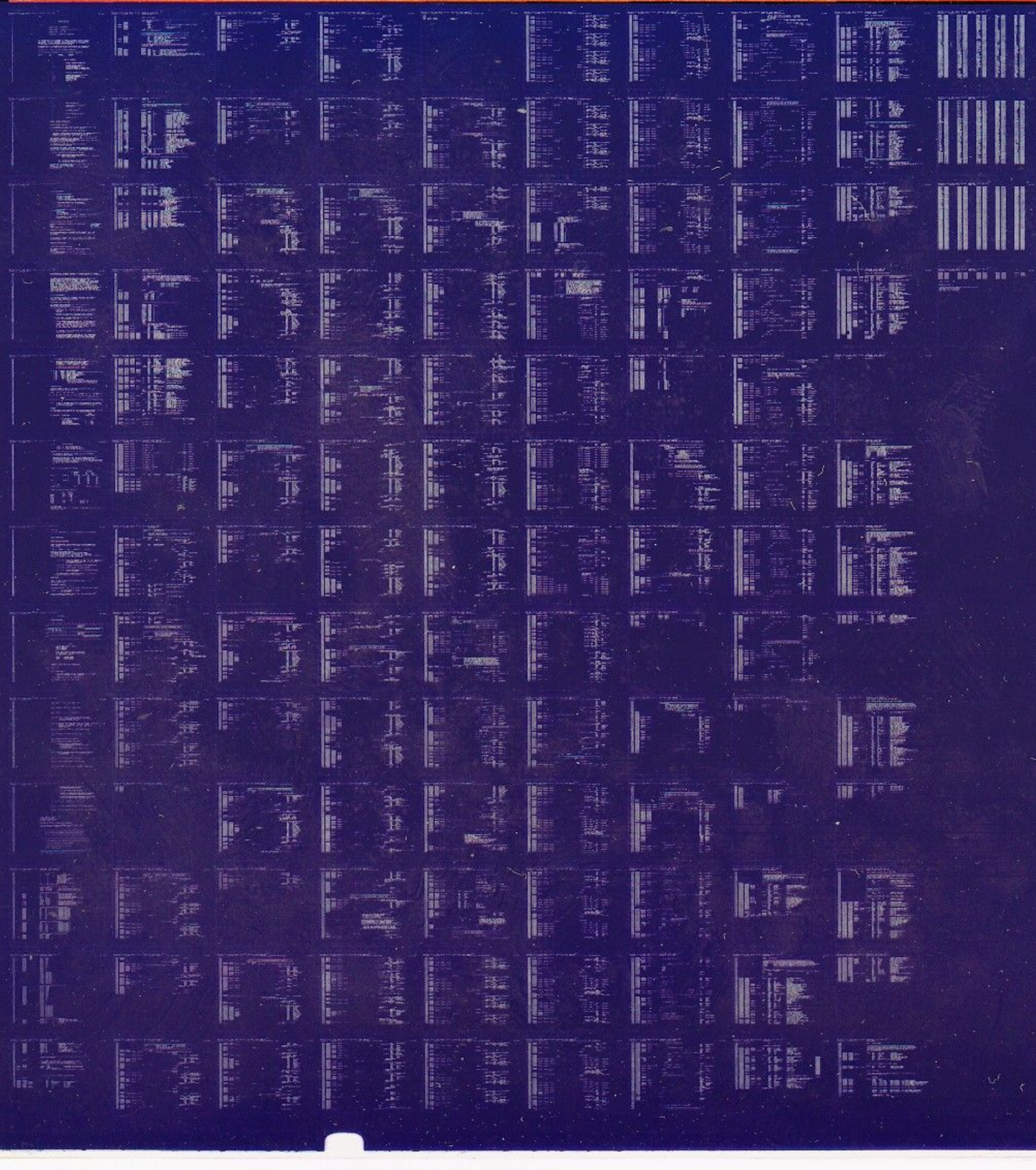
KDF11-B,
DLV11-J

KDF11-B SLU/DLV11-J
CJDLAA0

AH-S971A-MC
FICHE 1 OF 1



```
.REM %
```

IDENTIFICATION

PRODUCT CODE: AC-S970A-MC

PRODUCT NAME: CJDLAA0 KDF11B SLU/DLV11J TST

PRODUCT DATE: JULY 1981

MAINTAINER: DIAGNOSTIC ENGINEERING

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 SLU'S ON THE KDF11-B AND UP TO 2 DLV11-J MODULES.

1. ALL SELECTED CHANNELS ARE TESTED INDIVIDUALLY

2. THE DLV11-J MODULES ARE TESTED AS A WHOLE FOR CHANNEL INTERACTION PROBLEMS. THIS DIAGNOSTIC IS DESIGNED TO TEST AND DETECT ERRORS TO THE LOGIC LEVEL (NOT TO THE CHIP LEVEL).

THIS DIAGNOSTIC OPERATES UP TO 2 DLV11-J SERIAL LINE INTERFACES CONFIGURED AT CONSECUTIVE BASE ADDRESSES.
BITS 9 AND 10 OF THE USER SWITCH REGISTER, $USWR (1220), ARE USED TO DETERMINE WHAT DEVICES ARE PRESENT.

THE PROGRAM WILL PRINT OUT ALL MODULES & CHANNELS
TO BE TESTED BEFORE PROCEEDING.

THE OPERATOR MUST INSTALL DATA WRAP AROUND CONNECTORS TO DO DATA TESTING. TO BYPASS DATA TESTS, THE OPERATOR MUST MODIFY '$USWR (1220)' (USER SWITCH REGISTER) , SEE PROGRAM OPTIONS SEC. 2.4)

THE DEFAULT ADDRESSES & VECTORS ARE AS FOLLOWS:

177560 -CONSOLE INTERFACE DEVICE ADDRESS
176500 -FIRST SERIAL CHANNEL ADDRESS OF UP TO 8 CONSECUTIVE SERIAL LINE DEVICES.

60 - VECTOR FOR CONSOLE DEVICE INTERFACE.
300 - VECTOR FOR FIRST OF 8 DEVICES.

THIS PROGRAM IS DESIGNED TO RUN ON A PDP-11/23B WITH 8K OF MEMORY. IT CAN RUN UNDER XXDP & APT MONITORS, AND ON PROCESSORS WITH NO HARDWARE SWITCH REGISTER.

1.2 SYSTEM REQUIREMENTS.

HARDWARE REQUIREMENTS:

ANY PDP 11/23B PROCESSOR
8K MEMORY - MINIMUM
A SPECIAL DATA WRAP AROUND CONNECTOR OR EQUIVALENT
(REQ'D IF DATA WRAP AROUND TESTS DESIRED)

IF DATA WRAP AROUND TESTS ARE BYPASSED:
TESTS 7-12, 14-17, 21, 22 ARE BYPASSED.

SOFTWARE REQUIREMENTS:

THIS DIAGNOSTIC IS DESIGNED TO RUN IN ANY OF THE FOLLOWING WAYS:
STAND ALONE
WITH APT MONITOR
WITH XXDP MONITOR (CHAINABLE IF RENAMED TO .BIC EXTENSION)

THIS DIAGNOSTIC IS NOT DESIGNED TO RUN WITH THE DIAGNOSTIC SUPERVISOR.

1.3 RELATED DOCUMENTS AND STANDARDS.

DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS	175-003-009-02
APT	MD-11-DZZMA
SYSMAC	MD-11-DZQAC
SPMACJ USERS GUIDE	RAS - 81 015 - 0

DIAGNOSTIC HIERARCHY PREREQUISITES

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

THIS DIAGNOSTIC ASSUMES THAT THE OPERATOR WILL INITIALIZE LOCATION '$USWR (1220)' TO THE PROPER VALUE. (SEE SEC. 2.4)

2.0 OPERATING INSTRUCTIONS.

2.1 LOADING AND STARTING PROCEDURES.

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMAT
TED MEDIA.
THERE IS ONE STARTING LOCATION FOR THIS DIAGNOSTIC.
IT IS LOCATION 200.
AS SOON AS TESTING STARTS, THE OPERATOR CAN CHANGE THE SWITCH

REGISTER ONLY BY A 'BREAK' & MANUALLY LOADING LOCATION 176 (SWREG)
WITH THE DESIRED CONTENTS (SEE SEC. 2.3) THEN TYPING A 'P' TO PROCEED.

THE NORMAL STARTING PROCEDURE WOULD BE TO USE THE DEFAULT VALUE
ALREADY STORED IN LOCATION 1220($USWR). THEREFORE, THE
OPERATOR SHOULD JUST TYPE IN 'R CJDLAA'. HOWEVER, IF A VALUE OTHER
THAN THE DEFAULT VALUE (31) IS DESIRED IN $USWR(1220) THE OPERATOR
SHOULD TYPE IN 'L CJDLAA'. AFTER THE PROGRAM HAS BEEN LOADED INTO
MEMORY THE MONITOR WILL STILL HAVE CONTROL. THE OPERATOR
SHOULD HIT THE BREAK KEY. IN RESPONSE TO THE 'AT' SIGN, LOCATION
1220($USWR) CAN BE ALTERED BY TYPING IN '1220' FOLLOWED BY A SLASH. THE
CONTENTS OF THE LOCATION WILL BE TYPED OUT. THE NEW VALUE SHOULD BE
TYPED IN FOLLOWED BY A CARRIAGE-RETURN. ANOTHER 'AT' SIGN WILL BE
TYPED ON THE DEVICE. THE OPERATOR MUST THEN TYPE IN 200G, THE PROGRAM
SHOULD THEN RUN BY ITSELF.

2.2 SPECIAL ENVIRONMENTS.

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

2.3 OPERATIONAL SWITCH SETTINGS

THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED FOR ALL
OPERATIONAL SWITCH SETTINGS.
THIS CAN BE ACCOMPLISHED IN THE FOLLOWING WAY:

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

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

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

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

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

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

4) THE DIAGNOSTIC WILL CONTINUE ON TYPING <CR>.

NOTE: BECAUSE OF FREQUENT BUS RESETS IN THE PROGRAM, IT MAY BE NECESSARY TO DO 'CONTROL-G' SEVERAL TIMES. ALTERNATELY, A 'BREAK' & MANUALLY LOADING LOC. 176, FOLLOWED BY A 'P' TO PROCEED WILL ALSO WORK.

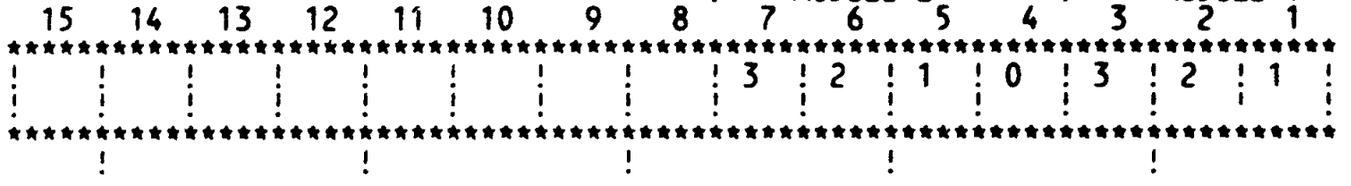
SOFTWARE SWITCH REGISTER OPTIONS (SWREG)

```
BIT 15 SET = 100000 = HALT ON ERROR
    14 SET =  40000 = LOOP ON TEST  (TO BE USED ONLY WHILE TESTING IN PR
    13 SET =  20000 = INHIBIT ERROR TYPEOUTS
    12 SET =  10000 = ENABLE PERFORMANCE REPORTS
    11 SET =   4000 = INHIBIT ITERATIONS
    10 SET =   2000 = BELL ON ERROR
     9 SET =   1000 = LOOP ON ERROR
     8 SET =    400 = LOOP ON TEST IN SWR<7:0>
     7:0   =        NUMBER OF TEST TO LOOP ON (USED WITH BIT 8)
                      (ALL TESTS PREVIOIUS TO THE SELECTED TEST
                      ARE EXECUTED FIRST WITH 1 ITERATION ONLY)
```

2.4 PROGRAM OPTIONS.

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

$DEVM (BITS 15 THRU 8 NOT USED)



NOTE: SIZING IS PERFORMED ONLY ONCE AT THE BEGINNING OF THE PROGRAM. IF $DEVM IS TO BE CHANGED, THE PROGRAM MUST BE RESTARTED AT 200 FOR IT TO BE EFFECTIVE.

OPTIONS

LOCATION $USWR (1220) CONTAINS ALL THE USER SELECTABLE OPTIONS. THE VALU IN THIS WORD MUST CONFORM TO THE ACTUAL BOARD CONFIGURATION.

BIT POSITION	DEFINITION	$USWR DEFAULT VALUE
0	# OF DATA BITS TRANSMITTED 0 = 7 BITS, 1 = 8 BITS	1 = 8 BITS
1	PARITY ENABLED	0 = NO
2	EVEN ODD PARITY	0 = ODD
3	BREAK DETECTION ENABLED	1 = YES
4	RUN DATA WRAP AROUND TESTS	1 = YES

<10:9> DEVICES PRESENT

<10:9> = 0 = ONLY TESTING KDF11-B
<10:9> = 1 = A KDF11-B AND ONE DLV11-J
<10:9> = 2 = A KDF11-B AND 2 DLV11-J'S

IMPORTANT:**********

1. FOR DIAGNOSTIC PURPOSES, ALL CHANNELS MUST BE CONFIGURED TO THE SAME BIT/WORD LENGTH.

2. WHEN BITS 10:9 ARE SET TO ONE, THE DLV11 ADDRESS MUST BE SET AT 176500 AND BE CONTIGUOUS. THE SECOND SLU ADRRESSS OF THE KDF11-B MUST BE 176540. IF BITS 10:9 ARE SET TO TWO THEN THE SECOND SLU ON THE KDF11-B MUST BE DESELECTED. THE DLV11 MODULES MUST START AT ADDRESS 176500 AND BE CONTIGUOUS.

SUMMARY OF DEVICE ADDRESSES & VECTORS.

			DEV ADDRESS (RCSR)	VECTORS REC	VECTORS XMIT
MODULE #1	CHANNEL	0	$BASE+00	$VECT1+00	$VECT1+4
		1	+10	+10	+14
		2	+20	+20	+24
		3	+30	+30	+34
MODULE #2	CHANNEL	0	+40	+40	+44
		1	+50	+50	+54
		2	+60	+60	+64
		3	+70	+70	+74

CONSOLE DEVICE $TKS TKVEC=60 TPVEC=64

SUMMARY OF USER LOCATIONS & DEFAULTS.

	LOC	DEFAULT	
$BASE	1250	176500	
$VECT1	1244	300	
$TKS	1144	177560	
TKVEC		60	
TPVEC		64	
$USWR	1220	0031	SEE SEC. 2.4
$DEVM	1252	0	SEE SEC. 2.4
SWREG	176	0	SEE SEC. 2.3

2.5 EXECUTION TIMES.
THE TEST TIME IS BAUD RATE DEPENDENT; HIGHER BAUD RATES RESULT IN SHORTER PASS TIMES.

2.6 POWER FAIL.

AUTO START FROM POWER FAIL IS NOT IMPLEMENTED IN THIS PROGRAM.

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.

SINCE THIS DIAGNOSTIC WAS DESIGNED TO FIT IN 8K 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 CHANNEL.
FOR FURTHER INFORMATION THE LISTING MUST BE CONSULTED.
BITS 15,13,10 AND 9 OF THE SWITCH REGISTER (SWREG) CONTROL THE SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.

BIT 15 SET: CAUSES THE PROGRAM TO HALT IN THE ERROR ROUTIN
IF THE PROGRAM IS CONTINUED, IT WILL PROCEED
FROM WHERE IT HALTED.

BIT 13 SET: DISABLES THE PRINTING OF THE ERROR MESSAGE.

BIT 10 SET: CAUSES THE BELL TO RING ON ERROR.

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

THE ERROR ROUTINE SUPPORTS THE CONTROL G <^G> FUNCTION.
REFER TO SECTION 2.3 FOR DETAILS.

3.2 ERROR HALTS.

ONLY IF BIT 15 OF THE SWITCH REGISTER (SWREG) IS SET WHEN AN ERROR OCCURS.

4.0 PERFORMANCE REPORTS.

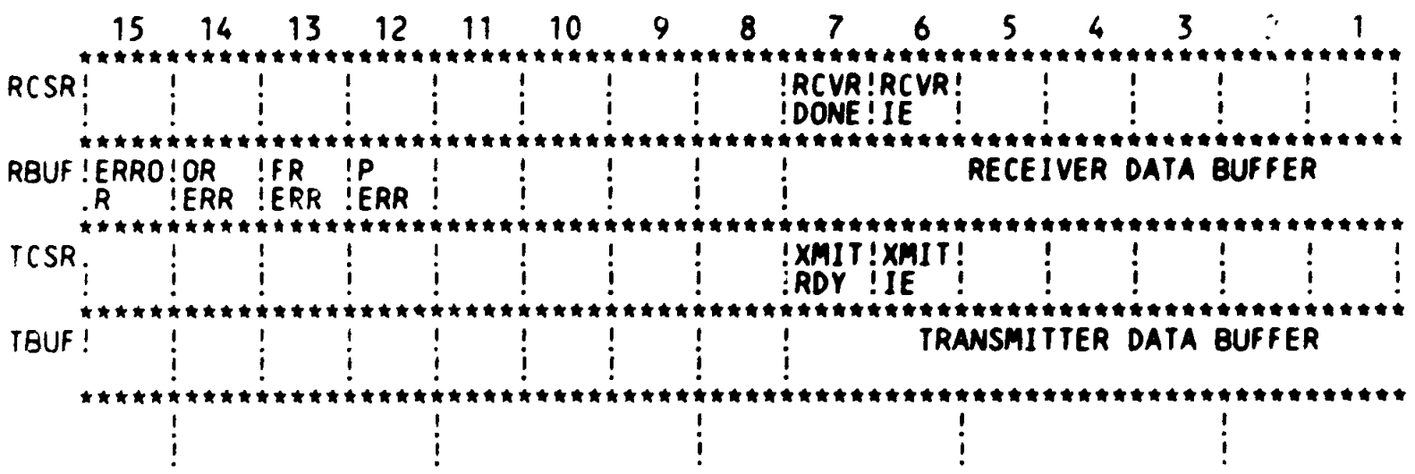
PERFORMANCE REPORTS. (BIT 12 SET IN THE SWITCH REGISTER 'SWREG')

'ERRORS:__' : THE TOTAL NUMBER OF ERRORS
ON THIS PASS.

AFTER ALL MODULES & CHANNELS TO BE TESTED HAVE BEEN EXERCISED,
AN END PASS STATEMENT IS TYPED:

'END PASS#_____.''

5.0 DEVICE REGISTERS.



NOTES:

1. RCSR AT BASE ADDRESS ($BASE)
   RBUF AT $BASE+2
   TCSR AT $BASE+4
   TBUF AT $BASE+6

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

3. ORERR = OVERRUN ERROR
   FRERR = FRAMING ERROR
   PERR = PARITY ERROR

6.0 SUMMARY OF TESTS AND SPECIAL SUBROUTINES.

PHASE 1 TESTS

TEST 1 ADDRESSABILITY

THIS TEST VERIFIES THAT ALL 4 REGISTERS OF THE CHANNEL UNDER TEST RESPOND TO THEIR ADDRESSES.

