          BIT     #RCVRDONE,@RCSR
2057  004614  001010                          BNE     $75
2058                                                                    ;RCVRDONE DID NOT SET IN RCSR
2059                                                                    ; CAN NOT LEAVE WITH MAINT SET
```

```
2060  004616
2061  004616  042777  000004  174440          BIC     #MAINT,@TCSR
2062  004624                                                                  LET    @TCSR := @TCSR CLR.BY #MAINT
2063  004624  104077                          ERROR   77                          ERRHRD 77,,DIDNOT
2064  004626                                                                    EXIT TEST
2065  004626  012767  000001  174324          MOV     #1,$TIMES
2066  004634  000415                          BR      TST15           ;;;EXIT THIS TEST
2067  004636                                                                  ENDIF
2068  004636                          $75:
2069                                                          ;TEST THAT READING THE RECEIVER
2070                                                          ;BUFFER CLEARS RCVRDONE
2071
2072
2073                                                                      ;READ CHAR.
2074  004636                                                            LET R0 := @RBUF
2075  004636  01770u  174420                  MOV     @RBUF,R0
2076
2077  004642                                                            IF #RCVRDONE SETIN @RCSR THEN
2078  004642  032777  000200  174410          BIT     #RCVRDONE,@RCSR
2079  004650  001404                          BEQ     $76
2080                                                                      ;RCVRDONE DID NOT CLEAR IN RCSR
2081                                                                      ; CAN NOT LEAVE WITH MAINT SET
2082  004652                                                            LET    @TCSR := @TCSR CLR.BY #MAINT
2083  004652  042     000004  174404          BIC     #MAINT,@TCSR
2084  004660                                                                  ERRHRD 100,,DIDNOT
2085  004660  104100                          ERROR   100
2086  004662                                                            ENDIF
2087  004662                          $76:
2088
2089  004662                                                                    EXIT
2090  004662  000402                                  BR      TST15          ;;;EXIT THIS TEST
2091  004664  070000                                  MAX:70000
2092  004666  000000                                  COUNT:  0
2093
2094  004670                                                                  ENDTST
2095
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 15       TEST THE OVERRUN BIT - RBUF 14
                                ;;*************************************************`         .****
004670  000004                  TST15:  SCOPE
004672  012767  000010  174260          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
004700  012767  000015  174272          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
004704
004706                                                  IF #ERRBITS SETIN $USWR ANDB APTCON EQ #FALSE TH
004706  032767  100000  174304          BIT     #ERRBITS,$USWR
004714  001416                          BEQ     $77
004716  126727  004344  000000          CMPB    APTCON,#FALSE
004724  001012                          BNE     $77
004726                                                      IF #MAINTJUMP SETIN $USWR THEN
004726  032767  040000  174264          BIT     #MAINTJUMP,$USWR
004734  001401                          BEQ     $100
004736                                                          ;; NULL ---EXECUTE TEST
                                                            ELSE
004736  000404                          BR      $101
004740                          $100:
004740                                                          EXIT TEST
004740  012767  000001  174212          MOV     #1,$TIMES
004746  000567                          BR      TST16           ;;;EXIT THIS TEST
004750                                                      ENDIF
004750                          $101:
004750                                                  ELSE
004750  000404                          BR      $102
004752                          $77:
004752                                                      EXIT TEST
004752  012767  000001  174200          MOV     #1,$TIMES
004760  000562                          BR      TST16           ;;;EXIT THIS TEST
004762                                                      ENDIF
004762                          $102:
004762                                                  LET @TCSR := @TCSR SET.BY #MAINT
004762  052777  000004  174274          BIS     #MAINT,@TCSR
004766
004770
004770                                                  BGNSUB
004770  012767  004776  174112          MOV     #64$,$LPERR
                                            ;OUTPUT 2 CHARACTERS WITH
                                            ;AMPLE DELAYS BETWEEN FOR RECEPTION.
                                            ;THIS SHOULD AN CAUSE OVERRUN ERROR.
004774
004776                                                      ;OUTPUT 1 CHARACTER
                                                        LET @TBUF :B= #0
004776  105077  174266                  CLRB    @TBUF
005000                                                      ;GO AWAY FOR 500 M SEC
005002                                                  WAITUS 50
005002  010546                          MOV     R5,-(SP)
005004  012745  000050                  MOV     #50,-(R5)
005010  004767  003724                  JSR     PC,WAIT
005014  012605                          MOV     (SP)+,R5
005016
```

```
                                                        ;OUTPUT 2ND CHARACTER
005016                                                  LET @TBUF :B= #0
005016  105077  174246          CLRB    @TBUF
                                                        ;LET OVERRUN HAPPEN
005022                                                  WAITUS 50
005022  010546                  MOV     R5,-(SP)
005024  012745  000050          MOV     #50,-(R5)
005030  004767  003704          JSR     PC,WAIT
005034  012605                  MOV     (SP)+,R5


                                                        ;READ BUFFER AND ERROR BITS
005036                                                  LET R4 := @RBUF
005036  017704  174220          MOV     @RBUF,R4

                                                        ;IT DIDN'T SET
005042                                                  IF #ORERR NOTSETIN R4 THEN
005042  032704  040000          BIT     #ORERR,R4
005046  001010                  BNE     $103

                                                        ;ORERR DID NOT SET IN RBUF
                                                        ; CAN NOT LEAVE WITH MAINT SET
005050                                                  LET   @TCSR := @TCSR CLR.BY #MAINT
005050  042777  000004  174206  BIC     #MAINT,@TCSR
005056                                                  ERRHRD 101,,DIDNOT
005056  104101                  ERROR   101


                                                        ;NO USE COMPOUNDING ERRORS
                                                        EXIT TST
005060
005060  012767  000001  174072  MOV     #1,$TIMES
005066  000517                  BR      TST16           ;;;EXIT THIS TEST
005070                                                  ENDIF
005070                  $103:
005070                                                  ENDSUB

                                ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
005070                                                  BGNSUB
005070  012767  005076  174012  MOV     #64$,$LPERR
005076                                                      IF #ERROR NOTSETIN R4 THEN
005076  032704  100000          BIT     #ERROR,R4
005102  001010                  BNE     $104


                                                        ;ERROR DID NOT SET IN RBUF
                                                        ; CAN NOT LEAVE WITH MAINT SET
005104                                                  LET   @TCSR := @TCSR CLR.BY #MAINT
005104  042777  000004  174152  BIC     #MAINT,@TCSR
005112                                                      ERRHRD 102,,DIDNOT
005112  104102                  ERROR   102


                                                        ;-WHEN ORERR SET.
                                                        ;GET OUT NOW.
005114                                                  EXIT TST
005114  012767  000001  174036  MOV     #1,$TIMES
005122  000501                  BR      TST16           ;;;EXIT THIS TEST
005124                                                  ENDIF
005124                  $104:
005124                                                  ENDSUB
```

N04


```
005124                                                          BGNSUB
005124 012767 005132 173756             MOV     #64$,$LPERR
                                            ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.

005132                                                              IF #ORERR NOTSETIN @RBUF THEN
005132 032777 040000 174122             BIT     #ORERR,@RBUF
005140 001010                           BNE     $105

                                                                    ;READING RBUF CLEARED ORERR.
                                                                    ; CAN NOT LEAVE WITH MAINT SET
005142                                                          LET   @TCSR := @TCSR CLR.BY #MAINT
005142 042777 000004 174114             BIC     #MAINT,@TCSR
005150 104103                           ERROR   103                 ERRHRD 103,ITCLRED

                                                                    ;SKIP REST OF TEST
                                                                    EXIT TEST
005152
005152 012767 000001 174000             MOV     #1,$TIMES
005160 000462                           BR      TST16           ;;;EXIT THIS TEST
005162                                                              ENDIF
005162                          $105:
005162                                                          ENDSUB

005162                                                          BGNSUB
005162 012767 005170 173720             MOV     #64$,$LPERR
                                            ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED

                                                                    ;SEND A CHARACTER AROUND.
005170                                                          LET @TBUF :B= #0
005170 105077 174074                    CLRB    @TBUF
                                                                    ;LET IT CIRCULATE
005174                                                          WAITUS 50
005174 010546                           MOV     R5,-(SP)
005176 012745 000050                    MOV     #50,-(R5)
005202 004767 003532                    JSR     PC,WAIT
005206 012605                           MOV     (SP)+,R5

005210                                                              IF #ORERR SETIN @RBUF THEN
005210 032777 040000 174044             BIT     #ORERR,@RBUF
005216 001410                           BEQ     $106

                                                                    ;ORERR DID NOT CLEAR IN RBUF
                                                                    ; CAN NOT LEAVE WITH MAINT SET
005220                                                          LET   @TCSR := @TCSR CLR.BY #MAINT
005220 042777 000004 174036             BIC     #MAINT,@TCSR
005226                                                              ERRHRD 104,,DIDNOT
005226 104104                           ERROR   104

                                                                    ;-AFTER RECEIVING ANOTHER CHAR
                                                                    ;SKIP AROUND REST
                                                                    EXIT TST
005230
005230 012767 000001 173722             MOV     #1,$TIMES
005236 000433                           BR      TST16           ;;;EXIT THIS TEST
005240                                                              ENDIF
005240                          $106:

005240                                                              IF #ERROR SETIN @RBUF THEN
005240 032777 100000 174014             BIT     #ERROR,@RBUF
```

```
2264  005246  001404                    BEQ     $107
2265                                                          ;ERROR DID NOT CLEAR IN RBUF
2266                                                          ; CA: NOT LEAVE WITH MAINT SET
2267  005250                                                LET     @TCSR := @TCSR CLR.BY #MAINT
2268  005250  042777  000004  174006    BIC     #MAINT,@TCSR
2269  005256                                                    ERRHRD 105,,DIDNOT
2270  005256  104105                    ERROR   105
2271                                                          ENDIF
2272  005260
2273  005260                        $107:                   ENDSUB
2274  005260                                                EXIT
2275  005260
2276  005260  000422                    BR      TST16       ;;;EXIT THIS TEST
2277  005262  042522  042101  047111                        ITCLRED:        .ASCIZ  /READING RBUF CLEARED OV
2278  005270  020107  041122  043125
2279  005276  041440  042514  051101
2280  005304  042105  047440  042526
2281  005312  051122  047125  042440
2282  005320  051122  051117  000056
2283                                                          .EVEN
2284  005326                                                ENDTST
2285
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 16        PROGRAMMABLE BAUD RATE TEST
                                ;*              TEST AT ALL SPEEDS AVAILABLE
                                ;*              A COMPARISON WILL BE MADE TO SEE
                                ;*              IF NEW TIME IS LESS THAN PREVIOUS.
                                ;;**********************************************************************
005326  000004                  TST16:  SCOPE
005330  012767  000010  173622          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005336  012767  000016  173634          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

                                        ;;NOTE: IF PROGRAMMABLE BAUDE RATE TEST IS TO BE RUN ON THE APT CONSOLE
                                        ;;      THE OPERATOR MUST MAKE CERTAIN THAT LOCATION $USWR IS
                                        ;;      SET WITH #16 IN BITS <11:8>(9600 BAUD).


005344                                                  IF #PBR SETIN $USWR AND #MAINTJUMP SETIN $USWR T
005344  032767  000200  173646          BIT     #PBR,$USWR
005352  001416                          BEQ     $110
005354  032767  040000  173636          BIT     #MAINTJUMP,$USWR
005362  001412                          BEQ     $110

005364                                              IFB APTCON EQ #FALSE THEN
005364  126727  003676  000000          CMPB    APTCON,#FALSE
005372  001001                          BNE     $111

                                                ;;;  NULL --- EXECUTE TEST
                                            ELSE
005374
005374  000404                          BR      $112
005376                          $111:
005376                                                  EXIT TST
005376  012767  000001  173554          MOV     #1,$TIMES
005404  000565                          BR      TST17           ;,;EXIT THIS TEST
                                            ENDIF
005406                          $112:
005406
005406  000404                          BR      $113
005410                          $110:
005410                                          EXIT TEST ; IF EITHER OF CASES FAIL
005410  012767  000001  173542          MOV     #1,$TIMES
005416  000560                          BR      TST17           ;;;EXIT THIS TEST
                                            ENDIF
005420
005420                          $113:

005420                                          LET ERRCHK := #0        ; CLEAR ERROR WORD
005420  005067  002710                  CLR     ERRCHK
005424                                          LET OLD := #-1
005424  012767  177777  000322          MOV     #-1,OLD
005432                                          LET OLD+2 := #-1
005432  012767  177777  000316          MOV     #-1,OLD+2
005440                                          LET @TCSR := @TCSR SET.BY #MAINT
005440  052777  000004  173616          BIS     #MAINT,@TCSR

                                                ;EACH BAUD RATE
005446                                  INCR R3 FROM #0 TO #15. BY #1
005446  005003                  CLR     R3
005450  000401                  BR      $114
005452                          $115:
```

```
2342  005452  005203                          INC     R3
2343  005454                          $114:
2344  005454  020327  000017                  CMP     R3,#15.
2345  005460  003062                          BGT     $116
2346  005462
2347  005462  017700  173574                  MOV     @RBUF,R0
2348
2349  005466
2350  005466  116377  005676  173572          MOVB    RATES(R3),@TCSRHI
2351
2352  005474
2353  005474  005002                          CLR     BIT
2354
2355  005476
2356  005476  005077  173566                  CLR     @TBUF
2357
2358  005502
2359  005502  005067  000242                  CLR     NEW
2360  005506
2361  005506  005067  000240                  CLR     NEW+2
2362  005512
2363  005512                          $117:
2364  005512  005702                          TST     BIT
2365  005514  001014                          BNE     $120
2366  005516
2367  005516  032777  000200  173534          BIT     #RCVRDONE,@RCSR
2368  005524  001403                          BEQ     $121
2369
2370  005526
2371  005526  012702  000001                  MOV     #1,BIT
2372  005532
2373  005532  000404                          BR      $122
2374  005534                          $121:
2375
2376  005534
2377  005534  005267  000210                  INC     NEW
2378  005540
2379  005540  005567  000206                  ADC     NEW+2
2380  005544
2381  005544                          $122:
2382
2383  005544
2384  005544  000762                          BR      $117
2385  005546                          $120:
2386
2387  005546
2388  005546  026767  000200  000202          CMP     NEW+2,OLD+2
2389  005554  103001                          BHIS    $'23
2390
2391  005556
2392  005556  000414                          BR      $124
2393  005560                          $123:
2394
2395  005560
2396  005560  026767  000166  000170          CMP     NEW+2,OLD+2
2397  005566  001005                          BNE     $125
```

```
LET R0 := @RBUF

    ;CHANGE BAUDE RATE
LET @TCSRHI :B= RATES(R3)

    ;FLAG
LET BIT := #0

    ;OUTPUT THE CHARACTER
LET @TBUF := #0

    ;INITIALIZE COUNTER
LET NEW := #0

LET NEW+2 := #0

WHILE BIT EQ #0 DO


    IF #RCVRDONE SETIN @RCSR THEN


    ;DONE - ITS READY
        LET BIT := #1

    ELSE

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

        LET NEW+2 := NEW+2 + CARRY

    ENDIF

    ;SIGNALS DONE
ENDDO


IF NEW+2 LO OLD+2 THEN


    ; OK
ELSE

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

```
2398  005570  026767  000154  000156          CMP     NEW,OLD
2399  005576  103001                          BHIS    $125
2400
2401  005600                                                                  ;OK
2402  005600  000403                          BR      $126                        ELSE
2403  005602                          $125:                                       ;NEW+2 > OLD+2 OR
2404                                                                              ;(NEW+2 = OLD+2 AND
2405                                                                              ; NEW >= OLD)
2406                                                                              ;BAUD RATE DIDN'T CHANGE
2407
2408  005602                                                                  LET ERRCHK := #BIT2 ; SET ERROR INDICATOR
2409  005602  012767  000004  002524          MOV     #BIT2,ERRCHK
2410  005610                                                                      ENDIF
2411  005610                          $126:
2412  005610                                                                  ENDIF
2413  005610                          $124:
2414                                                                              ;UPDATE OLD TIME
2415  005610                                                                  LET OLD := NEW
2416  005610  016767  000134  000136          MOV     NEW,OLD
2417  005616                                                                  LET OLD+2 := NEW+2
2418  005616  016767  000130  000132          MOV     NEW+2,OLD+2
2419
2420  005624                                                              ENDINC  ;BAUD RATE
2421  005624  000712                          BR      $115
2422  005626                          $116:
2423  005626                                                          LET R3 :B= $USWR+1 AND #17      ; PUT BAUD BACK
2424  005626  116703  173367                  MOVB    $USWR+1,R3
2425  005632  110346                          MOVB    R3,-(SP)
2426  005634  142716  000017                  BICB    #17,(SP)
2427  005640  142603                          BICB    (SP)+,R3
2428  005642                                                          LET R3 := R3 CLR.BY #177400
2429  005642  042703  177400                  BIC     #177400,R3
2430  005646                                                          LET @TCSRHI :B= RATES(R3)       ; LIKE HE WANTED IT
2431  005646  116377  005676  173412          MOVB    RATES(R3),@TCSRHI
2432
2433                                                                          ; CAN NOT LEAVE WITH MAINT SET
2434  005654                                                              LET   @TCSR := @TCSR CLR.BY #MAINT
2435  005654  042777  000004  173402          BIC     #MAINT,@TCSR
2436  005662                                                              IF #BIT2 SETIN ERRCHK THEN
2437  005662  032767  000004  002444          BIT     #BIT2,ERRCHK
2438  005670  001401                          BEQ     $127
2439                                                                          ; REPORT DEFERED ERROR
2440  005672                                                              ERRHRD      126
2441  005672  104126                          ERROR   126
2442  005674                                                              ENDIF
2443  005674                          $127:
2444  005674                                                              EXIT    ;SKIP TABLE
2445  005674  000431                          BR      TST17           ;;;EXIT THIS TEST
2446
2447  005676                          RATES:  ;A TABLE OF THE ACTUAL BYTES TO MOVE INTO THE
2448                                  ;UPPER BYTE OF XCSR FOR EACH BAUD RATE
2449                                  ;** NOTE::      THE VALUE INDICATED IN THE COLUMN 'OFFSET
2450                                  ;**             INTO TABLE' CAN BE PLACED INTO BITS<11:8>
2451                                  ;**             OF LOCATION 'SUSWR' TO CAUSE THE CORROSPONDING
2452                                  ;**             BAUD TO BE SELECTED IN THE DLV11-F UPON
2453                                  ;**             COMPLETION OF THIS TEST.
```

```
2454
2455                                                        ;       BAUD    OFFSET INTO TABLE
2456  005676      010                   R0050:  .BYTE   010 ;       50      0
2457  005677      030                   R0070:  .BYTE   030 ;       70      1
2458  005700      050                   R0110:  .BYTE   050 ;       110     2
2459  005701      070                   R0135:  .BYTE   070 ;       135     3
2460  005702      110                   R0150:  .BYTE   110 ;       150     4
2461  005703      130                   R0300:  .BYTE   130 ;       300     5
2462  005704      150                   R0600:  .BYTE   150 ;       600     6
2463  005705      170                   R1200:  .BYTE   170 ;       1200    7
2464  005706      210                   R1800:  .BYTE   210 ;       1800    10
2465  005707      230                   R2000:  .BYTE   230 ;       2000    11
2466  005710      250                   R2400:  .BYTE   250 ;       2400    12
2467  005711      270                   R3600:  .BYTE   270 ;       3600    13
2468  005712      310                   R4800:  .BYTE   310 ;       4800    14
2469  005713      330                   R7200:  .BYTE   330 ;       7200    15
2470  005714      350                   R9600:  .BYTE   350 ;       9600    16
2471  005715      370                   R10000: .BYTE   370 ;       19200   17
2472
2473  005716  040502  042125  051040  BAUDRATE:       .ASCIZ  /BAUD RATE DIDN'T CHANGE./
2474  005724  052101  020105  044504
2475  005732  047104  052047  041440
2476  005740  040510  043516  027105
2477  005746    000
2478          005750                .EVEN
2479  005750  000000  000000          NEW:    0,0
2480  005754  000000  000000          OLD:    0,0
2481  005760                                                      ENDTST
2482
2483
2484
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 17        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
                                ;;*****************************************************************
005760  000004                  TST17:  SCOPE
005762  012767  000010  173170          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
005770  012767  000017  173202          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
                                                                LET INTFLAG := #0
005776
005776  005067  003024                  CLR     INTFLAG

                                                                        ;GET VECTOR ADDRESS
006002                                                          LET R3 := DLVEC
006002  016703  173250                  MOV     DLVEC,R3
                                                                        ;FOR THE TRANSMITTER
006006                                                          LET R3 := R3 + #4
006006  062703  000004                  ADD     #4,R3
                                                                        ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
006012                                                          SETVEC R3, #INTSRV, #PR7
006012  010146                  MOV     R1,-(SP)
006014  010301                  MOV     R3,R1
006016  012721  011020          MOV     #INTSRV,(R1)+
006022  012711  000340          MOV     #PR7,(R1)
006026  012601                  MOV     (SP)+,R1
006030                                                  BGNSUB
006030  012767  006036  173052          MOV     #65$,$LPERR
                                                                ;; MAKE SURE THAT TRANSMITTER READY IS SET
006036                                                          CALL TIMER IN <#50,#XMITRDY,TCSR,#SET>
006036  010546                  MOV     R5,-(SP)
006040  012745  177777          MOV     #SET,-(R5)
006044  016745  173214          MOV     TCSR,-(R5)
006050  012745  000200          MOV     #XMITRDY,-(R5)
006054  012745  000050          MOV     #50,-(R5)
006060  004767  002376          JSR     PC,TIMER
006064  012605                  MOV     (SP)+,R5

                                                                        ;CLEAR INTERRUPT ENABLE
006066                                                          LET @TCSR := @TCSR CLR.BY #XMITIE
006066  042777  000100  173170          BIC     #XMITIE,@TCSR

                                                                        ;SET IT TO 0
006074  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
006100  012746  006106          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
006104  000002                  RTI                     ;;POP NEW PC AND PS
006106                  65$:

                                                                        ;NOW SET I.E. BIT
006106                                                          LET @TCSR := @TCSR SET.BY #XMITIE
006106  052777  000100  173150          BIS     #XMITIE,@TCSR

                                                                        ;LET INTERRUPT HAVE TIME TO OCCUR
```

```
2541  006114                                                          WAITUS 10
2542  006114  010546                  MOV     R5,-(SP)
2543  006116  012745  000010          MOV     #10,-(R5)
2544  006122  004767  002612          JSR     PC,WAIT
2545  006126  012605                  MOV     (SP)+,R5
2546
2547                                                                      ;DID EXACTLY 1 INTERRUPT OCCUR
2548  006130                                                          IF INTFLAG NE #1 THEN
2549  006130  026727  002672  000001  CMP     INTFLAG,#1
2550  006136  001406                  BEQ     $130
2551                                                                      ;NO - WAS IT 0 OR MORE THAN ONCE
2552  006140                                                              IF INTFLAG EQ #0 THEN
2553  006140  005767  002662          TST     INTFLAG
2554  006144  001002                  BNE     $131
2555                                                                      ;TRANSMITTER DID NOT INTERRUPT IN TIME
2556  006146                                                                  ERRHRD 106,,DIDNOT
2557  006146  104106                  ERROR   106
2558  006150                                                              ELSE
2559  006150  000401                  BR      $132
2560  006152                  $131:
2561                                                                      ;TWICE
2562                                                                      ;TRANSMITTER INTERRUPTED TWICE
2563  006152                                                                  ERRHRD 107,,TWICE
2564  006152  104107                  ERROR   107
2565  006154                                                              ENDIF
2566  006154                  $132:
2567  006154                                                          ENDIF
2568  006154                  $130:
2569  006154                                                      ENDSUB
2570                                                          ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
2571  006154                                                      BGNSUB
2572  006154  012767  006162  172726  MOV     #64$,$LPERR
2573                                                                      ;CLEAR 'INTERRUPT OCCURED' FLAG
2574  006162                                                          LET INTFLAG := #0
2575  006162  005067  002640          CLR     INTFLAG
2576                                                                      ;CLEAR INTERRUPT ENABLE
2577  006166                                                          LET @TCSR := @TCSR CLR.BY #XMITIE
2578  006166  042777  000100  173070  BIC     #XMITIE,@TCSR
2579                                                                      ;NO INTERRUPTS SHOULD OCCUR.
2580  006174  012746  000000          MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
2581  006200  012746  006206          MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
2582  006204  000002                  RTI                     ;;POP NEW PC AND PS
2583  006206                  65$:
2584                                                                      ;DARE IT TO HAPPEN
2585  006206                                                          WAITUS 10
2586  006206  010546                  MOV     R5,-(SP)
2587  006210  012745  000010          MOV     #10,-(R5)
2588  006214  004767  002520          JSR     PC,WAIT
2589  006220  012605                  MOV     (SP)+,R5
2590  006222
2591  006222  005767  002600          TST     INTFLAG         IF INTFLAG NE #0 THEN
2592  006226  001401                  BEQ     $133
2593                                                                      ;INTERRUPT OCCURED WITH I E CLEARED
2594  006230                                                              ERRHRD 110,NOTENAB
2595  006230  104110                  ERROR   110
2596  006232                                                          ENDIF
```

```
2597  006232                          $133:
2598  006232                                                          BRESET
2599  006232
2599  006232  000005                          RESET
2600  006234                                                        ENDSUB
2601                                                                        ;RESTORE VECTOR AREA
2602  006234                                                        CLRVEC R3
2603  006234  010146                          MOV     R1,-(SP)      ;;PUSH R1 ON STACK
2604  006236  010246                          MOV     R2,-(SP)      ;;PUSH R2 ON STACK
2605  006240  012701  000003                  MOV     #R3,R1
2606  006244  010102                          MOV     R1,R2
2607  006246  062702  000002                  ADD     #2,R2
2608  006252  010221                          MOV     R2,(R1)+
2609  006254  005011                          CLR     (R1)
2610  006256  012602                          MOV     (SP)+,R2      ;;POP STACK INTO R2
2611  006260  012601                          MOV     (SP)+,R1      ;;POP STACK INTO R1
2612
2613  006262                                                          ENDTST
2614
2615
2616
2617
2618
2619
```

```
                                ;;**********************************************************************
                                ;;**********************************************************************
                                ;*TEST 20        RECEIVER INTERRUPT LOGIC TEST
                                ;*              THIS TEST COVERS ALL OF THE RECEIVER
                                ;*              SIDE OF THE INTERRUPT LOGIC IN
                                ;*              CHARACTER MODE.
                                ;;**********************************************************************
006262  000004                  TST2F:  SCOPE
006264  012767  000010  172666          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
006272  012767  000020  172700          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
006300                                                          IF #MAINTJUMP NOTSETIN $USWR ORB APTCON EQ #TRUE
006300  032767  040000  172712          BIT     #MAINTJUMP,$USWR
006306  001404                          BEQ     $134
006310  126727  002752  000001          CMPB    APTCON,#TRUE
006316  001002                          BNE     $135
006320                          $134:
006320  000167  000422                  JMP     TST21                   ;       EXIT TEST
006324                                                          ENDIF
006324                          $135:
                                                                        ;CLEAR INTERRUPT OCCURED FLAG
                                                                        ;SET UP RECEIVER INTER.VECTOR
006324                                                          SETVEC DLVEC,#INTSRV,#PR7
006324  010146                          MOV     R1,-(SP)
006326  016701  172724                  MOV     DLVEC,R1
006332  012721  011020                  MOV     #INTSRV,(R1)+
006336  012711  000340                  MOV     #PR7,(R1)
006342  012601                          MOV     (SP)+,R1
                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
                                                                BGNSUB
006344
006344  012767  006352  172536          MOV     #64$,$LPERR
006352                                                          LET INTFLAG := #0
006352  005067  002450                  CLR     INTFLAG
                                                                        ;SET MAINT. BIT
006356                                                          LET @TCSR := @TCSR SET.BY #MAINT
006356  052777  000004  172700          BIS     #MAINT,@TCSR
                                                                        ;CLEAR INTERRUPTS
006364                                                              LET @RCSR := @RCSR CLR.BY #RCVRIE
006364  042777  000100  172666          BIC     #RCVRIE,@RCSR
                                                                        ;CHANGE PRIORITY
                                                                        ;..TO 0
006372  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
006376  012746  006404                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
006402  000002                          RTI                     ;;POP NEW PC AND PS
006404                          65$:
                                                                        ;SEND A CHARACTER
006404                                                              LET @TBUF :B= #0
006404  105077  172660                  CLRB    @TBUF
                                                                        ;WAIT A MAXIMUM
                                                                ;OF 50 MSEC FOR
                                                                ;XMIT RDY TO SET IN TCSR
006410                                                          CALL TIMER IN <#50,#XMITRDY,TCSR,#SET>
006410  010546                          MOV     R5,-(SP)
006412  012745  177777                  MOV     #SET,-(R5)
006416  016745  172642                  MOV     TCSR,-(R5)
```

KOL


```
006422  012745  000200          MOV     #XMITRDY,-(R5)
006426  012745  000050          MOV     #50,-(R5)
006432  004767  002024          JSR     PC,TIMER
006436  012605                  MOV     (SP)+,R5
006440
006440  052777  000100  172612  BIS     #RCVRIE,@RCSR
006446
006446  010546                  MOV     R5,-(SP)
006450  012745  000010          MOV     #10,-(R5)
006454  004767  002260          JSR     PC,WAIT
006460  012605                  MOV     (SP)+,R5
006462
006462  026727  002340  000001  CMP     INTFLAG,#1
006470  001411                  BEQ     $136
006472
006472  042777  000004  172564  BIC     #MAINT,@TCSR
006500
006500  005767  002322          TST     INTFLAG
006504  001002                  BNE     $137
006506
006506  104111                  ERROR   111
006510
006510  000401                  BR      $140
006512          $137:
006512
006512  104112                  ERROR   112
006514
006514          $140:
006514          $136:
006514
006514  042777  000004  172542  BIC     #MAINT,@TCSR
006522
006522
006522  012767  006530  172360  MOV     #64$,$LPERR
006530
006530  052777  000004  172526  BIS     #MAINT,@TCSR
```

```
    ;SET INTERRUPT ENABLE
LET @RCSR := @RCSR SET.BY #RCVRIE

    ;LET IT COME IN.
WAITUS 10



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


    ;NONE OCCURED
    ; CAN NOT LEAVE WITH MAINT SET
LET    @TCSR := @TCSR CLR.BY #MAINT

    IF INTFLAG EQ #0 THEN


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

    ;TWICE OR MORE
    ELSE

        ;RECEIVER INTERRUPTED TWICE
            ERRHRD 112,,TWICE

        ENDIF

    ENDIF

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

ENDSUB


;INTERRUPT WITHOUT I E SET.
BGNSUB

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

        ;CLEAR INTERRUPT FLAG
```

```
2732  006536
2733  006536  005067  002264          CLR     INTFLAG
2734
2735  006542
2736  006542  042777  000100  172510  BIC     #RCVRIE,@RCSR
2737
2738  006550
2739  006550  105077  172514          CLRB    @TBUF
2740
2741  006554
2742  006554  010546                  MOV     R5,-(SP)
2743  006556  012745  000010          MOV     #10,-(R5)
2744  006562  004767  002152          JSR     PC,WAIT
2745  006566  012605                  MOV     (SP)+,R5
2746  006570
2747  006570  005767  002232          TST     INTFLAG
2748  006574  001404                  BEQ     $141
2749
2750  006576
2751  006576  042777  000004  172460  BIC     #MAINT,@TCSR
2752  006604  104113                  ERROR   113
2753  006606
2754  006606                  $141:
2755  006606
2756  006606  042777  000004  172450  BIC     #MAINT,@TCSR
2757  006614
2758
2759                                  ; INTERRUPT WITH I E SET BUT AT PRIORITY OF 7
2760
2761  006614                                                  BGNSUB
2762  006614  012767  006622  172266          MOV     #64$,$LPERR
2763  006622                                                      LET @TCSR := @TCSR SET.BY #MAINT
2764  006622  052777  000004  172434          BIS     #MAINT,@TCSR
2765  006630                                                      LET INTFLAG := #0
2766  006630  005067  002172                  CLR     INTFLAG
2767                                                              ;DON'T LET THEM IN
2768  006634  012746  000340                  MOV     #PR7,-(SP)      ;;PUT NEW PS ON STACK
2769  006640  012746  006646                  MOV     #65$,-(SP)      ;;PUT NEW PC ON  STACK
2770  006644  000002                          RTI                     ;;POP NEW PC AND PS
2771  006646                  65$:
2772  006646                                                      LET @RCSR := @RCSR SET.BY #RCVRIE
2773  006646  052777  000100  172404          BIS     #RCVRIE,@RCSR
2774  006654                                                      LET @TBUF :B= #0
2775  006654  105077  172410                  CLRB    @TBUF
2776                                                              ;DARE IT
2777  006660                                                      WAITUS 10
2778  006660  010546                  MOV     R5,-(SP)
2779  006662  012745  000010          MOV     #10,-(R5)
2780  006666  004767  002046          JSR     PC,WAIT
2781  006672  012605                  MOV     (SP)+,R5
2782
2783  006674                                                      ;DID IT HAPPEN?
2784  006674  005767  002126          TST     INTFLAG         IF INTFLAG NE #0 THEN
2785  006700  001401                  BEQ     $142
2786                                                                  ;INTERRUPT OCCURED AT PRIORITY 7
2787  006702                                                          ERRHRD 115,NOTENAB
```

```
                                    LET INTFLAG := #0

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

                                        ; SEND A CHARACTER
                                    LET @TBUF :B= #0

                                        ; DARE IT
                                    WAITUS 10



                                    IF INTFLAG NE #0 THEN

                                        ; EXIT WITH MAINT CLEAR
                                        LET @TCSR := @TCSR CLR.BY #MAINT
                                ; INTERRUPT WITH I E CLEAR
                                    ENDIF

                                        LET @TCSR := @TCSR CLR.BY #MAINT

                                ENDSUB
```

```
2788  006702  104115                    ERROR   115
2789  006704                                                          ENDIF
2790  006704                    $142:
2791                                                                  ;CLEAR THE WORLD
2792  006704                                                          BRESET
2793  006704  000005                    RESET
2794
2795                                                                  ;RESET MAINT. BIT.
2796  006706                                                          LET @TCSR := @TCSR CLR.BY #MAINT
2797  006706  042777  000004  172350    BIC     #MAINT,@TCSR
2798  006714                                                          ENDSUB
2799
2800  006714                                                          LET R4 := @DLVEC
2801  006714  017704  172336            MOV     @DLVEC,R4
2802  006720                                                          CLRVEC R4
2803  006720  010146                    MOV     R1,-(SP)        ;;PUSH R1 ON STACK
2804  006722  010246                    MOV     R2,-(SP)        ;;PUSH R2 ON STACK
2805  006724  012701  000004            MOV     #R4,R1
2806  006730  010102                    MOV     R1,R2
2807  006732  062702  000002            ADD     #2,R2
2808  006736  010221                    MOV     R2,(R1)+
2809  006740  005011                    CLR     (R1)
2810  006742  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
2811  006744  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
2812  006746                                                          ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 21        TEST ACTUAL DATA TRANSFERRED
                                ;*               NON-INTERRUPT MAINTENANCE BIT SET
                                ;;*****************************************************************
006746  000004                  TST21:  SCOPE
006750  012767  000001  172202          MOV     #1,$TIMES       ;;DO 1 ITERATION
006756  012767  000021  172214          MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        IF #MAINTJUMP NOTSETIN $USWR ORB APTCON EQ #TRUE
006764  032767  040000  172226          BIT     #MAINTJUMP,$USWR
006772  001404                        BEQ     $143
006774  126727  002266  000001        CMPB    APTCON,#TRUE
007002  001004                        BNE     $144
007004                          $143:
                                                                            EXIT TEST
007004  012767  000001  172146          MOV     #1,$TIMES
007012  000520                        BR      TST22           ;;;EXIT THIS TEST
                                                                        ENDIF
007014                          $144:
                                                                        LET ERRCHK := #0
007014  005067  001314                  CLR     ERRCHK
                                                                                ;SET MAINT. BIT
                                                                        LET @TCSR := @TCSR SET.BY #MAINT
007020  052777  000004  172236          BIS     #MAINT,@TCSR
                                                                                ;CHANGE PRIORITY
                                                                                ;..TO 0
007026  012746  000000                  MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
007032  012746  007040                  MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
007036  000002                        RTI                     ;;POP NEW PC AND PS
007040                          64$:
                                                                                ;GET DATA MASK.
                                                                        CALL DATLNG OUT <R1>
007040  162705  000002                  SUB     #1*2,R5
007044  004767  001570                  JSR     PC,DATLNG
007050  012501                        MOV     (R5)+,R1
                                                                                ;ALL BINARY CHAR.
                                                                        INCR R2 FROM #0 TO #377 BY #1
007052  005002                        CLR     R2
007054  000401                        BR      $145
007056                          $146:
007056  005202                        INC     R2
007060                          $145:
007060  020227  000377                  CMP     R2,#377
007064  003056                        BGT     $147

                                                                        ;TRANSMIT CHAR IN R2

007066                                                                  CALL TIMER IN <#50,#XMITRDY,TCSR,#SET>
007066  010546                  MOV     R5,-(SP)
007070  012745  177777          MOV     #SET,-(R5)
007074  016745  172164          MOV     TCSR,-(R5)
007100  012745  000200          MOV     #XMITRDY,-(R5)
```

```
2869  007104  012745  000050            MOV     #50,-(R5)
2870  007110  004767  001346            JSR     PC,@TIMER
2871  007114  012605                    MOV     (SP)+,R5
2872  007116
2873  007116  103003                    BCC     $150
2874  007120
2875  007120  052767  000010  001206    BIS     #BIT3,ERRCHK
2876  007126
2877  007126                    $150:
2878
2879
2880  007126
2881  007126  110277  172136            MOVB    R2,@TBUF
2882
2883  007132
2884  007132  010546                    MOV     R5,-(SP)
2885  007134  012745  177777            MOV     #SET,-(R5)
2886  007140  016745  172114            MOV     RCSR,-(R5)
2887  007144  012745  000200            MOV     #RCVRDONE,-(R5)
2888  007150  012745  000050            MOV     #50,-(R5)
2889  007154  004767  001302            JSR     PC,@TIMER
2890  007160  012605                    MOV     (SP)+,R5
2891  007162
2892  007162  103003                    BCC     $151
2893  007164
2894  007164  052767  000020  001142    BIS     #BIT4,ERRCHK
2895  007172
2896  007172                    $151:
2897
2898  007172
2899  007172  017703  172064            MOV     @RBUF,R3
2900
2901
2902
2903
2904
2905
2906  007176
2907  007176  010204                    MOV     R2,R4
2908  007200  040104                    BIC     R1,R4
2909  007202
2910  007202  040103                    BIC     R1,R3
2911
2912
2913  007204
2914  007204  020403                    CMP     R4,R3
2915  007206  001404                    BEQ     $152
2916
2917
2918  007210
2919  007210  042777  000004  172046    BIC     #MAINT,@TCSR
2920  007216
2921  007216  104116                    ERROR   116
2922
2923  007220
2924  007220                    $152:
```

```
IF.ERROR THEN

    LET ERRCHK := ERRCHK SET.BY #BIT3

ENDIF


    ;TRANSMIT IT
LET @TBUF :B= R2

CALL TIMER IN (#50,#RCVRDONE,RCSR,#SET)




IF.ERROR THEN

    LET ERRCHK := ERRCHK SET.BY #BIT4

ENDIF

    ;AND SAVE IT
LET R3 := @RBUF


;COMPARE TO SEE IF WE RECEIVED IT ALL

    ;CLEAN OFF NON-DATA BITS
    ;ON BOTH TRANSMITTED AND
LET R4 := R2 CLR.BY R1

LET R3 := R3 CLR.BY R1


;RECEIVED DATA
IF R4 NE R3 THEN


    ;DATA COMPARE ERROR
    ; CAN NOT LEAVE WITH MAINT SET
LET    @TCSR := @TCSR CLR.BY #MAINT

    ERRHRD 116,COMP,SBWAS

    ;(TRANSMITTED) (RECEIVED)
ENDIF
```

```
2925  007220
2926  007220  000716                          BR      $146
2927  007222                          $147:
2928
2929
2930  007222
2931  007222  04277'  000004  172034          BIC     #MAINT,@TCSR
2932  007230
2933  007231  032767  000010  00 076          BIT     #BIT3,ERRCHK
2934  007236  001401                          BEQ     $153
2935  007240
2936  007240  104130                          ERROR   130
2937  007242
2938  007242                          $153:
2939  007242
2940  007242  032767  000020  001064          BIT     #BIT4,ERRCHK
2941  007250  001401                          BEQ     $154
2942  007252
2943  007252  104131                          ERROR   131
2944  007254
2945  007254                          $154:
2946  007254
2947
2948
2949
```

```
        ENDINC  ; R2


                ;RESET MAINT. BIT.
        LET @TCSR := @TCSR CLR.BY #MAINT
        IF #BIT3 SETIN ERRCHK THEN

                ERRHRD 130

        ENDIF
        IF #BIT4 SETIN ERRCHK THEN

                ERRHRD 131

        ENDIF
ENDTST
```

```
2950                                    ;;*****************************************************************
2951                                    ;;*****************************************************************
2952                                    ;*TEST 22        TEST DATA THROUGH WRAP
2953                                    ;;*****************************************************************
2954  007254  000004                    TST22:  SCOPE
2955  007256  012767  000001  171674            MOV     #1,$TIMES       ;;DO 1 ITERATION
2956  007264  012767  000022  171706            MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2957  007272                                                                    IF #WRAP NOTSETIN $USWR OR #COMSPD NOTSETIN $USW
2958  007272  032767  020000  171720            BIT     #WRAP,$USWR
2959  007300  001004                            BEQ     $155
2960  007302  032767  000100  171710            BIT     #COMSPD,$USWR
2961  007310  001014                            BNE     $156
2962  007312                                  $155:
2963                                                                              ;CAN'T TEST WITHOUT A WRAP
2964  007312                                                                        EXIT TST
2965  007312  012767  000001  171640            MOV     #1,$TIMES
2966  007320  000522                            BR      TST23           ;;;EXIT THIS TEST
2967  007322                                                                    IFB APTCON EQ #TRUE THEN
2968  007322  126727  001740  000001            CMPB    APTCON,#TRUE
2969  007330  001004                            BNE     $157
2970  007332                                                                        EXIT TEST
2971  007332  012767  000001  171620            MOV     #1,$TIMES
2972  007340  000512                            BR      TST23           ;;;EXIT THIS TEST
2973  007342                                                                    ENDIF
2974  007342                                  $157:
2975  007342                                                                    ENDIF
2976  007342                                  $156:
2977                                                                              ;DON'T USE MAINT.
2978  007342                                                                    LET @TCSR := @TCSR CLR.BY #MAINT
2979  007342  042777  000004  171714            BIC     #MAINT,@TCSR
2980                                                                            ; IF A SPECIAL TURN AROUND CARD IS
2981                                                                            ;   CONNECTED IN PLACE OF THE WRAP
2982                                                                            ;  SETTING READER RUN WILL ENABLE IT.
2983                                                                            ; THIS MODULE IS ONLY USED IN MANUFACTUR
2984                                                                            ; AND ONLY ON THE CONSOLE DLV11-F.
2985                                                                            ; IF NO SPECIAL MODULE IS AVAILABLE,
2986                                                                            ; AND THE WRAP BIT IS SET IN $USWR
2987                                                                            ; THEN THIS TEST WILL ERROR ON THE CONSO
2988
2989  007350                                                                    LET @RCSR := @RCSR SET.BY #11
2990  007350  052777  000011  171702            BIS     #11,@RCSR
2991                                                                              ;CHANGE PRIORITY
2992                                                                              ;..TO 0
2993  007356  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
2994  007362  012746  007370                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
2995  007366  000002                            RTI                     ;;POP NEW PC AND PS
2996  007370                                  64$:
2997                                                                              ;GET DATA MASK
2998  007370                                                                    CALL DATLNG OUT <R1>
2999  007370  162705  000002                    SUB     #1*2,R5
3000  007374  004767  001240                    JSR     PC,DATLNG
3001  007400  012501                            MOV     (R5)+,R1
3002  007402                                                                    LET R0 := @RBUF ; START CLEAN
3003  007402  017700  171654                    MOV     @RBUF,R0
3004                                                                              ;BINARY COUNT PATTERN
3005  007406                                                                    INCR R2 FROM #0 TO #377 BY #1
```

```
3006  007406  005002                          CLR     R2
3007  007410  000401                          BR      $160
3008  007412                          $161:
3009  007412  005202                          INC     R2
3010  007414                          $160:
3011  007414  020227  000377                  CMP     R2,#377
3012  007420  003057                          BGT     $162
3013
3014
3015                                                                ;TRANSMIT THE CHAR. IN R2.
3016
3017                                                                    ; MAKE SURE IT'S READY
3018  007422                                                        CALL TIMER IN <#50,#XMITRDY,TCSR,#SET>
3019  007422  010546                          MOV     R5,-(SP)
3020  007424  012745  177777                  MOV     #SET,-(R5)
3021  007430  016745  171630                  MOV     TCSR,-(R5)
3022  007434  012745  000200                  MOV     #XMITRDY,-(R5)
3023  007440  012745  000050                  MOV     #50,-(R5)
3024  007444  004767  001012                  JSR     PC,TIMER
3025  007450  012605                          MOV     (SP)+,R5
3026  007452                                                        IF.ERROR THEN
3027  007452  103005                          BCC     $163
3028  007454  104123                          ERROR   123                 ; TRANSMITTER NEVER BECAME READY
3029  007456                                                            EXIT TEST
3030  007456  012767  000001  171474          MOV     #1,$TIMES
3031  007464  000440                          BR      TST23       ;;;EXIT THIS TEST
3032  007466                                                        ENDIF
3033  007466                          $163:
3034
3035                                                                  ;START IT ON ITS WAY
3036  007466                                                        LET @TBUF :B= R2
3037  007466  110277  171576                  MOVB    R2,@TBUF
3038                                                                ; NOW WAIT FOR RECIEVER DONE
3039  007472                                                        CALL TIMER IN <#50,#RCVRDONE,RCSR,#SET>
3040  007472  010546                          MOV     R5,-(SP)
3041  007474  012745  177777                  MOV     #SET,-(R5)
3042  007500  016745  171554                  MOV     RCSR,-(R5)
3043  007504  012745  000200                  MOV     #RCVRDONE,-(R5)
3044  007510  012745  000050                  MOV     #50,-(R5)
3045  007514  004767  000742                  JSR     PC,TIMER
3046  007520  012605                          MOV     (SP)+,R5
3047  007522                                                            IF.ERROR THEN
3048  007522  103005                          BCC     $164
3049  007524                                                                ERRHRD 124
3050  007524  104124                          ERROR   124
3051                                                                ; RECIEVER NEVER BECAME READY
3052  007526                                                            EXIT TEST
3053  007526  012767  000001  171424          MOV     #1,$TIMES
3054  007534  000414                          BR      TST23       ;;;EXIT THIS TEST
3055  007536                                                        ENDIF
3056  007536                          $164:
3057
3058                                                                  ;RETRIEVE
3059  007536                                                        LET R3 := @RBUF
3060  007536  017703  171520                  MOV     @RBUF,R3
3061
```

```
3062
3063  007542
3064  007542  010204                    MOV     R2,R4
3065  007544  040104                    BIC     R1,R4
3066  007546
3067  007546  040103                    BIC     R1,R3
3068
3069
3070  007550
3071  007550  020403                    CMP     R4,R3
3072  007552  001401                    BEQ     $165
3073
3074  007554
3075  007554  104117                    ERROR   117
3076
3077  007556
3078  007556                      $165:
3079
3080  007556
3081  007556  000715                    BR      $161
3082  007560                      $162:
3083
3084
3085
3086  007560
3087  007560  052777  000011  171472    BIS     #11,@RCSR
3088
3089
3090
3091  007566                                                  ENDTST
3092
3093
3094
3095
```

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

    LET R3 := R3 CLR.BY R1

        ;WE HAVE TROUBLE
    IF R4 NE R3 THEN

        ;DATA COMPARE ERROR
        ERRHRD 117,COMP,SBWAS

        ;(R2) (R3)
    ENDIF

ENDINC  ; R2

        ; NOW THAT THE TEST IS DONE
        ;  WE WILL TOGGLE READER RUN
        ;  TO TURN OFF THE SPECIAL MODULE.
LET @RCSR := @RCSR SET.BY #11
```

```
3096                                    ;;*****************************************************************
3097                                    ;;*****************************************************************
3098                                    ;*TEST 23        FULL DATA TRANSFER WITH INTERRUPTS
3099                                    ;*              AND MAINTENANCE MODE.
3100                                    ;;*****************************************************************
3101  007566  000004                    TST23:  SCOPE
3102  007570  012767  000001  171362            MOV     #1,$TIMES       ;;DO 1 ITERATION
3103  007576  012767  000023  171374            MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3104                                                                            IF #MAINTJUMP NOTSETIN $USWR ORB APTCON EQ #TRUE
3105  007604
3106  007604  032767  040000  171406            BIT     #MAINTJUMP,$USWR
3107  007612  001404                            BEQ     $166
3108  007614  126727  001446  000001            CMPB    APTCON,#TRUE
3109  007622  001004                            BNE     $167
3110  007624                            $166:
3111  007624                                                                        EXIT TEST
3112  007624  012767  000001  171326            MOV     #1,$TIMES
3113  007632  000540                            BR      TST24           ;;;EXIT THIS TEST
3114  007634                                                                        ENDIF
3115  007634                            $167:
3116
3117                                                                                ;GET DATA MASK
3118  007634                                                                        CALL DATLNG OUT <R3>
3119  007634  162705  000002                    SUB     #1*2,R5
3120  007640  004767  000774                    JSR     PC,DATLNG
3121  007644  012503                            MOV     (R5)+,R3
3122
3123
3124                                    ;SINCE THIS WILL BE A FULL SPEED INTERRUPT TEST
3125                                    ;IF A DATA COMPARE ERROR OCCURS THE REPORT
3126                                    ;WILL BE DEFERRED UNTIL ALL DATA PATTERNS HAVE
3127                                    ;BEEN TRANSFERED.  IF THE NUMBER OF CHAR. TRANSMITTED
3128                                    ;DOES NOT EQUAL THE NUMBER RECEIVED THIS ALSO WILL
3129                                    ;BE REPORTED.
3130
3131                                                                                ;CHANGE PRIORITY
3132                                                                                ;..TO 0
3133  007646  012746  000000                    MOV     #PR0,-(SP)      ;;PUT NEW PS ON STACK
3134  007652  012746  007660                    MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
3135  007656  000002                            RTI                     ;;POP NEW PC AND PS
3136  007660                            64$:
3137                                                                                ;GET VECTOR ADDRESS
3138  007660                                                                    LET R1 := DLVEC
3139  007660  016701  171372                    MOV     DLVEC,R1
3140                                                                                ;RCVR VECTOR
3141  007664                                                                    LET (R1)+ := #REC
3142  007664  012721  010040                    MOV     #REC,(R1)+
3143  007670                                                                    LET (R1)+ := #PR7
3144  007670  012721  000340                    MOV     #PR7,(R1)+
3145                                                                                ;POINT TO TRANSMITTER VECTOR
3146                                                                                ;AND SET IT UP ALSO
3147  007674                                                                    LET (R1)+ := #TRAN
3148  007674  012721  007776                    MOV     #TRAN,(R1)+
3149  007700                                                                    LET (R1)  := #PR7
3150  007700  012711  000340                    MOV     #PR7,(R1)
3151
```