THE FOLLOWING 3 TESTS TEST ALL 'READ WRITE' BITS

TEST 2 BREAK - TCSR 0 SET, CLEAR, RESET

```
                                TEST 3 XMITIE - TCSR 6 SET, CLEAR, RESET
                                ---- -


                                TEST  4 RCVRIE - RCSR 6 SET, CLEAR, RESET
                                ----  -


                                TEST 5 XMITRDY - TCSR 7 - IS SET BY INIT
                                ---- --


                                TEST 6 XMIT RDY - TCSR 7 - CLEARS WHEN TBUF IS LOADED
                                ---- --    WITH  A  CHARACTER  AND  THAT IT SETS WITHIN A
                                           REASONABLE AMOUNT OF TIME.


                                TEST 7 OUTPUTTING A CHAR FROM TBUF (WITH  WRAP AROUND  CONNECTED)
                                ---- --   RESULTS   IN   RCVRDONE   SETTING   WITHIN   A
                                          REASONABLE  AMOUNT  OF  TIME  AND  THAT  RESET
                                          CLEARS THE BIT.

                                TEST 10  RCVRDONE IS CLEARED BY SETTING READER ENABLE
                                ---- --


                                TEST 11 RCVRDONE IS CLEARED BY READING RBUF
                                ---- --


                                TEST 12 OVERRUN & ERROR BIT - RBUF 14
                                ---- --
                                TEST 13 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 14 RECEIVER INTERRUPT LOGIC TEST THIS TEST COVERS ALL
                                ---- --   OF  THE RECEIVER SIDE OF THE INTERRUPT LOGIC IN
                                          CHARACTER MODE.


                                TEST 15  TEST DATA WRAP AROUND: FLAG MODE.
                                ---- --

                                TEST 16  TEST DATA WRAP AROUND: INTERRUPT MODE.
                                ---- --


                                TEST 17 TEST BREAK DETECTION LOGIC  TRANSMIT  KNOWN  CHAR
                                ---- --
```

```
                                        A)  TRANSMIT KNOWN CHAR WITH BREAK SET
                                            AND COMPARE RECEIVED WITH 0

                                        B)  TEST FOR FRAMING ERROR ON BREAK

                                        C)  IF PARITY IS ENABLED AND ODD PARITY IS SELECTED,
                                            CHECK TO BE SURE PARITY ERROR WAS GENERATED

                                        D)  IF PARITY IS ENABLED AND EVEN PARITY IS SELECTED,
                                            CHECK TO BE SURE NO PARITY ERROR OCCURRED


                                TEST 20 NOT A TEST - PREPARE FOR TESTS 21 & 22


                                TEST 21  TEST THAT CHANNELS INTERRUPT AT ASSIGNED PRIORITY
                                ---- --

                                TEST 22  TEST DATA TRANSFERS WITH ALL ACTIVE LINES INTERRUPTING.
                                ---- --


                                %

                        .TITLE  KDF11-B / DLV11-J TEST
                        ;*COPYRIGHT (C) 1980
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY GREGORY GLEZMAN & GARY PAPAZIAN
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-C5), JAN, 1981.
                        ;*
                        ;*****************************************************************
                        ;
                        ;       THIS PROGRAM WAS ASSEMBLED USING SPMACJ VERSION Y03.3
                        ;
                        ;*****************************************************************


                        ;*************************************************************************
                        ;
                        ;       THIS PROGRAM WAS FIRST COMPLETED BY G GLEZMAN AND G PAPAZIAN
                        ;
                        ;       IT WAS THEN MODIFIED BY RICHARD KIMBALL
                        ;
                        ;       AND ONCE AGAIN ALTERED BY E S REDNER
                        ;
                        ;       THERE IS ROOM FOR MORE
                        ;
                        ;*************************************************************************
                        ;************************** ********************************************* ****
                        .SBTTL  OPERATIONAL SWITCH SETTINGS
```

```
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                .SBTTL  BASIC DEFINITIONS

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

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

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

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13    20000
010000                          SW12=   10000
```

```
                        004000          SW11=   4000
                        002000          SW10=   2000
                        001000          SW09=   1000
                        000400          SW08=   400
                        000200          SW07=   200
                        000100          SW06=   100
                        000040          SW05=   40
                        000020          SW04=   20
                        000010          SW03=   10
                        000004          SW02=   4
                        000002          SW01=   2
                        000001          SW00=   1
                                        .EQUIV  SW09,SW9
                                        .EQUIV  SW08,SW8
                                        .EQUIV  SW07,SW7
                                        .EQUIV  SW06,SW6
                                        .EQUIV  SW05,SW5
                                        .EQUIV  SW04,SW4
                                        .EQUIV  SW03,SW3
                                        .EQUIV  SW02,SW2
                                        .EQUIV  SW01,SW1
                                        .EQUIV  SW00,SW0

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                        100000          BIT15=  100000
                        040000          BIT14=  40000
                        020000          BIT13=  20000
                        010000          BIT12=  10000
                        004000          BIT11=  4000
                        002000          BIT10=  2000
                        001000          BIT09=  1000
                        000400          BIT08=  400
                        000200          BIT07=  200
                        000100          BIT06=  100
                        000040          BIT05=  40
                        000020          BIT04=  20
                        000010          BIT03=  10
                        000004          BIT02=  4
                        000002          BIT01=  2
                        000001          BIT00=  1
                                        .EQUIV  BIT09,BIT9
                                        .EQUIV  BIT08,BIT8
                                        .EQUIV  BIT07,BIT7
                                        .EQUIV  BIT06,BIT6
                                        .EQUIV  BIT05,BIT5
                                        .EQUIV  BIT04,BIT4
                                        .EQUIV  BIT03,BIT3
                                        .EQUIV  BIT02,BIT2
                                        .EQUIV  BIT01,BIT1
                                        .EQUIV  BIT00,BIT0

                                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
                        000004          ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
                        000010          RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
                        000014          TBITVEC=14              ;;"T" BIT
                        000014          TRTVEC= 14              ;;TRACE TRAP
```

```
                    000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
                    000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
                    000024                  PWRVEC= 24              ;;POWER FAIL
                    000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
                    000034                  TRAPVEC=34              ;;"TRAP" TRAP
                    000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
                    000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
                    000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                    000001                  ADRS=   R1


                                            ;;*****************************************************************
                                            .SBTTL  TRAP CATCHER

                                                    .=0
                    000000                  ;*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  001330                             JMP     @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM


                                            ;
                                            ; OFFLINE TESTING FOR INTERRUPT VECTOR PROBLEMS
                                            ;
                    000400                          .=400

000400  005000                             1$:     CLR     R0
000402  012720  017632                             MOV     #INTSRV,(R0)+   ;SET INTR HANDLER PTR
000406  012720  000340                             MOV     #PR7,(R0)+      ;SET PRIORITY
000412  020027  000400                             CMP     R0,#400         ;ALL DONE?
000416  001371                                     BNE     1$              ;BR IF NO
000420  005037  000176                             CLR     176             ;CLEAN UP SWREG
000424  000137  001330                             JMP     START           ;GO DO IT
```

```
                                .SBTTL  ACT11 HOOKS

                                ;;******************************************************************
                                ;HOOKS REQUIRED BY ACT11
            000430                      $SVPC=.                 ;SAVE PC
            000046                      .=46
000046      021712                      $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
            000052                      .=52
000052      000000                      .WORD   0               ;;2)SET LOC.52 TO ZERO
            000430                      .=$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  000031                  $TSTM:  .WORD   25.     ;;RUN TIM OF LONGEST TEST
001006  000144                  $PASTM: .WORD   100.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000144                  $UNITM: .WORD   100.    ;;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  0^0             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  0U0             $ERRFLG:        .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED'
001157     000          $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
001160  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001162  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001164  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001170     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001171     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001172  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;**********************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

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

```
   801  001212  000000                 $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
   802  001214                         $ETABLE:                ;;APT ENVIRONMENT TABLE
   803  001214     000                 $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
   804  001215     000                 $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
   805  001216  000000                 $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
   806  001220  000031                 $USWR:  .WORD   AUSWR   ;;USER SWITCHES
   807  001222  000000                 $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
   808                                 ;*                      BITS 15-11=CPU TYPE
   809                                 ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
   810                                 ;*                              11/70=06,PDQ=07,Q=10
   811                                 ;*                      BIT 10=REAL TIME CLOCK
   812                                 ;*                      BIT  9=FLOATING POINT PROCESSOR
   813                                 ;*                      BIT  8=MEMORY MANAGEMENT
   814  001224     000                 $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
   815  001225     000                 $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
   816                                 ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
   817                                 ;*                              900 NSEC CORE=001
   818                                 ;*                              300 NSEC BIPOLAR=002
   819                                 ;*                              500 NSEC MOS=003
   820  001226  000000                 $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
   821                                 ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
   822  001230     000                 $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
   823  001231     000                 $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
   824  001232  000000                 $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
   825  001234     000                 $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
   826  001235     000                 $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
   827  001236  000000                 $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
   828  001240     000                 $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
   829  001241     000                 $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
   830  001242  000000                 $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
   831  001244  000300                 $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
   832  001246  000000                 $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
   833  001250  176500                 $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
   834  001252  000000                 $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
   835  001254                         $ETEND:
   836                                 .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  000000                     I:    .WORD   0
001256  176500                      DLADD:  .WORD       176500
001260  000300                      DLVEC:  .WORD       300
001262  001256                      RCSR:   DLADD  +  0
001264  001260                      RBUF:   DLADD  +  2
001266  001262                      TCSR:   DLADD  +  4
001270  001264                      TBUF:   DLADD  +  6

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

001272  000000                  CONTST: .WORD   0
001274  177560                  CONADR: .WORD   177560
001276  000000                  PHASE2: .WORD   0
001300  000000                  P1CNT:  .WORD   0
001302  000000                  SECSLU: .WORD   0
001304  000000                  NODLV:  .WORD   0
001306  000000                  KDDLV:  .WORD   0
001310  000000                  DLV211: .WORD   0
001312  000000                  DLBASE: .WORD   0
001314  000000                  KDFTST: .WORD   0
001316  000000                  SFTREG: .WORD   0
001320  000000                  MASK:   .WORD   0
001322  000000                  KDF11B: .WORD   0
001324  000000                  CONSOL: .WORD   0
001326  000000                  TMP9:   .WORD   0
001330                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001330  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001334  005026                  1$:     CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001336  022706  001140                  CMP     #SWR,R6 ;;DONE?
001342  001374                          BNE     .-6             ;;LOOP BACK IF NO
001344  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
001350  012737  023534  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001356  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001364  012737  023334  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001372  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001400  012737  024466  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001406  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
```

```
                                        MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
                                        CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
                                        CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
                                        MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
                                        MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                                        MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
                                        MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
                                        MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
                                        MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
                                        MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
                                        CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
                                        BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
                                        BR      65$             ;;BRANCH IF NO TIMEOUT
                                64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
                                        RTI
                                65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
                                        MOV     #DISPREG,DISPLAY
                                66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

                                        CLR     $PASS           ;;CLEAR PASS COUNT
                                        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
                                        BEQ     67$             ;;YES,USE NON-APT SWITCH
                                        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
                                67$:
                                .SBTTL  TYPE PROGRAM NAME
                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
                                        INC     #-1             ;;FIRST TIME?
                                        BNE     68$             ;;BRANCH IF NO
                                        CMP     #$ENDAD,@#42    ;;ACT-11?
                                        BEQ     68$             ;;BRANCH IF YES
                                        TYPE    ,69$            ;;TYPE ASCIZ STRING
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
                                        TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
                                        BNE     70$             ;;BRANCH IF YES
                                        CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
                                        BEQ     70$             ;;BRANCH IF YES
                                        CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
                                        BNE     71$             ;;BRANCH IF NO
                                        GTSWR                   ;;GET SOFT-SWR SETTINGS
                                        BR      71$
                                70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
                                71$:
                                        BR      68$             ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  <CRLF>*KDF11-B AND DLV11-J TEST*<CRLF>
                                68$:


                                LET $MXCNT := #1                                ;ITERATION CNT FOR SCOPE ROUTINE
                                                                                        MOV     #1,$MXCNT

                                LET P1CNT := #0                                         CLR     P1CNT
                                LET NODLV := #0
```

```
                                                                                        CLR     NODLV
                        LET KDDLV := #0                                                 CLR     KDDLV
                        LET DLV211 := #0                                                CLR     DLV211
                        LET MASK := #0                                                  CLR     MASK
                        INLINE <JSR     PC,SIZE>                        ;SIZE THE BUS
                                                                                        JSR     PC,SIZE
                        LET DLADD := #0                                                 CLR     DLADD
                        LET PHASE2 := #0                                                CLR     PHASE2
                        LET KDF11B := #0                                                CLR     KDF11B
                        LET CONTST := #0                                                CLR     CONTST
                        LET KDFTST := #0                                                CLR     KDFTST
                        LET DLBASE := #176500                                           MOV     #176500,DLBASE
                        LET CONSOL := #0                                ;SET UP INIT COND
                                                                                        CLR     CONSOL

002000  012706  001100  LOOP:   MOV     #STACK,SP       ;RESET STACK POINTER
002004  004737  020540          JSR     PC,CYCLE        ;SETUP MODULE
                                                                ;RCSR = DLADD  +  0
                                                        LET     RCSR  := DLADD
002010                                                                  MOV     DLADD,RCSR
002010  013737  001256  001262                          LET     RBUF  := DLADD + #2
002016                                                                  MOV     DLADD,RBUF
002016  013737  001256  001264                                          ADD     #2,RBUF
002024  062737  000002  001264                          LET     TCSR  := DLADD + #4
002032                                                                  MOV     DLADD,TCSR
002032  013737  001256  001266                                          ADD     #4,TCSR
002040  062737  000004  001266                          LET     TBUF  := DLADD + #6
002046                                                                  MOV     DLADD,TBUF
002046  013737  001256  001270                                          ADD     #6,TBUF
002054  062737  000006  001270                      LET $TSTNM := #0
002062                                                                  CLR     $TSTNM
002062  005037  001102
```

```
                        ;;*****************************************************************
                        ;*TEST 1        ADDRESSABILITY
                        ;*
                        ;*      THIS TEST VERIFIES THAT ALL 4 REGISTERS OF THE CHANNEL UNDER TEST
                        ;*      RESPOND TO THEIR ADDRESSES.
                        ;*
                        ;;*****************************************************************
002066  000004          TST1:   SCOPE
002070  012737  000001  001200          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002076                                                                  LET     ADRS := DLADD
002076  013701  001256                                                           MOV     DLADD,ADRS
                                                                        ; SET UP INTERRUPT
002102  012737  017632  000004          MOV     #INTSRV,ERRVEC
002110  012737  000340  000006          MOV     #PR7,ERRVEC+2
002116                                                                  LET     I := #0
002116  005037  001254                                                           CLR     I
002122                                                                  REPEAT
002122                                                                          $50000:
002122                                                              BGNSUB
002122  012737  002130  001110                                                   MOV     #64$,$LPERR
                                                                        ;CLEAR FLAG
                                                                        LET INTFLG := #0
002130                                                                          CLR     INTFLG
002130  005037  017640
                                                                        ;READ FLAG
002134  005711                          TST     @ADRS
002136                                                                  IF INTFLG NE #0 THEN
002136  005737  017640                                                           TST     INTFLG
002142  001002                                                                  BNE     .+6
002144  000137  002152                                                           JMP     $50001
                                                                        ; FATAL ERROR
002150                                                                             ERRDF 1
002150  104001                          ERROR   1
002152                                                                      ENDIF
002152                                                                          $50001:
002152                                                              ENDSUB
002152                                                                  LET     I := I + #2
002152  062737  000002  001254                                                   ADD     #2,I
002160                                                                  LET     ADRS := DLADD + I
002160  013701  001256                                                           MOV     DLADD,ADRS
002164  063701  001254                                                           ADD     I,ADRS
002170                                                              UNTIL I EQ #8.
002170  023727  001254  000010                                                   CMP     I,#8.
002176  001351                                                                  BNE     $50000
002200  032777  177476  177054          BIT     #177476,@RCSR   ;CHECK THAT ALL UNUSED BITS ARE 0
002206  001401                          BEQ     1$
002210  104027                          ERROR   27              ;RCSR HAS UNUSED BITS SET
002212  032777  007400  177044  1$:     BIT     #7400,@RBUF
002220  001401                          BEQ     2$
002222  104030                          ERROR   30              ;RBUF HAS UNUSED BITS SET
002224  032777  177476  177034  2$:     BIT     #177476,@TCSR
002232  001401                          BEQ     3$
002234  104031                          ERROR   31              ;TCSR HAS UNUSED BITS SET
```

```
1046
1047  002236  032777  177400  177024  3$:     BIT     #177400,@TBUF
1048  002244  001401                          BEQ     4$
1049  002246  104032                          ERROR   32              ;TBUF HAS ;UNUSED BITS SET
1050
1051  002250  012737  000006  000004  4$:     MOV     #6,ERRVEC       ;RESTORE
1052  002256  005037  000006                  CLR     ERRVEC+2
1053
1054                                  ;;*******************************************************************
1055                                  ;* THE FOLLOWING 3 TESTS  TEST ALL 'READ WRITE' BITS
1056                                  ;;*******************************************************************
1057
1058
1059                                  ;;*******************************************************************
1060                                  ;*TEST 2        BREAK - TCSR 0 SET, CLEAR, RESET
1061                                  ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
1062                                  ;*              THAT IS READ AND WRITE.
1063                                  ;;*******************************************************************
1064  002262  000004                  TST2:   SCOPE
1065  002264  012737  000002  001200          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1066
1067  002272                                                          IF #BIT03 NOTSETIN $USWR THEN
1068  002272  032737  000010  001220                                          BIT     #BIT03,$USWR
1069  002300  001402                                                          BEQ     .+6
1070  002302  000137  002310                                                  JMP     $50002
1071  002306                                                              EXIT                ; BREAK NOT INSTALLED
1072  002306  000503                          BR      TST3            ;;EXIT THIS TEST
1073  002310                                                          ENDIF
1074  002310                                                                  $50002:
1075  002310                          IF #BIT00 SETIN CONSOL THEN                     ;IS THIS THE CONSOL
1076  002310  032737  000001  001324                                          BIT     #BIT00,CONSOL
1077  002316  001002                                                          BNE     .+6
1078  002320  000137  002326                                                  JMP     $50003
1079                                                                                  ;  KDF11-B CONSOL DOES..
1080  002324                                                              EXIT        ; ..NOT IMPLEMENT BREAK
1081  002324  000474                  BR      TST3            ;;EXIT THIS TEST
1082  002326                                                          ENDIF
1083  002326                                                                  $50003:
1084
1085                                  ; SEE IF IT IS CLEAR
1086  002326                                                          BGNSUB
1087  002326  012737  002334  001110                                          MOV     #64$,$LPERR
1088
1089  002334                                                              IF      #BIT00 SETIN @TCSR THEN
1090  002334  032777  000001  176724                                          BIT     #BIT00,@TCSR
1091  002342  001002                                                          BNE     .+6
1092  002344  000137  002352                                                  JMP     $50004
1093                                                                    ; BREAK DID NOT RESET IN TCSR
1094  002350                                                            ERRHRD 2
1095  002350  104002                  ERROR   2
1096  002352                                                              ENDIF
1097  002352                                                                  $50004:
1098  002352                                                          ENDSUB
1099
1100                                  ; TRY TO SET BREAK BIT
1101  002352                                                          BGNSUB
```

```
1102  002352  012737  002360  001110                                                  MOV     #64$,$LPERR
1103  002360                                                                LET     @TCSR := @TCSR SET.BY #BIT00
1104  002360  052777  000001  176700                                                  BIS     #BIT00,@TCSR
1105                                                                            ; STUCK TO 0
1106  002366                                                                IF      #BIT00 NOTSETIN @TCSR THEN
1107  002366  032777  000001  176672                                                  BIT     #BIT00,@TCSR
1108  002374  001402                                                                 BEQ     .+6
1109  002376  000137  002404                                                         JMP     $50005
1110                                                                            ; BREAK DID NOT SET IN TCSR
1111  002402                                                                    ERRHRD 3
1112  002402  104003                          ERROR   3
1113  002404                                                                ENDIF
1114  002404                                                                          $50005:
1115  002404                                                            ENDSUB
1116
1117                                                              ; TRY TO CLEAR A SET BIT
1118  002404                                                            BGNSUB
1119  002404  012737  002412  001110                                                  MOV     #64$,$LPERR
1120
1121  002412                                                                LET     @TCSR := @TCSR CLR.BY #BIT00
1122  002412  042777  000001  176646                                                  BIC     #BIT00,@TCSR
1123                                                                            ; SHOULD HAVE CLEARED
1124  002420                                                                IF      #BIT00 SETIN @TCSR THEN
1125  002420  032777  000001  176640                                                  BIT     #BIT00,@TCSR
1126  002426  001002                                                                 BNE     .+6
1127  002430  000137  002436                                                         JMP     $50006
1128                                                                            ; BREAK  DID NOT CLEAR IN TCSR
1129  002434                                                                    ERRHRD 4
1130  002434  104004                          ERROR   4
1131  002436                                                                ENDIF
1132  002436                                                                          $50006:
1133  002436                                                            ENDSUB
1134
1135                                                                  ; NOW SEE IF RESET CLEARS IT
1136  002436                                                                BGNSUB
1137  002436  012737  002444  001110                                                  MOV     #64$,$LPERR
1138  002444                                  IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
1139  002444  032737  000002  001214                                                  BIT     #BIT01,$ENV
1140  002452  001406                                                                 BEQ     $50007
1141  002454  023727  001202  000001                                                  CMP     $PASS,#1
1142  002462  002402                                                                 BLT     .+6
1143  002464  000137  002516                                                         JMP     $50010
1144  002470                                                                          $50007:
1145
1146  002470                                                                LET     @TCSR := @TCSR SET.BY #BIT00
1147  002470  052777  000001  176570                                                  BIS     #BIT00,@TCSR
1148                                                                            ; ISSUE BUS RESET
1149  002476                                                                RESET
1150  002476  000005                          RESET
1151  002500                                                                IF      #BIT00 SETIN @TCSR THEN
1152  002500  032777  000001  176560                                                  BIT     #BIT00,@TCSR
1153  002506  001002                                                                 BNE     .+6
1154  002510  000137  002516                                                         JMP     $50011
1155                                                                            ; BREAK DID NOT RESET IN TCSR
1156  002514                                                                    ERRHRD 5
1157  002514  104005                          ERROR   5
```

```
1158  002516                                                    ENDIF
1159  002516                                                $50011:
1160  002516                          ENDIF
1161  002516                                                $50010:
1162  002516                                                ENDSUB
```