```
                                                                        ; CLEAR ERROR COUNTER
007704                                                                  LET ERRCNT := #0
007704  005067  000060                  CLR     ERRCNT
                                                                        ;START COUNT AT 0
007710                                                                  LET R1 := #0
007710  005001                          CLR     R1
                                                                        ;RECEIVER STORAGE
007712                                                                  LET R2 := #0
007712  005002                          CLR     R2
                                                                        ;# OF RECEIVED CHAR. COUNT.
007714                                                                  LET R4 := #0
007714  005004                          CLR     R4

007716                                                                  BRESET  ;SET UP ALL REGISTERS
007716  000005                          RESET
                                                                        ;SET UP MAINTENANCE
007720                                                                  LET @TCSR := @TCSR SET.BY #MAINT
007720  052777  000004  171336          BIS     #MAINT,@TCSR

                                                                        ;SET I.E. IN TRANSMITTER
007726                                                                  LET @TCSR := @TCSR SET.BY #XMITIE
007726  052777  000100  171330          BIS     #XMITIE,@TCSR
                                                                        ;AND RECEIVER
007734                                                                  LET @RCSR := @RCSR SET.BY #RCVRIE
007734  052777  000100  171316          BIS     #RCVRIE,@RCSR


                                                                                ; CLEAR OUR DONE AND ERROR FLAG WORD

                                                                        ;NOW WE WAIT UNTIL R4 COUNT (RECEIVED) IS EQUAL
007742                                                                  REPEAT
007742                          $170:
007742                                                                  UNTIL R4 HIS NUMBER
007742  020467  000024                  CMP     R4,NUMBER
007746  103775                          BLO     $170

007750                                                                  LET @TCSR := @TCSR CLR.BY #MAINT
007750  042777  000004  171306          BIC     #MAINT,@TCSR
                                                                        ; CHECK FOR DATA COMPARE ERRORS.
007756                                                                  IF ERRCNT NE #0 THEN
007756  005767  000006                  TST     ERRCNT
007762  001401                          BEQ     $171

                                                                        ;DATA COMPARE ERROR
007764                                                                      ERRHRD 120,COMP,FIRST
007764  104120                          ERROR   120

                                                                        ;<R3> OCCURED, FIRST: SB <S3>, WAS <WAS>
007766                                                                  ENDIF
007766                          $171:

007766                                                                  EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
007766  000462                          BR      TST24           ;;;EXIT   IS TEST

007770  000000                  ERRCNT: 0
007772  001000                  NUMBER: 1000
007774     000                  SB:     .BYTE 0
007775     000                  WAS:    .BYTE 0
```

```
                                                                        ;TRANSMIT INTERRUPT HANDLER
                                                                BGNSRV  TRAN

007776                                  TRAN:
                                                                        ;SET UP FOR TRANSFER
                                                                    LET HOLD := R1      CLR.BY R3
007776  010167  000032                          MOV     R1,HOLD
010002  040367  000026                          BIC     R3,HOLD
                                                                        ;AND SEND.
010006                                                              LET @TBUF := HOLD
010006  016777  000022  171254                  MOV     HOLD,@TBUF
                                                                        ;INCREMENT CHAR COUNT
010014                                                              LET R1 := R1 + #1
010014  005201                          INC     R1
                                                                        ;ALL DONE
010016                                                              IF R1 EQ NUMBER THEN
010016  020167  177750                  CMP     R1,NUMBER
010022  001003                          BNE     $172
                                                                        ;STOP   INTERRUPT PROCESSING
                                                                        LET @TCSR := @TCSR CLR.BY #XMITIE
010024  042777  000100  171232          BIC     #XMITIE,@TCSR
                                                                    ENDIF
010032                                  $172:

010032  000401                          BR      ZZZ                 ; EXIT SRV

010034  000000                      HOLD:0

010036                                  ZZZ:
010036  000002                          RTI

                                                                ENDSRV


                                                                        ;RECEIVER INTERRUPT HANDLER
                                                                BGNSRV  REC

010040                                  REC:

                                                                        ;GET CHAR IN + MASK IT
                                                                    LET R2 := @RBUF CLR.BY R3
010040  017702  171216                  MOV     @RBUF,R2
010044  040302                          BIC     R3,R2
                                                                        ;RHLD WILL CONTAIN EXPECTED INPUT
                                                                    LET RHLD := R4 CLR.BY R3
010046  010467  000056                  MOV     R4,RHLD
010052  040367  000052                  BIC     R3,RHLD

                                                                        ;DO THEY COMPARE
                                                                    IF R2 NE RHLD THEN
010056  020267  000046                  CMP     R2,RHLD
010062  001412                          BEQ     $173
                                                                        ;FIRST ERROR
                                                                        IF ERRCNT EQ #0 THEN
010064  005767  177700                  TST     ERRCNT
010070  001005                          BNE     $174
```

```
3264
3265  010072
3266  010072  116767  000032  177674          MOVB    RHLD,SB
3267  010100
3268  010100  110267  177671                  MOVB    R2,WAS
3269  010104
3270  010104                          $174:
3271
3272  010104
3273  010104  005267  177660                  INC     ERRCNT
3274  010110
3275  010110                          $173:
3276
3277
3278  010110
3279  010110  005204                          INC     R4
3280
3281  010112
3282  010112  020467  177654                  CMP     R4,NUMBER
3283  010116  001003                          BNE     $175
3284
3285  010120
3286  010120  042777  000100  171132          BIC     #RCVRIE,@RCSR
3287
3288
3289  010126
3290  010126                          $175:
3291
3292
3293  010126  000401                          BR      ZZZZ
3294
3295  010130  000000
3296  010132                          ZZZZ:
3297  010132
3298  010132  000002                          RTI
3299
3300  010134
3301
3302
3303
```

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

        LET WAS :B= R2

    ENDIF

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

ENDIF


    ;COUNT THIS CHAR.
LET R4 := R4 + #1

    ;ALL DONE?
IF R4 EQ NUMBER THEN


    ;STOP RECEIVER INTERRUPTS
    LET @RCSR := @RCSR CLR.BY #RCVRIE

            ; INDICATE ALL DONE TO TIMER
    ;MAIN REPEAT LOOP IS CHECKING
ENDIF


                                    ;FOR 'R4 = NUMBER' ALSO
                                ;   EXIT SRV

                                    RHLD:0

                                ENDSRV

                            ENDTST
```

```
                                ;;*****************************************************************
                                ;;*****************************************************************
                                ;*TEST 24        TEST BREAK GENERATION LOGIC
                                ;*              TRANSMIT KNOWN CHAR WITH BREAK SET
                                ;*              AND COMPARE RECEIVED WITH 0.
                                ;*              FRAMING ERROR WILL ALSO BE CHECKED
                                ;*              IF ERROR BITS ARE ENABLED.
                                ;;*****************************************************************
010134  000004                  TST24:  SCOPE
010136  012767  000010  171014          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
010144  012767  000024  171026          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        IF #MAINTJUMP NOTSETIN $USWR OR #BRK NOTSETIN $U
010152
010152  032767  040000  171040          BIT     #MAINTJUMP,$USWR
010160  001404                          BEQ     $176
010162  032767  010000  171030          BIT     #BRK,$USWR
010170  001004                          BNE     $177
010172                          $176:
                                                                            EXIT TEST
010172
010172  012767  000001  170760          MOV     #1,$TIMES
010200  000456                          BR      TST25           ;;;EXIT THIS TEST
                                                                        ENDIF
010202
010202                          $177:
                                ;
                                                                LET ERRCHK := #0        ; CLEAR ERROR WORD
010202
010202  005067  000126                  CLR     ERRCHK
                                ;
                                                                        ;SET MAINTENANCE BIT
010206                                                          LET @TCSR := @TCSR SET.BY #MAINT
010206  052777  000004  171050          BIS     #MAINT,@TCSR
                                ;
                                                                        ;SET BREAK BIT
010214                                                          LET @TCSR := @TCSR SET.BY #BREAK
010214  052777  000001  171042          BIS     #BREAK,@TCSR
                                ;
                                                                        ;NON-ZERO CHAR. '*'
010222                                                          LET @TBUF := #252
010222  012777  000252  171040          MOV     #252,@TBUF
                                ;
                                                                        ;WAIT FOR DONE
010230                                                          REPEAT
010230                          $200:
                                ;
                                                                        ; WAIT FOR DONE
010230                                                          UNTIL #RCVRDONE SETIN @RCSR
010230  032777  000200  171022          BIT     #RCVRDONE,@RCSR
010236  001774                          BEQ     $200
                                ;
                                                                IFB @RBUF NE #0 THEN
010240
010240  105777  171016                  TSTB    @RBUF
010244  001404                          BEQ     $201
                                ;
                                                                        ; BREAK DID NOT EQUAL 0
010246                                                              LET ERRCHK := ERRCHK SET.BY #BIT0
010246  052767  000001  000060          BIS     #BIT0,ERRCHK
010254                                                          ELSE
010254  000413                          BR      $202
010256                          $201:
                                ;
                                                                IF #ERRBITS SETIN $USWR THEN
010256
010256  032767  100000  170734          BIT     #ERRBITS,$USWR
010264  001407                          BEQ     $203
                                ;
                                                                        IF #FRERR NOTSETIN @RBUF THEN
010266
010266  032777  020000  170766          BIT     #FRERR,@RBUF
```

```
3360  010274  001003                        BNE     $204
3361  010276
3362  010276  052767  000002  000030        BIS     #BIT1,ERRCHK
3363  010304
3364  010304                          $204:
3365  010304
3366  010304                          $203:
3367  010304
3368  010304                          $202:
3369
3370  010304
3371  010304  000005                        RESET
3372
3373  010306
3374  010306  032767  000001  000020        BIT     #BIT0,ERRCHK
3375  010314  001401                        BEQ     $205
3376  010316
3377  010316  104121                        ERROR   121
3378  010320
3379  010320                          $205:
3380  010320
3381  010320  032767  000002  000006        BIT     #BIT1,ERRCHK
3382  010326  001401                        BEQ     $206
3383  010330
3384  010330  104122                        ERROR   122
3385  010332
3386  010332                          $206:
3387  010332
3388  010332  000401                        BR      TST25
3389  010334  000000                ERRCHK: .WORD 0
3390  010336
3391
```

```
            LET ERRCHK := ERRCHK SET.BY #BIT1
        ENDIF
ENDIF
ENDIF

BRESET  ;CLEAN UP

IF #BIT0 SETIN ERRCHK THEN

    ERRHRD 121  ;BREAK ERROR
ENDIF
IF #BIT1 SETIN ERRCHK THEN

    ERRHRD 122  ; FRAMING ERROR
ENDIF
                ENDIF

                EXIT
;;;EXIT THIS TEST
ENDTST
```

```
3392
3393
3394
3395
3396  010336  000004
3397  010340  012767  000001  170612          ;;**********************************************************************
3398  010346  104401  010354                  ;;*TEST 25       NOT A TEST - SEND BACK TO LOOP
3399  010352  000404                          ;;**********************************************************************
3400                                  TST25:  SCOPE
3401  010364                                  MOV     #1,$TIMES       ;;DO 1 ITERATION
3402  010364  016746  170664                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3403  010370  104402                          BR      64$             ;;GET OVER THE ASCIZ
3404  010372  104401  010400          ;;65$:  .ASCIZ  <CRLF>*CSR: *
3405  010376  000405                  64$:
3406                                          MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
3407  010412                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3408  010412  016746  170640                  TYPE    ,67$            ;;TYPE ASCIZ STRING
3409  010416  104402                          BR      66$             ;;GET OVER THE ASCIZ
3410  010420  104401  010426          ;;67$:  .ASCIZ  *,VECTOR: *
3411  010424  000405                  66$:
3412                                          MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
3413  010440                                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3414  010440  016746  170446                  TYPE    ,69$            ;;TYPE ASCIZ STRING
3415  010444  104405                          BR      68$             ;;GET OVER THE ASCIZ
3416  010446  005067  170440          ;;69$:  .ASCIZ  *,ERRORS: *
3417  010452  104401  001171          68$:
3418  010456  000167  171454                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
                                                TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
                                                CLR     $ERTTL          ; RESET FOR NEXT DEVICE/PASS
                                                TYPE    ,$CRLF
                                                JMP     LOOP            ; BACK UP TO THE BEGINNING
```

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

000001                                                  TRUE=   1
000000                                                  FALSE=  0

010462                                                  LET     REGSAV := REG(R5) ; GET POINTER TO REGIST
010462  016567  000004  000136          MOV     REG(R5),REGSAV
010470                                                  LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR
010470  016567  000000  000132          MOV     HOWLONG(R5),TIMSAV
010476                                                  LET     FLAG :B= #FALSE ; INITIALIZE THE EXIT FLA
010476  112767  000000  000126          MOVB    #FALSE,FLAG


                                                        ;       START OF AN INFINITE LOOP
010504                                                  LOOP
010504                          $211:
                                                            ; TEST TO SEE IF WHICHBIT IS SET
010504                                                      IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
010504  036577  000002  000114          BIT     WHICHBIT(R5),@REGSAV
010512  001004                          BNE     $213
010514                                                          LET     HOLDSC :B= #CLR
010514  112767  000000  000111          MOVB    #CLR,HOLDSC
010522                                                      ELSE
010522  000403                          BR      $214
010524                          $213:
                                                            LET     HOLDSC :B= #SET ; REMEMBER THIS
010524  112767  177777  000101          MOVB    #SET,HOLDSC
010532                                                      ENDIF
010532                          $214:
                                                        ; NOW SEE IF THAT WAS WHAT WE WANTED
010532                                                  IFB     HOLDSC EQ SETCLR(R5) THEN
```

```
3475  010532  126765  000075  000006          CMPB    HOLDSC,SETCLR(R5)
3476  010540  001003                          BNE     $215
3477
3478  010542                                                  ; JUST THE THING WE NEEDED
3479  010542                                                        LET     FLAG :B= #TRUE
3480  010542  112767  000001  000062          MOVB    #TRUE,FLAG
3481  010550                                                     ENDIF
3482  010550                          $215:
3483  010550                                                  EXIFB   FLAG EQ #TRUE OR TIMSAV LE #0
3484  010550  126727  000056  000001          CMPB    FLAG,#TRUE
3485  010556  001414                          BEQ     $212
3486  010560  005767  000044                  TST     TIMSAV
3487  010564  003411                          BLE     $212
3488
3489                                                    ; ONE WAY OR THE OTHER, WE ARE DONE
3490                                            ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
3491  010566                                                        WAITUS  10      ;WAIT FOR 10 MILLI-SECONDS
3492  010566  010546                          MOV     R5,-(SP)
3493  010570  012745  000010                  MOV     #10,-(R5)
3494  010574  004767  000140                  JSR     PC,WAIT
3495  010600  012605                          MOV     (SP)+,R5
3496  010602                                                     LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
3497  010602  005367  000022                  DEC     TIMSAV
3498  010606                                                  ENDLOOP                 ; CONTINUED AT THE TOP
3499  010606  000736                          BR      $211
3500  010610                          $212:
3501
3502                                          ; ONLY 2 WAYS TO GET HERE
3503                                          ;       1). WE RAN OUT OF TIME---ERROR ! !
3504                                          ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
3505
3506  010610                                                        IFB     FLAG EQ #TRUE THEN
3507  010610  126727  000016  000001          CMPB    FLAG,#TRUE
3508  010616  001001                          BNE     $216
3509  010620                                                            RETURN  NO.ERROR    ; GOOD
3510  010620  000405                          BR      $207
3511  010622                                                        ENDIF
3512  010622                          $216:
3513  010622                                                        RETURN  ERROR           ; BAD
3514  010622  000261                          SEC
3515  010624  000404                          BR      $210
3516
3517  010626  000000                                        REGSAV: .WORD 0
3518  010630  000000                                        TIMSAV: .WORD 0
3519  010632     000                                        FLAG:   .BYTE 0
3520  010633     000                                        HOLDSC: .BYTE 0
3521                                                  ; WE ARE DONE  GO BACK HOME
3522  010634                                                     ENDRTN
3523  010634                          $207:
3524  010634  000241                          CLC
3525  010636                          $210:
3526  010636  000207                          RTS     PC
```

```
                                ;;*****************************************************************
                                ;
                                ROUTINE DATLNG (MASK)
                                DATLNG:
                                ;* ROUTINE:DATLNG
                                ;*       THIS ROUTINE SETS UP A MASK FOR DATA, WITH
                                ;*  INPUT - NOTHING IS PASSED TO THIS ROUTINE
                                ;*      BUT GLOBAL INFORMATION IS ASSUMED TO EXIST:
                                ;*      $USWR-- THE WORD FOR SOFTWARE PARAMETERS
                                ;*      DATA-- A MASK FOR THE LOCATION OF THE OCTAL
                                ;*              NUMBER OF DATA BITS
                                ;*  OUTPUT----
                                ;*      MASK-- A MASK OF BINARY ONES RIGHT-JUSTIFIED
                                ;*              THE NUMBER OF WHICH IS DEFINED IN $USWR WORD.
                                ;
                                ;;*****************************************************************

010640                                                          LET MASK(R5) := #0              ; START
010640  005065  000000                  CLR     MASK(R5)
010644                                                          LET NUMBR := $USWR AND #DATA
010644  016767  170350  000062          MOV     $USWR,NUMBR
010652  016746  000056                  MOV     NUMBR,-(SP)
010656  042716  000017                  BIC     #DATA,(SP)
010662  042667  000046                  BIC     (SP)+,NUMBR

010666                                                          INCR I FROM #1 TO NUMBR BY #1
010666  012767  000001  170376          MOV     #1,I
010674  000402                          BR      $221
010676                          $222:
010676  005267  170370                  INC     I
010702                          $221:
010702  026767  170364  000024          CMP     I,NUMBR
010710  003006                          BGT     $223
010712                                      LET MASK(R5) := MASK(R5) SHIFT 1
010712  006365  000000                  ASL     MASK(R5)
010716                                      LET MASK(R5) := MASK(R5) SET.BY #1
010716  052765  000001  000000          BIS     #1,MASK(R5)
010724                                                      ENDINC
010724  000764                          BR      $222
010726                          $223:
010726                                                      LET MASK(R5) := COMP MASK(R5)
010726  005165  000000                  COM     MASK(R5)
010732                                                  RETURN
010732  000401                          BR      $217
010734  000000                          NUMBR:0
010736                                                  ENDRTN
010736                          $217:
010736                          $220:
010736  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.
                                ;;*****************************************************************
010740  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
010742  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
010744  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
010746                                  LET R1 := TIME(R5)
010746  016501  000000                  MOV     TIME(R5),R1
010752                                  INCRU R2 FROM #1 TO R1 BY #1
010752  012702  000001                  MOV     #1,R2
010756  000402                          BR      $225
010760                          $227:
010760  062702  000001                  ADD     #01,R2
010764                          $226:
010764  020201                          CMP     R2,R1
010766  101010                          BHI     $230
010770                                  INCR R3 FROM #0 TO #700 BY #1
010770  005003                          CLR     R3
010772  000401                          BR      $231
010774                          $232:
010774  005203                          INC     R3
010776                          $231:
010776  020327  000700                  CMP     R3,#700
011002  003001                          BGT     $233
011004                                  ENDINC
011004  000773                          BR      $232
011006                          $233:
011006                                  ENDINC
011006  000764                          BR      $227
011010                          $230:
011010  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
011012  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
011014  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
011016                          ENDRTN
011016                          $224:
011016                          $225:
011016  000207                          RTS     PC
```

```
3621
3622                                    .SBTTL  INTSRV  INTERRUPT SERVICE ROUTINE
3623                                    ;;*****************************************************************
3624  011020                            INTSRV:
3625                                    ;* SERVICE ROUTINE: INTSRV
3626                                    ;*       THIS GLOBAL ROUTINE DOES NOTHING BUT INCREMENT
3627                                    ;*       'INTFLAG'  EACH TIME IT IS CALLED.  IT ASSUMES
3628                                    ;*       THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
3629                                    ;*       TO LOOK FOR.
3630                                    ;;*****************************************************************
3631
3632                                                    ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
3633  011020                                                            LET INTFLAG := INTFLAG + #1
3634  011020  005267  000002                   INC     INTFLAG
3635  011024                                    ENDSRV                          ;THAT'S ALL
3636  011024  000002                           RTI
3637  011026  000000                   INTFLAG:  0
```

```
                        ROUTINE CYCLE
                        CYCLE:
                        ;;*****************************************************************
                        ;* ROUTINE:     CYCLE
                        ;*       THIS  ROUTINE  CAUSES  ADRS   TO POINT TO  THE
                        ;*       ADDRESS  OF  DLV11-F  UNDER  TEST,  ADRS +2 TO
                        ;*       POINT TO THE VECTOR OF THE DLV11-F UNDER TEST.
                        ;*      IT  KEEPS  TRACK  OF THE CURRENT DEVICE AND BIT
                        ;*      MASKS. THE CONSOLE IS TREATED SPECIAL BY THIS ROUTINE.
                        ;*      IT IS ONLY TESTED ONCE IF UNDER APT. IF NOT UNDER APT
                        ;*      ALL TESTS THAT REQUIRE THE MAINT BIT ARE NOT RUN.
                        ;;*****************************************************************
                                                        LET APTCON :B= #FALSE   ; SET DEFAULT VALUE
011030  112767  000000  000230          MOVB    #FALSE,APTCON
                                                        LET CONSOLE :B= #FALSE
011036  112767  000000  000223          MOVB    #FALSE,CONSOLE
                                                     REPEAT       ; UNTIL BITMASK SETIN $DEVM
                        $236:                         IF BITMASK EQ #0 THEN
011044  005767  000200                  TST     BITMASK
011050  001027                          BNE     $237
                                                         IF INITFLAG EQ #1 THEN
011052  026727  000174  000001          CMP     INITFLAG,#1
011060  001003                          BNE     $240
                                                             LET INITFLAG := #0
011062  005067  000164                  CLR     INITFLAG
                                                         ELSE
011066  000403                          BR      $241
                        $240:
                                                             CALL $EOP   ; AS A SUBROUTINE
011070  004767  000370                  JSR     PC,$EOP
                        SPECIALADDRESS:                ; BECAUSE $EOP RETURNS AS A JUMP
                                                             LET R0 := POP
011074  012600                          MOV     (SP)+,R0
                                                         ENDIF
                        $241:
                                                         LET BITMASK := #1
011076  012767  000001  000144          MOV     #1,BITMASK
                                                         LET $DEVCT := #1
011104  012767  000001  170072          MOV     #1,$DEVCT
                                                         LET ADDRESS := $BASE
011112  016767  170132  000134          MOV     $BASE,ADDRESS
                                                         LET VECTOR := $VECT1
011120  016767  170120  000130          MOV     $VECT1,VECTOR
                                                     ELSE
011126  000410                          BR      $242
                        $237:
                                                         LET R4 := #10
011130  012704  000010                  MOV     #10,R4
                                                         LET BITMASK := BITMASK ROTATE 1
011134  006167  000110                  ROL     BITMASK
                                                         LET ADDRESS  := ADDRESS + R4
011140  060467  000110                  ADD     R4,ADDRESS
                                                         LET VECTOR := VECTOR + R4
011144  060467  000106                  ADD     R4,VECTOR
```

```
3694  011150                                                          ENDIF
3695  011150                          $242:
3696  011150                                                      UNTIL   BITMASK SETIN $DEVM
3697  011150  036767  000074  170074          BIT     BITMASK,$DEVM
3698  011156  001732                          BEQ     $236
3699  011160                                                  IF BITMASK EQ #BIT15 THEN
3700  011160  026727  000064  100000          CMP     BITMASK,#BIT15
3701  011166  001023                          BNE     $243
3702  011170                                                  LET CONSOLE :B= #TRUE
3703  011170  112767  000001  000071          MOVB    #TRUE,CONSOLE
3704  011176                                                  LET ADDRESS := CONADR
3705  011176  016767  000060  000050          MOV     CONADR,ADDRESS
3706  011204                                                  LET VECTOR  := CONVECT
3707  011204  016767  000054  000044          MOV     CONVECT,VECTOR
3708                                          ;;;;;;;         IF #CONMAINT NOTSETIN $USWR THEN
3709                                          ;;;;;;;             LET NOCONMAINT :B= #TRUE
3710                                          ;;;;;;;         ENDIF
3711  011212                                                  IF #APTENV SETIN $ENV THEN ; APT MODE
3712  011212  032767  000001  167774          BIT     #APTENV,$ENV
3713  011220  001406                          BEQ     $244
3714  011222                                                      IF $PASS NE #0 THEN  ; NOT FIRST PASS
3715  011222  005767  167754                  TST     $PASS
3716  011226  001403                          BEQ     $245
3717                                                                      ; DEFINE DEVICE AS APT CONSOLE
3718  011230                                                              LET APTCON :B= #TRUE
3719  011230  112767  000001  000030          MOVB    #TRUE,APTCON
3720  011236                                                          ENDIF           ; FIRST PASS
3721  011236                          $245:
3722  011236                                                      ENDIF   ; APT
3723  011236                          $244:
3724  011236                                                  ENDIF                   ; BITMASK
3725  011236                          $243:
3726
3727  011236                                                  LET ADRS := #ADDRESS
3728  011236  012701  011254                  MOV     #ADDRESS,ADRS
3729  011242                                                      LET $DEVCT := $DEVCT + #1
3730  011242  005267  167736                  INC     $DEVCT
3731  011246                                                      RETURN
3732  011246  000411                          BR      $234
3733  011250  100000                  BITMASK:        100000                  ; CONSOLE FIRST
3734  011252  000001                  INITFLAG:       1
3735  011254  000000                  ADDRESS:        0
3736  011256  000000                  VECTOR:         0
3737  011260  000000                  OK:             0
3738  011262  177560                  CONADR:         177560                  ; CONSOLE ADDRESS
3739  011264  000060                  CONVECT:            60                  ; CONSOLE VECTOR
3740  011266     000                  APTCON:         .BYTE 0
3741  011267     000                  CONSOLE:        .BYTE 0
3742  011270     000                  NOCONMANT:      .BYTE 0
3743        011272                          .EVEN
3744
3745  011272                                                      ENDRTN
3746  011272                          $234:
3747  011272                          $235:
3748  011272  000207                          RTS     PC
3749
```

```
3750
```

```
3751
3752  011274                                ROUTINE MYTYPE
3753  011274                                MYTYPE:
3754                                        ;;*****************************************************************
3755  011274  104401  011302                        TYPE    ,65$            ;;TYPE ASCIZ STRING
3756  011300  000405                                BR      64$             ;;GET OVER THE ASCIZ
3757                                        ;;65$:  .ASCIZ  <CRLF>*TEST # *
3758  011314                                64$:
3759  011314  016746  167660                        MOV     $TESTN,-(SP)    ;;SAVE $TESTN FOR TYPEOUT
3760  011320  104402                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3761  011322  104401  011330                        TYPE    ,67$            ;;TYPE ASCIZ STRING
3762  011326  000405                                BR      66$             ;;GET OVER THE ASCIZ
3763                                        ;;67$:  .ASCIZ  *,ERROR # *
3764  011342                                66$:
3765  011342  116767  167546  167626                MOVB    $ITEMB,$FATAL   ; APT FATAL ERROR NUMBER
3766  011350  016746  167622                        MOV     $FATAL,-(SP)    ;;SAVE $FATAL FOR TYPEOUT
3767  011354  104403                                TYPOS                   ;;GO TYPE--OCTAL ASCII
3768  011356     006                                .BYTE   6               ;;TYPE 6 DIGITS
3769  011357     000                                .BYTE   0               ;;SUPPRESS LEADING ZEROS
3770  011360  104401  011366                        TYPE    ,69$            ;;TYPE ASCIZ STRING
3771  011364  000404                                BR      68$             ;;GET OVER THE ASCIZ
3772                                        ;;69$:  .ASCIZ  *,PC = *
3773  011376                                68$:
3774  011376  016746  167514                        MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
3775  011402  104402                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3776  011404  104401  011412                        TYPE    ,71$            ;;TYPE ASCIZ STRING
3777  011410  000404                                BR      70$             ;;GET OVER THE ASCIZ
3778                                        ;;71$:  .ASCIZ  *,CSR: *
3779  011422                                70$:
3780  011422  016746  167626                        MOV     DLADD,-(SP)     ;;SAVE DLADD FOR TYPEOUT
3781  011426  104402                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3782  011430  104401  011436                        TYPE    ,73$            ;;TYPE ASCIZ STRING
3783  011434  000405                                BR      72$             ;;GET OVER THE ASCIZ
3784                                        ;;73$:  .ASCIZ  *,VECTOR: *
3785  011450                                72$:
3786  011450  016746  167602                        MOV     DLVEC,-(SP)     ;;SAVE DLVEC FOR TYPEOUT
3787  011454  104402                                TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3788  011456  104401  001171                        TYPE    ,$CRLF
3789  011462                                ENDRTN
3790  011462                                $246:
3791  011462                                $247:
3792  011462  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

011464                          $EOP:
011464  000004                         SCOPE
011466  005067  167410                 CLR     $TSTNM          ;;ZERO THE TEST NUMBER
011472  005067  167462                 CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
011476  005267  167500                 INC     $PASS           ;;INCREMENT THE PASS NUMBER
011502  042767  100000  167472         BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
011510  005327                         DEC     (PC)+           ;;LOOP?
011512  000001                 $EOPCT: .WORD   1
011514  003022                         BGT     $DOAGN          ;;YES
011516  012737                         MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
011520  000001                 $ENDCT: .WORD   1
011522  011512                         $EOPCT
011524  104401  011571                 TYPE    ,$ENDMG         ;;TYPE "END PASS #"
011530  016746  167446                 MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
011534  104405                         TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011536  104401  011566                 TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
011542  013700  000042         $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
011546  001405                         BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
011550  000005                         RESET                   ;;CLEAR THE WORLD
011552  004710                 $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
011554  000240                         NOP                     ;;SAVE ROOM
011556  000240                         NOP                     ;;FOR
011560  000240                         NOP                     ;;ACT11
011562                          $DOAGN:
011562  000137                         JMP     @(PC)+          ;;RETURN
011564  011074                 $RTNAD: .WORD   SPECIALADDRESS
011566     377     377     000 $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
011571     015  042412  042116 $ENDMG: .ASCIZ  <15><12>/END PASS #/
011576  050040  051501  020123
011604  000043
```

```
                                .SBTTL  POWER DOWN AND UP ROUTINES

                                ;;**********************************************************************
                                ;POWER DOWN ROUTINE
011606  012737  011752  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
011614  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
011622  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
011624  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
011626  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
011630  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
011632  010446                        MOV     R4,-(SP)        ;;PUSH R4 ON STACK
011634  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
011636  017746  167276                MOV     @$WR,-(SP)      ;;PUSH @$WR ON STACK
011642  010667  000110                MOV     SP,$SAVR6       ;;SAVE SP
011646  012737  011660  000024        MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
011654  000000                        HALT
011656  000776                        BR      .-2             ;;HANG UP

                                ;;**********************************************************************
                                ;POWER UP ROUTINE
011660  012737  011752  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
011666  016706  000064                MOV     $SAVR6,SP       ;;GET SP
011672  005067  000060                CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
011676  005267  000054         1$:    INC     $SAVR6          ;;WAIT FOR THE INC
011702  001375                        BNE     1$              ;;OF  WORD
011704  012677  167230                MOV     (SP)+,@$WR      ;;POP STACK INTO @$WR
011710  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
011712  012604                        MOV     (SP)+,R4        ;;POP STACK INTO R4
011714  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
011716  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
011720  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
011722  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
011724  012737  011606  000024        MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
011732  012737  000340  000026        MOV     #340,@#PWRVEC+2 ;;PRIO:7
011740  104401                        TYPE                    ;REPORT THE POWER FAILURE
011742  011760                $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
011744  012716                        MOV     (PC)+,(SP)      ;;RESTART AT LOOP
011746  002136                $PWRAD: .WORD   LOOP            ;;RESTART ADDRESS
011750  000002                        RTI
011752  000000                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
011754  000776                        BR      .-2             ;;  BEFORE THE POWER DOWN WAS COMPLETE
011756  000000                $SAVR6: 0                       ;;PUT THE SP HERE
011760  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
011766  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
                                ;*

011770  105767  167163          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
011774  100002                          BPL     1$              ;;BR IF YES
011776  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
012000  000430                          BR      3$              ;;LEAVE
012002  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
012004  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
012010  122767  000001  167176          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
012016  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
012020  132767  000100  167167          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
012026  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
012030  010067  000004                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
012034  004767  000774                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
012040  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
012042  132767  000040  167145  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
012050  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
012052  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
012054  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
012056  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
012060  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
012062  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
012066  000002                          RTI                     ;;RETURN
012070  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
012074  001430                          BEQ     8$
012076  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
012102  001006                          BNE     5$
012104  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
012106  104401                          TYPE                    ;;TYPE A CR AND LF
012110  001171                          $CRLF
012112  105067  000130                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
012116  000755                          BR      2$              ;;GET NEXT CHARACTER
012120  004767  000056          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
012124  126726  167026          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
012130  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
012132  016746  167016                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
012136  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
012142  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
012144  004767  000032                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
012150  105367  000072                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
3933  012154  000770                        BR      7$              ;;LOOP
3934
3935                                        ;HORIZONTAL TAB PROCESSOR
3936
3937  012156  112716  000040                8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
3938  012162  004767  000014                9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
3939  012166  132767  000007  000052                BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
3940  012174  001372                              BNE     9$              ;;TAB STOP
3941  012176  005726                              TST     (SP)+           ;;POP SPACE OFF STACK
3942  012200  000724                              BR      2$              ;;GET NEXT CHARACTER
3943  012202  105777  166742                $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
3944  012206  100375                              BPL     $TYPEC
3945  012210  116677  000002  166734              MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
3946  012216  122766  000015  000002              CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
3947  012224  001003                              BNE     1$              ;;BRANCH IF NO
3948  012226  1C5067  000014                      CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
3949  012232  000406                              BR      $TYPEX          ;;EXIT
3950  012234  122766  000012  000002  1$:       CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
3951  012242  001402                              BEQ     $TYPEX          ;;BRANCH IF YES
3952  012244  105227                              INCB    (PC)+           ;;COUNT THE CHARACTER
3953  012246  000000                $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
3954  012250  000207                $TYPEX: RTS     PC
3955
```

```
                                .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.
012252  022767  000176  166660  $CKSWR: CMP     @#SWREG,SWR     ;;IS THE SOFT-SWR SELECTED?
012260  001074                          BNE     15$             ;;BRANCH IF NO
012262  105777  166656                  TSTB    @#TKS           ;;CHAR THERE?
012266  100071                          BPL     15$             ;;IF NO, DON'T WAIT AROUND
012270  117746  166652                  MOVB    @#TKB,-(SP)     ;;SAVE THE CHAR
012274  042716  177600                  BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
012300  022726  000007                  CMP     #7,(SP)+        ;;IS IT A CONTROL G?
012304  001062                          BNE     15$             ;;NO, RETURN TO USER
012306  126727  166622  000001          CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
012314  001456                          BEQ     15$             ;;BRANCH IF YES
                                
012316  104401  012777                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
012322  104401  013004          $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
012326  016746  165644                  MOV     $WREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
012332  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012334  104401  013015                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
012340  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
012342  005046                          CLR     -(SP)           ;;THE NEW SWR
012344  105777  166574          7$:     TSTB    @#TKS           ;;CHAR THERE?
012350  100375                          BPL     7$              ;;IF NOT TRY AGAIN
                                
012352  117746  166570                  MOVB    @#TKB,-(SP)     ;;PICK UP CHAR
012356  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
                                
                                
012362  021627  000025          9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
012366  001005                          BNE     10$             ;;BRANCH IF NOT
012370  104401  012772                  TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
012374  062706  000006          20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
012400  000757                          BR      19$             ;;LET'S TRY IT AGAIN
                                
012402  021627  000015          10$:    CMP     (SP),#15        ;;IS IT A <CR>?
012406  001022                          BNE     16$             ;;BRANCH IF NO
012410  005766  000004                  TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
012414  001403                          BEQ     11$             ;;BRANCH IF YES
012416  016677  000002  166514          MOV     2(SP),@#SWR     ;;SAVE NEW SWR
012424  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
012430  104401  001171          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
012434  126727  166475  000001          CMPB    @#INTAG,#1      ;;RE-ENABLE TTY KBD INTERRUPTS?
012442  001003                          BNE     15$             ;;BRANCH IF NOT
012444  012777  000100  166472          MOV     #100,@#TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
012452  000002                  15$:    RTI                     ;;RETURN
012454  004767  177522          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
012460  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
```

```
4012  012464  002420                        BLT     18$             ;;BRANCH IF YES
4013  012466  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
4014  012472  003015                        BGT     18$             ;;BRANCH IF YES
4015  012474  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
4016  012500  005766  000002                TST     2(SP,           ;;IS THIS THE FIRST CHAR
4017  012504  001403                        BEQ     17$             ;;BRANCH IF YES
4018  012506  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
4019  012510  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
4020  012512  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
4021  012514  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
4022  012520  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
4023  012524  000707                        BR      7$              ;;GET THE NEXT ONE
4024  012526  104401  001170        16$:    TYPE    ,$QUES          ;;TYPE ?(CR)(LF)
4025  012532  000720                        BR      20$             ;;SIMULATE CONTROL-U
4026                                .DSABL  LSB
4027
4028
4029                                ;;**********************************************************************
4030                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
4031                                ;*CALL:
4032                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
4033                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
4034                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
4035                                ;
4036
4037  012534  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
4038  012536  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
4039  012544  105777  166374        1$:     TSTB    @$TKS           ;;WAIT FOR
4040  012550  100375                        BPL     1$              ;;A CHARACTER
4041  012552  117766  166370  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
4042  012560  042766  177600  000004        BIC     #^C(177),4(SP)  ;;GET RID OF JUNK IF ANY
4043  012566  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
4044  012574  001013                        BNE     3$              ;;BRANCH IF NO
4045  012576  105777  166342        2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
4046  012602  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
4047  012604  117746  166336                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
4048  012610  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
4049  012614  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
4050  012620  001366                        BNE     2$              ;;IF NOT DISCARD IT
4051  012622  000750                        BR      1$              ;;YES, RESUME
4052  012624  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
4053  012632  002407                        BLT     4$              ;;BRANCH IF YES
4054  012634  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
4055  012642  003003                        BGT     4$              ;;BRANCH IF YES
4056  012644  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
4057  012652  000002                4$:     RTI                     ;;GO BACK TO USER
4058                                ;;**********************************************************************
4059                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4060                                ;*CALL:
4061                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
4062                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4063                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
4064
4065  012654  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
4066  012656  012703  012762        1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
4067  012662  022703  012772        2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
```

```
4068  012666  101405                          BLOS    4$              ;;BR IF YES
4069  012670  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
4070  012672  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
4071  012674  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
4072  012700  001003                          BNE     3$              ;;SKIP IF NOT
4073  012702  104401  001170          4$:     TYPE    ,$QUES          ;;TYPE A '?'
4074  012706  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
4075  012710  111367  000044          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
4076  012714  104401  012760                  TYPE    ,9$
4077  012720  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
4078  012724  001356                          BNE     2$              ;;LOOP IF NOT RETURN
4079  012726  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
4080  012732  104401  001172                  TYPE    ,$LF            ;;TYPE A LINE FEED
4081  012736  012603                          MOV     (SP)+,R3        ;;RESTORE R3
4082  012740  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
4083  012742  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
4084  012750  012766  012762  000004          MOV     #$TTYIN,4(SP)
4085  012756  000002                          RTI                     ;;RETURN
4086  012760  000000          9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
4087  012761  000000                  .BYTE   0                       ;;TERMINATOR
4088  012762  000010          $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
4089  012772  052536  005015  000000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
4090  012777  000136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
4091  013004  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
4092  013012  020075  000000
4093  013015  000040  047040  053505  $MNEW:  .ASCIZ  /  NEW = /
4094  013022  036440  000040
```

```
                                .SBTTL  APT COMMUNICATIONS ROUTINE

                                ;;*****************************************************************
013026  112767  000001  000236  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
013034  112767  000001  000226  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
013 42  000403                  BR      $ATYC
013044  112767  000001  000220  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
013052                          $ATYC:
013052  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013 54  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013056  105767  000206          TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
013062  001450                  BEQ     5$              ;;IF NOT:  BR
013064  122767  000001  166122  CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
013072  001031                  BNE     3$              ;;IF NOT:  BR
013 74  132767  000100  166113  BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
013102  001425                  BEQ     3$              ;;IF NOT:  BR
013104  017600  000004          MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
013110  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
013116  005767  166052          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
013122  001375                  BNE     1$              ;;IF NOT:  WAIT
013124  010067  166060          MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
013130  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
013132  001376                  BNE     2$
013134  166700  166050          SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
013140  006200                  ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
013142  010067  166044          MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
013146  012767  000004  166020  MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
013154  000413                  BR      5$
013156  017667  000004  000016  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
013164  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
013172  016746  164600          MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
013176  004767  176566          JSR     PC,$TYPE        ;;CALL TYPE MACRO
013202  000000                  4$:     .WORD   0
013204                          5$:
013204  105767  000062          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
013210  001416                  BEQ     12$             ;;IF NOT:  BR
013212  005767  165776          TST     $ENV            ;;RUNNING UNDER APT?
013216  001413                  BEQ     12$             ;;IF NOT:  BR
013220  005767  165750          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
013224  001375                  BNE     11$             ;;IF NOT:  WAIT
013226  017667  000004  165742  MOV     @4(SP),$FATAL   ;;GET ERROR #
013234  062766  000002  000004  ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
013242  005267  165726          INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
013246  105067  000020          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
013252  105067  000013          CLRB    $LFLG           ;;CLEAR LOG FLAG
013256  105067  000006          CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
013262  012601                  MOV     (SP)+,R1        ;;POP STACK INTO R1
013264  012600                  MOV     (SP)+,R0        ;;POP STACK INTO R0
013266  000207                  RTS     PC              ;;RETURN
013270     000                  $MFLG:  .BYTE   0       ;;MESSG. FLAG
013271     000                  $LFLG:  .BYTE   0       ;;LOG FLAG
013272     000                  $FFLG:  .BYTE   0       ;;FATAL FLAG
            013274                          .EVEN
            000200          APTSIZE=200
            000001          APTENV=001
            000100          APTSPOOL=100
```

```
M3INDEC-11-DVDVC-A      MACY11 27(1006)  08-AUG-77  09:20  PAGE 96
DVDVCA.P11    08-AUG-77 09:16            APT COMMUNICATIONS ROUTINE                    SEQ 0095

  4151          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

013274                          $ERROR:
013274  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
013276  105267  165601          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
013302  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
013304  016777  165572  165630          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
013312  032777  002000  165620          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
013320  001402                          BEQ     1$              ;;NO - SKIP
013322  104401  001164                  TYPE    ,$BELL          ;;RING BELL
013326  005267  165560          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
013332  011667  165560                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
013336  162767  000002  165552          SUB     #2,$ERRPC
013344  117767  165546  165542          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
013352  032777  020000  165560          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
013360  001004                          BNE     20$             ;;SKIP TYPEOUTS
013362  004767  175706                  JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
013366  104401  001171                  TYPE    ,$CRLF
013372                          20$:
013372  122767  000001  165614          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
013400  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
013402  116767  165506  000004          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
013410  004767  177430                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
013414     000                  21$:    .BYTE   0
013415     000                          .BYTE   0
013416  000777                  22$:    BR      22$             ;;APT ERROR LOOP
013420  005777  165514          2$:     TST     @SWR            ;;HALT ON ERROR
013424  100002                          BPL     3$              ;;SKIP IF CONTINUE
013426  000000                          HALT                    ;;HALT ON ERROR!
013430  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
013432  032777  001000  165500  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
013440  001402                          BEQ     4$              ;;BR IF NO
013442  016716  165442                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
013446  005767  165510          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
013452  001402                          BEQ     5$              ;;BR IF NONE
013454  016716  165502                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
013460                          5$:
013460  022737  011552  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
013466  001001                          BNE     6$              ;;BRANCH IF NO
013470  000000                          HALT                    ;;YES
013472                          6$:
013472  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

013474                          $SCOPE:
013474  104407                         CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
013476  032777  040000  165434  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
013504  001114                         BNE     $OVER           ;;YES IF SW14=1
                                ;*****START OF CODE FOR THE XOR TESTER*****
013506  000416                 $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
013510  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
013514  012737  013534  000004         MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
013522  005737  177060                 TST     @#177060        ;;TIME OUT ON XOR?
013526  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
013532  000463                         BR      $SVLAD          ;;GO TO THE NEXT TEST
013534  022626                 5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
013536  012637  000004                 MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
013542  000423                         BR      7$              ;;LOOP ON THE PRESENT TEST
013544                          6$:;*****END OF CODE FOR THE XOR TESTER*****
013544  032777  000400  165366         BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
013552  001404                         BEQ     2$              ;;BR IF NO
013554  127767  165360  165320         CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
013562  001465                         BEQ     $OVER           ;;BR IF YES
013564  105767  165313         2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
013570  001421                         BEQ     3$              ;;BR IF NO
013572  126767  165317  165303         CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
013600  101015                         BHI     3$              ;;BR IF NO
013602  032777  001000  165330         BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
013610  001404                         BEQ     4$              ;;BR IF NO
013612  016767  165272  165266  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
013620  000446                         BR      $OVER
013622  105067  165255         4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
013626  005067  165326                 CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
013632  000415                         BR      1$              ;;ESCAPE TO THE NEXT TEST
013634  032777  004000  165276  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
013642  001011                         BNE     1$              ;;BR IF YES
013644  005767  165332                 TST     $PASS           ;;IF FIRST PASS OF PROGRAM
013650  001406                         BEQ     1$              ;;      INHIBIT ITERATIONS
013652  005267  165226                 INC     $ICNT           ;;INCREMENT ITERATION COUNT
013656  026767  165276  165220         CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
013664  002024                         BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
013666  012767  000001  165210  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
013674  016767  000052  165256         MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
013702  105267  165174         $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
013706  116767  165170  165264         MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 4262  013714  011667  165166                 MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 4263  013720  011667  165164                 MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 4264  013724  005067  165232                 CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 4265  013730  112767  000001  165157         MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 4266  013736  016777  165140  165176 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 4267  013744  016716  165136                 MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 4268  013750  000002                         RTI                     ;;FIXES PS
 4269  013752  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

013754                          $TYPDS:
013754  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013756  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013760  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
013762  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
013764  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
013766  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
013772  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
013776  100004                         BPL     1$              ;;BR IF INPUT IS POS.
014000  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
014002  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
014010  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
014012  012703  014170                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
014016  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
014022  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
014024  016001  014160                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
014030  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
014032  002402                         BLT     4$              ;;BR IF DONE
014034  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
014036  000774                         BR      3$
014040  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
014042  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
014044  001002                         BNE     5$              ;;FALL THROUGH IF 0
014046  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
014050  100407                         BMI     7$              ;;BR IF YES
014052  106316                 5$:     ASLB    (SP)            ;;MSD?
014054  103003                         BCC     6$              ;;BR IF NO
014056  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
014064  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
014070  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
014074  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
014076  005720                         TST     (R0)+           ;;JUST INCREMENTING
014100  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
014104  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
014106  003002                         BGT     8$              ;;GO TO EXIT
014110  010502                         MOV     R5,R2           ;;GET THE LSD
014112  000764                         BR      6$              ;;GO CHANGE TO ASCII
014114  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
014116  100003                         BPL     9$              ;;BR IF NO
014120  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
014126  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
014130  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
014132  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
014134  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 4326  014136  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 4327  014140  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 4328  014142  104401  014170                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4329  014146  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4330  014154  012616                        MOV     (SP)+,(SP)
 4331  014156  000002                        RTI                     ;;RETURN TO USER
 4332  014160  023420                $DTBL:  10000.
 4333  014162  001750                        1000.
 4334  014164  000144                        100.
 4335  014166  000012                        10.
 4336  014170  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

014200  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
014204  116667  000001  000211          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
014212  112667  000207                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
014216  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
014222  000406                          BR      $TYPON
014224  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
014232  112767  000006  000165          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
014240  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
014246  010346                          MOV     R3,-(SP)        ;;SAVE R3
014250  010446                          MOV     R4,-(SP)        ;;SAVE R4
014252  010546                          MOV     R5,-(SP)        ;;SAVE R5
014254  116704  000145                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
014260  005404                          NEG     R4
014262  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
014266  110467  000132                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
014272  116704  000125                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
014276  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
014302  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
014304  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
014306  000404                          BR      3$              ;;GO DO MSB
014310  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
014312  006105                          ROL     R5
014314  006105                          ROL     R5
014316  010503                          MOV     R5,R3
014320  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
014322  105367  000076                  DECB    $OMODE          ;;TYPE THIS DIGIT?
014326  100016                          BPL     7$              ;;BR IF NO
014330  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
014334  001002                          BNE     4$              ;;TEST FOR 0
014336  005704                          TST     R4              ;;SUPPRESS THIS 0?
014340  001403                          BEQ     5$              ;;BR IF YES
```

```
4393  014342  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
4394  014344  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
4395  014350  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
4396  014354  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
4397  014360  104401  014420                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
4398  014364  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
4399  014370  003347                  BGT     2$                      ;;BR IF MORE TO DO
4400  014372  002402                  BLT     6$                      ;;BR IF DONE
4401  014374  005204                  INC     R4                      ;;INSURE LAST DIGIT ISN'T A BLANK
4402  014376  000744                  BR      2$                      ;;GO DO THE LAST DIGIT
4403  014400  012605          6$:     MOV     (SP)+,R5                ;;RESTORE R5
4404  014402  012604                  MOV     (SP)+,R4                ;;RESTORE R4
4405  014404  012603                  MOV     (SP)+,R3                ;;RESTORE R3
4406  014406  016666  000002  000004  MOV     2(SP),4(SP)             ;;SET THE STACK FOR RETURNING
4407  014414  012616                  MOV     (SP)+,(SP)
4408  014416  000002                  RTI                             ;;RETURN
4409  014420     000          8$:     .BYTE   0                       ;;STORAGE FOR ASCII DIGIT
4410  014421     000                  .BYTE   0                       ;;TERMINATOR FOR TYPE ROUTINE
4411  014422     000          $OCNT:  .BYTE   0                       ;;OCTAL DIGIT COUNTER
4412  014423     000          $OFILL: .BYTE   0                       ;;ZERO FILL SWITCH
4413  014424  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.