```
1163
1164                                    ;;*****************************************************************
1165                                    ;*TEST 3          XMITIE - TCSR 6          SET, CLEAR, RESET
1166                                    ;;*****************************************************************
1167  002516  000004                    TST3:   SCOPE
1168  002520  012737  000003  001200            MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1169                                                                    ; USE PRIORITY OF 7
1170  002526                                    SETPRI #PR7
1171
1172                                                    ; SEE IF IT IS CLEAR
1173  002540                                                    BGNSUB
1174  002540  012737  002546  001110                                    MOV     #65$,$LPERR
1175
1176  002546                                                        IF      #BIT06 SETIN @TCSR THEN
1177  002546  032777  000100  176512                                        BIT     #BIT06,@TCSR
1178  002554  001002                                                       BNE     .+6
1179  002556  000137  002564                                               JMP     $50012
1180                                                                ; XMITIE DID NOT RESET IN TCSR
1181  002562                                                            ERRHRD 12
1182  002562  104012                            ERROR   12
1183  002564                                                        ENDIF
1184  002564                                                                $50012:
1185  002564                                                    ENDSUB
1186
1187                                                    ; TRY TO SET XMITIE BIT
1188  002564                                                    BGNSUB
1189  002564  012737  002572  001110                                    MOV     #64$,$LPERR
1190  002572                                                        LET     @TCSR := @TCSR SET.BY #BIT06
1191  002572  052777  000100  176466                                        BIS     #BIT06,@TCSR
1192                                                                ; STUCK TO 0
1193  002600                                                        IF      #BIT06 NOTSETIN @TCSR THEN
1194  002600  032777  000100  176460                                        BIT     #BIT06,@TCSR
1195  002606  001402                                                       BEQ     .+6
1196  002610  000137  002616                                               JMP     $50013
1197                                                                ; XMIT DID NOT SET IN TCSR
1198  002614                                                            ERRHRD 13
1199  002614  104013                            ERROR   13
1200  002616                                                        ENDIF
1201  002616                                                                $50013:
1202  002616                                                    ENDSUB
1203
1204                                                    ; TRY TO CLEAR A SET BIT
1205  002616                                                    BGNSUB
1206  002616  012737  002624  001110                                    MOV     #64$,$LPERR
1207
1208  002624                                                        LET     @TCSR := @TCSR CLR.BY #BIT06
1209  002624  042777  000100  176434                                        BIC     #BIT06,@TCSR
1210                                                                ; SHOULD HAVE CLEARED
1211  002632                                                        IF      #BIT06 SETIN @TCSR THEN
1212  002632  032777  000100  176426                                        BIT     #BIT06,@TCSR
1213  002640  001002                                                       BNE     .+6
1214  002642  000137  002650                                               JMP     $50014
1215                                                                ;  XMIT DID NOT CLEAR IN TCSR
1216  002646                                                            ERRHRD 14
1217  002646  104014                            ERROR   14
1218  002650                                                        ENDIF
```

```
1219  002650                                                                                    $50014:
1220  002650                                                                          ENDSUB
1221
1222                                                                          ; NOW SEE IF RESET CLEARS IT
1223  002650                                                                          BGNSUB
1224  002650  012737  002656  001110                                                          MOV     #64$,$LPERR
1225  002656                        IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
1226  002656  032737  000002  001214                                                        BIT     #BIT01,$ENV
1227  002664  001406                                                                       BEQ     $50015
1228  002666  023727  001202  000001                                                       CMP     $PASS,#1
1229  002674  002402                                                                       BLT     .+6
1230  002676  000137  002730                                                               JMP     $50016
1231  002702                                                                              $50015:
1232
1233  002702                                                                    LET     @TCSR := @TCSR SET.BY #BIT06
1234  002702  052777  000100  176356                                                      BIS     #BIT06,@TCSR
1235                                                                          ; ISSUE BUS RESET
1236  002710                                                                    BRESET
1237  002710  000005                        RESET
1238  002712                                                                    IF      #BIT06 SETIN @TCSR THEN
1239  002712  032777  000100  176346                                                      BIT     #BIT06,@TCSR
1240  002720  001002                                                                     BNE     .+6
1241  002722  000137  002730                                                             JMP     $50017
1242                                                                          ; XMIT DID NOT RESET IN TCSR
1243  002726                                                                          ERRHRD 15
1244  002726  104015                        ERROR   15
1245  002730                                                                    ENDIF
1246  002730                                                                          $50017:
1247  002730                        ENDIF
1248  002730                                                                          $50016:
1249  002730                                                                  ENDSUB

                                                                                    ENDSUB
```

```
                                ;;*****************************************************************
                                ;*TEST 4         RCVRIE - RCSR 6        SET, CLEAR, RESET
                                ;*              THIS BIT IS THE ONLY ONE IN THIS POSITION
                                ;*              THAT IS READ AND WRITE.
                                ;;*****************************************************************
002730  000004                  TST4:   SCOPE
002732  012737  000004  001200          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
                                                                ; SEE IF IT IS CLEAR
                                                                BGNSUB
002740                                                                  MOV     #64$,$LPERR
002740  012737  002746  001110
                                                                IF      #BIT06 SETIN @RCSR THEN
002746                                                                  BIT     #BIT06,@RCSR
002746  032777  000100  176306                                          BNE     .+6
002754  001002                                                          JMP     $50020
002756  000137  002764                                              ; RCVRIE DID NOT RESET IN RCSR
                                                                    ERRHRD 35
002762
002762  104035                  ERROR   35
                                                                ENDIF
002764                                                                  $50020:
002764                                                          ENDSUB
002764
                                                        ; TRY TO SET RCVRIE BIT
002764                                                          BGNSUB
002764  012737  002772  001110                                        MOV     #64$,$LPERR
002772                                                          LET     @RCSR := @RCSR SET.BY #BIT06
002772  052777  000100  176262                                        BIS     #BIT06,@RCSR
                                                                ; STUCK TO 0
003000                                                          IF      #BIT06 NOTSETIN @RCSR THEN
003000  032777  000100  176254                                        BIT     #BIT06,@RCSR
003006  001402                                                          BEQ     .+6
003010  000137  003016                                                  JMP     $50021
                                                                ; RCVRIE DID NOT SET IN RCSR
003014                                                          ERRHRD 36
003014  104036                  ERROR   36
003016                                                          ENDIF
003016                                                                  $50021:
003016                                                          ENDSUB
                                                        ; TRY TO CLEAR A SET BIT
003016                                                          BGNSUB
003016  012737  003024  001110                                        MOV     #64$,$LPERR
003024                                                          LET     @RCSR := @RCSR CLR.BY #BIT06
003024  042777  000100  176230                                        BIC     #BIT06,@RCSR
                                                                ; SHOULD HAVE CLEARED
003032                                                          IF      #BIT06 SETIN @RCSR THEN
003032  032777  000100  176222                                        BIT     #BIT06,@RCSR
003040  001002                                                          BNE     .+6
003042  000137  003050                                                  JMP     $50022
                                                                ; RCVRIE DID NOT CLEAR IN RCSR
003046                                                          ERRHRD 37
003046  104037                  ERROR   37
003050                                                          ENDIF
003050                                                                  $50022:
```

```
1306  003050                                                                ENDSUB
1307
1308                                                                ; NOW SEE IF RESET CLEARS IT
1309  003050                                                                BGNSUB
1310  003050  012737  003056  001110                                              MOV     #64$,$LPERR
1311  003056                        IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
1312  003056  032737  000002  001214                                              BIT     #BIT01,$ENV
1313  003064  001406                                                              BEQ     $50023
1314  003066  023727  001202  000001                                              CMP     $PASS,#1
1315  003074  002402                                                              BLT     .+6
1316  003076  000137  003130                                                      JMP     $50024
1317  003102                                                                      $50023:
1318
1319  003102                                                          LET     @RCSR := @RCSR SET.BY #BIT06
1320  003102  052777  000100  176152                                              BIS     #BIT06,@RCSR
1321                                                                      ; ISSUE BUS RESET
1322  003110                                                          BRESET
1323  003110  000005                        RESET
1324  003112                                                          IF      #BIT06 SETIN @RCSR THEN
1325  003112  032777  000100  176142                                              BIT     #BIT06,@RCSR
1326  003120  001002                                                              BNE     .+6
1327  003122  000137  003130                                                      JMP     $50025
1328                                                                      ; RCVRIE DID NOT RESET IN RCSR
1329  003126                                                                    ERRHRD 40
1330  003126  104040                        ERROR   40
1331  003130                                                          ENDIF
1332  003130                                                                      $50025:
1333  003130                        ENDIF
1334  003130                                                                      $50024:
1335  003130                                                          ENDSUB
```

```
1336
1337                                    ;;*****************************************************************
1338                                    ;*TEST 5          TEST THAT XMITRDY - TCSR 7 - IS SET BY INIT
1339                                    ;;*****************************************************************
1340   003130  000004                   TST5:   SCOPE
1341   003132  012737  000005  001200           MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
1342
1343   003140                           IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
1344   003140  032737  000002  001214                                                   BIT     #BIT01,$ENV
1345   003146  001406                                                                  BEQ     $50026
1346   003150  023727  001202  000001                                                  CMP     $PASS,#1
1347   003156  002402                                                                  BLT     .+6
1348   003160  000137  003230                                                          JMP     $50027
1349   003164                                                                          $50026:
1350   003164                               INLINE <RESET>                 ;THIS SHOULD SET THE BIT
1351   003164  000005                                                                  RESET
1352   003166                               IF #BIT07 NOTSETIN @TCSR THEN
1353   003166  032777  000200  176072                                                  BIT     #BIT07,@TCSR
1354   003174  001402                                                                  BEQ     .+6
1355   003176  000137  003204                                                          JMP     $50030
1356   003202                                   ERRHRD 42
1357   003202  104042                           ERROR   42
1358   003204                               ENDIF
1359   003204                                                                          $50030:
1360   003204                               LET @TCSR := @TCSR CLR.BY #BIT07    ;TRY TO CLR IT
1361   003204  042777  000200  176054                                                  BIC     #BIT07,@TCSR
1362   003212                               IF #BIT07 NOTSETIN @TCSR THEN
1363   003212  032777  000200  176046                                                  BIT     #BIT07,@TCSR
1364   003220  001402                                                                  BEQ     .+6
1365   003222  000137  003230                                                          JMP     $50031
1366   003226                                   ERRHRD 11
1367   003226  104011                           ERROR   11
1368   003230                               ENDIF
1369   003230                                                                          $50031:
1370   003230                           ENDIF
1371   003230                                                                          $50027:
```

```
                                ;;*****************************************************************
                                ;*TEST 6         TEST THAT XMIT RDY - TCSR 7 - CLEARS
                                ;*              WHEN TBUF IS LOADED WITH A CHARACTER
                                ;*              AND THAT IT SETS WITHIN A REASONABLE AMOUNT OF TIME.
                                ;;*****************************************************************
003230  000004                  TST6:   SCOPE
003232  012737  000006  001200          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

003240                                                          IF CONSOL EQ #BIT00 THEN
003240  023727  001324  000001                                           CMP     CONSOL,#BIT00
003246  001402                                                    BEQ     .+6
003250  000137  003256                                           JMP     $50032
003254                                                          EXIT
003254  000555                  BR      TST7            ;;EXIT THIS TEST
003256                                                          ENDIF
003256                                                          $50032:

003256                                                          LET PASS := #0  ;INIT COUNT OF TIMES THRU
003256  005037  003602                                           CLR     PASS
003262                                                          LOOP            ; START OF LOOP
003262                                                          $50033:
                                                                ;  MAX OF 2 TIMES THRU
003262                                                             LET ERRORFLAG := #0
003262  005037  003604                                           CLR     ERRORFLAG
003266                                                           LET EXITFLAG := #0
003266  005037  003606                                           CLR     EXITFLAG
                                ; LOAD TBUF WITH ONE CHARACTER
                                ; WAIT FOR READY TO SET
                                ; (SHOULD BE VERY SHORT WAIT
                                ; SINCE UART DOUBLE BUFFERS ITS INPUT)

                                                                ;SEND A CHARACTER
003272                                                       LET @TBUF :B= #0
003272  105077  175772                                           CLRB    @TBUF
                                                                ;WAIT A MAXIMUM
                                                                ;OF 500 MSEC FOR
                                                                ;XMIT RDY TO SET IN TCSR
003276                                                       CALL TIMER IN <#500,#BIT07,TCSR,#-1>
003276  010546                                                  MOV     R5,-(SP)
003300  010605                                                  MOV     SP,R5
003302  162706  000010                                         SUB     #10,SP
003306  010546                                                  MOV     R5,-(SP)
003310  012745  177777                                         MOV     #-1,-(R5)
003314  013745  001266                                         MOV     TCSR,-(R5)
003320  012745  000200                                         MOV     #BIT07,-(R5)
003324  012745  000500                                         MOV     #500,-(R5)
003330  004737  016272                                         JSR     PC,TIMER
003334  012605                                                  MOV     (SP)+,R5
003336  010506                                                  MOV     R5,SP
003340  012605                                                  MOV     (SP)+,R5
                                                                ;TIMER RETURNS AN ERROR IF BIT DID
                                                                ;NOT MEET CONDITION WITHIN TIME LIMIT
003342                                                       IF.ERROR THEN
003342  103402                                                  BCS     .+6
003344  000137  003356                                         JMP     $50035
```

```
1428                                                                    ;XMIT RDY DID NOT SET IN TCSR
1429  003350                                                            ERRHRD 66
1430  003350  104066                                 ERROR   66
1431  003352  000137  003610                         JMP     TST7
1432  003356                                                        ENDIF
1433  003356                                                              $50035:
1434  003356  105077  175706                         CLRB    @TBUF   ;SHIP 1'ST CHAR
1435  003362  105777  175700                 1$:     TSTB    @TCSR   ;WAIT FOR RDY
1436  003366  100375                                 BPL     1$
1437                                                  ; LOAD TBUF WITH A SECOND CHARACTER (TO DOUBLE BUFFER)
1438                                                  ; CHECK IMMEDIATELY THAT XMITRDY IS CLEAR
1439                                                  ; AND THEN WAIT FOR IT TO SET
1440
1441                                                                    ;SEND SECOND CHARACTER
1442  003370                                                          LET @TBUF :B= #0
1443  003370  105077  175674                                                 CLRB    @TBUF
1444  003374  000240                                 NOP             ; GIVE IT TIME TO CLEAR
1445                                                                  ; XMITRDY SHOULD HAVE CLEARED UPON
1446                                                                  ; RECEIPT OF A CHARACTER
1447  003376                                                          IF #BIT07 SETIN @TCSR THEN
1448  003376  032777  000200  175662                                         BIT     #BIT07,@TCSR
1449  003404  001002                                                        BNE     .+6
1450  003406  000137  003424                                                JMP     $50036
1451                                                                  ; XMITRDY DID NOT CLEAR IN TCSR
1452                                                                  ; WILL RESULT IN ERR 67 IF FAILS 2X
1453  003412                                                          LET ERRORFLAG := #-1
1454  003412  012737  177777  003604                                         MOV     #-1,ERRORFLAG
1455                                                                        ; DEFER ERROR TYPEOUT
1456
1457  003420                                                          ELSE
1458  003420  000137  003500                                                JMP     $50037
1459  003424                                                                $50036:
1460                                                                  ;WAIT A MAXIMUM
1461                                                                  ;OF 100 MSEC FOR
1462                                                                  ;XMIT RDY TO SET IN TCSR
1463  003424                                                          CALL TIMER IN <#1000,#BIT07,TCSR,#-1>
1464  003424  010546                                                        MOV     R5,-(SP)
1465  003426  010605                                                        MOV     SP,R5
1466  003430  162706  000010                                                SUB     #10,SP
1467  003434  010546                                                        MOV     R5,-(SP)
1468  003436  012745  177777                                                MOV     #-1,-(R5)
1469  003442  013745  001266                                                MOV     TCSR,-(R5)
1470  003446  012745  000200                                                MOV     #BIT07,-(R5)
1471  003452  012745  001000                                                MOV     #1000,-(R5)
1472  003456  004737  016272                                                JSR     PC,TIMER
1473  003462  012605                                                        MOV     (SP)+,R5
1474  003464  010506                                                        MOV     R5,SP
1475  003466  012605                                                        MOV     (SP)+,R5
1476  003470                                                          IF.ERROR THEN
1477  003470  103402                                                        BCS     .+6
1478  003472  000137  003500                                                JMP     $50040
1479                                                                  ;XMIT RDY DID NOT SET IN TCSR
1480  003476                                                          ERRHRD 70
1481  003476  104070                                 ERROR   70
1482  003500                                                          ENDIF
1483  003500                                                                $50040:
```

```
1484  003500                                                                    ENDIF ; OF DEFERED ERROR CALL
1485  003500                                                                                  $50037:
1486  003500                                                                  IF ERRORFLAG EQ #-1 THEN
1487  003500  023727  003604  177777                                                         CMP     ERRORFLAG,#-1
1488  003506  001402                                                                        BEQ     .+6
1489  003510  000137  003550                                                                JMP     $50041
1490  003514                                                                    LET PASS := PASS + #1
1491  003514  005237  003602                                                                INC     PASS
1492  003520                                                                      IF PASS GT #1 THEN
1493  003520  023727  003602  000001                                                         CMP     PASS,#1
1494  003526  003002                                                                        BGT     .+6
1495  003530  000137  003544                                                                JMP     $50042
1496                                                                                  ; CALL ERROR IF 2ND TRY
1497                                                                                  ; ON XMIT RDY NOT CLEARING
1498  003534                                                                          ERRHRD 67
1499  003534  104067                                  ERROR   67
1500  003536                                                                          LET EXITFLAG := #-1
1501  003536  012737  177777  003606                                                         MOV     #-1,EXITFLAG
1502  003544                                                                      ENDIF
1503  003544                                                                                  $50042:
1504  003544                                                                  ELSE            ; NO ERROR
1505  003544  000137  003556                                                                JMP     $50043
1506  003550                                                                                  $50041:
1507  003550                                                                      LET EXITFLAG := #-1
1508  003550  012737  177777  003606                                                         MOV     #-1,EXITFLAG
1509  003556                                                                  ENDIF
1510  003556                                                                                  $50043:
1511  003556                                                              EXIF        EXITFLAG EQ #-1
1512  003556  023727  003606  177777                                                         CMP     EXITFLAG,#-1
1513  003564  001002                                                                        BNE     $50044
1514  003566  000137  003574                                                                JMP     $50034
1515  003572                                                                                  $50044:
1516  003572                                                                  ENDLOOP
1517  003572  000633                                                                        BR      $50033
1518  003574                                                                                  $50034:
1519  003574                                  LET R0 :B= @RBUF                                ;CLR RBUF BY READING IT
1520  003574  117700  175464                                                                MOVB    @RBUF,R0
1521  003600                                                                  EXIT ; SKIP AROUND FLAG WORDS
1522  003600  000403                                  BR      TST7           ;;EXIT THIS TEST
1523  003602  000000                          PASS:       0
1524  003604  000000                          ERRORFLAG:  0
1525  003606  000000                          EXITFLAG:   0
```

```
                                ;;*****************************************************************
                                ;;*TEST 7         TEST THAT OUTPUTTING A CHAR FROM TBUF (WITH WRAP CONNECTED)
                                ;;*               RESULTS IN RCVRDONE SETTING WITHIN A REASONABLE AMOUNT OF TIME
                                ;;*               AND THAT RESET CLEARS THE BIT.
                                ;;*****************************************************************
003610  000004                TST7:   SCOPE
003612  012737  000007  001200        MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX

003620                                                          IF #BIT04 NOTSETIN $USWR OR CONSOL EQ #BIT00 THE
003620  032737  000020  001220                                          BIT     #BIT04,$USWR
003626  001406                                                          BEQ     $50045
003630  023727  001324  000001                                        CMP     CONSOL,#BIT00
003636  001402                                                          BEQ     .+6
003640  000137  003646                                                 JMP     $50046
003644                                                                  $50045:
003644                                                      EXIT
003644  000462                        BR      TST10           ;;EXIT THIS TEST
003646                                                      ENDIF
003646                                                                  $50046:

003646                                                      BGNSUB
003646  012737  003654  001110                                          MOV     #64$,$LPERR
                                ; SEND A CHARACTER AND LET IT WRAP AROUND

003654                                                          LET @TBUF :B= #0
003654  105077  175410                                                  CLRB    @TBUF

                                ; WAIT A MAXIMUM OF 50 MSEC
                                ; FOR RCVR DONE TO SET IN
                                ; RCSR
003660                          CALL TIMER IN <#500,#BIT07,RCSR,#-1>
003660  010546                                 MOV     R5,-(SP)
003662  010605                                 MOV     SP,R5
003664  162706  000010                         SUB     #10,SP
003670  010546                                 MOV     R5,-(SP)
003672  012745  177777                         MOV     #-1,-(R5)
003676  013745  001262                         MOV     RCSR,-(R5)
003702  012745  000200                         MOV     #BIT07,-(R5)
003706  012745  000500                         MOV     #500,-(R5)
003712  004737  016272                         JSR     PC,TIMER
003716  012605                                 MOV     (SP)+,R5
003720  010506                                 MOV     R5,SP
003722  012605                                 MOV     (SP)+,R5

                                    ;DIDN'T SET IN TIME
003724                          IF.ERROR THEN
003724  103402                                 BCS     .+6
003726  000137  003734                         JMP     $50047
                                    ; RCVRDONE DID NOT SET IN RCSR
003732                              ERRHRD 71
003732  104071                                 ERROR   71
003734
                                ENDIF
003734                                                  $50047:

003734                          ENDSUB
```

```
1582
1583  003734                                                    BGNSUB
1584  003734  012737  003742  001110                                    MOV     #64$,$LPERR
1585                                        ; NOW THAT IT IS SET SEE IF IT CAN BE RESET
1586  003742                              IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
1587  003742  032737  000002  001214                                    BIT     #BIT01,$ENV
1588  003750  001406                                                   BEQ     $50050
1589  003752  023727  001202  000001                                    CMP     $PASS,#1
1590  003760  002402                                                   BLT     .+6
1591  003762  000137  003770                                           JMP     $50051
1592  003766                                                          $50050:
1593  003766                                                  BRESET
1594  003766  000005                         RESET
1595  003770                              ENDIF
1596  003770                                                          $50051:
1597
1598  003770                                                  IF #BIT07 SETIN @RCSR THEN
1599  003770  032777  000200  175264                                    BIT     #BIT07,@RCSR
1600  003776  001002                                                   BNE     .+6
1601  004000  000137  004006                                           JMP     $50052
1602                                                              ; RCVRDONE DID NOT RESET IN RCSR.
1603  004004                                                      ERRHRD 72
1604  004004  104072                         ERROR   72
1605  004006                                                  ENDIF
1606  004006                                                          $50052:
1607  004006                              LET R0 :B= @RBUF                    ;READ RBUF TO CLR IT
1608  004006  117700  175252                                           MOVB    @RBUF,R0
1609  004012                                                  ENDSUB
```

```
                                        ;;*****************************************************************
                                        ;*TEST 10        TEST THAT RCVRDONE IS CLEARED BY SETTING READER ENABLE
                                        ;;*****************************************************************
004012  000004                          TST10:  SCOPE
004014  012737  000010  001200                  MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

004022                                                                  IF #BIT04 NOTSETIN $USWR OR CONSOL EQ #BIT00 THE
004022  032737  000020  001220                                                  BIT     #BIT04,$USWR
004030  001406                                                                  BEQ     $50053
004032  023727  001324  000001                                                  CMP     CONSOL,#BIT00
004040  001402                                                                  BEQ     .+6
004042  000137  004050                                                          JMP     $50054
004046                                                                          $50053:
004046                                                                  EXIT
004046  000456                                  BR      TST11           ;;EXIT THIS TEST
004050                                                                  ENDIF
004050                                                                          $50054:

004050                                                                  BGNSUB
004050  012737  004056  001110                                                  MOV     #64$,$LPERR
                                        ; OUTPUT A CHARACTER AND WAIT FOR XMITRDY TO SET.