014426  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
014430  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
014434  005740                          TST     -(R0)           ;;BACKUP BY 2
014436  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
014440  006300                          ASL     R0              ;;POSITION FOR INDEXING
014442  016000  014462                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
014446  000200                          RTS     R0              ;;GO TO ROUTINE

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

014450  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
014452  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
014460  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
014462  014450                  $TRPAD: .WORD   $TRAP2
014464  011770                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
014466  014224                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
014470  014200                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
014472  014240                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
014474  013754                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

014476  012322                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

014500  012252                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
014502  012534                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
014504  012654                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
        000001                  .END
```

```
ABORSE = 175610            1#    945     986
ACOW1  = 000000          945
ACOW2  = 000000          945
ACPUOP= 000000           945     960
ADDRES  011254          3680*  3691*  3705*  3728   3735#
ADOW0  = 000000          945
ADOW1  = 000000          945
ADOW10= 000000           945
ADOW11= 000000           945
ADOW12= 000000           945
ADOW13= 000000           945
ADOW14= 000000           945
ADOW15= 000000           945
ADOW2  = 000000          945
ADOW3  = 000000          945
ADOW4  = 000000          945
ADOW5  = 000000          945
ADOW6  = 000000          945
ADOW7  = 000000          945
ADOW8  = 000000          945
ADOW9  = 000000          945
ADEVCT= 000000           945     951
ADEVM  = 100000            1#    945     987
AENV   = 000000          945     956
AENVM  = 000000          945     957
AFATAL= 000000           945     948
AMADR1= 000000           945     973
AMADR2= 000000           945     977
AMADR3= 000000           945     980
AMADR4= 000000           945     983
AMANS1= 000000           945     967
AMANS2= 000000           945     975
AMANS3= 000000           945     978
AMANS4= 000000           945     981
AMSCAD= 000000           945     953
AMSCLR= 000000           945     954
AMSCTY= 000000           945     947
AMTYP1= 000000           945     968
AMTYP2= 000000           945     976
AMTYP3= 000000           945     979
AMTYP4= 000000           945     982
APASS  = 000000          945     950
APRIOR= 000000           945
APTCON  011266          1673    1805    1879   1958   2108   2309   2633   2824   2968   3108   3651*  3719*  3740#
APTCSU= 000040          3907    4151#
APTENV= 000001          3712    3900    4107   4149#  4183
APTSIZ= 000200          1056    4148#
APTSPO= 000100          3902    4109    4150#
ASHREG= 000000           945     958
ATESTN= 000000           945     949
AUNIT  = 000000          945     952
AUSHR  = 011110            1#    945     959
AVECT1= 000300             1#    945     984
AVECT2= 000000           945     985
BAUD   = 007400          848#
BAUDRA  005716          2473#
```

```
BITMAS  011250          1095#   3657    3676#   3689#   3697    3700    3733#
BIT0   = 000001          723#   3351    3374
BIT00  = 000001          713#    723     772     792     813     834
BIT01  = 000002          712#    722     791     833
BIT02  = 000004          711#    721     790     811     832
BIT03  = 000010          710#    720     789     831
BIT04  = 000020          709#    719     788     830
BIT05  = 000040          708#    718     787     829
BIT06  = 000100          707#    717     766     786     807     828
BIT07  = 000200          706#    716     765     785     806     827
BIT08  = 000400          705#    715    4236
BIT09  = 001000          704#    714    4194    4244
BIT1   = 000002          722#   3362    3381
BIT10  = 002000          703#   4171
BIT11  = 004000          702#    761     801    4251
BIT12  = 010000          701#    780     800
BIT13  = 020000          700#    779     799    4178
BIT14  = 040000          699#    778     798    4222
BIT15  = 100000          698#    777     797    1095    3700
BIT2   = 000004          721#   2409    2437
BIT3   = 000010          720#   2875    2933
BIT4   = 000020          719#   2894    2940
BIT5   = 000040          718#
BIT6   = 000100          717#
BIT7   = 000200          716#
BIT8   = 000400          715#
BIT9   = 001000          714#
BPTVEC= 000014           730#
BPEAK  = 000001          813#   1212    1225    1228    1242    1245    1259    1264    3334
BRK    = 010000          849#   1199    3318
CKSWR  = 104407         4167    4193    4221    4453#
CLR    = 000000          751#   1687    1689    1974    3464
COMSPD= 000100           843#   2960
CONADR  011262          3705    3738#
CONSOL  011267          1571    3653*   3703*   3741#
CONVEC  011264          3707    3739#
COUNT   004666          1976*   1994*   2000    2004    2092#
CR     = 000015          638#   3946    3956
CRLF   = 000200          639#   1080    1086    1087    1088    3401    3758    3917    3956
CYCLE   011030          1098    3639#
DATA   = 000017          840#   3548    3549
DATLNG  010640          2848    3000    3120    3530#
DDISP  = 177570          645#    925    1044
DISPLA  001142           926#   1044*   1052*   4170*   4266*
DISPRE  000174           861#   1052
DLADD   001254          1006#   1102*   1107    1110    1112    1115    1118    1121    1137    1169    3402    3780
DLADDR= 175610           747#   1006    1008    1009    1010    1011    1012
DLVEC   001256          1007#   1105*   2503    2644    2801    3139    3408    3786
DSWR   = 177570          644#    925    1043
EMTVEC= 000030           733#   1027*   1028*
ERRBIT= 100000           852#   2106    3356
ERRCHK  010334          2330*   2409*   2437    2834*   2875*   2894*   2933    2940    3328*   3351*   336?*   3374    3381
                        3389#
ERRCNT  007770          3154*   3191    3203#   3262    3273*
ERROR  = 100000           777#   2189    2263
ERRORF  003776          1687*   1734*   1761    1789#
```

```
ERRVEC= 000004          726#  1041  1042*  1053*  4227  4228*  4230*  4233*
EVENOD= 000040          842#
EXITFL  004000         1689*  1770*  1777*  1781  1790#
FALSE = 000000         1958   2108   2309   3442#  3452  3651  3653
FLAG    010632         1974*  1989*  1998   3452*  3479*  3484  3507  3519#
FRERR = 020000          779#  3359
GNS   = ****** U        860   1079   1086   1087   1088  3400  3406  3412  3757  3763  3772  3778  3784
                       4445   4446   4447   4448   4449  4451  4453  4454  4455
GTSWR = 104406         1074   4451#
HOLD    010034         3216*  3217*  3220   3236#
HOLDSC  010633         3464*  3469*  3475   3520#
HOWLON= 000000         3424#  3450
HT    = 000011          636#  3915   3956
I       001272         1013#  1146*  1167*  1170  1172  3553*  3556*  3558
ILLMEM= 000004          738#  1141   1177
INITFL  011252         1093*  3660   3663*  3734#
INTFLA  011026         1153*  1158   2499*  2549  2553  2575*  2591  2652*  2692  2699  2733*  2747  2766*
                       2784   3634*  3637#
INTSRV  011020         1142   2511   2645   3624#
IOTVEC= 000020          731#  1025*  1026*
ITCLRE  005262         2277#
LF    = 000012          637#  3950   3956
LOOP    002136         1096#  3411   3869
MAINT = 000004          811#  1297   1310   1313  1327  1330  1344  1349  1585  1610  1816  1843  1890
                       1916   1971   1939   1944  1972  2010  2039  2061  2083  2132  2173  2195  2219
                       2250   2268   2336   2435  2635  2697  2717  2729  2751  2756  2764  2797  2837
                       2919   2931   2979   3169  3188  3331
MAINTJ= 040000          851#  128    1803   1877  1960  2111  2306  2631  2822  3106  3316
MASK  = 000000         3531#  3545*  3561*  3563*  3568*
MAX     004664         2000   2004   2091#
MYTYPE  011274         3753#  4180
NEW     005750         2359*  2361*  2377*  2379*  2388  2396  2398  2416  2418  2479#
NOCONM  011270         3742#
NUMBER  007772         3184   3204#  3226   3282
NUMBR   010734         3547*  3548   3550*  3558   3571#
OK      011260         3737#
OLD     005754         2332*  2334*  2388   2396   2398  2416*  2418*  2480#
ORERR = 040000          778#  2168   2213   2245
PARITY= 000020          841#
PASS    003774         1682*  1764   1788#
PBAUDS= 004000          801#
PBAUD0= 010000          800#
PBAUD1= 020000          799#
PBAUD2= 040000          798#
PBAUD3= 100000          797#
PBR   = 000200          844#  2304
PERR  = 010000          780#
PIRQ  = 177772          643#
PIRQVE= 000240          737#
PR0   = 000000          660#  2531   2580   2661   2841  2993  3133
PR1   = 000040          661#
PR2   = 000100          662#
PR3   = 000140          663#
PR4   = 000200          664#
PR5   = 000240          665#
PR6   = 000300          666#
```

```
PR7     = 000340   667#  1143   1369   2512   2646   2769   3144   3150
PS      = 177776   640#   641
PSW     = 177776   641#
PWRVEC= 000024   732#  1031*  1032*  3836*  3837*  3846*  3852*  3864*  3865*
RATES   005676  2350   2431   2447#
RBUF    001262  1009#  1112*  1113*  1931   2075   2164   2213   2245   2263   2347   2899   3003   3060
                3249   3347   3359
RCSR    001260  1008#  1110*  1457   1470*  1473   1487*  1490   1504*  1509   1540   1605   1831   1859
                1905   1934   1986   2026   2030   2033   2056   2078   2367   2658*  2682*  2736*  2773*
                2886   2990*  3042   3087*  3176*  3286*  3343
RCVRAC= 004000   761#  1605   1986   2026   2033
RCVRDO= 000200   765#  1540   1832   1859   1906   1934   2030   2056   2078   2367   2887   3043   3343
RCVRIE= 000100   766#  1457   1470   1473   1487   1490   1504   1509   2658   2682   2736   2773   3176
                3286
RDATA0= 000001   792#
RDATA1= 000002   791#
RDATA2= 000004   790#
RDATA3= 000010   789#
RDATA4= 000020   788#
RDATA5= 000040   787#
RDATA6= 000100   786#
RDATA7= 000200   785#
ROCHR   = 104410  4069   4454#
ROLIN   = 104411  4455#
RORRUN= 000001   772#
REC     010040  3142   3245#
REG     = 000004  3424#  3448
REGSAV  010626  3448*  3461   3517#
RESVEC= 000010   727#
RHLD    010130  3253*  3254*  3258   3266   3295#
R0050   005676  2456#
R0070   005677  2457#
R0110   005700  2458#
R0135   005701  2459#
R0150   005702  2460#
R0200   005705  2463#
R0300   005703  2461#
R0600   005704  2462#
R10000  005715  2471#
R1800   005706  2464#
R2000   005707  2465#
R2400   005710  2466#
R3600   005711  2467#
R4800   005712  2468#
RSSTAC  001334  1015#  1124
R7200   005713  2469#
R9600   005714  2470#
SB      007774  3205#  3266*
SET     = 177777   750#  1703   1734   1745   1761   1770   1777   1781   1830   1904   1989   1998   2519
                2674   2866   2885   3020   3041   3469
SETCLR= 000006  3424#  3475
SPECIA  011074  3670#  3827
STACK   = 001100   631#  1023
START   001336   864   1016#
STKLMT= 177774   642#
SWR     001140   925#  1021   1043*  1045   1051*  1058*  1072   3844   3857*  3966   4003*  4171   4178
```

```
                        4190    4194    4222    4236    4238    4244    4251
SWREG   000176           862#    1051    1072    3966    3979
SW0     = 000001         695#
SW00    = 000001         685#     695
SW01    = 000002         684#     694
SW02    = 000004         683#     693
SW03    = 000010         682#     692
SW04    = 000020         681#     691
SW05    = 000040         680#     690
SW06    = 000100         679#     689
SW07    = 000200         678#     688
SW08    = 000400         677#     687
SW09    = 001000         676#     686
SW1     = 000002         694#
SW10    = 002000         675#
SW11    = 004000         674#
SW12    = 010000         673#
SW13    = 020000         672#
SW14    = 040000         671#
SW15    = 100000         670#
SW2     = 000004         693#
SW3     = 000010         692#
SW4     = 000020         691#
SW5     = 000040         690#
SW6     = 000100         689#
SW7     = 000200         688#
SW8     = 000400         687#
SW9     = 001000         686#
TBITVE= 000014           728#
TBUF    001270          1012#   1121*   1122*   1697*   1725*   1823*   1898*   1982*   2144*   2154*   2236*   2356*   2668*
                        2739*   2775*   2881*   3037*   3220*   3337*
TCSR    001264          1010#   1115*   1116*   1212    1225*   1228    1242*   1245    1259*   1264    1297    1310*   1313
                        1327*   1330    1344*   1349    1379    1392*   1395    1409*   1412    1426*   1431    1610*   1643
                        1704    1730    1746    1816*   1843*   1890*   1916*   1921*   1939*   1944*   1972*   2010*   2039*
                        2061*   2083*   2132*   2173*   2195*   2219*   2250*   2268*   2336*   2435*   2520    2528*   2538*
                        2578*   2655*   2675    2697*   2717*   2729*   2751*   2756*   2764*   2797*   2837*   2867    2919*
                        2931*   2979*   3021    3169*   3173*   3188*   3230*   3331*   3334*
TCSRHI  001266          1011#   1118*   1119*   2350*   2431*
TDATA0= 000001           834#
TDATA1= 000002           833#
TDATA2= 000004           832#
TDATA3= 000010           831#
TDATA4= 000020           830#
TDATA5= 000040           829#
TDATA6= 000100           828#
TDATA7= 000200           827#
TIME    = 000000        3580#   3591
TIMER   010462          1707    1749    1834    1908    2523    2678    2870    2889    3024    3045    3423#
TIMSAV  010630          3450*   3486    3497*   3518#
TKVEC   = 000060         735#
TPVEC   = 000064         736#
TRAN    007776          3148    3212#
TRAPVE= 000034           734#    1029*   1030*
TRTVEC= 000014           729#
TRUE    = 000001        1571    1673    1805    1879    2633    2824    2968    3108    3444#   3479    3484    3507    3703
                        3719
```

F09


```
TST1    002252          1133#
TST10   003456          1593    1632#
TST11   003516          1668#
TST12   004002          1677    1787    1798#
TST13   004154          1810    1872#
TST14   004344          1884    1954#
TST15   004670          1968    2015    2045    2066    2090    2101#
TST16   005326          2119    2127    2180    2203    2225    2258    2276    2294#
TST17   005760          2317    2325    2445    2494#
TST2    002420          1194#
TST20   006262          2627#
TST21   006746          2636    2818#
TST22   007254          2829    2954#
TST23   007566          2966    2972    3031    3054    3101#
TST24   010134          3113    3201    3312#
TST25   010336          3323    3388    3396#
TST3    002602          1203    1279#
TST4    002764          1288    1365#
TST5    003140          1449#
TST6    003302          1530#
TST7    003342          1563#
TYPOS = 104405          3415    3816    4449#
TYPE  = 104401          1066    1086    1087    1088    3398    3404    3410    3417    3755    3761    3770    3776    3782
                        3788    3814    3817    3866    3920    3977    3978    3981    3994    4005    4024    4073    4076
                        4080    4173    4181    4328    4397    4445#
TYPOC = 104402          3403    3409    3760    3775    3781    3787    3980    4446#
TYPON = 104404          4448#
TYPOS = 104403          3767    4447#
VECTOR  011256          3682*   3693*   3707*   3736#
WAIT    010740          2149    2159    2241    2544    2588    2687    2744    2780    3494    3579#
WAS     007775          3206#   3268*
WHICHB= 000002          3424#   3461
WRAP  = 020000           850#   1578    2958
XMITIE= 000100           807#   1379    1392    1395    1409    1412    1426    1431    2528    2538    2578    3173    3230
XMITRD= 000200           806#   1643    1705    1730    1747    2521    2676    2868    3022
ZZZ     010036          3234    3238#
ZZZZ    010132          3293    3296#
$APTHD  001000           885     891#
$ASTAT= ****** U        4129    4144
$ATYC   013052          4100    4102#
$ATY1   013026          4098#
$ATY3   013034          3905    4099#
$ATY4   013044          4101#   4186
$AUTOB  001134           922#   1076*   3974    4095
$BASE   001250           986#   3680
$ERROR  001122           917#
$FDAT   001126           919#
$FILL   001164           937#   4173    4206
$GCMLE= 177777             1#
$CHARC  012246          3922*   3932*   3939    3948*   3953#
$CKSWR  012252          3966#   4453
$CNTFG  001100           905#   1018    1019    1027    1033    1034    1035
$CNT3 = 000000           935#
$CNTLG  012777          3977    4090#
$CNTLU  012772          3994    4089#
$CPUOP  001222           960#
```

$CRLF 001171 939# 3417 3788 3921 3956 4005 4089 4181 4206
$ CLK 014170 4294 4328 4336#
$ VCT 001204 951# 3678# 3730#
$ VM 001252 987# 1083 3697
$ MGN 011562 3810 3819 3825#
$ PC 014160 4297 4332#
$ PS0 011552 871 1064 3821# 4201
$ PCT 011520 1033 3812#
$ PMG 011571 3814 3[illegible]9#
$ PULL 011566 3817 3828#
$ PV 001214 956# 1070 3712 3900 4107 4131 4183
$ PVM 001215 957# 1056 3902 3907 4109
$ PS 011464 3668 3802#
$ SPCT 011512 1033# 3809# 3813
$ SFLG 001103 908# 4168# 4206 4211 4240 4242 4248# 4270
$SC-MAX 001115 914# 1036# 4242 4265# 4270
$ERRFL= 000000 1093# 1094 1095# 1096 1098# 1102# 1103 1105# 1106 1107# 1108 1110# 1111
1112# 1114 1115# 1117 1118# 1120 1121# 1123 1124# 1125 1137# 1138 1140#
1141# 1142# 1143# 1144# 1145 1146# 1147 1150# 1151 1153# 1154 1167# 1168
1169# 1171 1177# 1178# 1180# 1181# 1182 1202# 1203 1209# 1210 1223# 1224
1225# 1226 1239# 1240 1242# 1243 1256# 1257 1259# 1260 1287# 1288 1294#
1295 1308# 1309 1310# 1311 1324# 1325 1327# 1328 1341# 1342 1344# 1345
1376# 1377 1390# 1391 1392# 1393 1406# 1407 1409# 1410 1423# 1424 1426#
1427 1454# 1455 1468# 1469 1470# 1471 1484# 1485 1487# 1488 1501# 1502
1504# 1505 1538# 1539 1592# 1593 1602# 1603 1610# 1611 1640# 1641 1676#
1677 1682# 1683 1687# 1688 1689# 1690 1697# 1698 1702# 1725# 1726 1734#
1735 1744# 1770# 1771 1777# 1778 1809# 1810 1816# 1817 1819# 1820 1823#
1824 1829# 1843# 1844 1852# 1853 1883# 1884 1890# 1891 1892# 1893 1898#
1899 1903# 1916# 1917 1921# 1922 1931# 1932 1939# 1940 1944# 1945 1967#
1968 1972# 1973 1974# 1975 1976# 1977 1982# 1983 1989# 1990 1994# 1995
2010# 2011 2014# 2015 2039# 2040 2044# 2045 2061# 2062 2065# 2066 2075#
2076 2083# 2084 2116# 2119 2126# 2127 2132# 2133 2137# 2138 2144# 2145
2147# 2154# 2155 2157# 2164# 2165 2173# 2174 2179# 2180 2187# 2188 2195#
2196 2202# 2203 2209# 2210 2219# 2220 2224# 2225 2231# 2232 2236# 2237
2239# 2250# 2251 2257# 2258 2268# 2269 2316# 2317 2324# 2325 2330# 2331
2332# 2333 2334# 2335 2336# 2337 2339# 2340 2342 2346 2347# 2348 2350#
2351 2353# 2354 2356# 2357 2359# 2360 2361# 2362 2371# 2372 2377# 2378
2379# 2380 2409# 2410 2416# 2417 2418# 2419 2424# 2428 2429# 2430 2431#
2432 2435# 2436 2499# 2500 2503# 2504 2506# 2507 2509# 2510# 2511# 2512#
2513# 2514 2515# 2516 2518# 2528# 2529 2538# 2539 2542# 2572# 2573 2575#
2576 2578# 2579 2586# 2605# 2606# 2608# 2609# 2610 2643# 2644# 2645# 2646#
2647# 2648 2650# 2651 2652# 2653 2655# 2656 2658# 2659 2668# 2669 2673#
2682# 2683 2685# 2697# 2698 2717# 2718 2727# 2728 2729# 2730 2733# 2734
2736# 2737 2739# 2740 2742# 2751# 2752 2756# 2757 2762# 2763 2764# 2765
2766# 2767 2773# 2774 2775# 2776 2778# 2797# 2798 2801# 2802 2805# 2806#
2808# 2809# 2810 2828# 2829 2834# 2835 2837# 2838 2847# 2853# 2854 2856
2860 2865# 2875# 2876 2881# 2882 2884# 2894# 2895 2899# 2900 2907# 2909
2910# 2911 2919# 2920 2931# 2932 2965# 2966 2971# 2972 2979# 2980 2990#
2991 2999# 3003# 3004 3006# 3007 3009 3013 3019# 3030# 3031 3037# 3038
3040# 3053# 3054 3060# 3061 3064# 3066 3067# 3068 3087# 3088 3112# 3113
3119# 3139# 3140 3142# 3143 3144# 3145 3148# 3149 3150# 3151 3154# 3155
3157# 3158 3160# 3161 3163# 3164 3169# 3170 3173# 3174 3176# 3177 3188#
3189 3216# 3218 3220# 3221 3223# 3224 3230# 3231 3249# 3251 3253# 3255
3266# 3267 3268# 3269 3273# 3274 3279# 3280 3286# 3287 3322# 3323 3328#
3329 3331# 3332 3334# 3335 3337# 3338 3351# 3352 3362# 3363 3448# 3449
3450# 3451 3452# 3453 3464# 3465 3469# 3470 3479# 3480 3492# 3497# 3498

```
                    3510#  3511   3514#  3516   3545#  3546   3547#  3551   3553#  3554   3556   3560   3561#
                    3562   3563#  3564   3568#  3569   3591#  3592   3593#  3594   3596   3600   3601#  3602
                    3604   3608   3634#  3635   3651#  3652   3653#  3654   3663#  3664   3668#  3672#  3673
                    3676#  3677   3678#  3679   3680#  3681   3682#  3683   3687#  3688   3689#  3690   3691#
                    3692   3693#  3694   3703#  3704   3705#  3706   3707#  3708   3719#  3720   3728#  3729
                    3730#  3731
SERROR  013274      1027   4166#
SERRPC  001116       915#  3774   4175*  4176*  4177   4206
SERRTB  001254      1004#
SERTTL  001112       912#  3414   3416*  4174*  4206
SESCAP  001162       936#  1035*  4197   4199   4206   4264*
SETABL  001214       955#
SETEND  001254       897    988#
SFATAL  001176       948#  3765*  3766   4135*
SFFLG   013272      4098*  4101*  4129   4138*  4146#
SFILLC  001156       933#  3925   3956
SFILLS  001155       932#  3956
SFSAND= 000310         1#  1085   1160   1201   1214   1230   1247   1266   1286   1299   1315   1332   1351
                    1381   1397   1414   1433   1459   1475   1492   1511   1542   1573   1580   1587   1607
                    1645   1675   1732   1763   1766   1805   1861   1879   1936   1960   1988   2006   2028
                    2032   2035   2058   2080   2108   2113   2170   2191   2215   2247   2265   2306   2311
                    2366   2369   2390   2398   2439   2551   2555   2593   2633   2694   2701   2749   2786
                    2824   2916   2935   2942   2960   2970   3073   3108   3193   3228   3260   3264   3284
                    3318   3349   3358   3361   3376   3383   3463   3477   3509   3659   3662   3702   3714
                    3717
SFSBAD= 000401         1#  1085   1160   1201   1214   1230   1247   1266   1286   1299   1315   1332   1351
                    1381   1397   1414   1433   1459   1475   1492   1511   1542   1573   1580   1587   1607
                    1645   1675   1732   1763   1766   1808   1861   1882   1936   1962   1988   2006   2028
                    2032   2035   2058   2080   2110   2113   2170   2191   2215   2247   2265   2308   2311
                    2366   2369   2390   2400   2439   2551   2555   2593   2636   2694   2701   2749   2786
                    2827   2916   2935   2942   2963   2970   3073   3111   3193   3228   3260   3264   3284
                    3321   3349   3358   3361   3376   3383   3463   3477   3509   3659   3662   3702   3714
                    3717
SFSBLA= 000170         1#
SFSCAS= 000150         1#
SFSDEC= 000220         1#
SFSGOO= 000400         1#  1082   1083   1158   1199   1212   1228   1245   1264   1284   1297   1313   1330
                    1349   1379   1395   1412   1431   1457   1473   1490   1509   1540   1571   1578   1585
                    1605   1643   1673   1712   1730   1752   1761   1764   1803   1839   1859   1877   1912
                    1934   1958   1986   2004   2025   2026   2030   2033   2056   2078   2106   2111   2168
                    2189   2213   2245   2263   2304   2309   2363   2364   2367   2388   2396   2437   2549
                    2553   2591   2631   2692   2699   2747   2784   2822   2873   2892   2914   2933   2940
                    2958   2968   3027   3048   3071   3106   3191   3226   3258   3262   3282   3316   3347
                    3356   3359   3374   3381   3461   3475   3507   3657   3660   3700   3712   3715
SFSIF = 000110         1#  1158   1164   1199   1205   1212   1218   1228   1234   1245   1251   1264   1270
                    1284   1290   1297   1303   1313   1319   1330   1336   1349   1355   1379   1385   1395
                    1401   1412   1418   1431   1437   1457   1463   1473   1479   1490   1496   1509   1515
                    1540   1551   1571   1575   1578   1582   1585   1589   1595   1597   1599   1605   1622
                    1643   1654   1673   1679   1712   1717   1730   1738   1752   1757   1759   1761   1764
                    1772   1774   1779   1803   1805   1812   1839   1847   1859   1865   1877   1879   1886
                    1912   1923   1934   1946   1958   1960   1964   1970   1986   1991   1996   2004   2017
                    2030   2033   2047   2049   2056   2068   2078   2087   2106   2108   2111   2115   2121
                    2123   2129   2168   2182   2189   2205   2213   2227   2245   2260   2263   2273   2304
                    2306   2309   2313   2319   2321   2327   2367   2373   2381   2388   2392   2396   2398
                    2402   2411   2413   2437   2443   2549   2553   2559   2566   2568   2591   2597   2631
                    2633   2638   2692   2699   2706   2712   2714   2747   2754   2784   2790   2822   2824
```

```
                    2831    2873    2877    2892    2896    2914    2924    2933    2938    2940    2945    2958    2960
                    2968    2974    2976    3027    3033    3048    3056    3071    3078    3106    3108    3115    3191
                    3198    3226    3232    3258    3262    3270    3275    3282    3290    3316    3318    3325    3347
                    3353    3356    3359    3364    3366    3368    3374    3379    3381    3386    3461    3466    3471
                    3475    3481    3507    3512    3657    3660    3665    3674    3684    3695    3700    3712    3715
                    3721    3723    3725
$FSINC= 000210        1#    2339    2421    2853    2926    3006    3081    3553    3565    3593    3601    3609    3612
$FSLOO= 000200        1#    1684    1781    1784    3458    3484    3499
$FSNAM= 000160        1#
$FSNO = 0004C3        1#    1083    1158    1199    1212    1228    1245    1264    1284    1297    1313    1330    1349
                    1379    1395    1412    1431    1457    1473    1490    1509    1540    1571    1578    1585    1605
                    1643    1730    1761    1764    1803    1859    1877    1934    1960    1986    2004    2026    2030
                    2033    2056    2078    2106    2111    2168    2189    2213    2245    2263    2304    2306    2364
                    2367    2388    2396    2398    2437    2549    2553    2591    2631    2692    2699    2747    2784
                    2822    2914    2933    2940    2958    2960    3071    3106    3191    3226    3258    3262    3282
                    3316    3318    3356    3359    3374    3381    3461    3657    3660    3700    3712    3715
$FSOR = 000320        1#    1085    1160    1201    1214    1230    1247    1266    1286    1299    1315    1332    1351
                    1381    1397    1414    1433    1459    1475    1492    1511    1542    1573    1580    1587    1607
                    1645    1675    1732    1763    1766    1805    1861    1879    1936    1960    1962    1988    2006
                    2028    2032    2035    2058    2080    2108    2110    2113    2170    2191    2215    2247    2265
                    2306    2308    2311    2366    2369    2390    2398    2400    2439    2551    2555    2593    2633
                    2694    2701    2749    2786    2824    2916    2935    2942    2960    2970    3073    3108    3193
                    3228    3260    3264    3284    3318    3349    3358    3361    3376    3383    3463    3477    3509
                    3659    3662    3702    3714    3717
$FSRTN= 000300        1#    3424    3523    3531    3573    3580    3618    3640    3746    3754    3790
$FSSEL= 000140        1#
$FSUNT= 000130        1#    1148    1172    1984    1998    3182    3184    3340    3343    3655    3697
$FSWHI= 000120        1#    1082    1083    1090    1805    1879    1900    2025    2026    2051    2108    2306    2363
                    2364    2384    2398    2633    2824    2960    3108    3318
$FSYES= 000402        1#    1083    1158    1199    1212    1228    1245    1264    1284    1297    1313    1330    1349
                    1379    1395    1412    1431    1457    1473    1490    1509    1540    1571    1578    1585    1605
                    1643    1673    1730    1761    1764    1803    1805    1859    1877    1879    1934    1958    1960
                    1986    2004    2026    2030    2033    2056    2078    2106    2108    2111    2168    2189    2213
                    2245    2263    2304    2306    2309    2364    2367    2388    2396    2398    2437    2549    2553
                    2591    2631    2633    2692    2699    2747    2784    2822    2824    2914    2933    2940    2958
                    2960    2968    3071    3106    3108    3191    3226    3258    3262    3282    3316    3318    3347
                    3356    3359    3374    3381    3461    3475    3507    3657    3660    3700    3712    3715
$GOADR  001120       916#
$GOOAT  001124       918#
$GET42  011542      3818#
$GTSWR  012322      3978#    4451
$HO   = 000000       616
$HIBTS  001000       892#
$ICNT   001104       909#    4255*   4256    4258*   4269
$IFLEV= 177777        1#    1158#   1164#   1199#   1205#   1212#   1218#   1228#   1234#   1245#   1251#   1264#   1270#
                    1284#   1290#   1297#   1303#   1313#   1319#   1330#   1336#   1349#   1355#   1379#   1385#   1395#
                    1401#   1412#   1418#   1431#   1437#   1457#   1463#   1473#   1479#   1490#   1496#   1509#   1515#
                    1540#   1551#   1571#   1578#   1585#   1595#   1597#   1599#   1605#   1622#   1643#   1654#   1673#
                    1679#   1712#   1717#   1730#   1752#   1757#   1759#   1761#   1764#   1772#   1779#   1803#   1812#
                    1839#   1847#   1859#   1865#   1877#   1886#   1912#   1923#   1934#   1946#   1958#   1970#   1986#
                    1996#   2004#   2017#   2030#   2033#   2047#   2049#   2056#   2068#   2078#   2087#   2106#   2111#
                    2121#   2129#   2168#   2182#   2189#   2205#   2213#   2227#   2245#   2260#   2263#   2273#   2304#
                    2309#   2319#   2327#   2367#   2381#   2388#   2396#   2411#   2413#   2437#   2443#   2549#   2553#
                    2566#   2568#   2591#   2597#   2631#   2638#   2692#   2699#   2712#   2714#   2747#   2754#   2784#
                    2790#   2822#   2831#   2873#   2877#   2892#   2896#   2914#   2924#   2933#   2938#   2940#   2945#
                    2958#   2968#   2974#   2976#   3027#   3033#   3048#   3056#   3071#   3078#   3106#   3115#   3191#
```

```
                    3198#  3226#  3232#  3258#  3262#  3270#  3275#  3282#  3290#  3316#  3325#  3347#  3356#
                    3359#  3364#  3366#  3368#  3374#  3379#  3381#  3386#  3461#  3471#  3475#  3481#  3507#
                    3512#  3657#  3660#  3674#  3695#  3700#  3712#  3715#  3721#  3723#  3725#
$ILLUP  011752      3836   3852   3871#
$INTAG  001135       923#  4006   4095
$ISK0 = 000001      1158#  1164   1199#  1205   1212#  1218   1228#  1234   1245#  1251   1264#  1270   1284#
                    1290   1297#  1303   1313#  1319   1330#  1336   1349#  1355   1379#  1385   1395#  1401
                    1412#  1418   1431#  1437   1457#  1463   1473#  1479   1490#  1496   1509#  1515   1540#
                    1551   1571#  1599   1605#  1622   1643#  1654   1673#  1679   1712#  1717   1730#  1759
                    1761#  1779   1803#  1812   1839#  1847   1859#  1865   1877#  1886   1912#  1923   1934#
                    1946   1958#  1970   1986#  1996   2004#  2017   2030#  2049   2056#  2068   2078#  2087
                    2106#  2129   2168#  2182   2189#  2205   2213#  2227   2245#  2260   2263#  2273   2304#
                    2327   2367#  2381   2388#  2413   2437#  2443   2549#  2568   2591#  2597   2631#  2638
                    2692#  2714   2747#  2754   2784#  2790   2822#  2831   2873#  2877   2892#  2896   2914#
                    2924   2933#  2938   2940#  2945   2958#  2976   3027#  3033   3048#  3056   3071#  3078
                    3106#  3115   3191#  3198   3226#  3232   3258#  3275   3282#  3290   3316#  3325   3347#
                    3368   3374#  3379   3381#  3386   3461#  3471   3475#  3481   3507#  3512   3657#  3695
                    3700#  3725
$ISK1 = 000001      1578#  1597   1752#  1757   1764#  1772   2033#  2047   2111#  2121   2309#  2319   2396#
                    2411   2553#  2566   2699#  2712   2968#  2974   3262#  3270   3356#  3366   3660#  3674
                    3712#  3723
$ISK2 = 000001      1585#  1595   3359#  3364   3715#  3721
$ITEMB  001114       913#  3765   4177*  4185   4206
$LF     001172       940#  3956   4080   4089   4206
$LFLG   013271      4139*  4145#
$LOCTA= 177777         1#  1082   1083   1084   1085   1090   1091   1092   1148   1149   1159   1160   1164
                    1165   1173   1174   1200   1201   1205   1206   1213   1214   1218   1219   1229   1230
                    1234   1235   1246   1247   1251   1252   1265   1266   1270   1271   1285   1286   1290
                    1291   1298   1299   1303   1304   1314   1315   1319   1320   1331   1332   1336   1337
                    1350   1351   1355   1356   1380   1381   1385   1386   1396   1397   1401   1402   1413
                    1414   1418   1419   1432   1433   1437   1438   1458   1459   1463   1464   1474   1475
                    1479   1480   1491   1492   1496   1497   1510   1511   1515   1516   1541   1542   1551
                    1552   1572   1573   1575   1576   1577   1579   1580   1582   1583   1584   1586   1587
                    1589   1590   1591   1595   1596   1597   1598   1599   1600   1606   1607   1622   1623
                    1644   1645   1654   1655   1674   1675   1679   1680   1684   1685   1712   1713   1717
                    1718   1731   1732   1738   1739   1740   1752   1753   1757   1758   1759   1760   1762
                    1763   1765   1766   1772   1773   1774   1775   1776   1779   1780   1782   1783   1784
                    1785   1786   1804   1805   1806   1807   1808   1812   1813   1839   1840   1847   1848
                    1860   1861   1865   1866   1878   1879   1880   1881   1882   1886   1887   1912   1913
                    1923   1924   1935   1936   1946   1947   1959   1960   1961   1962   1964   1965   1966
                    1970   1971   1984   1985   1987   1988   1991   1992   1993   1996   1997   1999   2000
                    2001   2002   2003   2005   2006   2017   2018   2025   2026   2027   2028   2031   2032
                    2034   2035   2047   2048   2049   2050   2051   2052   2053   2057   2058   2068   2069
                    2079   2080   2087   2088   2107   2108   2109   2110   2112   2113   2115   2116   2117
                    2121   2122   2123   2124   2125   2129   2130   2169   2170   2182   2183   2190   2191
                    2205   2206   2214   2215   2227   2228   2246   2247   2260   2261   2264   2265   2273
                    2274   2305   2306   2307   2308   2310   2311   2313   2314   2315   2319   2320   2321
                    2322   2323   2327   2328   2340   2341   2342   2343   2344   2345   2346   2363   2364
                    2365   2366   2368   2369   2373   2374   2375   2381   2382   2384   2385   2386   2389
                    2390   2392   2393   2394   2397   2398   2399   2400   2402   2403   2404   2411   2412
                    2413   2414   2421   2422   2423   2438   2439   2443   2444   2550   2551   2554   2555
                    2559   2560   2561   2566   2567   2568   2569   2592   2593   2597   2598   2632   2633
                    2634   2635   2636   2638   2639   2693   2694   2700   2701   2706   2707   2708   2712
                    2713   2714   2715   2748   2749   2754   2755   2785   2786   2790   2791   2823   2824
                    2825   2826   2827   2831   2832   2854   2855   2856   2857   2858   2859   2860   2873
                    2874   2877   2878   2892   2893   2896   2897   2915   2916   2924   2925   2926   2927
```

```
                    2928    2934    2935    2938    2939    2941    2944    2945    2946    2959    2960    2961    2962
                    2963    2969    2970    2974    2975    2976    2977    3007    3008    3009    3010    3011    3012
                    3013    3027    3028    3033    3034    3048    3049    3056    3057    3072    3073    3078    3079
                    3081    3082    3083    3107    3108    3109    3110    3111    3115    3116    3182    3183    3185
                    3186    3192    3193    3198    3199    3227    3228    3232    3233    3259    3260    3263    3264
                    3270    3271    3275    3276    3283    3284    3290    3291    3317    3318    3319    3320    3321
                    3325    3326    3340    3341    3344    3345    3348    3349    3353    3354    3355    3357    3358
                    3360    3361    3364    3365    3366    3367    3368    3369    3375    3376    3379    3380    3382
                    3383    3386    3387    3423    3458    3459    3462    3463    3466    3467    3468    3471    3472
                    3476    3477    3481    3482    3485    3486    3487    3488    3499    3500    3501    3508    3509
                    3510    3511    3512    3513    3515    3516    3523    3524    3525    3526    3530    3554    3555
                    3556    3557    3558    3559    3560    3565    3566    3567    3570    3571    3573    3574    3575
                    3579    3594    3595    3596    3597    3598    3599    3600    3602    3603    3604    3605    3606
                    3607    3608    3609    3610    3611    3612    3613    3614    3618    3619    3620    3639    3655
                    3656    3658    3659    3661    3662    3665    3666    3667    3674    3675    3684    3685    3686
                    3695    3696    3698    3699    3701    3702    3713    3714    3716    3717    3721    3722    3723
                    3724    3725    3726    3732    3733    3746    3747    3748    3753    3790    3791    3792
$LPAOR  001106       910#   1037*   4246*   4262*   4267    4269
$LPERR  001110       911#   1038*   1150*   1209*   1223*   1239*   1256*   1294*   1308*   1324*   1341*   1376*   1390*
                    1406*   1423*   1454*   1468*   1484*   1501*   1538*   1602*   1640*   1819*   1852*   1892*   2137*
                    2187*   2209*   2231*   2515*   2572*   2650*   2727*   2762*   4196    4246    4263*   4269
$LSTCN= 177777        1#    1082    1083    1085    1090    1091    1148    1149    1158    1160    1164    1172    1199
                    1201    1205    1212    1214    1218    1228    1230    1234    1245    1247    1251    1264    1266
                    1270    1284    1286    1290    1297    1299    1303    1313    1315    1319    1330    1332    1336
                    1349    1351    1355    1379    1381    1385    1395    1397    1401    1412    1414    1418    1431
                    1433    1437    1457    1459    1463    1473    1475    1479    1490    1492    1496    1509    1511
                    1515    1540    1542    1551    1571    1573    1576    1577    1578    1580    1583    1584    1585
                    1587    1590    1591    1595    1597    1599    1605    1607    1622    1643    1645    1654    1673
                    1675    1679    1684    1685    1712    1713    1717    1730    1732    1739    1740    1752    1753
                    1757    1759    1761    1763    1764    1766    1772    1775    1776    1779    1784    1785    1803
                    1808    1812    1839    1840    1847    1859    1861    1865    1877    1882    1886    1912    1913
                    1923    1934    1936    1946    1958    1962    1965    1966    1970    1984    1985    1986    1988
                    1992    1993    1996    1998    2003    2004    2006    2017    2025    2026    2028    2030    2032
                    2033    2035    2047    2049    2051    2052    2056    2058    2068    2078    2080    2087    2106
                    2110    2111    2113    2116    2117    2121    2124    2125    2129    2168    2170    2182    2189
                    2191    2205    2213    2215    2227    2245    2247    2260    2263    2265    2273    2304    2308
                    2309    2311    2314    2315    2319    2322    2323    2327    2339    2341    2342    2343    2346
                    2363    2364    2366    2367    2369    2374    2375    2381    2384    2385    2388    2390    2393
                    2394    2396    2400    2403    2404    2411    2413    2421    2422    2437    2439    2443    2549
                    2551    2553    2555    2560    2561    2566    2568    2591    2593    2597    2631    2636    2638
                    2692    2694    2699    2701    2707    2708    2712    2714    2747    2749    2754    2784    2786
                    2790    2822    2827    2831    2853    2855    2856    2857    2860    2873    2874    2877    2892
                    2893    2896    2914    2916    2924    2926    2927    2933    2935    2938    2940    2942    2945
                    2958    2963    2968    2970    2974    2976    3006    3008    3009    3010    3013    3027    3028
                    3033    3048    3049    3056    3071    3073    3078    3081    3082    3106    3111    3115    3182
                    3183    3184    3191    3193    3198    3226    3228    3232    3258    3260    3262    3264    3270
                    3275    3282    3284    3290    3316    3321    3325    3340    3341    3343    3347    3349    3354
                    3355    3356    3358    3359    3361    3364    3366    3368    3374    3376    3379    3381    3383
                    3386    3424    3458    3459    3461    3463    3467    3468    3471    3475    3477    3481    3499
                    3500    3507    3509    3512    3523    3531    3553    3555    3556    3557    3560    3565    3566
                    3573    3580    3593    3595    3596    3597    3600    3601    3603    3604    3605    3608    3609
                    3610    3612    3613    3618    3640    3655    3656    3657    3659    3660    3662    3666    3667
                    3674    3685    3686    3695    3697    3700    3702    3712    3714    3715    3717    3721    3723
                    3725    3746    3754    3790
$LSTIN= 000000        1#    1083    1084    1085    1090    1091    1093    1094    1095    1096    1098    1099    1102
                    1103    1105    1106    1107    1108    1110    1111    1112    1113    1114    1115    1116    1117
```

1118	1119	1120	1121	1122	1123	1124	1125	1137	1138	1140	1141	1142
1143	1144	1145	1146	1147	1150	1151	1153	1154	1158	1159	1160	1167
1168	1169	1170	1171	1172	1173	1174	1177	1178	1179	1180	1181	1182
1199	1200	1201	1202	1203	1209	1210	1212	1213	1214	1223	1224	1225
1226	1228	1229	1230	1239	1240	1242	1243	1245	1246	1247	1256	1257
1259	1260	1264	1265	1266	1284	1285	1286	1287	1288	1294	1295	1297
1298	1299	1308	1309	1310	1311	1313	1314	1315	1324	1325	1327	1328
1330	1331	1332	1341	1342	1344	1345	1349	1350	1351	1376	1377	1379
1380	1381	1390	1391	1392	1393	1335	1396	1397	1406	1407	1409	1410
1412	1413	1414	1423	1424	1426	1427	1431	1432	1433	1454	1455	1457
1458	1459	1468	1469	1470	1471	1473	1474	1475	1484	1485	1487	1488
1490	1491	1492	1501	1502	1504	1505	1509	1510	1511	1538	1539	1540
1541	1542	1571	1572	1573	1575	1576	1578	1579	1580	1582	1583	1585
1586	1587	1589	1590	1592	1593	1602	1603	1605	1606	1607	1610	1611
1640	1641	1643	1644	1645	1673	1674	1675	1676	1677	1682	1683	1687
1688	1689	1690	1697	1698	1702	1703	1704	1705	1706	1707	1708	1709
1712	1713	1725	1726	1730	1731	1732	1734	1735	1738	1739	1744	1745
1746	1747	1748	1749	1750	1751	1752	1753	1761	1762	1763	1764	1765
1766	1770	1771	1774	1775	1777	1778	1781	1782	1783	1784	1785	1803
1804	1805	1806	1807	1809	1810	1816	1817	1819	1820	1823	1824	1829
1830	1831	1832	1833	1834	1835	1836	1839	1840	1843	1844	1852	1853
1859	1860	1861	1877	1878	1879	1880	1881	1883	1884	1890	1891	1892
1893	1898	1899	1903	1904	1905	1906	1907	1908	1909	1910	1912	1913
1916	1917	1921	1922	1931	1932	1934	1935	1936	1939	1940	1944	1945
1958	1959	1960	1961	1962	1964	1965	1967	1968	1972	1973	1974	1975
1976	1977	1982	1983	1986	1987	1988	1989	1990	1991	1992	1994	1995
1998	1999	2000	2001	2002	2004	2005	2006	2010	2011	2014	2015	2026
2027	2028	2030	2031	2032	2033	2034	2035	2039	2040	2044	2045	2051
2052	2056	2057	2058	2061	2062	2065	2066	2075	2076	2078	2079	2080
2083	2084	2106	2107	2108	2109	2110	2111	2112	2113	2115	2116	2118
2119	2123	2124	2126	2127	2132	2133	2137	2138	2144	2145	2147	2148
2149	2150	2151	2154	2155	2157	2158	2159	2160	2161	2164	2165	2168
2169	2170	2173	2174	2179	2180	2187	2188	2189	2190	2191	2195	2196
2202	2203	2209	2210	2213	2214	2215	2219	2220	2224	2225	2231	2232
2236	2237	2239	2240	2241	2242	2243	2245	2246	2247	2250	2251	2257
2258	2263	2264	2265	2268	2269	2304	2305	2306	2307	2308	2309	2310
2311	2313	2314	2316	2317	2321	2322	2324	2325	2330	2331	2332	2333
2334	2335	2336	2337	2339	2340	2341	2342	2343	2344	2345	2346	2347
2348	2350	2351	2353	2354	2356	2357	2359	2360	2361	2362	2364	2365
2366	2367	2368	2369	2371	2372	2373	2374	2377	2378	2379	2380	2384
2385	2388	2389	2390	2392	2393	2396	2397	2398	2399	2400	2402	2403
2409	2410	2416	2417	2418	2419	2421	2422	2424	2425	2426	2427	2428
2429	2430	2431	2432	2435	2436	2437	2438	2439	2499	2500	2503	2504
2506	2507	2509	2510	2511	2512	2513	2514	2515	2516	2518	2519	2520
2521	2522	2523	2524	2525	2528	2529	2538	2539	2542	2543	2544	2545
2546	2549	2550	2551	2553	2554	2555	2559	2560	2572	2573	2575	2576
2578	2579	2586	2587	2588	2589	2590	2591	2592	2593	2605	2606	2607
2608	2609	2610	2631	2632	2633	2634	2635	2643	2644	2645	2646	2647
2648	2650	2651	2652	2653	2655	2656	2658	2659	2668	2669	2673	2674
2675	2676	2677	2678	2679	2680	2682	2683	2685	2686	2687	2688	2689
2692	2693	2694	2697	2698	2699	2700	2701	2706	2707	2717	2718	2727
2728	2729	2730	2733	2734	2736	2737	2739	2740	2742	2743	2744	2745
2746	2747	2748	2749	2751	2752	2756	2757	2762	2763	2764	2765	2766
2767	2773	2774	2775	2776	2778	2779	2780	2781	2782	2784	2785	2786
2797	2798	2801	2802	2805	2806	2807	2808	2809	2810	2822	2823	2824
2825	2826	2828	2829	2834	2835	2837	2838	2847	2848	2849	2850	2853

	2854	2855	2856	2857	2858	2859	2860	2865	2866	2867	2868	2869	2870
	2871	2872	2873	2874	2875	2876	2881	2882	2884	2885	2886	2887	2888
	2889	2890	2891	2892	2893	2894	2895	2899	2900	2907	2908	2909	2910
	2911	2914	2915	2916	2919	2920	2926	2927	2931	2932	2933	2934	2935
	2940	2941	2942	2958	2959	2960	2961	2962	2965	2966	2968	2969	2970
	2971	2972	2979	2980	2990	2991	2999	3000	3001	3002	3003	3004	3006
	3007	3008	3009	3010	3011	3012	3013	3019	3020	3021	3022	3023	3024
	3025	3026	3027	3028	3030	3031	3037	3038	3040	3041	3042	3043	3044
	3045	3046	3047	3048	3049	3053	3054	3060	3061	3064	3065	3066	3067
	3068	3071	3072	3073	3081	3082	3087	3088	3106	3107	3108	3109	3110
	3112	3113	3119	3120	3121	3122	3139	3140	3142	3143	3144	3145	3148
	3149	3150	3151	3154	3155	3157	3158	3160	3161	3163	3164	3169	3170
	3173	3174	3176	3177	3184	3185	3186	3188	3189	3191	3192	3193	3216
	3217	3218	3220	3221	3223	3224	3226	3227	3228	3230	3231	3249	3250
	3251	3253	3254	3255	3258	3259	3260	3262	3263	3264	3266	3267	3268
	3269	3273	3274	3279	3280	3282	3283	3284	3286	3287	3316	3317	3318
	3319	3320	3322	3323	3328	3329	3331	3332	3334	3335	3337	3338	3343
	3344	3345	3347	3348	3349	3351	3352	3353	3354	3356	3357	3358	3359
	3360	3361	3362	3363	3374	3375	3376	3381	3382	3383	3448	3449	3450
	3451	3452	3453	3461	3462	3463	3464	3465	3466	3467	3469	3470	3475
	3476	3477	3479	3480	3484	3485	3486	3487	3488	3492	3493	3494	3495
	3496	3497	3498	3499	3500	3507	3508	3509	3510	3511	3514	3515	3516
	3524	3525	3526	3527	3545	3546	3547	3548	3549	3550	3551	3553	3554
	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3568
	3569	3570	3571	3575	3576	3591	3592	3593	3594	3595	3596	3597	3598
	3599	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3612
	3613	3620	3621	3634	3635	3651	3652	3653	3654	3657	3658	3659	3660
	3661	3662	3663	3664	3665	3666	3668	3669	3672	3673	3676	3677	3678
	3679	3680	3681	3682	3683	3684	3685	3687	3688	3689	3690	3691	3692
	3693	3694	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707
	3708	3712	3713	3714	3715	3716	3717	3719	3720	3728	3729	3730	3731
	3732	3733	3748	3749	3792	3793							
$LSTST= 177777	1#	1082	1083	1085	1090	1091	1146	1149	1158	1160	1164	1172	1199
	1201	1205	1212	1214	1218	1228	1230	1234	1245	1247	1251	1264	1266
	1270	1284	1286	1290	1297	1299	1303	1313	1315	1319	1330	1332	1336
	1349	1351	1355	1379	1381	1385	1395	1397	1401	1412	1414	1418	1431
	1433	1437	1457	1459	1463	1473	1475	1479	1490	1492	1496	1509	1511
	1515	1540	1542	1551	1571	1573	1575	1576	1577	1578	1580	1582	1583
	1584	1585	1587	1589	1590	1591	1595	1597	1599	1605	1607	1622	1643
	1645	1654	1673	1675	1679	1684	1685	1712	1713	1717	1730	1732	1738
	1739	1740	1752	1753	1757	1759	1761	1763	1764	1766	1772	1774	1775
	1776	1779	1781	1784	1785	1803	1808	1812	1839	1840	1847	1859	1861
	1865	1877	1882	1886	1912	1913	1923	1934	1936	1946	1958	1962	1964
	1965	1966	1970	1984	1985	1986	1988	1991	1992	1993	1996	1998	2004
	2006	2017	2025	2026	2028	2030	2032	2033	2035	2047	2049	2051	2052
	2056	2058	2068	2078	2080	2087	2106	2110	2111	2113	2115	2116	2117
	2121	2123	2124	2125	2129	2168	2170	2182	2189	2191	2205	2213	2215
	2227	2245	2247	2260	2263	2265	2273	2304	2308	2309	2311	2313	2314
	2315	2319	2321	2322	2323	2327	2339	2341	2342	2343	2346	2363	2364
	2366	2367	2369	2373	2374	2375	2381	2384	2385	2388	2390	2392	2393
	2394	2396	2400	2402	2403	2404	2411	2413	2421	2422	2437	2439	2443
	2549	2551	2553	2555	2559	2560	2561	2566	2568	2591	2593	2597	2631
	2636	2638	2692	2694	2699	2701	2706	2707	2708	2712	2714	2747	2749
	2754	2784	2786	2790	2822	2827	2831	2853	2855	2856	2857	2860	2873
	2874	2877	2892	2893	2896	2914	2916	2924	2926	2927	2933	2935	2938
	2940	2942	2945	2958	2963	2968	2970	2974	2976	3006	3008	3009	3010