004056                                                                      ; OUTPUT A CHARACTER
004056  105077  175206                                                  LET @TBUF :B= #0
                                                                                CLRB    @TBUF
                                                                        ; WAIT MAXIMUM OF 500 MSEC
                                                                        ; FOR RCVRDONE TO SET IN
                                                                        ; RCSR
004062                                                                  CALL TIMER IN <#500,#BIT07,RCSR,#-1>
004062  010546                                                                  MOV     R5,-(SP)
004064  010605                                                                  MOV     SP,R5
004066  162706  000010                                                          SUB     #10,SP
004072  010546                                                                  MOV     R5,-(SP)
004074  012745  177777                                                          MOV     #-1,-(R5)
004100  013745  001262                                                          MOV     RCSR,-(R5)
004104  012745  000200                                                          MOV     #BIT07,-(R5)
004110  012745  000500                                                          MOV     #500,-(R5)
004114  004737  016272                                                          JSR     PC,TIMER
004120  012605                                                                  MOV     (SP)+,R5
004122  010506                                                                  MOV     R5,SP
004124  012605                                                                  MOV     (SP)+,R5
                                                                        ; DID IT BECOME READY?
004126                                                                  IF.ERROR  THEN
004126  103402                                                                  BCS     .+6
004130  000137  004136                                                          JMP     $50055
                                                                        ;RCVRDONE DID NOT SET IN RCSR
004134                                                                  ERRHRD 25
004134  104025                          ERROR   25
                                                                        ; SET IT BACK TO CONTINUE
004136                                                                  ENDIF
004136                                                                          $50055:
004136                                                          ENDSUB

                                        ;NOW THAT IT IS SET LETS SEE IF SETTING
                                        ;READER ENABLE CLEARS RCVRDONE
```

```
1666
1667  004136
1668  004136  023727  001322  000001                                        IF KDF11B EQ #BIT00 THEN
1669  004144  001402                                                                CMP     KDF11B,#BIT00
1670  004146  000137  004154                                                        BEQ     .+6
1671  004152                                                                        JMP     $50056
1672  004152  000414                          BR      TST11          ;;EXIT THIS TEST  EXIT                    ;  KDF11-B SLU'S DON'T..
1673                                                                                ;  ..HAVE 'READER RUN'
1674  004154                                                                    ENDIF
1675  004154                                                                $50056:
1676  004154                                                                LET @RCSR := @RCSR SET.BY #BIT00
1677  004154  052777  000001  175100                                               BIS     #BIT00,@RCSR
1678  004162                                                                    IF #BIT07 SETIN @RCSR THEN
1679  004162  032777  000200  175072                                               BIT     #BIT07,@RCSR
1680  004170  001002                                                               BNE     .+6
1681  004172  000137  004200                                                       JMP     $50057
1682                                                                                ;RCVRDONE DID NOT CLEAR IN RCSR
1683  004176                                                                        ERRHRD 26
1684  004176  104026                          ERROR   26                           ; SET IT BACK TO CONTINUE
1685                                                                            ENDIF
1686  004200                                                                $50057:
1687  004200
1688  004200                          LET R0 :B= @RBUF                      ;READ RBUF TO CLR IT
1689  004200  117700  175060                                                       MOVB    @RBUF,R0
```

```
                                ;;*****************************************************************
                                ;*TEST 11        TEST THAT RCVRDONE IS CLEARED BY READING RBUF
                                ;;*****************************************************************
004204  000004                  TST11:  SCOPE
004206  012737  000011  001200          MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX

004214                                                  IF #BIT04 NOTSETIN $USWR OR CONSOL EQ #BIT00 THE
004214  032737  000020  001220                                   BIT     #BIT04,$USWR
004222  001406                                                    BEQ     $50060
004224  023727  001324  000001                                   CMP     CONSOL,#BIT00
004232  001402                                                    BEQ     .+6
004234  000137  004242                                           JMP     $50061
004240                                                          $50060:
004240                                                  EXIT
004240  000444                          BR      TST12           ;;EXIT THIS TEST
004242                                                  ENDIF
004242                                                          $50061:

004242                                                  BGNSUB
004242  012737  004250  001110                                   MOV     #64$,$LPERR
                                ; OUTPUT A CHARACTER AND WAIT FOR RCVRDONE TO SET.

                                        ; OUTPUT A CHARACTER
004250                                  LET @TBUF :B= #0
004250  105077  175014                                  CLRB    @TBUF
                                        ; WAIT MAXIMUM OF 500 MSEC
                                        ; FOR RCVRDONE TO SET IN
                                        ; RCSR
004254                                  CALL TIMER IN <#500,#BIT07,RCSR,#-1>
004254  010546                                 MOV     R5,-(SP)
004256  010605                                 MOV     SP,R5
004260  162706  000010                         SUB     #10,SP
004264  010546                                 MOV     R5,-(SP)
004266  012745  177777                         MOV     #-1,-(R5)
004272  013745  001262                         MOV     RCSR,-(R5)
004276  012745  000200                         MOV     #BIT07,-(R5)
004302  012745  000500                         MOV     #500,-(R5)
004306  004737  016272                         JSR     PC,TIMER
004312  012605                                 MOV     (SP)+,R5
004314  010506                                 MOV     R5,SP
004316  012605                                 MOV     (SP)+,R5
                                        ; DID IT BECOME READY?
004320                                  IF.ERROR  THEN
004320  103402                                 BCS     .+6
004322  000137  004330                         JMP     $50062
                                        ;RCVRDONE DID NOT SET IN RCSR
004326                                  ERRHRD 73
004326  104073                  ERROR   73
                                        ; SET IT BACK TO CONTINUE
004330                                  ENDIF
004330                                          $50062:
004330                                  ENDSUB