```
                    3013    3027    3028    3033    3048    3049    3056    3071    3073    3078    3081    3082    3106
                    3111    3115    3182    3183    3184    3191    3193    3198    3226    3228    3232    3258    3260
                    3262    3264    3270    3275    3282    3284    3290    3316    3321    3325    3340    3341    3343
                    3347    3349    3353    3354    3355    3356    3358    3359    3361    3364    3366    3368    3374
                    3376    3379    3381    3383    3386    3424    3458    3459    3461    3463    3466    3467    3468
                    3471    3475    3477    3481    3484    3499    3500    3507    3509    3512    3523    3531    3553
                    3555    3556    3557    3560    3565    3566    3573    3580    3593    3595    3596    3597    3600
                    3601    3603    3604    3605    3608    3609    3610    3612    3613    3618    3640    3655    3656
                    3657    3659    3660    3662    3665    3666    3667    3674    3684    3685    3686    3695    3697
                    3700    3702    3712    3714    3715    3717    3721    3723    3725    3746    3754    3790
$LSTTA= 000000        1#    1082    1083    1091    1092    1148    1149    1164    1165    1205    1206    1218    1219
                    1234    1235    1251    1252    1270    1271    1290    1291    1303    1304    1319    1320    1336
                    1337    1355    1356    1385    1386    1401    1402    1418    1419    1437    1438    1463    1464
                    1479    1480    1496    1497    1515    1516    1551    1552    1576    1577    1583    1584    1590
                    1591    1595    1596    1597    1598    1599    1600    1622    1623    1654    1655    1679    1680
                    1684    1685    1717    1718    1739    1740    1757    1758    1759    1760    1772    1773    1775
                    1776    1779    1780    1785    1786    1807    1808    1812    1813    1847    1848    1865    1866
                    1881    1882    1886    1887    1923    1924    1946    1947    1965    1966    1970    1971    1984
                    1985    1992    1993    1996    1997    2002    2003    2017    2018    2025    2026    2047    2048
                    2049    2050    2052    2053    2068    2069    2087    2088    2116    2117    2121    2122    2124
                    2125    2129    2130    2182    2183    2205    2206    2227    2228    2260    2261    2273    2274
                    2314    2315    2319    2320    2322    2323    2327    2328    2341    2342    2343    2344    2363
                    2364    2374    2375    2381    2382    2385    2386    2393    2394    2403    2404    2411    2412
                    2413    2414    2422    2423    2443    2444    2560    2561    2566    2567    2568    2569    2597
                    2598    2635    2636    2638    2639    2707    2708    2712    2713    2714    2715    2754    2755
                    2790    2791    2826    2827    2831    2832    2855    2856    2857    2858    2877    2878    2896
                    2897    2924    2925    2927    2928    2938    2939    2945    2946    2962    2963    2974    2975
                    2976    2977    3008    3009    3010    3011    3033    3034    3056    3057    3078    3079    3082
                    3083    3110    3111    3115    3116    3182    3183    3198    3199    3232    3233    3270    3271
                    3275    3276    3290    3291    3320    3321    3325    3326    3340    3341    3354    3355    3364
                    3365    3366    3367    3368    3369    3379    3380    3386    3387    3423    3424    3458    3459
                    3467    3468    3471    3472    3481    3482    3500    3501    3512    3513    3523    3524    3525
                    3526    3530    3531    3555    3556    3557    3558    3566    3567    3573    3574    3575    3579
                    3580    3595    3596    3597    3598    3603    3604    3605    3606    3610    3611    3613    3614
                    3618    3619    3620    3639    3640    3655    3656    3666    3667    3674    3675    3685    3686
                    3695    3696    3721    3722    3723    3724    3725    3726    3746    3747    3748    3753    3754
                    3790    3791    3792
SMADR1  001226       973#
SMADR2  001232       977#
SMADR3  001236       980#
SMADR4  001242       983#
SMAIL   001174       893     897     946#    1055    1070    1135    1196    1281    1367    1451    1532    1565    1634
                    1670    1800    1874    1956    2103    2296    2496    2629    2820    2956    3103    3314    3900
                    4183    4261
SMAMS1  001224       967#
SM      001230       975#
SM      001234       978#
SM      001240       981#
S       001002       893#
SMCALL= 000000         1#
SMFLG   013270      4099#   4105    4140#   4144#
SMMEM   013015      3981    4093#
SMSGAD  001210       953#   4115#   4118
SMSGLG  001212       954#   4120#
SMSGTY  001174       947#   4113    4121#   4133    4137#
SMSWR   013004      3978    4091#
```

```
$MTYP1  001225          968#
$MTYP2  001231          976#
$MTYP3  001235          979#
$MTYP4  001241          9[illegible]#
$MXCNT  013752         4259    4269#
$NESTL= 177777            1#   1082#  1090#  1148#  1158#  1164#  1172#  1199#  1205#  1212#  1218#  1228#  1234#
                       1245#  1251#  1264#  1270#  1284#  1290#  1297#  1303#  1313#  1319#  1330#  1336#  1349#
                       1355#  1379#  1385#  1395#  1401#  1412#  1418#  1431#  1437#  1457#  1463#  1473#  1479#
                       1490#  1496#  1509#  1515#  1540#  1551#  1571#  1575   1578#  1582   1585#  1589   1595#
                       1597#  1599#  1605#  1622#  1643#  1654#  1673#  1679#  1684#  1712#  1717#  1730#  1738
                       1752#  1757#  1759#  1761#  1764#  1772#  1774   1779#  1781   1784#  1803#  1812#  1839#
                       1847#  1859#  1865#  1877#  1886#  1912#  1923#  1934#  1946#  1958#  1964   1970#  1984#
                       1986#  1991   1996#  1998#  2004#  2017#  2025#  2030#  2043#  2047#  2049#  20[illegible]#  2056#
                       2068#  2078#  2087#  2106#  2111#  2115   2121#  2123   2129#  2168#  2182#  21[illegible]#  2[illegible]5#
                       2213#  2227#  2245#  2260#  2263#  2273#  2304#  2309#  2313   2319#  2321   2[illegible]7#  2339#
                       2363#  2367#  2373   2381#  2384#  2388#  2392   2396#  2402   2411#  2413#  2421#  2437#
                       2443#  2549#  2553#  2559   2566#  2568#  2591#  2597#  2631#  2638#  2692#  2699#  2706
                       2712#  2714#  2747#  2754#  2784#  2790#  2822#  2831#  2953#  2873#  2877#  2892#  2896#
                       2914#  2924#  2926#  2933#  2938#  2940#  2945#  2958#  2968#  2974#  2976#  3006#  3027#
                       3033#  3048#  3056#  3071#  3078#  3081#  3106#  3115#  3182#  3184#  3191#  3198#  3226#
                       3232#  3258#  3262#  3270#  3275#  3282#  3290#  3316#  3325#  3340#  3343#  3347#  3353
                       3356#  3359#  3364#  3366#  3368#  3374#  3379#  3381#  3386#  3424#  3458#  3461#  3466
                       3471#  3475#  3481#  3484   3499#  3507#  35[illegible]#  3523#  3531#  3553#  3565#  3573#  3580#
                       3593#  3601#  3609#  3612#  3618#  3640#  3655#  3657#  3660#  3665   3674#  3684   3695#
                       3697#  3700#  3712#  3715#  3721#  3723#  3725#  3746#  3754#  3790#
$NSK0 = 000300         1082#  1090   1148#  1172   1199#  1205   1212#  1218   1228#  1234   1245#  1251   1264#
                       1270   1284#  1290   1297#  1303   1313#  1319   1330#  1336   1349#  1355   1379#  1385
                       1395#  1401   1412#  1418   1431#  1437   1457#  1463   1473#  1479   1490#  1496   1509#
                       1515   1[illegible]0#  1551   1571#  1575   1599   1605#  1622   1643#  1654   1673#  1679   1684#
                       1781   1784   1803#  1812   1839#  1847   1859#  1865   1877#  1886   1912#  1923   1934#
                       1946   1958#  1964   1970   1984#  1998   2004#  2017   2025#  2051   2056#  2068   2078#
                       2087   2106#  2123   2129   2168#  2182   2189#  2205   2213#  2227   2245#  2260   2263#
                       2273   2304#  2321   2327   2339#  2421   2437#  2443   2549#  2568   2591#  2597   2631#
                       2638   2692#  2714   2747#  2754   2784#  2790   2822#  2831   2853#  2926   2933#  2938
                       2940#  2945   2958#  2976   3006#  3081   3106#  3115   3182#  3184   3191#  3198   3226#
                       3232   3258#  3275   3282#  3290   3316#  3325   3340#  3343   3347#  3353   3368   3374#
                       3379   3381#  3386   3424#  3523   3531#  3573   3580#  3618   3640#  3746   3754#  3790
$NSK1 = 000110         1158#  1164   1578#  1582   1597   1712#  1717   1730#  1738   1759   1761#  1774   1779
                       1986#  1991   1996   2030#  2049   2111#  2115   2121   2309#  2313   2319   2363#  2384
                       2388#  2392   2413   2553#  2559   2566   2699#  2706   2712   2873#  2877   2892#  2896
                       2914#  2924   2968#  2974   3027#  3033   3048#  3056   3071#  3078   3262#  3270   3356#
                       3366   3458#  3484   3499   3507#  3512   3553#  3565   3593#  3612   3655#  3697   3700#
                       3725
$NSK2 = 000110         1585#  1589   1595   1752#  1757   1764#  1772   2033#  2047   2367#  2373   2381   2396#
                       2402   2411   3359#  3364   3461#  3466   3471   3475#  3481   3601#  3609   3657#  3684
                       3695   3712#  3723
$NSK3 = 000110         3660#  3665   3674   3715#  3721
$NULL   001154          931#  3927   3956
$NWTST= 000001         1127#  1129   1189#  1191   1276#  1362#  1444#  1446   1527#  1560#  1629#  1663#  166[illegible]
                       1793#  1795   1869#  1949#  1951   2098#  2288#  2290   2486#  2488   2621#  2623   2814#
                       2816   2951#  3097#  3099   3305#  3307   339[illegible]#
$OCNT   014422         4369*  4398*  4411#
$OMODE  014424         [illegible]364*  4368*  4373   4376*  4387*  4413#
$OVER   013736         4[illegible]23   4239   4247   4257   4266#
$PASS   001202          950#  1055*  3715   3806*  3807*  3815   3828   4253   4270
$PASTM  001006          895#
```

```
$POWER  011760          3867    3874#
$PWRAD  011746          3869#
$PWRON  011606          1031    3836#   3864
$PWRPC  011742          3867#
$PWRUP  011660          3846    3852#
$SCALE  001170           938#   3956    4024    4073    4089    4206
$SPOCHR 012534          4037#   4454
$SPODEC= ****** U       4456
$SPOLIN 012654          4065#   4455
$SPOOCT= ****** U       4456
$STKSZ = 000010         4058#
$STRNAD 011564          3827#
$SR2A  = ****** U       4456
$SAVLE= 177777             1#   1090#   1091#   1784#   1785#   2051#   2052#   2342#   2346#   2384#   2385#   2856#   2860#
                        3009#   3013#   3499#   3500#   3556#   3560#   3596#   3600#   3604#   3608#
$SAVRE= ****** U        4456
$SAVR6  011756          3845#   3853    3854#   3855#   3873#
$SCOPE  013474          1025    4220#
$SETUP= 000137          1016#   1024    1025    1027    1029    1031    1033    1034    1035    1037    1064    1067    3804
                        3961    4095    4167    4193    4201    4221
$SSKO = 000232          1090#   1091    1784#   1785    2051#   2052    2342#   2346    2384#   2385    2856#   2860    3009#
                        3013    3499#   3500    3556#   3560    3596#   3600    3604#   3608
$STUP = 177777          1016#
$SVLAD  013702          4231    4260#
$SVPC = 000204           869#    874
$SWR  = 167400              1#   616     620     621     622     623     624     625     626     935     936     937     1034
                        1035    1037    1038    1134    1195    1280    1366    1450    1571    1564    1633    1669    1799
                        1873    1955    2102    2295    2495    2628    2819    2955    3102    3313    3397    3799    3805
                        3820    3826    3828    3870    4158    4159    4160    4161    4162    4171    4178    4190    4194
                        4206    4212    4213    4214    4215    4216    4222    4234    4236    4237    4240    4241    4242
                        4249    4250    4251    4263    4266    4269
$SWREG  001216           958#   1058
$SWRMK= 000000           626     627    4216    4217    4238
$TAGLE= 177777              1#   1083#   1085#   1090#   1149#   1160#   1174#   1172#   1201#   1205#   1214#   1218#   1230#
                       1234#   1247#   1251#   1266#   1270#   1286#   1290#   1299#   1303#   1315#   1319#   1332#   1336#
                       1351#   1355#   1381#   1385#   1397#   1401#   1414#   1418#   1433#   1437#   1459#   1463#   1475#
                       1479#   1492#   1496#   1511#   1515#   1542#   1551#   1573#   1576#   1577#   1580#   1583#   1584#
                       1587#   1590#   1591#   1595#   1597#   1599#   1607#   1622#   1645#   1654#   1675#   1679#   1685#
                       1713#   1717#   1732#   1739#   1740#   1753#   1757#   1759#   1763#   1766#   1772#   1775#   1776#
                       1779#   1781    1784#   1808#   1812#   1840#   1847#   1861#   1865#   1882#   1886#   1913#   1923#
                       1936#   1946#   1962#   1965#   1966#   1970#   1985#   1988#   1992#   1993#   1996#   1998#   2006#
                       2017#   2026#   2028#   2032#   2035#   2047#   2049#   2051#   2058#   2068#   2080#   2087#   2110#
                       2113#   2116#   2117#   2121#   2124#   2125#   2129#   2170#   2182#   2191#   2205#   2215#   2227#
                       2247#   2260#   2265#   2273#   2308#   2311#   2314#   2315#   2319#   2322#   2323#   2327#   2341#
                       2343#   2346#   2364#   2366#   2369#   2374#   2375#   2381#   2384#   2390#   2393#   2394#   2400#
                       2403#   2404#   2411#   2413#   2421#   2422#   2439#   2443#   2551#   2555#   2560#   2561#   2566#
                       2568#   2593#   2597#   2636#   2638#   2694#   2701#   2707#   2708#   2712#   2714#   2749#   2754#
                       2786#   2790#   2827#   2831#   2855#   2857#   2860#   2874#   2877#   2893#   2896#   2916#   2924#
                       2926#   2927#   2935#   2938#   2942#   2945#   2963#   2970#   2974#   2976#   3008#   3010#   3013#
                       3028#   3033#   3049#   3056#   3073#   3078#   3081#   3082#   3111#   3115#   3183#   3184#   3193#
                       3198#   3228#   3232#   3260#   3264#   3270#   3275#   3284#   3290#   3321#   3325#   3341#   3343#
                       3349#   3354#   3355#   3358#   3361#   3364#   3366#   3368#   3376#   3379#   3383#   3386#   3459#
                       3463#   3467#   3468#   3471#   3477#   3481#   3484    3499#   3509#   3512#   3555#   3557#   3560#
                       3565#   3566#   3595#   3597#   3600#   3603#   3605#   3608#   3609#   3610#   3612#   3613#   3656#
                       3659#   3662#   3666#   3667#   3674#   3685#   3686#   3695#   3697#   3702#   3714#   3717#   3721#
                       3723#   3725#
```

$TAGMU= 000250
1# 1082 1083# 1084 1085# 1148 1149# 1159 1160# 1200 1201# 1213 1214#
1229 1230# 1246 1247# 1265 1266# 1285 1286# 1298 1299# 1314 1315# 1331
1332# 1350 1351# 1380 1381# 1396 1397# 1413 1414# 1432 1433# 1458 1459#
1474 1475# 1491 1492# 1510 1511# 1541 1542# 1572 1573# 1575 1577# 1579
1580# 1582 1584# 1586 1587# 1589 1591# 1606 1607# 1644 1645# 1674 1675#
1684 1685# 1712 1713# 1731 1732# 1738 1740# 1752 1753# 1762 1763# 1765
1766# 1774 1776# 1804 1806 1807 1808# 1839 1840# 1860 1861# 1878 1880
1881 1882# 1912 1913# 1935 1936# 1959 1961 1962# 1964 1966# 1984 1985#
1987 19?8# 1991 1993# 1999 2002 2003# 2005 2006# 2025 2026# 2027 2028#
2031 2032# 2034 2035# 2057 2058# 2079 2080# 2107 2109 2110# 2112 2113#
2115 2117# 2123 2125# 2169 2170# 2190 2191# 2214 2215# 2246 2247# 2264
2265# 2305 2307 2308# 2310 2311# 2313 2315# 2321 2323# 2340 2341# 2342#
2345 2346# 2363 2364# 2365 2366# 2368 2369# 2373 2375# 2389 2390# 2392
2394# 2397 2399 2400# 2402 2404# 2438 2439# 2550 2551# 2554 2555# 2559
2561# 2592 2593# 2632 2634 2635 2636# 2693 2694# 2700 2701# 2706 2708#
2748 2749# 2795 2786# 2823 2825 2826 2827# 2854 2855# 2856# 2859 2860#
2873 2874# 2892 2893# 2915 2916# 2934 2935# 2941 2942# 2959 2961 2962
2963# 2969 2970# 3007 3008# 3009# 3012 3013# 3027 3028# 3048 3049# 3072
3073# 3107 3109 3110 3111# 3182 3183# 3192 3193# 3227 3228# 3259 3260#
3263 3264# 3283 3284# 3317 3319 3320 3321# 3340 3341# 3348 3349# 3353
3355# 3357 3358# 3360 3361# 3375 3376# 3382 3383# 3424# 3458 3459# 3462
3463# 3466 3468# 3476 3477# 3508 3509# 3531# 3554 3555# 3556# 3559 3560#
3580# 3594 3595# 3596# 3599 3600# 3602 3603# 3604# 3607 3608# 3640# 3655
3656# 3658 3659# 3661 3662# 3665 3667# 3684 3686# 3701 3702# 3713 3714#
3716 3717# 3754#

$TEMP = 000300
1090# 1091# 1093# 1094# 1095# 1096# 1102# 1103# 1105# 1106# 1107# 1108# 1110#
1111# 1112# 1114# 1115# 1117# 1118# 1120# 1121# 1123# 1124# 1125# 1137# 1138#
1140# 1141# 1142# 1143# 1144# 1145# 1146# 1147# 1150# 1151# 1153# 1154# 1164#
1167# 1168# 1169# 1171# 1172# 1173 1177# 1178# 1180# 1181# 1182# 1202# 1203#
1205# 1209# 1210# 1218# 1223# 1224# 1225# 1226# 1234# 1239# 1240# 1242# 1243#
1251# 1256# 1257# 1259# 1260# 1270# 1287# 1288# 1290# 1294# 1295# 1303# 1308#
1309# 1310# 1311# 1319# 1324# 1325# 1327# 1328# 1336# 1341# 1342# 1344# 1345#
1355# 1376# 1377# 1385# 1390# 1391# 1392# 1393# 1401# 1406# 1407# 1409# 1410#
1418# 1422# 1424# 1426# 1427# 1437# 1454# 1455# 1463# 1468# 1469# 1470# 1471#
1479# 1484# 1485# 1487# 1488# 1496# 1501# 1502# 1504# 1505# 1515# 1538# 1539#
1551# 1575# 1576# 1582# 1583# 1589# 1590# 1592# 1593# 1595# 1597# 1599# 1602#
1603# 1610# 1611# 1622# 1640# 1641# 1654# 1676# 1677# 1679# 1682# 1683# 1687#
1688# 1689# 1690# 1697# 1698# 1717# 1725# 1726# 1734# 1735# 1738# 1739# 1757#
1759# 1770# 1771# 1772# 1774# 1775# 1777# 1778# 1779# 1781# 1782 1784# 1785#
1809# 1810# 1812# 1816# 1817# 1819# 1820# 1823# 1824# 1843# 1844# 1847# 1852#
1853# 1865# 1883# 1884# 1886# 1890# 1891# 1892# 1893# 1898# 1899# 1916# 1917#
1921# 1922# 1923# 1931# 1932# 1939# 1940# 1944# 1945# 1946# 1964# 1965# 1967#
1968# 1970# 1972# 1973# 1974# 1975# 1976# 1977# 1982# 1983# 1989# 1990# 1991#
1992# 1994# 1995# 1996# 1998# 2001 2010# 2011# 2014# 2015# 2017# 2039# 2040#
2044# 2045# 2047# 2049# 2051# 2052# 2061# 2062# 2065# 2066# 2068# 2075# 2076#
2083# 2084# 2087# 2115# 2116# 2118# 2119# 2121# 2123# 2124# 2126# 2127# 2129#
2132# 2133# 2137# 2138# 2144# 2145# 2154# 2155# 2164# 2165# 2173# 2174# 2179#
2180# 2182# 2187# 2188# 2195# 2196# 2202# 2203# 2205# 2209# 2210# 2219# 2220#
2224# 2225# 2227# 2231# 2232# 2236# 2237# 2250# 2251# 2257# 2258# 2260# 2268#
2269# 2273# 2313# 2314# 2316# 2317# 2319# 2321# 2322# 2324# 2325# 2327# 2330#
2331# 2332# 2333# 2334# 2335# 2336# 2337# 2343# 2346# 2347# 2348# 2350# 2351#
2353# 2354# 2356# 2357# 2359# 2360# 2361# 2362# 2371# 2372# 2373# 2374# 2377#
2378# 2379# 2380# 2381# 2384# 2385# 2392# 2393# 2402# 2403# 2409# 2410# 2411#
2413# 2416# 2417# 2418# 2419# 2421# 2422# 2424# 2428# 2429# 2430# 2431# 2432#
2435# 2436# 2443# 2499# 2500# 2503# 2504# 2506# 2507# 2509# 2510# 2511# 2512#
2513# 2514# 2515# 2516# 2528# 2529# 2538# 2539# 2559# 2560# 2566# 2568# 2572#