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

```
 1746
 1747
 1748  004330
 1749  004330  117700  174730                                    .READ BUFFER
 1750                                                        LET R0 :B= @RBUF
 1751  004334                                                        MOVB    @RBUF,R0
 1752  004334  032777  000200  174720                        IF #BIT07 SETIN @RCSR THEN
 1753  004342  001002                                               BIT     #BIT07,@RCSR
 1754  004344  000137  004352                                       BNE     .+6
 1755                                                                JMP     $50063
 1756  004350                                                    ;RCVRDONE DID NOT CLEAR IN RCSR
 1757  004350  104074                    ERROR   74             ERRHRD 74
 1758
 1759  004352                                                    ; SET IT BACK TO CONTINUE
 1760  004352                                                ENDIF
                                                                    $50063:
```

```
1761
1762                                    ;;*****************************************************************
1763                                    ;*TEST 12       TEST THE OVERRUN & ERROR BITS - RBUF 14
1764                                    ;;*****************************************************************
1765  004352  000004                    TST12:  SCOPE
1766  004354  012737  000012  001200            MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1767
1768  004362                                                    IF CONSOL EQ #0 THEN
1769  004362  005737  001324                                                   TST     CONSOL
1770  004366  001402                                                           BEQ     .+6
1771  004370  000137  004424                                                   JMP     $50064
1772  004374                                                        IF #BIT04 SETIN $USWR THEN
1773  004374  032737  000020  001220                                           BIT     #BIT04,$USWR
1774  004402  001002                                                           BNE     .+6
1775  004404  000137  004414                                                   JMP     $50065
1776                                                                    ;; NULL ---EXECUTE TEST
1777  004410                                                        ELSF
1778  004410  000137  004420                                                   JMP     $50066
1779  004414                                                                    $50065:
1780  004414  000137  005000                                               JMP     TST13
1781  004420                                                        ENDIF
1782  004420                                                                    $50066:
1783  004420                                                    ELSE
1784  004420  000137  004430                                                   JMP     $50067
1785  004424                                                                    $50064:
1786  004424  000137  005000                                           JMP TST13
1787  004430                                                    ENDIF
1788  004430                                                                    $50067:
1789
1790
1791
1792  004430                                                            BGNSUB
1793  004430  012737  004436  001110                                           MOV     #64$,$LPERR
1794                                    ;OUTPUT 2 CHARACTERS
1795                                    ;THIS SHOULD AN CAUSE OVERRUN ERROR.
1796
1797                                                                ;OUTPUT 1 CHARACTER
1798  004436                                                    LET @TBUF :B= #0
1799  004436  105077  174626                                                   CLRB    @TBUF
1800
1801  004442                                                    CALL TIMER IN <#500,#BIT07,RCSR,#-1>
1802  004442  010546                                                           MOV     R5,-(SP)
1803  004444  010605                                                           MOV     SP,R5
1804  004446  162706  000010                                                   SUB     #10,SP
1805  004452  010546                                                           MOV     R5,-(SP)
1806  004454  012745  177777                                                   MOV     #-1,-(R5)
1807  004460  013745  001262                                                   MOV     RCSR,-(R5)
1808  004464  012745  000200                                                   MOV     #BIT07,-(R5)
1809  004470  012745  000500                                                   MOV     #500,-(R5)
1810  004474  004737  016272                                                   JSR     PC,TIMER
1811  004500  012605                                                           MOV     (SP)+,R5
1812  004502  010506                                                           MOV     R5,SP
1813  004504  012605                                                           MOV     (SP)+,R5
1814
1815                                                                ;DID IT SET IN TIME?
1816  004506                                                    IF.ERROR THEN
```

```
1817  004506  103402                                                                  BCS     .+6
1818  004510  000137  004516                                                          JMP     $50070
1819                                                                          ;RCVRDONE DID NOT SET IN RCSR
1820  004514                                                                  ERRHRD 16
1821  004514  104016                        ERROR   16
1822  004516                                                                ENDIF
1823  004516                                                                          $50070:
1824
1825                                                                          ;OUTPUT 2ND CHARACTER
1826  004516                                                                LET @TBUF :B= #0
1827  004516  105077  174546                                                         CLRB    @TBUF
1828                                                                          ;LET OVERRUN HAPPEN
1829  004522                                                                  WAITMS 300.
1830  004522  010546                                                                 MOV     R5,-(SP)
1831  004524  010605                                                                 MOV     SP,R5
1832  004526  162706  000002                                                        SUB     #2,SP
1833  004532  010546                                                                 MOV     R5,-(SP)
1834  004534  012745  000454                                                        MOV     #300.,-(R5)
1835  004540  004737  016524                                                        JSR     PC,WAIT
1836  004544  012605                                                                 MOV     (SP)+,R5
1837  004546  010506                                                                 MOV     R5,SP
1838  004550  012605                                                                 MOV     (SP)+,R5
1839
1840                                                                          ;READ BUFFER AND ERROR BITS
1841  004552                                                                LET R4 := @RBUF
1842  004552  017704  174506                                                         MOV     @RBUF,R4
1843
1844                                                                          ;IT DIDN'T SET
1845  004556                                                                IF #BIT14 NOTSETIN R4 THEN
1846  004556  032704  040000                                                         BIT     #BIT14,R4
1847  004562  001402                                                                 BEQ     .+6
1848  004564  000137  004574                                                        JMP     $50071
1849                                                                          ;ORERR DID NOT SET IN RBUF
1850  004570                                                                  ERRHRD 101
1851  004570  104101                        ERROR   101
1852
1853                                                                          ;NO USE COMPOUNDING ERRORS
1854  004572                                                                  EXIT
1855  004572  000502                        BR      TST13           ;;EXIT THIS TEST
1856  004574                                                                ENDIF
1857  004574                                                                          $50071:
1858  004574                                                              ENDSUB
1859
1860                                        ;NOW SEE IF ERROR BIT SET WITH OVERRUN  ERROR:
1861  004574                                                              BGNSUB
1862  004574  012737  004602  001110                                                 MOV     #64$,$LPERR
1863  004602                                                                IF #ERROR NOTSETIN R4 THEN
1864  004602  032704  104000                                                         BIT     #ERROR,R4
1865  004606  001402                                                                 BEQ     .+6
1866  004610  000137  004620                                                        JMP     $50072
1867
1868                                                                          ;ERROR DID NOT SET IN RBUF
1869  004614                                                                  ERRHRD 102
1870  004614  104102                        ERROR   102
1871
1872                                                                          ;-WHEN ORERR SET.
```

```
1873                                                                    ;GET OUT NOW.
1874  004616                                                            EXIT
1875  004616  000470                    BR      TST13           ;;EXIT THIS TEST
1876  004620                                                            ENDIF
1877  004620                                                                  $50072:
1878  004620                                                          ENDSUB
1879
1880  004620                                                          BGNSUB
1881  004620  012737  004626  001110                                         MOV     #64$,$LPERR
1882                                                  ;CHECK REAL RBUF TO SEE IF ORERR IS STILL SET.
1883
1884  004626                                                              IF #BIT14 NOTSETIN @RBUF THEN
1885  004626  032777  040000  174430                                         BIT     #BIT14,@RBUF
1886  004634  001402                                                        BEQ     .+6
1887  004636  000137  004646                                                JMP     $50073
1888
1889                                                                    ;READING RBUF CLEARED ORERR.
1890  004642                                                            ERRHRD 103
1891  004642  104103                    ERROR   103
1892                                                                    ;SKIP REST OF TEST
1893  004644                                                            EXIT
1894  004644  000455                    BR      TST13           ;;EXIT THIS TEST
1895  004646                                                            ENDIF
1896  004646                                                                  $50073:
1897  004646                                                          ENDSUB
1898
1899  004646                                                          BGNSUB
1900  004646  012737  004654  001110                                         MOV     #64$,$LPERR
1901                                                  ;READING RBUF ABOVE SHOULD ENABLE ERROR TO BE CLEARED
1902                                                  ;BY NEXT TRANSFER.
1903                                                  ;NOW SEE IF THEY CLEAR WHEN ANOTHER CHAR. IS RECEIVED
1904
1905                                                                    ;SEND A CHARACTER AROUND.
1906  004654                                                        LET @TBUF :B= #0
1907  004654  105077  174410                                               CLRB    @TBUF
1908  004660                                                      CALL TIMER IN <#500,#BIT07,RCSR,#-1>
1909  004660  010546                                                       MOV     R5,-(SP)
1910  004662  010605                                                       MOV     SP,R5
1911  004664  162706  000010                                              SUB     #10,SP
1912  004670  010546                                                       MOV     R5,-(SP)
1913  004672  012745  177777                                              MOV     #-1,-(R5)
1914  004676  013745  001262                                              MOV     RCSR,-(R5)
1915  004702  012745  000200                                              MOV     #BIT07,-(R5)
1916  004706  012745  000500                                              MOV     #500,-(R5)
1917  004712  004737  016272                                              JSR     PC,TIMER
1918  004716  012605                                                       MOV     (SP)+,R5
1919  004720  010506                                                       MOV     R5,SP
1920  004722  012605                                                       MOV     (SP)+,R5
1921
1922                                                                    ;DID IT SET IN TIME?
1923  004724                                                      IF.ERROR THEN
1924  004724  103402                                                       BCS     .+6
1925  004726  000137  004734                                              JMP     $50074
1926                                                                  ;RCVRDONE DID NOT SET IN RCSR
1927  004732                                                            ERRHRD 20
1928  004732  104020                    ERROR   20
```

```
1929  004734                                                          ENDIF
1930  004734                                                                  $50074:
1931
1932  004734                                                      IF #BIT14 SETIN @RBUF THEN
1933  004734  032777  040000  174322                                        BIT     #BIT14,@RBUF
1934  004742  001002                                                        BNE     .+6
1935  004744  000137  004754                                                JMP     $50075
1936                                                                  ;URERR DID NOT CLEAR IN RBUF
1937  004750                                                            ERRHRD 104
1938  004750  104104                      ERROR   104
1939
1940                                                                    ;-AFTER RECEIVING ANOTHER CHAR
1941                                                                    ;SKIP AROUND REST
1942  004752                                                            EXIT
1943  004752  000412                      BR      TST13           ;;EXIT THIS TEST
1944  004754                                                          ENDIF
1945  004754                                                                  $50075:
1946
1947  004754                                                      IF #ERROR SETIN @RBUF THEN
1948  004754  032777  104000  174302                                        BIT     #ERROR,@RBUF
1949  004762  001002                                                        BNE     .+6
1950  004764  000137  004772                                                JMP     $50076
1951                                                                  ;ERROR DID NOT CLEAR IN RBUF
1952  004770                                                            ERRHRD 105
1953  004770  104105                      ERROR   105
1954
1955  004772                                                          ENDIF
1956  004772                                                                  $50076:
1957  004772                          LET R0 :B= @RBUF                ;READ RBUF TO CLR IT
1958  004772  117700  174266                                                MOVB    @RBUF,R0
1959  004776                                                          ENDSUB
1960  004776                                                          EXIT
1961  004776  000400                      BR      TST13           ;;EXIT THIS TEST
```

```
                                        ;;*****************************************************************
                                        ;*TEST 13       TRANSMITTER INTERRUPT LOGIC TEST
                                        ;*             LOGICALLY 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
                                        ;;*****************************************************************
005000  000004                          TST13:  SCOPE
005002  012737  000013  001200                  MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                                        ;CLEAR 'INTERRUPT OCCURED' FLAG
005010                                  ESR:                                    LET INTFLG := #0
005010  005037  017640                                                          CLR     INTFLG
                                                                        ;GET VECTOR ADDRESS
005014                                                                  LET R3 := DLVEC
005014  013703  001260                                                          MOV     DLVEC,R3
                                                                        ;FOR THE TRANSMITTER
005020                                                                  LET R3 := R3 + #4
005020  062703  000004                                                          ADD     #4,R3
                                                                        ;SET VECTOR TO POINT TO TRANS.SRV AT PRI
005024  012723  017632                          MOV     #INTSRV,(R3)+
005030  012713  000340                          MOV     #PR7,(R3)
005034                                                                  BGNSUB
005034  012737  005042  001110                                                  MOV     #64$,$LPERR
                                                                        ;; MAKE SURE THAT TRANSMITTER READY IS SET
005042                                                                  CALL TIMER IN <#500,#BIT07,TCSR,#-1>
005042  010546                                                                  MOV     R5,-(SP)
005044  010605                                                                  MOV     SP,R5
005046  162706  000010                                                          SUB     #10,SP
005052  010546                                                                  MOV     R5,-(SP)
005054  012745  177777                                                          MOV     #-1,-(R5)
005060  013745  001266                                                          MOV     TCSR,-(R5)
005064  012745  000200                                                          MOV     #BIT07,-(R5)
005070  012745  000500                                                          MOV     #500,-(R5)
005074  004737  016272                                                          JSR     PC,TIMER
005100  012605                                                                  MOV     (SP)+,R5
005102  010506                                                                  MOV     R5,SP
005104  012605                                                                  MOV     (SP)+,R5
                                                                        ;CLEAR INTERRUPT ENABLE
005106                                                                  LET @TCSR := @TCSR CLR.BY #BIT06
005106  042777  000100  174152                                                  BIC     #BIT06,@TCSR
                                                                        ;SET IT TO 0
005114                                  SETPRI #PR0
                                                                        ;NOW SET I.E. BIT
005126                                                                  LET @TCSR := @TCSR SET.BY #BIT06
005126  052777  000100  174132                                                  BIS     #BIT06,@TCSR
005134                                                                  CALL TIMER IN <#500,#1,#INTFLG,#-1>
005134  010546                                                                  MOV     R5,-(SP)
005136  010605                                                                  MOV     SP,R5
005140  162706  000010                                                          SUB     #10,SP
```

```
2018  005144  010546                                                          MOV     R5,-(SP)
2019  005146  012745  177777                                                  MOV     #-1,-(R5)
2020  005152  012745  017640                                                  MOV     #INTFLG,-(R5)
2021  005156  012745  000001                                                  MOV     #1,-(R5)
2022  005162  012745  000500                                                  MOV     #500,-(R5)
2023  005166  004737  016272                                                  JSR     PC,TIMER
2024  005172  012605                                                          MOV     (SP)+,R5
2025  005174  010506                                                          MOV     R5,SP
2026  005176  012605                                                          MOV     (SP)+,R5
2027
2028                                                                  ;DID IT SET IN TIME?
2029  005200                                                        IF.ERROR THEN
2030  005200  103402                                                          BCS     .+6
2031  005202  000137  005210                                                  JMP     $50077
2032                                                                  ;INTERRUPT DID NOT OCCUR
2033  005206                                                                  ERRHRD 106
2034  005206  104106                         ERROR   106
2035  005210                                                        ENDIF
2036  005210                                                                  $50077:
2037
2038                                                                  ;LET POSSIBLE 2'ND INTERR OCCUR
2039  005210                                                              WAITMS 500.
2040  005210  010546                                                          MOV     R5,-(SP)
2041  005212  010605                                                          MOV     SP,R5
2042  005214  162706  000002                                                  SUB     #2,SP
2043  005220  010546                                                          MOV     R5,-(SP)
2044  005222  012745  000764                                                  MOV     #500.,-(R5)
2045  005226  004737  016524                                                  JSR     PC,WAIT
2046  005232  012605                                                          MOV     (SP)+,R5
2047  005234  010506                                                          MOV     R5,SP
2048  005236  012605                                                          MOV     (SP)+,R5
2049
2050                                                                      ;DID EXACTLY 1 INTERRUPT OCCUR
2051  005240                                                              IF INTFLG GT #1 THEN
2052  005240  023727  017640  000001                                              CMP     INTFLG,#1
2053  005246  003002                                                              BGT     .+6
2054  005250  000137  005256                                                      JMP     $50100
2055                                                                          ;TRANSMITTER INTERRUPTED TWICE
2056  005254                                                                      ERRHRD 107
2057  005254  104107                         ERROR   107
2058  005256                                                              ENDIF
2059  005256                                                                      $50100:
2060  005256                                                            ENDSUB
2061                                                              ;INTERRUPT WITHOUT INTERRUPT ENABLE SET
2062  005256                                                            BGNSUB
2063  005256  012737  005264  001110                                              MOV     #64$,$LPERR
2064                                                                      ;CLEAR INTERRUPT ENABLE
2065  005264                                                              LET @TCSR := @TCSR CLR.BY #BIT06
2066  005264  042777  000100  173774                                              BIC     #BIT06,@TCSR
2067                                                                      ;CLEAR 'INTERRUPT OCCURED' FLAG
2068  005272                                                              LET INTFLG := #0
2069  005272  005037  017640                                                     CLR     INTFLG
2070                                                                  ;NO INTERRUPTS SHOULD OCCUR, PSW STILL AT 0.
2071                                                                  ;DARE IT TO HAPPEN
2072  005276                                                                  LET @TBUF := #0
2073  005276  005077  173766                                                  CLR     @TBUF
```

```
                                                        ;SEE IF INT FLAG EVER SETS
                                                        CALL TIMER IN <#1000,#1,#INTFLG,#-1>
005302                                                              MOV     R5,-(SP)
005302  010546                                                     MOV     SP,R5
005304  010605                                                     SUB     #10,SP
005306  162706  000010                                             MOV     R5,-(SP)
005312  010546                                                     MOV     #-1,-(R5)
005314  012745  177777                                             MOV     #INTFLG,-(R5)
005320  012745  017640                                             MOV     #1,-(R5)
005324  012745  000001                                             MOV     #1000,-(R5)
005330  012745  001000                                             JSR     PC,TIMER
005334  004737  016272                                             MOV     (SP)+,R5
005340  012605                                                     MOV     R5,SP
005342  010506                                                     MOV     (SP)+,R5
005344  012605
                                                        ;C BIT SHOULD BE SET
005346  103403                  BCS     1$
                                                                ;;NTERR STILL OCCURED WITH IE DISABLED
                                                                ERRHRD 110
005350
005350  104110                  ERROR   110
005352                  LET R0 :B= @RBUF                ;READ RBUF TO CLR IT
005352  117700  173706                                             MOVB    @RBUF,R0
005356                  1$:

                        ;;**********************************************************************
                        ;*TEST 14       RECEIVER INTERRUPT LOGIC TEST
                        ;*              THIS TEST COVERS ALL OF THE RECEIVER
                        ;*              SIDE OF THE INTERRUPT LOGIC IN
                        ;*              CHARACTER MODE.
                        ;;**********************************************************************
005356  000004          TST14:  SCOPE
005360  012737  000014  001200          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                            IF #BIT04 NOTSETIN $USWR OR CONSOL EQ #BIT00 THEN
005366  032737  000020  001220                                             BIT     #BIT04,$USWR
005374  001406                                                             BEQ     $50101
005376  023727  001324  000001                                             CMP     CONSOL,#BIT00
005404  001402                                                             BEQ     .+6
005406  000137  005416                                                     JMP     $50102
005412                                                                     $50101:
005412  000137  006034              JMP     TST15           ;       EXIT TEST
005416                              ENDIF
005416                                                                     $50102:
                                                                ;CLEAR INTERRUPT OCCURED FLAG
                                                                ;SET UP RECEIVER INTER.VECTOR
005416                                                  SETVEC DLVEC,#INTSRV,#PR7
005416  010146                                                     MOV     R1,-(SP)
005420  013701  001260                                             MOV     DLVEC,R1
005424  012721  017632                                             MOV     #INTSRV,(R1)+
005430  012711  000340                                             MOV     #PR7,(R1)
005434  012601                                                     MOV     (SP)+,R1
                                        ;PRIORITY 0 AND MULTIPLE INTERRUPT TEST.-RCVRIE
005436                                                  BGNSUB
005436  012737  005444  001110                                     MOV     #64$,$LPERR
005444                                                  LET INTFLG := #0
005444  005037  017640                                             CLR     INTFLG
                                                                ;CLEAR INTERRUPTS
                                                            LET @RCSR := @RCSR CLR.BY #BIT06
005450
```

```
2130  005450  042777  000100  173604                          BIC     #BIT06,@RCSR
2131                                                      ;CHANGE PRIORITY
2132                                                      ;..TO 0
2133  005456                          SETPRI #PR0
2134
2135                                                      ;SEND A CHARACTER
2136  005470                                            LET @TBUF :B= #0
2137  005470  105077  173574                                   CLRB    @TBUF
2138                                                      ;WAIT A MAXIMUM
2139                                                    ;OF 50 MSEC FOR
2140                                                    ;XMIT RDY TO SET IN TCSR
2141  005474                                            CALL TIMER IN <#500,#BIT07,TCSR,#-1>
2142  005474  010546                                           MOV     R5,-(SP)
2143  005476  010605                                           MOV     SP,R5
2144  005500  162706  000010                                   SUB     #10,SP
2145  005504  010546                                           MOV     R5,-(SP)
2146  005506  012745  177777                                   MOV     #-1,-(R5)
2147  005512  013745  001266                                   MOV     TCSR,-(R5)
2148  005516  012745  000200                                   MOV     #BIT07,-(R5)
2149  005522  012745  000500                                   MOV     #500,-(R5)
2150  005526  004737  016272                                   JSR     PC,TIMER
2151  005532  012605                                           MOV     (SP)+,R5
2152  005534  010506                                           MOV     R5,SP
2153  005536  012605                                           MOV     (SP)+,R5
2154                                                      ;SET INTERRUPT ENABLE
2155  005540                                            LET @RCSR := @RCSR SET.BY #BIT06
2156  005540  052777  000100  173514                           BIS     #BIT06,@RCSR
2157                                                      ;LET IT COME IN.
2158
2159  005546                                            CALL TIMER IN <#500,#1,#INTFLG,#-1>
2160  005546  010546                                           MOV     R5,-(SP)
2161  005550  010605                                           MOV     SP,R5
2162  005552  162706  000010                                   SUB     #10,SP
2163  005556  010546                                           MOV     R5,-(SP)
2164  005560  012745  177777                                   MOV     #-1,-(R5)
2165  005564  012745  017640                                   MOV     #INTFLG,-(R5)
2166  005570  012745  000001                                   MOV     #1,-(R5)
2167  005574  012745  000500                                   MOV     #500,-(R5)
2168  005600  004737  016272                                   JSR     PC,TIMER
2169  005604  012605                                           MOV     (SP)+,R5
2170  005606  010506                                           MOV     R5,SP
2171  005610  012605                                           MOV     (SP)+,R5
2172
2173                                                      ;DID IT SET IN TIME?
2174  005612                                            IF.ERROR THEN
2175  005612  103402                                           BCS     .+6
2176  005614  000137  005622                                   JMP     $50103
2177                                                      ;INTERRUPT DID NOT OCCUR
2178  005620                                                    ERRHRD 111
2179  005620  104111                  ERROR   111
2180  005622                                            ENDIF
2181  005622                                                    $50103:
2182                                                    ;LET POSSIBLE 2'ND INTERR OCCUR
2183  005622                                                    WAITMS 500
2184  005622  010546                                           MOV     R5,-(SP)
2185  005624  010605                                           MOV     SP,R5
```

```
2186  005626  162706  000002                                          SUB     #2,SP
2187  005632  010546                                                  MOV     R5,-(SP)
2188  005634  012745  000500                                          MOV     #500,-(R5)
2189  005640  004737  016524                                          JSR     PC,WAIT
2190  005644  012605                                                  MOV     (SP)+,R5
2191  005646  010506                                                  MOV     R5,SP
2192  005650  012605                                                  MOV     (SP)+,R5
2193
2194                                                          ;EXACTLY 1 INTERRUPT?
2195  005652                                                IF INTFLG GT #1 THEN
2196  005652  023727  017640  000001                                  CMP     INTFLG,#1
2197  005660  003002                                                  BGT     .+6
2198  005662  000137  005670                                          JMP     $50104
2199                                                      ;RECEIVER INTERRUPTED TWICE
2200  005666                                                  ERRHRD 112
2201  005666  104112                          ERROR   112
2202  005670                                                ENDIF
2203  005670                                                          $50104:
2204  005670                                              FNDSUB
2205
2206
2207
2208
2209
2210
2211                                        ;INTERRUPT WITHOUT IE SET.
2212  005670                                                BGNSUB
2213  005670  012737  005676  001110                                  MOV     #64$,$LPERR
2214
2215                                                      ;CLEAR INTERRUPT
2216  005676                                            LET @RCSR := @RCSR CLR.BY #BIT06
2217  005676  042777  000100  173356                                  BIC     #BIT06,@RCSR
2218                                                      ;CLEAR INTERRUPT FLAG
2219  005704                                            LET INTFLG := #0
2220  005704  005037  017640                                          CLR     INTFLG
2221                                                      ; SEND A CHARACTER
2222  005710                                            LET @TBUF :B= #0
2223  005710  105077  173354                                          CLRB    @TBUF
2224                                                      ; DARE IT
2225
2226  005714                                          CALL TIMER IN <#500,#1,#INTFLG,#CLR>
2227  005714  010546                                                  MOV     R5,-(SP)
2228  005716  010605                                                  MOV     SP,R5
2229  005720  162706  000010                                          SUB     #10,SP
2230  005724  010546                                                  MOV     R5,-(SP)
2231  005726  012745  005000                                          MOV     #CLR,-(R5)
2232  005732  012745  017640                                          MOV     #INTFLG,-(R5)
2233  005736  012745  000001                                          MOV     #1,-(R5)
2234  005742  012745  000500                                          MOV     #500,-(R5)
2235  005746  004737  016272                                          JSR     PC,TIMER
2236  005752  012605                                                  MOV     (SP)+,R5
2237  005754  010506                                                  MOV     R5,SP
2238  005756  012605                                                  MOV     (SP)+,R5
2239                                                      ;DID IT CLEAR IN TIME?
2240  005760                                          IF.ERROR THEN
2241  005760  103402                                                  BCS     .+6
```

```
2242  005762  000137  005770                                                        JMP     $50105
2243                                                                          ;INTERR STILL OCCURED WITH IE DISABLED
2244  005766                                                                  ERRHRD 113
2245  005766  104113                 ERROR   113
2246  005770                                                                ENDIF
2247  005770                                                                        $50105:
2248  005770                        LET R0 :B= @RBUF                        ;READ RBUF TO CLR IT
2249  005770  117700  173270                                                       MOVB    @RBUF,R0
2250  005774                                                                ENDSUB
2251  005774                        SETPRI  #PR7    ;RAISE CPU PRIORITY
2252  006006                        IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
2253  006006  032737  000002  001214                                               BIT     #BIT01,$ENV
2254  006014  001406                                                               BEQ     $50106
2255  006016  023727  001202  000001                                               CMP     $PASS,#1
2256  006024  002402                                                               BLT     .+6
2257  006026  000137  006034                                                       JMP     $50107
2258  006032                                                                        $50106:
2259                                                                        ;CLEAR THE WORLD
2260  006032                                                                    BRESET
2261  006032  000005                 RESET
2262  006034                        ENDIF
2263  006034                                                                        $50107:
2264
2265
2266
2267                                ;;*******************************************************************
2268                                ;*TEST 15       TEST DATA  WRAP AROUND: FLAG MODE
2269                                ;;*******************************************************************
2270  006034  000004                TST15:  SCOPE
2271  006036  012737  000015  001200         MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2272  006044                                                                IF #BIT04 NOTSETIN $USWR OR CONSOL EQ #BIT00 THE
2273  006044  032737  000020  001220                                               BIT     #BIT04,$USWR
2274  006052  001406                                                               BEQ     $50110
2275  006054  023727  001324  000001                                               CMP     CONSOL,#BIT00
2276  006062  001402                                                               BEQ     .+6
2277  006064  000137  006072                                                       JMP     $50111
2278  006070                                                                        $50110:
2279                                                                          ;CAN'T TEST WITHOUT A WRAP
2280  006070                                                                    EXIT
2281  006070  000560                 BR      TST16           ;;EXIT THIS TEST
2282  006072                                                                ENDIF
2283  006072                                                                        $50111:
2284  006072                        IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
2285  006072  032737  000002  001214                                               BIT     #BIT01,$ENV
2286  006100  001406                                                               BEQ     $50112
2287  006102  023727  001202  000001                                               CMP     $PASS,#1
2288  006110  002402                                                               BLT     .+6
2289  006112  000137  006120                                                       JMP     $50113
2290  006116                                                                        $50112:
2291  006116                                                                BRESET
2292  006116  000005                 RESET
2293  006120                        ENDIF
2294  006120                                                                        $50113:
2295                                                                          ;BINARY COUNT PATTERN
2296  006120                                                                INCR R2 FROM #0 TO #377 BY #1
2297  006120  005002                                                               CLR     R2
```

```
2298  006122  000401                                                          BR      $50115
2299  006124                                                                  $50114:
2300  006124  005202                                                          INC     R2
2301  006126                                                                  $50115:
2302  006126  020227  000377                                                 CMP     R2,#377
2303  006132  003402                                                          BLE     $50116
2304  006134  000137  006432                                                 JMP     $50117
2305  006140                                                                  $50116:
2306
2307
2308
2309                                                                      ; MAKE SURE IT'S READY
2310  006140                                                            CALL TIMER IN <#500,#BIT07,TCSR,#-1>
2311  006140  010546                                                          MOV     R5,-(SP)
2312  006142  010605                                                          MOV     SP,R5
2313  006144  162706  000010                                                 SUB     #10,SP
2314  006150  010546                                                          MOV     R5,-(SP)
2315  006152  012745  177777                                                 MOV     #-1,-(R5)
2316  006156  013745  001266                                                 MOV     TCSR,-(R5)
2317  006162  012745  000200                                                 MOV     #BIT07,-(R5)
2318  006166  012745  000500                                                 MOV     #500,-(R5)
2319  006172  004737  016272                                                 JSR     PC,TIMER
2320  006176  012605                                                          MOV     (SP)+,R5
2321  006200  010506                                                          MOV     R5,SP
2322  006202  012605                                                          MOV     (SP)+,R5
2323  006204                                                            IF.ERROR THEN
2324  006204  103402                                                          BCS     .+6
2325  006206  000137  006216                                                 JMP     $50120
2326                                                                      ; TRANSMITTER NEVER BECAME READY
2327  006212                                                              ERRHRD 123
2328  006212  104123                  ERROR   123
2329  006214                                                              EXIT
2330  006214  000506                  BR      TST16           ;;EXIT THIS TEST
2331  00621C                                                            ENDIF
2332  006216                                                                  $50120:
2333
2334                                                                      ;START IT ON ITS WAY
2355  006216                                                            LET @TBUF :B= R2
2336  006216  110277  173046                                                  MOVB    R2,@TBUF
2337                                                                    ; NOW WAIT FOR RECIEVER DONE
2338  006222                                                            CALL TIMER IN <#500,#BIT07,RCSR,#-1>
2339  006222  010546                                                          MOV     R5,-(SP)
2340  006224  010605                                                          MOV     SP,R5
2341  006226  162706  000010                                                 SUB     #10,SP
2342  006232  010546                                                          MOV     R5,-(SP)
2343  006234  012745  177777                                                 MOV     #-1,-(R5)
2344  006240  013745  001262                                                 MOV     RCSR,-(R5)
2345  006244  012745  000200                                                 MOV     #BIT07,-(R5)
2346  006250  012745  000500                                                 MOV     #500,-(R5)
2347  006254  004737  016272                                                 JSR     PC,TIMER
2348  006260  012605                                                          MOV     (SP)+,R5
2349  006262  010506                                                          MOV     R5,SP
2350  006264  012605                                                          MOV     (SP)+,R5
2351  006266                                                            IF.ERROR THEN
2352  006266  103402                                                          BCS     .+6
2353  006270  000137  006300                                                 JMP     $50121
```

```
2354  006274                                                                    ERRHRD 124
2355  006274  104124                          ERROR   124
2356                                                                          ; RECIEVER NEVER BECAME READY
2357  006276                                                                  EXIT
2358  006276  000455                          BR      TST16           ;;EXIT THIS TEST
2359  006300                                                                  ENDIF
2360  006300                                                                          $50121:
2361
2362                                                                          ;RETRIEVE
2363  006300                                                                  LET R3 := @RBUF
2364  006300  017703  172760                                                         MOV     @RBUF,R3
2365                                                                          ;CHECK FOR ERROR DURING TRANSFER
2366  006304                                                                  IF #ERROR SETIN R3 THEN
2367  006304  032703  104000                                                         BIT     #ERROR,R3
2368  006310  001002                                                                 BNE     .+6
2369  006312  000137  006320                                                        JMP     $50122
2370                                                                          ;ERROR BIT SET IN HIGH BYTE OF RBUF
2371  006316                                                                  ERRHRD 200
2372  006316  104200                          ERROR   200
2373  006320                                                                  ENDIF
2374  006320                                                                          $50122:
2375
2376
2377                                                                          ;COMPARE DATA
2378  006320                                                                IF R3 NE R2 THEN
2379  006320  020302                                                                 CMP     R3,R2
2380  006322  001002                                                                 BNE     .+6
2381  006324  000137  006376                                                        JMP     $50123
2382  006330                                                                  IF #BIT0 NOTSETIN $USWR THEN
2383  006330  032737  000001  001220                                                BIT     #BIT0,$USWR
2384  006336  001402                                                                BEQ     .+6
2385  006340  000137  006366                                                        JMP     $50124
2386  006344                                                                  IF R2 LT #200 THEN
2387  006344  020227  000200                                                         CMP     R2,#200
2388  006350  002402                                                                BLT     .+6
2389  006352  000137  006362                                                        JMP     $50125
2390                                                                          ;DATA COMPARE ERR IN 7 BIT WORD
2391  006356                                                                  ERRHRD 117
2392  006356  104117                          ERROR   117
2393  006360                                                                  EXIT
2394  006360  000424                          BR      TST16           ;;EXIT THIS TEST
2395  006362                                                                  ENDIF
2396  006362                                                                          $50125:
2397  006362                                                                ELSE
2398  006362  000137  006372                                                        JMP     $50126
2399  006366                                                                          $50124:
2400                                                                          ;DATA COMP ERR IN 8 BIT WORD
2401  006366                                                                  ERRHRD 17
2402  006366  104017                          ERROR   17
2403  006370                                                                  EXIT
2404  006370  000420                          BR      TST16           ;;EXIT THIS TEST
2405  006372                                                                  ENDIF
2406  006372                                                                          $50126:
2407  006372                                                                ELSE IF #BIT0 NOTSETIN $USWR AND R2 GT #177 THEN
2408  006372  000137  006430                                                        JMP     $50127
2409  006376                                                                          $50123:
```

```
006376  032737  000001  001220                                                  BIT     #BIT0,$USWR
006404  001402                                                                  BEQ     .+6
006406  000137  006430                                                          JMP     $50130
006412  020227  000177                                                          CMP     R2,#177
006416  003002                                                                  BGT     .+6
006420  000137  006430                                                          JMP     $50130
                                                                        ;GETTING 8 BITS ON 7 BIT XMIT
                                                                        ;MAKE SURE $USWR SETUP CORRECTLY.
006424                                                                  ERRHRD 22
006424  104022                          ERROR   22
006426                                                                  EXIT
006426  000401                          BR      TST16           ;;EXIT THIS TEST
006430                                                          ENDIF
006430                                                                          $50130:
006430                                                                          $50127:
006430                                                  ENDINC  ;R2
006430  000635                                                                  BR      $50114
006432                                                                          $50117:

                                ;;*****************************************************************
                                ;*TEST 16        TEST DATA WRAP AROUND: INTEPRUPT MODE
                                ;;*****************************************************************
006432  000004                  TST16:  SCOPE
006434  012737  000016  001200          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
006442                                                          IF #BIT04 NOTSETIN $USWR OR CONSOL EQ #BIT00 THE
006442  032737  000020  001220                                                  BIT     #BIT04,$USWR
006450  001406                                                                  BEQ     $50131
006452  023727  001324  000001                                                  CMP     CONSOL,#BIT00
006460  001402                                                                  BEQ     .+6
006462  000137  006472                                                          JMP     $50132
006466                                                                          $50131:
006466  000137  007156                  JMP     TST17                   ;EXIT TEST
006472                                                          ENDIF
006472                                                                          $50132:

                                        ; THIS TEST WILL RUN BOTH TRANSMITTER AND
                                        ; RECIEVER AT FULL SPEED TESTING
                                        ; THE ABILITY OF THE MODULE
                                        ; TO HANDLE INTERRUPTS FROM BOTH SIDES AT ONCE.
                                        ;
                                        ;DOUBLE BUFFERING IS NOT FULLY TESTED BECAUSE OF
                                        ;APT CONSIDERATIONS. I.E. 'BREAK' FROM APT
                                        ;CAUSES OVERRUN ERRORS. THEREFORE TRANSMIT INTR IS
                                        ;ENABLED ONLY AFTER THE RECVR HAS OBTAINED THE LAST WORD

                                        ;  THIS TEST WILL TRANSFER A MAXIMUM OF 400(8)
                                        ; CHARACTERS THROUGH THE MODULE, BUT IF AN ERROR
                                        ; IS DETECTED BY THE TEST A PREMATURE SHUTDOWN OCCURS.

                                                                        ;CHANGE PRIORITY
                                                                        ;..TO 0
006472                          SETPRI #PR0
                                                                        ;GET VECTOR ADDRESS
006504                                                          LET R1 := DLVEC
006504  013701  001260                                                          MOV     DLVEC,R1
                                                                        ;RCVR VECTOR
```

```
2466  006510
2467  006510  012721  007044                          LET (R1)+ := #REC
                                                              MOV     #REC,(R1)+
2468  006514                                          LET (R1)+ := #PR7
2469  006514  012721  000340                               MOV     #PR7,(R1)+
2470                                                        ;POINT TO TRANSMITTER VECTOR
2471                                                        ;AND SET IT UP ALSO
2472  006520                                          LET (R1)+ := #TRAN
2473  006520  012721  006776                               MOV     #TRAN,(R1)+
2474  006524                                          LET (R1)  := #PR7
2475  006524  012711  000340                               MOV     #PR7,(R1)
2476
2477                                                        ; CLEAR ERROR COUNTER
2478  006530                                          LET ERRCNT := #0
2479  006530  005037  006772                               CLR     ERRCNT
2480  006534                                          LET DATA := #0  ;XMIT DATA
2481  006534  005037  007042                               CLR     DATA
2482  006540                                          IF #BIT0 NOTSET IN $USWR THEN
2483  006540  032737  000001  001220                       BIT     #BIT0,$USWR
2484  006546  001402                                       BEQ     .+6
2485  006550  000137  006566                               JMP     $50133
2486  006554                                             LET NUMBER := #200
2487  006554  012737  000200  006774                       MOV     #200,NUMBER
2488  006562                                          ELSE
2489  006562  000137  006574                               JMP     $50134
2490  006566                                               $50133:
2491  006566                                             LET NUMBER := #400
2492  006566  012737  000400  006774                       MOV     #400,NUMBER
2493  006574                                          ENDIF
2494  006574                                               $50134:
2495
2496  006574                                          BRESET  ;SET UP ALL REGISTERS
2497  006574  000005                          RESET
2498
2499                                                        ;SET I.E. IN TRANSMITTER
2500  006576                                          LET @TCSR := @TCSR SET.BY #BIT06
2501  006576  052777  000100  172462                       BIS     #BIT06,@TCSR
2502                                                        ;AND RECEIVER
2503  006604                                          LET @RCSR := @RCSR SET.BY #BIT06
2504  006604  052777  000100  172450                       BIS     #BIT06,@RCSR
2505
2506
2507                                                    ;NOW WE WAIT
2508  006612                                          REPEAT
2509  006612                                               $50135:
2510  006612                                          UNTIL DATA EQ NUMBER OR ERRCNT GT #0
2511  006612  023737  007042  006774                       CMP     DATA,NUMBER
2512  006620  001403                                       BEQ     $50136
2513  006622  005737  006772                               TST     ERRCNT
2514  006626  003771                                       BLE     $50135
2515  006630                                               $50136:
2516
2517                                                    ;NOW LETS CHECK.
2518                                                    ;TURN OFF ALL INTR ENABLE
2519  006630                                          LET @TCSR := @TCSR CLR.BY #BIT06
2520  006630  042777  000100  172430                       BIC     #BIT06,@TCSR
2521  006636                                          LET @RCSR := @RCSR CLR.BY #BIT06
```

```
006636  042777  000100  172416                                               BIC     #BIT06,@RCSR
006644                                                       IF ERRCNT NE #0 THEN
006644  005737  006772                                                       TST     ERRCNT
006650  001002                                                               BNE     .+6
006652  000137  006770                                                       JMP     $50137
006656                                                          IF #ERROR SETIN RHLD THEN
006656  032737  104000  007164                                               BIT     #ERROR,RHLD
006664  001002                                                               BNE     .+6
006666  000137  006766                                                       JMP     $50140
006672                                                             IF #BIT14 SETIN RHLD THEN
006672  032737  040000  007164                                               BIT     #BIT14,RHLD
006700  001002                                                               BNE     .+6
006702  000137  006714                                                       JMP     $50141
                                                                     ;OVERRUN ERROR
006706                                                               ERRHRD 220
006706  104220                          ERROR   220
006710                                                             ELSE IF #BIT13 SETIN RHLD THEN
006710  000137  006762                                                       JMP     $50142
006714                                                                       $50141:
006714  032737  020000  007164                                               BIT     #BIT13,RHLD
006722  001002                                                               BNE     .+6
006724  000137  006736                                                       JMP     $50143
                                                                     ;FRAMING ERROR
006730                                                               ERRHRD 221
006730  104221                          ERROR   221
006732                                                             ELSE IF #BIT12 SETIN RHLD THEN
006732  000137  006762                                                       JMP     $50144
006736                                                                       $50143:
006736  032737  010000  007164                                               BIT     #BIT12,RHLD
006744  001002                                                               BNE     .+6
006746  000137  006760                                                       JMP     $50145
                                                                     ;PARITY ERROR
006752                                                               ERRHRD 222
006752  104222                          ERROR   222
006754                                                             ELSE
006754  000137  006762                                                       JMP     $50146
006760                                                                       $50145:
                                                                     ;UNKNOWN ERROR
006760                                                               ERRHRD 24
006760  104024                          ERROR   24
006762                                                             ENDIF
006762                                                                       $50146:
006762                                                                       $50144:
006762                                                                       $50142:
006762                                                          ELSE
006762  000137  006770                                                       JMP     $50147
006766                                                                       $50140:
                                                                      ;DATA COMPARE ERROR
006766                                                                ERRHRD 120
006766  104120                          ERROR   120
006770                                                          ENDIF
006770                                                                       $50147:
006770                                                       ENDIF
006770                                                                       $50137:
006770                                                       EXIT    ;SKIP OVER SUPPORT ROUTINES & STORAGE
006770  000476                          BR      TST17           ;;EXIT THIS TEST
```

```
2578
2579
2580  006772  000000                ERRCNT: 0
2581  006774  000000                NUMBER: 0
```

```
                                ;*****************************************************************
                                ;TRANSMIT INTERRUPT HANDLER
                                ;*****************************************************************
                                        BGNSRV  TRAN
006776                          TRAN:
                                            IF DATA NE NUMBER AND ERRCNT EQ #0 THEN
006776  023737  007042  006774                        CMP     DATA,NUMBER
007004  001002                                        BNE     .+6
007006  000137  007032                                JMP     $50150
007012  005737  006772                                TST     ERRCNT
007016  001402                                        BEQ     .+6
007020  000137  007032                                JMP     $50150
                                            ;SHIP OUT WORD
007024                                      LET @TBUF := DATA
007024  013777  007042  172236                        MOV     DATA,@TBUF
007032                                      ENDIF
007032                                              $50150:
                                            ;STOP INTERR, NOT EXER DOUBL BUFFER
007032                                      LET @TCSR := @TCSR CLR.BY #BIT06
007032  042777  000100  172226                        BIC     #BIT06,@TCSR
007040                                      ENDSRV
007040  000002                  RTI
                                
007042  000000                DATA:   0
                                ;*****************************************************************
                                ;RECEIVER INTERRUPT HANDLER
                                ;*****************************************************************
                                        BGNSRV  REC
007044                          REC:
                                            ;GET CHAR
007044                                      LET RHLD := @RBUF
007044  017737  172214  007164                       MOV     @RBUF,RHLD
                                            ;CHECK ERROR
007052                                      IF #ERROR SETIN RHLD OR RHLD NE  DATA THEN
007052  032737  104000  007164                       BIT     #ERROR,RHLD
007060  001006                                      BNE     $50151
007062  023737  007164  007042                       CMP     RHLD,DATA
007070  001002                                      BNE     .+6
007072  000137  007122                              JMP     $50152
007076                                              $50151:
                                               ;STOP ALL INTERR PROC & GET OUT
007076                                         LET DATA := NUMBER
007076  013737  006774  007042                       MOV     NUMBER,DATA
007104                                         LET @RCSR := @RCSR CLR.BY #BIT06
007104  042777  000100  172150                       BIC     #BIT06,@RCSR
007112                                         LET ERRCNT := ERRCNT + #1
007112  005237  006772                              INC     ERRCNT
007116                                      ELSE
007116  000137  007162                              JMP     $50153
007122                                              $50152:
007122                                         LET DATA := DATA + #1
007122  005237  007042                              INC     DATA