```
                2574#  2575#  2576#  2578#  2579#  2597#  2605#  2606#  2608#  2609#  2610#  2638#  2643#
                2644#  2645#  2646#  2647#  2648#  2650#  2651#  2652#  2653#  2655#  2656#  2658#  2659#
                2668#  2669#  2682#  2683#  2697#  2698#  2706#  2707#  2712#  2714#  2717#  2718#  2727#
                2728#  2729#  2730#  2733#  2734#  2736#  2737#  2739#  2740#  2751#  2752#  2754#  2756#
                2757#  2762#  2763#  2764#  2765#  2766#  2767#  2773#  2774#  2775#  2776#  2790#  2797#
                2798#  2801#  2802#  2805#  2806#  2808#  2809#  2810#  2828#  2829#  2831#  2834#  2835#
                2837#  2838#  2857#  2860#  2875#  2876#  2877#  2881#  2882#  2894#  2895#  2896#  2899#
                2900#  2907#  2909#  2910#  2911#  2919#  2920#  2924#  2926#  2927#  2931#  2932#  2938#
                2945#  2965#  2966#  2971#  2972#  2974#  2976#  2979#  2980#  2990#  2991#  3003#  3004#
                3010#  3013#  3030#  3031#  3033#  3037#  3038#  3053#  3054#  3056#  3060#  3061#  3064#
                3066#  3067#  3068#  3078#  3081#  3082#  3087#  3088#  3112#  3113#  3115#  3139#  3140#
                3142#  3143#  3144#  3145#  3148#  3149#  3150#  3151#  3154#  3155#  3157#  3158#  3160#
                3161#  3163#  3164#  3169#  3170#  3173#  3174#  3176#  3177#  3184#  3185   3188#  3189#
                3198#  3216#  3218#  3220#  3221#  3223#  3224#  3230#  3231#  3232#  3249#  3251#  3253#
                3255#  3266#  3267#  3268#  3269#  3270#  3273#  3274#  3275#  3279#  3280#  3286#  3287#
                3290#  3322#  3323#  3325#  3328#  3329#  3331#  3332#  3334#  3335#  3337#  3338#  3343#
                3344   3351#  3352#  3353#  3354#  3362#  3363#  3364#  3366#  3368#  3379#  3386#  3448#
                3449#  3450#  3451#  3452#  3453#  3464#  3465#  3466#  3467#  3469#  3470#  3471#  3479#
                3480#  3481#  3484#  3485   3487   3497#  3498#  3499#  3500#  3512#  3523#  3545#  3546#
                3547#  3551#  3557#  3560#  3561#  3562#  3563#  3564#  3565#  3566#  3568#  3569#  3573#
                3591#  3592#  3597#  3600#  3605#  3608#  3609#  3610#  3612#  3613#  3613#  3634#  3635#
                3651#  3652#  3653#  3654#  3663#  3664#  3665#  3666#  3672#  3673#  3674#  3676#  3677#
                3678#  3679#  3680#  3681#  3682#  3683#  3684#  3685#  3687#  3688#  3689#  3690#  3691#
                3692#  3693#  3694#  3695#  3697#  3698   3703#  3704#  3705#  3706#  3707#  3708#  3719#
                3720#  3721#  3723#  3725#  3728#  3729#  3730#  3731#  3746#  3790#
$TESTN  001200   949#  1135*  1196*  1281*  1367*  1451*  1532*  1565*  1634*  1670*  1800*  1874*  1956*
                2103*  2296*  2496*  2629*  2820*  2956*  3103*  3314*  3759   4261*
$TIMES  001160   935#  1034*  1134*  1195*  1202*  1280*  1287*  1366*  1450*  1531*  1564*  1592*  1633*
                1669*  1676*  1799*  1809*  1873*  1883*  1955*  1967*  2014*  2044*  2065*  2102*  2118*
                2126*  2179*  2202*  2224*  2257*  2295*  2316*  2324*  2495*  2628*  2819*  2828*  2955*
                2965*  2971*  3030*  3053*  3102*  3112*  3313*  3322*  3397*  3805*  4249*  4256   4259*
                4269
$TKB    001146   928#  3959   3971   3987   4041   4047
$TKS    001144   927#  3959   3968   3984   4008*  4039   4045
$TN   = 000026     1#   616   1127   1134#  1135   1189   1195#  1196   1203   1276   1280#  1281   1288
                1362   1366#  1367   1444   1450#  1451   1527   1531#  1532   1560   1564#  1565   1593
                1629   1633#  1634   1663   1669#  1670   1677   1787   1793   1799#  1800   1810   1869
                1873#  1874   1884   1949   1955#  1956   1968   2015   2045   2066   2090   2098   2102#
                2103   2119   2127   2180   2203   2225   2258   2276   2288   2295#  2296   2317   2325
                2445   2486   2495#  2496   2621   2628#  2629   2814   2819#  2820   2829   2951   2955#
                2956   2966   2972   3031   3054   3097   3102#  3103   3113   3201   3305   3313#  3314
                3323   3388   3393   3397#
$TPB    001152   930#  3945*  3956
$TPFLG  001157   934#  3894   3956
$TPS    001150   929#  3943   3956
$TRAP   014426  1029   4422#
$TRAP2  014450  4433#  4444
$TRP  = 000012  4437#  4446#  4447#  4448#  4449#  4450#  4451   4452#  4453   4454#  4455#  4456#
$TRP90  014462  4427   4444#
$TSKD = 000243  1083#  1090   1149#  1172   1201#  1205   1214#  1218   1230#  1234   1247#  1251   1266#
                1270   1286#  1290   1299#  1303   1315#  1319   1332#  1336   1351#  1355   1381#  1385
                1397#  1401   1414#  1418   1433#  1437   1459#  1463   1475#  1479   1492#  1496   1511#
                1515   1542#  1551   1573#  1576   1577#  1599   1607#  1622   1645#  1654   1675#  1679
                1685#  1784   1808#  1812   1840#  1847   1861#  1865   1882#  1886   1913#  1923   1936#
                1946   1962#  1965   1966#  1970   1985#  1998   2006#  2017   2026#  2051   2058#  2068
                2080#  2087   2110#  2124   2125#  2129   2170#  2182   2191#  2205   2215#  2227   2247#
```

F10


```
                    2260    2265#   2273    2308#   2322    2323#   2327    2341#   2343    2346#   2422    2439#   2443
                    2551#   2568    2593#   2597    2636#   2638    2694#   2714    2749#   2754    2786#   2790    2827#
                    2831    2855#   2857    2860#   2927    2935#   2938    2942#   2945    2963#   2976    3008#   3010
                    3013#   3082    3111#   3115    3183#   3184    3193#   3198    3228#   3232    3260#   3275    3284#
                    3290    3321#   3325    3341#   3343    3349#   3354    3355#   3368    3376#   3379    3383#   3386
                    3459#   3499    3509#   3512    3555#   3557    3560#   3566    3595#   3597    3600#   3613    3656#
                    3697    3702#   3725
$.SK1 = 000244      1085#   1090    1160#   1164    1580#   1583    1584#   1597    1665#   1781    1784    1988#   1992
                    1993#   1996    2028#   2051    2113#   2116    2117#   2121    2311#   2314    2315#   2319    2346#
                    2421    2555#   2560    2561#   2566    2701#   2707    2708#   2712    2860#   2926    2970#   2974
                    3013#   3081    3264#   3270    3358#   3366    3459#   3484    3499    3560#   3565    3600#   3612
                    3659#   3685    3686#   3695    3714#   3723
$TSK2 = 000245      1587#   1590    1591#   1595    1713#   1717    1732#   1739    1740#   1759    1763#   1775    1776#
                    1779    2032#   2049    2364#   2384    2390#   2393    2394#   2413    2874#   2877    2893#   2896
                    2916#   2924    3028#   3033    3049#   3056    3073#   3078    3361#   3364    3463#   3467    3468#
                    3471    3477#   3481    3603#   3605    3608#   3610    3662#   3666    3667#   3674    3717#   3721
$TSK3 = 000232      1753#   1757    1766#   1772    2035#   2047    2366#   2384    2400#   2403    2404#   2411    3608#
                    3609
$TSK4 = 000122      2369#   2374    2375#   2381
$TSTM   001004       894#
$TSTNM  001102       907#   3804#   4170    4206    4211    4238    4260#   4261    4266    4270
$TTYIN  012762      4066    4067    4084    4088#
$TYPBN= ****** U    4450
$TYPDS  013754      4282#   4449
$TYPE   011770      3894#   4126    4437    4445
$TYPEC  012202      3924    3931    3938    3943#   3944    4010
$TYPEX  012250      3949    3951    3954#
$TYPOC  014224      4367#   4446
$TYPON  014240      4366    4369#   4448
$TYPOS  014200      4362#   4447
$UNIT   001206       952#
$UNITM  001010       896#
$USWR   001220       959#   1199    1284    1578    1585    1803    1877    1960    2106    2111    2304    2306    2424
                    2631    2822    2958    2960    3106    3316    3318    3356    3547
$VECT1  001244       984#   3682
$VECT2  001246       985#
$XTSTR  013506      4225#
$YESNO= 000001      1113#   1114#   1116#   1117#   1119#   1120#   1122#   1123#   1167#   1168#   1170#   1171#   1179#
                    1180#   1225#   1226#   1242#   1243#   1259#   1260#   1310#   1311#   1327#   1328#   1344#   1345#
                    1392#   1393#   1409#   1410#   1426#   1427#   1470#   1471#   1487#   1488#   1504#   1505#   1610#
                    1611#   1816#   1817#   1843#   1844#   1890#   1891#   1916#   1917#   1921#   1922#   1939#   1940#
                    1944#   1945#   1972#   1973#   1994#   1995#   2010#   2011#   2039#   2040#   2061#   2062#   2083#
                    2084#   2132#   2133#   2173#   2174#   2195#   2196#   2219#   2220#   2250#   2251#   2268#   2269#
                    2336#   2337#   2342#   2343#   2377#   2378#   2379#   2380#   2425#   2428#   2429#   2430#   2435#
                    2436#   2506#   2507#   2528#   2529#   2538#   2539#   2578#   2579#   2607#   2608#   2655#   2656#
                    2658#   2659#   2682#   2683#   2697#   2698#   2717#   2718#   2729#   2730#   2736#   2737#   2751#
                    2752#   2756#   2757#   2764#   2765#   2773#   2774#   2797#   2798#   2807#   2808#   2837#   2838#
                    2856#   2857#   2875#   2876#   2894#   2895#   2908#   2909#   2910#   2911#   2919#   2920#   2931#
                    2932#   2979#   2980#   2990#   2991#   3009#   3010#   3065#   3066#   3067#   3068#   3087#   3088#
                    3169#   3170#   3173#   3174#   3176#   3177#   3188#   3189#   3217#   3218#   3223#   3224#   3230#
                    3231#   3250#   3251#   3254#   3255#   3273#   3274#   3279#   3280#   3286#   3287#   3331#   3332#
                    3334#   3335#   3351#   3352#   3362#   3363#   3497#   3498#   3548#   3551#   3556#   3557#   3561#
                    3562#   3563#   3564#   3568#   3569#   3596#   3597#   3604#   3605#   3634#   3635#   3689#   3690#
                    3691#   3692#   3693#   3694#   3730#   3731#
$$ARGC= 000000      3424#   3531#   3580#   3640#   3754#
$$BYTE= 000403      1083#   1158#   1199#   1212#   1228#   1245#   1264#   1284#   1297#   1313#   1330#   1349#   1379#
```

```
                    1395#  1412#  1431#  1457#  1473#  1490#  1509#  1540#  1571#  1578#  1585#  1605#  1643#
                    1673#  1730#  1761#  1764#  1803#  1805#  1859#  1877#  1879#  1934#  1958#  1960#  1986#
                    2004#  2026#  2030#  2033#  2056#  2078#  2106#  2108#  2111#  2168#  2189#  2213#  2245#
                    2263#  2304#  2306#  2309#  2364#  2367#  2394#  2396#  2398#  2437#  2549#  2553#  2591#
                    2631#  2633#  2692#  2699#  2747#  2784#  2822#  2824#  2914#  2933#  2940#  2958#  2960#
                    2968#  3071#  3106#  3108#  3191#  3226#  3258#  3262#  3282#  3316#  3318#  3347#  3356#
                    3359#  3374#  3381#  3461#  3475#  3507#  3657#  3660#  3700#  3712#  3715#
$$DST = 000067      2425#  3548#
$$FLAG= 000001      1082#  1083#  1085   1158#  1160   1164#  1199#  1201   1205#  1212#  1214   1218#  1228#
                    1230   1234#  1245#  1247   1251#  1264#  1266   1270#  1284#  1286   1290#  1297#  1299
                    1303#  1313#  1315   1319#  1330#  1332   1336#  1349#  1351   1355#  1379#  1381   1385#
                    1395#  1397   1401#  1412#  1414   1418#  1431#  1433   1437#  1457#  1459   1463#  1473#
                    1475   1479#  1490#  1492   1496#  1509#  1511   1515#  1540#  1542   1551#  1571#  1573
                    1578#  1580   1585#  1587   1595#  1597#  1599#  1605#  1607   1622#  1643#  1645   1654#
                    1673#  1675   1679#  1712#  1717#  1730#  1732   1752#  1757#  1759#  1761#  1763   1764#
                    1766   1772#  1779#  1803#  1805#  1808   1812#  1839#  1847#  1859#  1861   1865#  1877#
                    1879#  1882   1896#  1912#  1923#  1934#  1936   1946#  1958#  1960#  1962   1970#  1986#
                    1988   1996#  2004#  2006   2017#  2025#  2026#  2028   2030#  2032   2033#  2035   2047#
                    2049#  2056#  2058   2068#  2078#  2080   2087#  2106#  2108#  2110   2111#  2113   2121#
                    2129#  2168#  2170   2182#  2189#  2191   2205#  2213#  2215   2227#  2245#  2247   2260#
                    2263#  2265   2273#  2304#  2306#  2308   2309#  2311   2319#  2327#  2363#  2364#  2366
                    2367#  2369   2381#  2388#  2390   2396#  2398#  2400   2411#  2413#  2437#  2439   2443#
                    2549#  2551   2553#  2555   2566#  2568#  2591#  2593   2597#  2631#  2633#  2636   2638#
                    2692#  2694   2699#  2701   2712#  2714#  2747#  2749   2754#  2784#  2786   2790#  2822#
                    2824#  2827   2831#  2873#  2877#  2892#  2896#  2914#  2916   2924#  2933#  2935   2938#
                    2940#  2942   2945#  2958#  2960#  2963   2968#  2970   2974#  2976#  3027#  3033#  3048#
                    3056#  3071#  3073   3078#  3106#  3108#  3111   3115#  3191#  3193   3198#  3226#  3228
                    3232#  3258#  3260   3262#  3264   3270#  3275#  3282#  3284   3290#  3316#  3318#  3321
                    3325#  3347#  3349   3356#  3358   3359#  3361   3364#  3366#  3368#  3374#  3376   3379#
                    3381#  3383   3 86#  3461#  3463   3471#  3475#  3477   3481#  3507#  3509   3512#  3657#
                    3659   3660#  3662   3674#  3695#  3700#  3702   3712#  3714   3715#  3717   3721#  3723#
                    3725#
$$FROM= 000000      1098#  1702#  1744#  1829#  1903#  2147#  2157#  2239#  2518#  2542#  2586#  2673#  2685#
                    2742#  2778#  2847#  2865#  2884#  2999#  3019#  3040#  3119#  3492#  3668#
$$GET4= 000000      3820#
$$LOC = 011226      1084#  1085   1159#  1160   1173#  1174   1200#  1201   1213#  1214   1229#  1230   1246#
                    1247   1265#  1266   1285#  1286   1298#  1299   1314#  1315   1331#  1332   1350#  1351
                    1380#  1381   1396#  1397   1413#  1414   1432#  1433   1458#  1459   1474#  1475   1491#
                    1492   1510#  1511   1541#  1542   1572#  1573   1579#  1580   1586#  1587   1606#  1607
                    1644#  1645   1674#  1675   1712#  1713   1731#  1732   1752#  1753   1762#  1763   1765#
                    1766   1806#  1807   1839#  1840   1860#  1861   1880#  1881   1912#  1913   1935#  1936
                    1959#  1960   1961#  1962   1987#  1988   2001#  2002   2005#  2006   2027#  2028   2031#
                    2032   2034#  2035   2057#  2058   2079#  2080   2107#  2108   2109#  2.10   2112#  2113
                    2169#  2170   2190#  2191   2214#  2215   2246.  2247   2264#  2265   2305#  2306   2307#
                    2308   2310#  2311   2365#  2366   2368#  2369   2389#  2390   2397#  2398   2399#  2400
                    2438#  2439   2550#  2551   2554#  2555   2592#  2593   2634#  2635   2693#  2694   2700#
                    2701   2748#  2749   2785#  2786   2825#  2826   2873#  2874   2892#  2893   2915#  2916
                    2934#  2935   2941#  2942   2961#  2962   2969#  2970   3027#  3028   3048#  3049   3072#
                    3073   3109#  3110   3185#  3186   3192#  3193   3227#  3228   3259#  3260   3263#  3264
                    3283#  3284   3319#  3320   3344#  3345   3348#  3349   3357#  3358   3360#  3361   3375#
                    3376   3382#  3383   3462#  3463   3476#  3477   3508#  3509   3658#  3659   3661#  3662
                    3698#  3699   3701#  3702   3713#  3714   3716#  3717
$$LOCN= 000000      3424#  3526   3531#  3575   3580#  3620   3640#  3748   3754#  3792
$$RETU= 000000      3424#  3510#  3514#  3524   3531#  3574   3580#  3619   3640#  3747   3754#  3791
$$RTN1= 000246      3424#  3510   3523   3531#  3570   3573   3580#  3618   3640#  3732   3746   3754#  3790
$$RTN2= 000247      3424#  3515   3525   3531#  3574   3580#  3619   3640#  3747   3754#  3791
```

$$SRC = 000027 2425# 3548#
$$TO = 000000 1098# 1099 1702# 1708 1744# 1750 1829# 1835 1903# 1909 2147# 2150 2157#
2160 2239# 2242 2518# 2524 2542# 2545 2586# 2589 2673# 2679 2685# 2688
2742# 2745 2778# 2781 2847# 2848 2849 2865# 2871 2884# 2890 2999# 3000
3001 3019# 3025 3040# 3046 3119# 3120 3121 3492# 3495 3668# 3669
$OFILL 014423 4363# 4367# 4377 4412#
$1 001712 1082# 1090
$10 002552 1246 1251#
$100 004740 2112 2116#
$101 004750 2115 2121#
$102 004762 2123 2129#
$103 005070 2169 2182#
$104 005124 2190 2205#
$105 005162 2214 2227#
$106 005240 2246 2260#
$107 005260 2264 2273#
$11 002602 1265 1270#
$110 005410 2305 2307 2322#
$111 005376 2310 2314#
$112 005406 2313 2319#
$113 005420 2321 2327#
$114 005454 2340 2343#
$115 005452 2341# 2421
$116 005626 2345 2422#
$117 005512 2363# 2384
$12 002640 1285 1290#
$120 005546 2365 2385#
$121 005534 2368 2374#
$122 005544 2373 2381#
$123 005560 2389 2393#
$124 005610 2392 2413#
$125 005602 2397 2399 2403#
$126 005610 2402 2411#
$127 005674 2438 2443#
$13 002660 1298 1303#
$130 006154 2550 2568#
$131 006152 2554 2560#
$132 006154 2559 2566#
$133 006232 2592 2597#
$134 006320 2632 2635#
$135 006324 2634 2638#
$136 006514 2693 2714#
$137 006512 2700 2707#
$14 002706 1314 1319#
$140 006514 2706 2712#
$141 006606 2748 2754#
$142 006704 2785 2790#
$143 007004 2823 2826#
$144 007014 2825 2831#
$145 007060 2854 2857#
$146 007056 2855# 2926
$147 007222 2859 2927#
$15 002734 1331 1336#
$150 007126 2873 2877#
$151 007172 2892 2896#
$152 007220 2915 2924#

$153	007242	2934	2938#	
$154	007254	2941	2945#	
$155	007312	2959	2962#	
$156	007342	2961	2976#	
$157	007342	2969	2974#	
$16	002764	1350	1355#	
$160	007414	3007	3010#	
$161	007412	30 3#	3081	
$162	007560	3012	3082#	
$163	007466	3027	3033#	
$164	007536	3048	3056#	
$165	007556	3072	3078#	
$166	007624	3107	3110#	
$167	007634	3109	3115#	
$17	003034	1380	1385#	
$170	007742	3182#	3185	
$171	007766	3192	3198#	
$172	010032	3227	3232#	
$173	010110	3259	3275#	
$174	010104	3263	3270#	
$175	010126	3283	3290#	
$176	010172	3317	3320#	
$177	010202	3319	3325#	
$2	002122	1084	1091#	
$20	003062	1396	1401#	
$200	010230	3340#	3344	
$201	010256	3348	3354#	
$202	010304	3353	3368#	
$203	010304	3357	3366#	
$204	010304	3360	3364#	
$205	010320	3375	3379#	
$206	010332	3382	3386#	
$207	010634	3510	3523#	
$21	003110	1413	1418#	
$210	010636	3515	3525#	
$211	010504	3458#	3499	
$212	010610	3485	3487	3500#
$213	010524	3462	3467#	
$214	010532	3466	3471#	
$215	010550	3476	3481#	
$216	010622	3508	3512#	
$217	010736	3570	3573#	
$22	003140	1432	1437#	
$220	010736	3574#		
$221	010702	3554	3557#	
$222	010676	3555#	3565	
$223	010726	3559	3566#	
$224	011016	3618#		
$225	011016	3619#		
$226	010764	3594	3597#	
$227	010760	3595#	3612	
$23	003176	1458	1463#	
$230	011010	3599	3613#	
$231	010776	3602	3605#	
$232	010774	3603#	3609	
$233	011006	3607	3610#	

```
$234    011272          3732    3746#
$235    011272          3747#
$236    011044          3655#   3698
$237    011130          3658    3685#
$24     003224          1474    1479#
$240    011070          3661    3666#
$241    011076          3665    3674#
$242    011150          3684    3695#
$243    011236          3701    3725#
$244    011236          3713    3723#
$245    011236          3716    3721#
$246    011462          3790#
$247    011462          3791#
$25     003252          1491    1496#
$26     003302          1510    1515#
$27     003342          1541    1551#
$3      002320          1148#   1173
$30     003372          1572    1576#
$31     003426          1575    1599#
$32     003404          1579    1583#
$33     003426          1582    1597#
$34     003416          1586    1590#
$35     003426          1589    1595#
$36     003456          1606    1622#
$37     003516          1644    1654#
$4      002344          1159    1164#
$40     003554          1674    1679#
$40CAT= ****** U        4:80    4222
$41     003562          1684#   1784
$42     003772          1782    1785#
$43     003636          1712    1717#
$44     003664          1731    1739#
$45     003720          1738    1759#
$46     003720          1752    1757#
$47     003752          1762    1775#
$5      002456          1200    1205#
$50     003750          1765    1772#
$51     003760          1774    1779#
$52     004040          1804    1807#
$53     004050          1806    1812#
$54     004132          1839    1847#
$55     004154          1860    1865#
$56     004212          1878    1881#
$57     004222          1880    1886#
$6      002476          1213    1218#
$60     004312          1912    1923#
$61     004344          1935    1946#
$62     004404          1959    1961    1965#
$63     004414          1964    1970#
$64     004440          1984#   2001
$65     004460          1987    1992#
$66     004464          1991    1996#
$67     004504          1999    2002#
$7      002524          1229    1234#
$70     004534          2005    2017#
$71     004534          2025#   2051
```

```
$72     004606          2027    2052#
$73     004634          2031    2049#
$74     004604          2034    2047#
$75     004636          2057    2068#
$76     004662          2079    2087#
$77     004752          2107    2109    2124#
.       = 014506         856#    860#    869     870#    872#    874#    875#    881     882#    884#    886#    904#    941
                        1014#   1022    1037    1038    1084    1085    1086#   1088#   1159    1160    1173    1174    1200
                        1201    1213    1214    1229    1230    1246    1247    1265    1266    1285    1286    1298    1299
                        1314    1315    1331    1332    1350    1351    1380    1381    1396    1397    1413    1414    1432
                        1433    1458    1459    1474    1475    1491    1492    1510    1511    1541    1542    1572    1573
                        1579    1580    1586    1587    1606    1607    1644    1645    1674    1675    1712    1713    1731
                        1732    1752    1753    1762    1763    1765    1766    1806    1807    1839    1840    1860    1861
                        1880    1881    1912    1913    1935    1936    1959    1960    1961    1962    1987    1988    2001
                        2002    2005    2006    2027    2028    2031    2032    2034    2035    2057    2058    2079    2080
                        2107    2108    2109    2110    2112    2113    2169    2170    2190    2191    2214    2215    2246
                        2247    2264    2265    2305    2306    2307    2308    2310    2311    2365    2366    2368    2369
                        2389    2390    2397    2398    2399    2400    2438    2439    2478#   2550    2551    2554    2555
                        2592    2593    2634    2635    2693    2694    2700    2701    2748    2749    2785    2786    2825
                        2826    2873    2874    2892    2893    2915    2916    2934    2935    2941    2942    2961    2962
                        2969    2970    3027    3028    3048    3049    3072    3073    3109    3110    3185    3196    3192
                        3193    3227    3228    3259    3260    3263    3264    3283    3284    3319    3320    3344    3345
                        3348    3349    3357    3358    3360    3361    3375    3376    3382    3383    3401#   3462    3463
                        3476    3477    3508    3509    3658    3659    3661    3662    3698    3699    3701    3702    3713
                        3714    3716    3717    3743#   3758#   3773#   3779#   3828    3832    3948    3872    3956    3959
                        4088#   4089    4095    4147#   4206    4269    4270    4336#
.$ASTA= ****** U        4099    4102
.$X   = 001000           881#    886

. ABS.  014506     000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DVDVCA,DVDVCA=SPMAC.SML,DVDVCA.P11
 RUN-TIME: 89 84 4 SECONDS
 RUN-TIME RATIO: 1407/177=7.9
 CORE USED:  34K  (68 PAGES)
```

L10