007126                                            IF DATA EQ NUMBER THEN
007126  023737  007042  006774                       CMP     DATA,NUMBER
```

```
2638  007134  001402                                                                  BEQ     .+6
2639  007136  000137  007154                                                          JMP     $50154
2640  007142                                                                    LET @RCSR := @RCSR CLR.BY #BIT06
2641  007142  042777  000100  172112                                                  BIC     #BIT06,@RCSR
2642  007150                                                                  ELSE
2643  007150  000137  007162                                                          JMP     $50155
2644  007154                                                                          $50154:
2645                                                                      ;ALLOW NEXT XMIT INTERR
2646  007154                                                              LET @TCSR := @TCSR SET.BY #BIT06
2647  007154  052777  000100  172104                                                  BIS     #BIT06,@TCSR
2648  007162                                                                  ENDIF
2649  007162                                                                          $50155:
2650  007162                                                          ENDIF
2651  007162                                                                          $50153:
2652  007162                                                      ENDSRV
2653  007162  000002                  RTI
2654
2655  007164  000000                  RHLD:   0
2656
2657
2658                                   ;;*****************************************************************
2659                                   ;*TEST 17       TEST BREAK DETECTION LOGIC
2660                                   ;*             TRANSMIT KNOWN CHAR WITH BREAK SET
2661                                   ;*             AND COMPARE RECEIVED WITH 0.
2662                                   ;*             FRAMING ERROR WILL ALSO BE CHECKED
2663                                   ;*             IF ERROR BITS ARE ENABLED.
2664                                   ;;*****************************************************************
2665  007166  000004                  TST17:  SCOPE
2666  007170  012737  000017  001200          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2667                                                                  ;DONT DO THIS TEST IF 'BREAK' GENERATION
2668                                                                  ;ENABLED, ELSE WILL HALT TO CONSOL ODT.
2669                                                                  ;DO IF BREAK 'DETECTION' IS ENABLED.
2670  007176                                                          IF #BIT04 NOTSETIN $USWR OR #BIT03 NOTSETIN $USWR THEN
2671  007176  032737  000020  001220                                                  BIT     #BIT04,$USWR
2672  007204  001406                                                                 BEQ     $50156
2673  007206  032737  000010  001220                                                  BIT     #BIT03,$USWR
2674  007214  001402                                                                 BEQ     .+6
2675  007216  000137  007232                                                          JMP     $50157
2676  007222                                                                          $50156:
2677  007222  000137  002000          JMP     LOOP                            ;EXIT TEST
2678  007226                                                          ELSE
2679  007226  000137  007252                                                          JMP     $50160
2680  007232                                                                          $50157:
2681  007232                                                                  IF CONSOL EQ #BIT00 THEN
2682  007232  023727  001324  000001                                                  CMP     CONSOL,#BIT00
2683  007240  001402                                                                 BEQ     .+6
2684  007242  000137  007252                                                          JMP     $50161
2685                                                                        ;CAN'T TEST CONSOL
2686  007246  000137  002000          JMP     LOOP                          ;EXIT TEST
2687  007252                                                                  ENDIF
2688  007252                                                                          $50161:
2689  007252                                                          ENDIF
2690  007252                                                                          $50160:
2691  007252                                                          BGNSUB
2692  007252  012737  007260  001110                                                  MOV     #64$,$LPERR
2693
```

```
2694  007260
2695  007260  005037  010100
2696
2697  007264
2698  007264  052777  000001  171774
2699
2700  007272
2701  007272  012777  000125  171770
2702
2703  007300
2704  007300  010546
2705  007302  010605
2706  007304  162706  000010
2707  007310  010546
2708  007312  012745  177777
2709  007316  013745  001262
2710  007322  012745  000200
2711  007326  012745  000500
2712  007332  004737  016272
2713  007336  012605
2714  007340  010506
2715  007342  012605
2716  007344
2717  007344  103402
2718  007346  000137  007354
2719
2720  007352
2721  007352  104115                  ERROR   115
2722  007354
2723  007354
2724  007354
2725  007354  017700  171704
2726  007360
2727  007360  105700
2728  007362  001002
2729  007364  000137  007376
2730
2731  007370
2732  007370  052737  000001  010100
2733  007376
2734  007376
2735  007376
2736  007376  032700  020000
2737  007402  001402
2738  007404  000137  007416
2739  007410
2740  007410  052737  000002  010100
2741  007416
2742  007416
2743  007416
2744  007416  032737  004000  001220
2745  007424  001002
2746  007426  000137  007512
2747
2748  007432
2749  007432  032737  000004  001220
```

```
LET ERRCHK := #0          ; CLEAR ERROR WORD
                    CLR     ERRCHK
          ;SET BREAK BIT
 LET @TCSR := @TCSR SET.BY #BIT00
                    BIS     #BIT00,@TCSR
          ;NON-ZERO CHAR. '1'
LET @TBUF := #125
                    MOV     #125,@TBUF
          ;WAIT FOR DONE
CALL TIMER IN <#500,#BIT07,RCSR,#-1>
                    MOV     R5,-(SP)
                    MOV     SP,R5
                    SUB     #10,SP
                    MOV     R5,-(SP)
                    MOV     #-1,-(R5)
                    MOV     RCSR,-(R5)
                    MOV     #BIT07,-(R5)
                    MOV     #500,-(R5)
                    JSR     PC,TIMER
                    MOV     (SP)+,R5
                    MOV     R5,SP
                    MOV     (SP)+,R5
IF.ERROR THEN
                    BCS     .+6
                    JMP     $50162
    ; RECIEVER DONE DID NOT SET
    ERRHRD 115

ENDIF
                    $50162:
LET R0 := @RBUF
                    MOV     @RBUF,R0
IFB R0 NE #0 THEN
                    TSTB    R0
                    BNE     .+6
                    JMP     $50163
        ; BREAK DID NOT EQUAL 0
    LET ERRCHK := ERRCHK SET.BY #BIT0
                    BIS     #BIT0,ERRCHK
ENDIF
                    $50163:
IF #BIT13 NOTSETIN R0 THEN
                    BIT     #BIT13,R0
                    BEQ     .+6
                    JMP     $50164
          LET ERRCHK := ERRCHK SET.BY #BIT1
                    BIS     #BIT1,ERRCHK
ENDIF
                    $50164:
IF #BIT11 SETIN $USWR THEN
                    BIT     #BIT11,$USWR
                    BNE     .+6
                    JMP     $50165
               ;ODD PARITY ENABLED
    IF #BIT02 NOTSETIN $USWR THEN
                    BIT     #BIT02,$USWR
```

```
2750  007440  001402                                                                    BEQ     .+6
2751  007442  000137  007472                                                            JMP     $50166
2752                                                                                ;BREAK SHOULD GENERATE A PARITY ERRO
2753  007446                                                                      IF #BIT12 NOTSETIN R0 THEN
2754  007446  032700  010000                                                            BIT     #BIT12,R0
2755  007452  001402                                                                    BEQ     .+6
2756  007454  000137  007466                                                            JMP     $50167
2757                                                                                  ;NO PAR ERROR WHEN THERE SHOULD
2758  007460                                                                       LET ERRCHK := ERRCHK SET.BY #BIT2
2759  007460  052737  000004  010100                                                    BIS     #BIT2,ERRCHK
2760  007466                                                                      ENDIF
2761  007466                                                                            $50167:
2762  007466                                                                    ELSE
2763  007466  000137  007512                                                            JMP     $50170
2764  007472                                                                            $50166:
2765  007472                                                                      IF #BIT12 SETIN R0 THEN
2766  007472  032700  010000                                                            BIT     #BIT12,R0
2767  007476  001002                                                                    BNE     .+6
2768  007500  000137  007512                                                            JMP     $50171
2769  007504                                                                       LET ERRCNT := ERRCNT SET.BY #BIT3
2770  007504  052737  000010  006772                                                    BIS     #BIT3,ERRCNT
2771  007512                                                                      ENDIF
2772  007512                                                                            $50171:
2773  007512                                                                    ENDIF
2774  007512                                                                            $50170:
2775  007512                                                                  ENDIF
2776  007512                                                                            $50165:
2777
2778  007512                          IF #BIT01 NOTSETIN $ENV OR $PASS LT #1 THEN
2779  007512  032737  000002  001214                                                    BIT     #BIT01,$ENV
2780  007520  001406                                                                    BEQ     $50172
2781  007522  023727  001202  000001                                                    CMP     $PASS,#1
2782  007530  002402                                                                    BLT     .+6
2783  007532  000137  007552                                                            JMP     $50173
2784  007536                                                                            $50172:
2785  007536                                                                    BRESET  ;CLEAN UP
2786  007536  000005                  RESET
2787  007540  032777  170000  171516  BIT     #170000,@RBUF
2788  007546  001401                  BEQ     ESR1
2789  007550  104033                  ERROR   33              ;RESET DID NOT CLEAR ERROR,FR ERR,OR PERR IN RBUF
2790  007552                          ENDIF
2791  007552                                                                            $50173:
2792  007552                          ESR1:
2793
2794  007552                                                                      IF #BIT0 SETIN ERRCHK THEN
2795  007552  032737  000001  010100                                                    BIT     #BIT0,ERRCHK
2796  007560  001002                                                                    BNE     .+6
2797  007562  000137  007570                                                            JMP     $50174
2798  007566                                                                        ERRHRD 121  ;BREAK ERROR
2799  007566  104121                  ERROR   121
2800  007570                                                                    ENDIF
2801  007570                                                                          $50174:
2802  007570                                                                    IF #BIT1 SETIN ERRCHK THEN
2803  007570  032737  000002  010100                                                    BIT     #BIT1,ERRCHK
2804  007576  001002                                                                    BNE     .+6
2805  007600  000137  007606                                                            JMP     $50175
```

```
2806  007604
2807  007604  104122                          ERROR   122
2808  007606
2809  007606
2810  007606
2811  007606  032737  000004  010100
2812  007614  001002
2813  007616  000137  007624
2814  007622
2815  007622  104235                          ERROR   235
2816
2817
2818  007624
2819  007624
2820  007624
2821  007624  032737  000010  010100
2822  007632  001002
2823  007634  000137  007642
2824  007640
2825  007640  104236                          ERROR   236
2826
2827
2828
2829  007642
2830  007642
2831  007642                                                  ENDSUB
2832  007642                                                  BGNSUB
2833  007642  012737  007650  001110
2834
2835  007650
2836  007650  052777  000001  171410
2837
2838  007656
2839  007656  010546
2840  007660  010605
2841  007662  162706  000010
2842  007666  010546
2843  007670  012745  177777
2844  007674  013745  001266
2845  007700  012745  000001
2846  007704  012745  000500
2847  007710  004737  016272
2848  007714  012605
2849  007716  010506
2850  007720  012605
2851
2852
2853  007722
2854  007722  103402
2855  007724  000137  007732
2856
2857  007730
2858  007730  104021                  ERROR   21
2859  007732
2860  007732
2861
```

```
        ERRHRD 122  ; FRAMING ERROR
ENDIF
                    $50175:
IF #BIT2 SETIN ERRCHK THEN
                    BIT     #BIT2,ERRCHK
                    BNE     .+6
                    JMP     $50176
        ERRHRD  235
            ;NO PARITY ERROR WHEN
            ;THERE SHOULD BE
ENDIF
                    $50176:
IF #BIT3 SETIN ERRCHK THEN
                    BIT     #BIT3,ERRCHK
                    BNE     .+6
                    JMP     $50177
        ERRHRD 236
            ;PARITY ERROR SHOULD NOT HAVE
            ;OCCURED WITH EVEN PARITY
            ;ENABLED AND BREAK SET
ENDIF
                    $50177:

                    MOV     #64$,$LPERR
            ;SET BREAK BIT
LET @TCSR := @TCSR SET.BY #BIT00
                    BIS     #BIT00,@TCSR

CALL TIMER IN <#500,#BIT00,TCSR,#-1>
                    MOV     R5,-(SP)
                    MOV     SP,R5
                    SUB     #10,SP
                    MOV     R5,-(SP)
                    MOV     #-1,-(R5)
                    MOV     TCSR,-(R5)
                    MOV     #BIT00,-(R5)
                    MOV     #500,-(R5)
                    JSR     PC,TIMER
                    MOV     (SP)+,R5
                    MOV     R5,SP
                    MOV     (SP)+,R5

            ;DID IT SET?
IF.ERROR THEN
                    BCS     .+6
                    JMP     $50200
            ;BREAK DID NOT SET
            ERRHRD 21
ENDIF
                    $50200:
```

```
2862
2863  007732
2864  007732  042777  000001  171326          ;CLEAR  BREAK BIT
2865  007740                          LET @TCSR := @TCSR CLR.BY #BIT00
2866  007740  010546                          BIC     #BIT00,@TCSR
2867  007742  010605                  WAITMS 100.
2868  007744  162706  000002                  MOV     R5,-(SP)
2869  007750  010546                          MOV     SP,R5
2870  007752  012745  000144                  SUB     #2,SP
2871  007756  004737  016524                  MOV     R5,-(SP)
2872  007762  012605                          MOV     #100.,-(R5)
2873  007764  010506                          JSR     PC,WAIT
2874  007766  012605                          MOV     (SP)+,R5
2875                                          MOV     R5,SP
2876                                          MOV     (SP)+,R5
2877
2878  007770                          ;READ RBUF TO CLEAR ERRORS & REC DONE
2879  007770  017700  171270          LET R0 := @RBUF
2880                                          MOV     @RBUF,R0
2881  007774                          ;SEND CHAR
2882  007774  012777  000125  171266  LET @TBUF := #125
2883                                          MOV     #125,@TBUF
2884  010002                          ;WAIT FOR DONE BIT
2885  010002  010546                  CALL  TIMER IN <#500,#BIT07,RCSR,#-1>
2886  010004  010605                          MOV     R5,-(SP)
2887  010006  162706  000010                  MOV     SP,R5
2888  010012  010546                          SUB     #10,SP
2889  010014  012745  177777                  MOV     R5,-(SP)
2890  0 0020  013745  001262                  MOV     #-1,-(R5)
2891  010024  012745  000200                  MOV     RCSR,-(R5)
2892  010030  012745  000500                  MOV     #BIT07,-(R5)
2893  010034  004737  016272                  MOV     #500,-(R5)
2894  010040  012605                          JSR     PC,TIMER
2895  010042  010506                          MOV     (SP)+,R5
2896  010044  012605                          MOV     R5,SP
2897  010046                                  MOV     (SP)+,R5
2898  010046  103402                  IF.ERROR THEN
2899  010050  000137  010056                  BCS     .+6
2900                                          JMP     $50201
2901  010054                          ;RECEIVER NEVER CAME READY
2902  010054  104230          ERROR   230     ERRHRD 230
2903  010056                          ENDIF
2904  010056                                  $50201:
2905
2906                                  ;WAS CHAR AFTER BREAK RECEIVED
2907  010056                          IFB @RBUF NE #125 THEN
2908  010056  127727  171202  000125          CMPB    @RBUF,#125
2909  010064  001002                          BNE     .+6
2910  010066  000137  010074                  JMP     $50202
2911                                  ;CHAR AFTER BREAK NOT RECEIVED CORRECTLY
2912  010072                          ERRHRD 231
2913  010072  104231          ERROR   231
2914  010074                          ENDIF
2915  010074                                  $50202:
2916  010074                  ENDSUB
2917  010074                  INLINE <JMP     LOOP>
```

```
2918  010074  000137  002000                                                    JMP     LOOP
2919  010100  000000                ERRCHK: .WORD 0
2920
2921  010102                        TEST20:
2922                                ;;*****************************************************************
2923                                ;*TEST 20       NOT A TEST -
2924                                ;;*****************************************************************
2925  010102  000004                TST20:  SCOPE
2926                                                        ;DLVEC IS 340, DLADD IS 176540
2927                                                        ;OR
2928                                                        ;DLVEC IS 400, DLADD IS 176600
2929
2930
2931  010104                        LET $TESTN := #20
2932  010104  012737  000020  001200                                           MOV     #20,$TESTN
2933
2934  010112                        IF SECSLU EQ #176500 THEN
2935  010112  023727  001302  176500                                           CMP     SECSLU,#176500
2936  010120  001402                                                           BEQ     .+6
2937  010122  000137  010146                                                  JMP     $50203
2938  010126
2939  010126  162737  000040  001260            LET DLVEC := DLVEC - #40       SUB     #40,DLVEC
2940  010134
2941  010134  162737  000040  001256            LET DLADD := DLADD - #40       SUB     #40,DLADD
2942  010142
2943  010142  000137  010162        ELSE                    ;EITHER ZERO OR 176540
                                                                               JMP     $50204
2944  010146                                                                   $50203:
2945  010146
2946  010146  013737  001244  001260            LET DLVEC := $VECT1  ;300      MOV     $VECT1,DLVEC
2947  010154
2948  010154  013737  001250  001256            LET DLADD := $BASE   ;176500   MOV     $BASE,DLADD
2949  010162                        ENDIF
2950  010162                                                                   $50204:
2951
2952  010162                        MODTST:
2953                                ;;*****************************************************************
2954                                ;*TEST 21       TEST THAT CHANNELS INTR AT ASSIGNED PRIORITY
2955                                ;*             INTERRUPTS WILL BE ENABLED ON ALL ACTIVE CHANNELS.
2956                                ;*             RECEIVER AND TRANSMITTER. THEN WE'LL CHECK TO
2957                                ;*             SEE IF THEY INTERRUPTED IN THE ASSIGNED SEQUENCE.
2958                                ;*             THIS TEST WILL BE BYPASSED IF THERE ARE NO DLV11-J'S.
2959                                ;;*****************************************************************
2960  010162  000004                TST21:  SCOPE
2961  010164  012737  000021  001200        MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
2962
2963
2964                                                        ;CLEAR OUT INTERRUPT TABLE
2965  010172                                                LET R0 := #INTRTABLE
2966  010172  012700  011452                                                   MOV     #INTRTABLE,R0
2967  010176                                                REPEAT
2968  010176                                                                   $50205:
2969  010176                                                    LET (R0)+ := #0
2970  010176  005020                                                           CLR     (R0)+
2971  010200                                                UNTIL R0 EQ #TABEND
2972  010200  020027  011512                                                   CMP     R0,#TABEND
2973  010204  001374                                                           BNE     $50205
```

```
2974
2975
2976  010206                                    SETPRI #PR7
2977
2978
2979  010220
2980  010220  013700  001260
2981  010224
2982  010224  012701  010612
2983  010230
2984  010230
2985  010230
2986  010230  010120
2987  010232
2988  010232  012720  000340
2989  010236
2990  010236  062701  000030
2991  010242
2992  010242  020127  011412
2993  010246  001370
2994
2995
2996
2997
2998
2999  010250
3000  010250  013737  001252  011512
3001  010256
3002  010256  005037  011514
3003  010262
3004  010262  013700  001250
3005  010266
3006  010266
3007  010266
3008  010266  032737  000001  011512
3009  010274  001002
3010  010276  000137  010406
3011
3012  010302
3013  010302  052710  000100
3014  010306
3015  010306  062700  000004
3016
3017  010312
3018  010312  052710  000100
3019
3020  010316
3021  010316  012760  000252  000002
3022
3023  010324
3024  010324  162700  000004
3025
3026  010330
3027  010330  010546
3028  010332  010605
3029  010334  162706  000010
```

```
        ;SET PRIORITY TO 7

        ;SET UP ALL INTERRUPT VECTORS
LET R0 := DLVEC
                                  MOV     DLVEC,R0
LET R1 := #RCVR0SRV
                                  MOV     #RCVR0SRV,R1
REPEAT
                                  $50206:
    LET (R0)+ := R1
                                  MOV     R1,(R0)+
    LET (R0)+ := #PR7
                                  MOV     #PR7,(R0)+
    LET R1 := R1 + #30
                                  ADD     #30,R1
UNTIL R1 EQ #SRVEND
                                  CMP     R1,#SRVEND
                                  BNE     $50206

        ;ENABLE INTERRUPTS ON ALL ACTIVE LINES,
        ;ALSO, XMIT CHAR'S TO PRIME RECEIVERS

    ;COPY DEVICES
LET CHMASK := $DEVM
                                  MOV     $DEVM,CHMASK
LET CHCNT := #0
                                  CLR     CHCNT
LET R0 := $BASE
                                  MOV     $BASE,R0
REPEAT
                                  $50207:
    IF #BIT0 SETIN CHMASK THEN
                                  BIT     #BIT0,CHMASK
                                  BNE     .+6
                                  JMP     $50210
        ;SET RCSR IE
        LET (R0) := (R0) SET.BY #BIT6
                                  BIS     #BIT6,(R0)
        LET R0 := R0 + #4
                                  ADD     #4,R0
        ;SET XCSR IE
        LET (R0) := (R0) SET.BY #BIT6
                                  BIS     #BIT6,(R0)
        ;LOAD XBUF
        LET 2(R0) := #252
                                  MOV     #252,2(R0)
        ;GO BACK TO RCSR
        LET R0 := R0 - #4
                                  SUB     #4,R0
        ;LOOK FOR RCVR DONE
        CALL TIMER IN <#500,#BIT07,R0,#-1>
                                  MOV     R5,-(SP)
                                  MOV     SP,R5
                                  SUB     #10,SP
```

```
3030  010340  010546                                                          MOV     R5,-(SP)
3031  010342  012745  177777                                                  MOV     #-1,-(R5)
3032  010346  010045                                                          MOV     R0,-(R5)
3033  010350  012745  000200                                                  MOV     #BIT07,-(R5)
3034  010354  012745  000500                                                  MOV     #500,-(R5)
3035  010360  004737  016272                                                  JSR     PC,TIMER
3036  010364  012605                                                          MOV     (SP)+,R5
3037  010366  010506                                                          MOV     R5,SP
3038  010370  012605                                                          MOV     (SP)+,R5
3039                                                              ;DID IT SET IN TIME?
3040  010372                                                    IF.ERROR THEN
3041  010372  103402                                                          BCS     .+6
3042  010374  000137  010402                                                  JMP     $50211
3043                                                            ;RCVRDONE DID NOT SET IN RCSR
3044  010400                                                    ERRHRD 23
3045  010400  104023                          ERROR   23
3046  010402                                                    ENDIF
3047  010402                                                                  $50211:
3048
3049  010402                                                    LET CHCNT := CHCNT + #1
3050  010402  005237  011514                                                  INC     CHCNT
3051  010406                                                  ENDIF
3052  010406                                                                  $50210:
3053  010406                                                    LET R0 := R0 + #10
3054  010406  062700  000010                                                  ADD     #10,R0
3055                                                          ;SETUP FOR NEXT CHANNEL
3056  010412                                                  LET CHMASK := CHMASK SHIFT -1
3057  010412  006237  011512                                                  ASR     CHMASK
3058  010416                  IF #BIT00 SETIN KDDLV THEN
3059  010416  032737  000001  001306                                          BIT     #BIT00,KDDLV
3060  010424  001002                                                          BNE     .+6
3061  010426  000137  010440                                                  JMP     $50212
3062  010432                      LET .+12 := #4                      ;TEST 4 MODULES
3063  010432  012737  000004  010444                                          MOV     #4,..+12
3064  010440                  ENDIF
3065  010440                                                                  $50212:
3066  010440                                                  UNTIL CHCNT EQ #10
3067  010440  023727  011514  000010                                          CMP     CHCNT,#10
3068  010446  001307                                                          BNE     $50207
3069
3070                                                  ;ALL XMIT & REC INTERRUPTS SHOULD BE
3071                                                  ;IN CONTENTION.
3072                                                  ;RECEIVE INTERRUPTS HAVE PRIORITY
3073                                                  ;OVER XMIT INTERRUPTS.
3074                                                  ;LOW CHANNELS HAVE PRIORITY OVER HIGH CHANNELS.
3075                                                  ;THEREFORE, ONCE CPU PRIORITY IS LOWERED,
3076                                                  ;INTERRUPTS S/B IN THE ORDER SHOWN IN THE
3077                                                  ;CHANNEL IDENTIFIER TABLE (RCH0:)
3078
3079                                                                  ;SET UP POINTER FOR SERVICE ROUTINE
3080  010450                                                  LET R4 := #INTRTABLE
3081  010450  012704  011452                                                  MOV     #INTRTABLE,R4
3082  010454                  LET RCSR := DLADD                               ;GET BACK TO ORG ADRS
3083  010454  013737  001256  001262                                          MOV     DLADD,RCSR
3084
3085                                                                  ;LET EM GO
```

```
3086  010462                                SETPRI #PR0
3087
3088                                                        ;DISABLE ALL INTERRUPTS
3089  010474                                                BRESET
3090  010474  000005                       RESET
3091
3092  010476                                                LET TCNT := #0
3093  010476  005037  011520                                                CLR     TCNT
3094                                                        ;CLEAR ERR FLAG
3095  010502                                                LET PRIERR := #0
3096  010502  005037  011516                                                CLR     PRIERR
3097  010506                                                LET R0 := #INTRTABLE
3098  010506  012700  011452                                                MOV     #INTRTABLE,R0
3099                                                        ;SETUP EXPECTED VALUE
3100  010512                                                LET R1 := #INTRTABLE+2
3101  010512  012701  011454                                                MOV     #INTRTABLE+2,R1
3102  010516  013737  001252  011512       MOV     $DEVM,CHMASK    ;GET # OF CHANNELS IN TCNT
3103
3104  010524  006237  011512       1$:     ASR     CHMASK
3105  010530  103002               BCC     2$
3106  010532  005237  011520               INC     TCNT            ;GET TOTAL CHANNEL CT
3107  010536  005737  011512       2$:     TST     CHMASK
3108  010542  001370               BNE     1$
3109
3110  010544  005337  011520       4$:     DEC     TCNT
3111  010550  001410               BEQ     5$
3112  010552  023727  011520  177777       CMP     TCNT,#-1        ;ALL 1'S ?
3113  010560  001404               BEQ     5$              ;BR IF YES
3114  010562  022120               CMP     (R1)+,(R0)+     ;R1-R0
3115  010564  003367               BGT     4$              ;BR IF R1>R0
3116  010566  005237  011516               INC     PRIERR          ;ELSE ERROR
3117
3118  010572                        5$:
3119  010572                                                IF PRIERR NE #0 THEN
3120  010572  005737  011516                                                TST     PRIERR
3121  010576  001002                                                        BNE     .+6
3122  010600  000137  010606                                                JMP     $50213
3123  010604                                                    ERRHRD 250
3124  010604  104250                       ERROR   250
3125                                                        ;CHANNELS DID NOT INTR ACCORDING TO
3126                                                        ;ASSIGNED PRIORITY. THE BYTE ENTRIES
3127                                                        ;IN INTRTABLE: SHOULD BE IN THE ORDER
3128                                                        ;THAT THEY APPEAR IN THE CHANNEL #
3129                                                        ;TABLE .(EXCLUDING CHANNELS NOT ACTIVE)
3130  010606                                                ENDIF
3131  010606                                                                  $50213:
3132  010606                        INLINE <JMP     TST22>
3133  010606  000137  011524                                                JMP     TST22
3134
3135                                ;;*****************************************************************
3136                                ;*               START OF SERVICE ROUTINES
3137                                ;;*****************************************************************
3138
3139  010612                                                BGNSRV RCVR0SRV
3140  010612                        RCVR0SRV:
3141                                                                ;PUT IN R IDENTIFIER IN INTRTABLE
```

```
3142  010612
3143  010612  013724  011412                                                    LET (R4)+ := RCH0
                                                                                        MOV     RCH0,(R4)+
3144                                                                            ;GENERATE CSR ADDRESS
3145  010616                                                                    LET TEMP := RCSR
3146  010616  013737  001262  011522                                                MOV     RCSR,TEMP
3147  010624                                                                    LET TEMP := TEMP + #0
3148  010624  062737  000000  011522                                                ADD     #0,TEMP
3149                                                                            ;ONE INTR IS ALL WE WANT FROM HERE
3150  010632                                                                    LET @TEMP := @TEMP CLR.BY #BIT6
3151  010632  042777  000100  000662                                                BIC     #BIT6,@TEMP
3152  010640                                                                  ENDSRV
3153  010640  000002                      RTI
3154
3155
3156  010642                                                                  BGNSRV  XMIT0SRV
3157  010642                            XMIT0SRV:
3158                                                                            ;PUT INTR IDENTIFIER IN INTRTABLE
3159  010642                                                                    LET (R4)+ := TCH0
3160  010642  013724  011432                                                        MOV     TCH0,(R4)+
3161                                                                            ;GENERATE CSR ADDRESS
3162  010646                                                                    LET TEMP := RCSR
3163  010646  013737  001262  011522                                                MOV     RCSR,TEMP
3164  010654                                                                    LET TEMP := TEMP + #4
3165  010654  062737  000004  011522                                                ADD     #4,TEMP
3166                                                                            ;ONE INTR IS ALL WE WANT FROM HERE
3167  010662                                                                    LET @TEMP := @TEMP CLR.BY #BIT6
3168  010662  042777  000100  000632                                                BIC     #BIT6,@TEMP
3169  010670                                                                  ENDSRV
3170  010670  000002                      RTI
3171
3172  010672                                                                  BGNSRV  RCVR1SRV
3173  010672                            RCVR1SRV:
3174                                                                            ;PUT INTR IDENTIFIER IN INTRTABLE
3175  010672                                                                    LET (R4)+ := RCH1
3176  010672  013724  011414                                                        MOV     RCH1,(R4)+
3177                                                                            ;GENERATE CSR ADDRESS
3178  010676                                                                    LET TEMP := RCSR
3179  010676  013737  001262  011522                                                MOV     RCSR,TEMP
3180  010704                                                                    LET TEMP := TEMP + #10
3181  010704  062737  000010  011522                                                ADD     #10,TEMP
3182                                                                            ;ONE INTR IS ALL WE WANT FROM HERE
3183  010712                                                                    LET @TEMP := @TEMP CLR.BY #BIT6
3184  010712  042777  000100  000602                                                BIC     #BIT6,@TEMP
3185  010720                                                                  ENDSRV
3186  010720  000002                      RTI
3187
3188  010722                                                                  BGNSRV  XMIT1SRV
3189  010722                            XMIT1SRV:
3190                                                                            ;PUT INTR IDENTIFIER IN INTRTABLE
3191  010722                                                                    LET (R4)+ := TCH1
3192  010722  013724  011434                                                        MOV     TCH1,(R4)+
3193                                                                            ;GENERATE CSR ADDRESS
3194  010726                                                                    LET TEMP := RCSR
3195  010726  013737  001262  011522                                                MOV     RCSR,TEMP
3196  010734                                                                    LET TEMP := TEMP + #14
3197  010734  062737  000014  011522                                                ADD     #14,TEMP
```

```
3198
3199  010742
3200  010742  042777  000100  000552                    ;ONE INTR IS ALL WE WANT FROM HERE
                                                      LET @TEMP := @TEMP CLR.BY #B:T6
                                                                   BIC     #BIT6,@TEMP
                                                    ENDSRV
3201  010750
3202  010750  000002                        RTI
3203
3204  010752                                          BGNSRV  RCVR2SRV
3205  010752                          RCVR2SRV:
3206                                                      ;PUT INTR IDENTIFIER IN INTRTABLE
3207  010752                                            LET (R4)+ := RCH2
3208  010752  013724  011416                                    MOV     RCH2,(R4)+
3209                                                      ;GENERATE CSR ADDRESS
3210  010756                                            LET TEMP := RCSR
3211  010756  013737  001262  011522                            MOV     RCSR,TEMP
3212  010764                                            LET TEMP := TEMP + #20
3213  010764  062737  000020  011522                            ADD     #20,TEMP
3214                                                      ;ONE INTR IS ALL WE WANT FROM HERE
3215  010772                                            LET @TEMP := @TEMP CLR.BY #BIT6
3216  010772  042777  000100  000522                            BIC     #BIT6,@TEMP
3217  011000                                            ENDSRV
3218  011000  000002                        RTI
3219
3220  011002                                          BGNSRV  XMIT2SRV
3221  011002                          XMIT2SRV:
3222                                                      ;PUT INTR IDENTIFIER IN INTRTABLE
3223  011002                                            LET (R4)+ := TCH2
3224  011002  013724  011436                                    MOV     TCH2,(R4)+
3225                                                      ;GENERATE CSR ADDRESS
3226  011006                                            LET TEMP := RCSR
3227  011006  013737  001262  011522                            MOV     RCSR,TEMP
3228  011014                                            LET TEMP := TEMP + #24
3229  011014  062737  000024  011522                            ADD     #24,TEMP
3230                                                      ;ONE INTR IS ALL WE WANT FROM HERE
3231  011022                                            LET @TEMP := @TEMP CLR.BY #BIT6
3232  011022  042777  000100  000472                            BIC     #BIT6,@TEMP
3233  011030                                            ENDSRV
3234  011030  000002                        RTI
3235
3236  011032                                          BGNSRV RCVR3SRV
3237  011032                          RCVR3SRV:
3238                                                      ;PUT INTR IDENTIFIER IN INTRTABLE
3239  011032                                            LET (R4)+ := RCH3
3240  011032  013724  011420                                    MOV     RCH3,(R4)+
3241                                                      ;GENERATE CSR ADDRESS
3242  011036                                            LET TEMP := RCSR
3243  011036  013737  001262  011522                            MOV     RCSR,TEMP
3244  011044                                            LET TEMP := TEMP + #30
3245  011044  062737  000030  011522                            ADD     #30,TEMP
3246                                                      ;ONE INTR IS ALL WE WANT FROM HERE
3247  011052                                            LET @TEMP := @TEMP CLR.BY #BIT6
3248  011052  042777  000100  000442                            BIC     #BIT6,@TEMP
3249  011060                                            ENDSRV
3250  011060  000002                        RTI
3251
3252  011062                                          BGNSRV  XMIT3SRV
3253  011062                          XMIT3SRV:
```

```
3254
3255  011062
3256  011062  013724  011440
3257
3258  011066
3259  011066  013737  001262  011522
3260  011074
3261  011074  062737  000034  011522
3262
3263  011102
3264  011102  042777  000100  000412
3265  011110
3266  011110  000002                        RTI
3267  011112
3268  011112                        RCVR4SRV:
3269
3270  011112
3271  011112  013724  011422
3272
3273  011116
3274  011116  013737  001262  011522
3275  011124
3276  011124  062737  000040  011522
3277
3278  011132
3279  011132  042777  000100  000362
3280  011140
3281  011140  000002                        RTI
3282  011142
3283  011142                        XMIT4SRV:
3284
3285  011142
3286  011142  013724  011442
3287
3288  011146
3289  011146  013737  001262  011522
3290  011154
3291  011154  062737  000044  011522
3292
3293  011162
3294  011162  042777  000100  000332
3295  011170
3296  011170  000002                        RTI
3297  011172
3298  011172                        RCVR5SRV:
3299
3300  011172
3301  011172  013724  011424
3302
3303  011176
3304  011176  013737  001262  011522
3305  011204
3306  011204  062737  000050  011522
3307
3308  011212
3309  011212  042777  000100  000302
```

```
                    ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := TCH3
                                MOV     TCH3,(R4)+
                    ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #34
                                ADD     #34,TEMP
                    ;ONE INTR IS ALL WE WANT FROM HERE
                LET @TEMP := @TEMP CLR.BY #BIT6
                                BIC     #BIT6,@TEMP
            ENDSRV

            BGNSRV  RCVR4SRV

                    ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := RCH4
                                MOV     RCH4,(R4)+
                    ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #40
                                ADD     #40,TEMP
                    ;ONE INTR IS ALL WE WANT FROM HERE
                LET @TEMP := @TEMP CLR.BY #BIT6
                                BIC     #BIT6,@TEMP
            ENDSRV

            BGNSRV  XMIT4SRV

                    ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := TCH4
                                MOV     TCH4,(R4)+
                    ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #44
                                ADD     #44,TEMP
                    ;ONE INTR IS ALL WE WANT FROM HERE
                LET @TEMP := @TEMP CLR.BY #BIT6
                                BIC     #BIT6,@TEMP
            ENDSRV

            BGNSRV  RCVR5SRV

                    ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := RCH5
                                MOV     RCH5,(R4)+
                    ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #50
                                ADD     #50,TEMP
                    ;ONE INTR IS ALL WE WANT FROM HERE
                LET @TEMP := @TEMP CLR.BY #BIT6
                                BIC     #BIT6,@TEMP
```

```
3310  011220
3311  011220  000002                          RTI
3312  011222
3313  011222                            XMIT5SRV:
3314
3315  011222
3316  011222  013724  011444
3317
3318  011226
3319  011226  013737  001262  011522
3320  011234
3321  011234  062737  000054  011522
3322
3323  011242
3324  011242  042777  000100  000252
3325  011250
3326  011250  000002                          RTI
3327  011252
3328  011252                            RCVR6SRV:
3329
3330  011252
3331  011252  013724  011426
3332
3333  011256
3334  011256  013737  001262  011522
3335  011264
3336  011264  062737  000060  011522
3337
3338  011272
3339  011272  042777  000100  000222
3340  011300
3341  011300  000002                          RTI
3342  011302
3343  011302                            XMIT6SRV:
3344
3345  011302
3346  011302  013724  011446
3347
3348  011306
3349  011306  013737  001262  011522
3350  011314
3351  011314  062737  000064  011522
3352
3353  011322
3354  011322  042777  000100  000172
3355  011330
3356  011330  000002                          RTI
3357  011332
3358  011332                            RCVR7SRV:
3359
3360  011332
3361  011332  013724  011430
3362
3363  011336
3364  011336  013737  001262  011522
3365  011344
```

```
        ENDSRV

        BGNSRV  XMIT5SRV

                ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := TCH5
                                MOV     TCH5,(R4)+
                ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #54
                                ADD     #54,TEMP
                ;ONE INTR IS ALL WE WANT FROM HERE
                LET @TEMP := @TEMP CLR.BY #BIT6
                                BIC     #BIT6,@TEMP
        ENDSRV

        BGNSRV  RCVR6SRV

                ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := RCH6
                                MOV     RCH6,(R4)+
                ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #60
                                ADD     #60,TEMP
                ;ONE INTR IS ALL WE WANT FROM HERE
                LET @TEMP := @TEMP CLR.BY #BIT6
                                BIC     #BIT6,@TEMP
        ENDSRV

        BGNSRV  XMIT6SRV

                ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := TCH6
                                MOV     TCH6,(R4)+
                ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #64
                                ADD     #64,TEMP
                ;ONE INTR IS ALL WE WANT FROM HERE
                LET @TEMP := @TEMP CLR.BY #BIT6
                                BIC     #BIT6,@TEMP
        ENDSRV

        BGNSRV  RCVR7SRV

                ;PUT INTR IDENTIFIER IN INTRTABLE
                LET (R4)+ := RCH7
                                MOV     RCH7,(R4)+
                ;GENERATE CSR ADDRESS
                LET TEMP := RCSR
                                MOV     RCSR,TEMP
                LET TEMP := TEMP + #70
```

```
3366  011344  062737  000070  011522                                                  ADD     #70,TEMP
3367                                                                          ;ONE INTR IS ALL WE WANT FROM HERE
3368  011352                                                            LET @TEMP := @TEMP CLR.BY #BIT6
3369  011352  042777  000100  000142                                                  BIC     #BIT6,@TEMP
3370  011360                                                      ENDSRV
3371  011360  000002                      RTI
3372  011362                                                      BGNSRV  XMIT7SRV
3373  011362                            XMIT7SRV:
3374                                                                      ;PUT INTR IDENTIFIER IN INTRTABLE
3375  011362                                                            LET (R4)+ := TCH7
3376  011362  013724  011450                                                          MOV     TCH7,(R4)+
3377                                                                      ;GENERATE CSR ADDRESS
3378  011366                                                            LET TEMP := RCSR
3379  011366  013737  001262  011522                                                  MOV     RCSR,TEMP
3380  011374                                                            LET TEMP := TEMP + #74
3381  011374  062737  000074  011522                                                  ADD     #74,TEMP
3382                                                                          ;ONE INTR IS ALL WE WANT FROM HERE
3383  011402                                                            LET @TEMP := @TEMP CLR.BY #BIT6
3384  011402  042777  000100  000112                                                  BIC     #BIT6,@TEMP
3385  011410                                                      ENDSRV
3386  011410  000002                      RTI
3387  011412                            SRVEND:
3388
3389
3390
3391                                        ;**CHANNEL IDENTIFIER TABLE**
3392                                        ;THE ENTRIES FROM THIS
3393                                        ;TABLE ARE PLACED IN THE INTRTABLE
3394                                        ;IN THE ORDER THAT THE INTR'S
3395                                        ;OCCUR. (EXCLUDING NON-ACTIVE CH'S)
3396  011412  000000                    RCH0:    .WORD 0
3397  011414  000001                    RCH1:    .WORD 1
3398  011416  000002                    RCH2:    .WORD 2
3399  011420  000003                    RCH3:    .WORD 3
3400  011422  000004                    RCH4:    .WORD   4
3401  011424  000005                    RCH5:    .WORD   5
3402  011426  000006                    RCH6:    .WORD   6
3403  011430  000007                    RCH7:    .WORD   7
3404  011432  000010                    TCH0:    .WORD 10
3405  011434  000011                    TCH1:    .WORD 11
3406  011436  000012                    TCH2:    .WORD 12
3407  011440  000013                    TCH3:    .WORD 13
3408  011442  000014                    TCH4:    .WORD 14
3409  011444  000015                    TCH5:    .WORD 15
3410  011446  000016                    TCH6:    .WORD 16
3411  011450  000017                    TCH7:    .WORD 17
3412
3413                                        ;THIS TABLE WILL CONTAIN ENTRIES
3414                                        ;REPRESENTING EACH INTR IN THE ORDER
3415                                        ;THAT IT OCCURED
3416                                        ; S/B IN THE ABOVE ORDER
3417                                        ;
3418  011452  000020                    INTRTABLE:      .BLKW   16.
3419  011512                            TABEND:
3420
3421  011512  000000                    CHMASK: .WORD 0         ; INDICATE WHICH CH'S ARE ACTIVE FOR THIS TEST
```

```
3422  011514  000000                  CHCNT:  .WORD 0          ;CHANNEL COUNTER
3423  011516  000000                  PRIERR: .WORD 0          ;ERROR WORD, CONTAINS NO. OF INTR OUT OF ORDER
3424  011520  000000                  TCNT:   .WORD 0          ;TABLE PASS CTR
3425  011522  000000                  TEMP:   .WORD 0          ;FOR SERVICE ROUTINES
3426
3427
3428                                  ;;*************************************************************
3429                                  ;*TEST 22       TEST DATA TRANSFERS WITH ALL ACTIVE LINES INTERRUPTING
3430                                  ;*             IN THIS WE'LL ENABLE INTERRUPTS ON ALL CHANNELS
3431                                  ;*             FOR THIS DLV11-J. THEN WE'LL XMIT AN INCREMENTING
3432                                  ;*             DATA PATERN VIA INTERRUPTS AND RECORD THE RECEIVER
3433                                  ;*             INTR. IN THE RECEIVER STATUS TABLE.
3434                                  ;*      NOTE:  DOUBLE BUFFERING CANNOT BE TESTED AT ITS MAX SPEED
3435                                  ;*             BECAUSE OF APT CONSIDERATIONS. I.E. APT SENDS
3436                                  ;*             'BREAKS' WHICH CAUSE OVERRUN ERRORS. THEREFORE
3437                                  ;*             THE XMIT IE IS NOT ENABLED AGAIN UNTIL THE RECVR
3438                                  ;*             HAS OBTAINED THE PREVIOUS WORD. THIS TEST WILL
3439                                  ;*             NOT BE EXECUTED IF THERE ARE NO DLV11-J'S.
3440                                  ;;*************************************************************
3441  011524  000004                  TST22:  SCOPE
3442  011526  012737  000022  001200          MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3443                                                                       ;CLR ERROR WORD,
3444  011534                                                           LET ERWRD := #0
3445  011534  005037  016122                                                  CLR     ERWRD
3446
3447
3448                                                                       ;CLEAR OUT RECEIVER TABLES
3449  011540                                                           LET R0 := #CHOTAB
3450  011540  012700  014114                                                  MOV     #CHOTAB,R0
3451  011544                                                           REPEAT
3452  011544                                                                  $50214:
3453  011544                                                               LET (R0)+ := #0
3454  011544  005020                                                          CLR     (R0)+
3455  011546                                                           UNTIL R0 EQ #STATEND
3456  011546  020027  016114                                                  CMP     R0,#STATEND
3457  011552  001374                                                          BNE     $50214
3458
3459
3460  011554                                                               ;SET UP ALL VECTORS
3461  011554  012701  012534                                           LET R1 := #R0SRV
3462  011560                                                                  MOV     #R0SRV,R1
3463  011560  013700  001260                                           LET R0 := DLVEC
3464  011564                                                                  MOV     DLVEC,R0
3465  011564                                                           REPEAT
3466  011564                                                                  $50215:
3467  011564  010120                                                      LET (R0)+ := R1
3468  011566                                                                  MOV     R1,(R0)+
3469  011566  012720  000340                                              LET (R0)+ := #PR7
3470  011572                                                                  MOV     #PR7,(R0)+
3471  011572  062701  000072                                              LET R1 := R1 + #72
3472  011576                                                                  ADD     #72,R1
3473  011576  010120                                                      LET (R0)+ := R1
3474  011600                                                                  MOV     R1,(R0)+
3475  011600  012720  000340                                              LET (R0)+ := #PR7
3476  011604                                                                  MOV     #PR7,(R0)+
3477  011604  062701  000044                                              LET R1 := R1 + #44
                                                                            ADD     #44,R1
```

```
3478  011610
3479  011610  020127  014114                                  UNTIL R1 EQ #SEREND
3480  011614  001363                                                      CMP     R1,#SEREND
3481                                                                      BNE     $50215
3482  011616                                SETPRI #PR7             ;SET PRIORTY TO 7
3483
3484                                                              ;ENABLE INTR'S ON ALL ACTIVE LINES
3485                                                              ;THIS MODULE
3486  011630                                                      LET R0 := $BASE
3487  011630  013700  001250                                                  MOV     $BASE,R0
3488                                                                  ;INITIALIZE
3489  011634                                                      LET CHMSK := $DEVM
3490  011634  013737  001252  016114                                          MOV     $DEVM,CHMSK
3491  011642                                                      LET CHCTR := #0
3492  011642  005037  016116                                                  CLR     CHCTR
3493                                                              ;ACTIVE CHANNEL CTR
3494  011646                                                      LET ACTCH := #0
3495  011646  005037  016120                                                  CLR     ACTCH
3496  011652                                                      REPEAT
3497  011652                                                                  $50216:
3498  011652                                                          IF #BIT0 SETIN CHMSK THEN
3499  011652  032737  000001  016114                                          BIT     #BIT0,CHMSK
3500  011660  001002                                                          BNE     .+6
3501  011662  000137  011716                                                  JMP     $50217
3502                                                                      ;# OF ACTIVE CH
3503  011666                                                              LET ACTCH := ACTCH + #1
3504  011666  005237  016120                                                      INC     ACTCH
3505                                                                      ;SET RCSR IE
3506  011672                                                              LET (R0) := (R0) SET.BY #BIT6
3507  011672  052710  000100                                                      BIS     #BIT6,(R0)
3508  011676                                                              LET R0 := R0 + #4
3509  011676  062700  000004                                                      ADD     #4,R0
3510                                                                      ;SET XCSR IE
3511  011702                                                              LET (R0) := (R0) SET.BY #BIT6
3512  011702  052710  000100                                                      BIS     #BIT6,(R0)
3513  011706                                                              LET R0 := R0 - #4
3514  011706  162700  000004                                                      SUB     #4,R0
3515  011712                                                              LET CHCTR := CHCTR + #1
3516  011712  005237  016116                                                      INC     CHCTR
3517  011716                                                          ENDIF
3518  011716                                                                  $50217:
3519  011716                                                          LET R0 := R0 + #10
3520  011716  062700  000010                                                  ADD     #10,R0
3521  011722                                                      LET CHMSK := CHMSK SHIFT -1
3522  011722  006237  016114                                                  ASR     CHMSK
3523
3524
3525  011726                                IF #BIT00 SETIN KDDLV THEN        ;4 OR 10 MODULES
3526  011726  032737  000001  001306                                          BIT     #BIT00,KDDLV
3527  011734  001002                                                          BNE     .+6
3528  011736  000137  011750                                                  JMP     $50220
3529  011742                                    LET .+12 := #4
3530  011742  012737  000004  011754                                         MOV     #4,.+12
3531  011750                                ENDIF
3532  011750                                                                  $50220:
3533  011750                                                      UNTIL CHCTR EQ #10
```

```
3534  011750  023727  016116  000010                                    CMP     CHCTR,#10
3535  011756  001335                                                    BNE     $50216
3536
3537                                                        ;INIT BUFFER POINTERS FOR SERVICE ROUTINES
3538  011760                                              LET X0 := #CH0TAB
3539  011760  012737  014114  016252                                    MOV     #CH0TAB,X0
3540  011766                                              LET X1 := #CH1TAB
3541  011766  012737  014314  016254                                    MOV     #CH1TAB,X1
3542  011774                                              LET X2 := #CH2TAB
3543  011774  012737  014514  016256                                    MOV     #CH2TAB,X2
3544  012002                                              LET X3 := #CH3TAB
3545  012002  012737  014714  016260                                    MOV     #CH3TAB,X3
3546  012010                                              LET X4 := #CH4TAB
3547  012010  012737  015114  016262                                    MOV     #CH4TAB,X4
3548  012016                                              LET X5 := #CH5TAB
3549  012016  012737  015314  016264                                    MOV     #CH5TAB,X5
3550  012024                                              LET X6 := #CH6TAB
3551  012024  012737  015514  016266                                    MOV     #CH6TAB,X6
3552  012032                                              LET X7 := #CH7TAB
3553  012032  012737  015714  016270                                    MOV     #CH7TAB,X7
3554
3555                                                        ;FILL XBUF WORDS
3556                                                        ;FROM TMP0 THRU TMP7E
3557                                                        ;START WORD CH 0 ALWAYS 0
3558  012040                                              LET TMP0 := #0
3559  012040  005037  016126                                            CLR     TMP0
3560                                                        ;OFFSET TO INDEX
3561  012044                                              LET R4 := #0
3562  012044  005004                                                    CLR     R4
3563                                                        ;PTR TO TABLE TO BE FILLED
3564  012046                                              LET R5 := #TMP0E
3565  012046  012705  016130                                            MOV     #TMP0E,R5
3566  012052                                              REPEAT
3567  012052                                                          $50221:
3568  012052                                                IF #BIT0 NOTSETIN $USWR THEN
3569  012052  032737  000001  001220                                    BIT     #BIT0,$USWR
3570  012060  001402                                                    BEQ     .+6
3571  012062  000137  012102                                            JMP     $50222
3572  012066                                                  LET (R5)+ := TABL7(R4)
3573  012066  016425  016212                                            MOV     TABL7(R4),(R5)+
3574  012072                                                  LET (R5)+ := TABL7(R4)
3575  012072  016425  016212                                            MOV     TABL7(R4),(R5)+
3576  012076                                                ELSE
3577  012076  000137  012112                                            JMP     $50223
3578  012102                                                          $50222:
3579  012102                                                  LET (R5)+ := TABL8(R4)
3580  012102  016425  016232                                            MOV     TABL8(R4),(R5)+
3581  012106                                                  LET (R5)+ := TABL8(R4)
3582  012106  016425  016232                                            MOV     TABL8(R4),(R5)+
3583  012112                                                ENDIF
3584  012112                                                          $50223:
3585  012112                                                  LET R4 := R4 + #2
3586  012112  062704  000002                                            ADD     #2,R4
3587  012116                                      IF #BIT00 SETIN KDDLV THEN    ;4 OR 10 MODULES
3588  012116  032737  000001  001306                                    BIT     #BIT00,KDDLV
3589  012124  001002                                                    BNE     .+6
```

```
3590  012126  000137  012140
3591  012132                        LET .+10 := #10
3592  012132  012737  000010  012142
3593  012140                        ENDIF
3594  012140
3595  012140
3596  012140  020427  000020
3597  012144  001342
3598
3599
3600  012146
3601  012146  013737  016120  016170
3602
3603  012154                        SETPRI #PR0
3604
3605  012166
3606  012166
3607  012166
3608  012166  005737  016170
3609  012172  001375
3610
3611
3612  012174
3613  012174  012737  100000  016170
3614  012202
3615  012202
3616  012202
3617  012202  005337  016170
3618  012206
3619  012206  005737  016170
3620  012212  001373
3621
3622
3623  012214
3624  012214  000005                RESET
3625
3626
3627
3628  012216
3629  012216  005037  016126
3630  012222
3631  012222  013737  016130  016132
3632  012230
3633  012230  013737  016134  016136
3634  012236
3635  012236  013737  016140  016142
3636  012244                        LET TMP4 := TMP3E
3637  012244  013737  016144  016146
3638  012252                        LET TMP5 := TMP4E
3639  012252  013737  016150  016152
3640  012260                        LET TMP6 := TMP5E
3641  012260  013737  016154  016156
3642  012266                        LET TMP7 := TMP6E
3643  012266  013737  016160  016162
3644
3645  012274
```

```
                                    JMP     $50224
                        ;TEST 4 MODULES
                                    MOV     #10,..+10

                                    $50224:
UNTIL R4 EQ #20
                                    CMP     R4,#20
                                    BNE     $50221
;INIT TMP10
;WILL BE DECR BY RECVR SERVICE ROUTINES TILL 0
LET TMP10 := ACTCH
                                    MOV     ACTCH,TMP10
    ;SET PRIORITY TO 0

REPEAT
                                    $50225:
UNTIL TMP10 EQ #0
                                    TST     TMP10
                                    BNE     $50225

;WAIT FOR RECV'S TO CATCH UP
LET TMP10 := #100000
                                    MOV     #100000,TMP10
REPEAT
                                    $50226:
   LET TMP10 := TMP10 - #1
                                    DEC     TMP10
UNTIL TMP10 EQ #0
                                    TST     TMP10
                                    BNE     $50226

    ;DISABLE ALL INTR'S
BRESET

    ;CHECK RECEIVER TABLE
    ;RESET TMP0-7
    LET TMP0 := #0
                                    CLR     TMP0
    LET TMP1 := TMP0E
                                    MOV     TMP0E,TMP1
    LET TMP2 := TMP1E
                                    MOV     TMP1E,TMP2
    LET TMP3 := TMP2E
                                    MOV     TMP2E,TMP3
                                    MOV     TMP3E,TMP4
                                    MOV     TMP4E,TMP5
                                    MOV     TMP5E,TMP6
                                    MOV     TMP6E,TMP7
    ;INITIALIZE
    LET CHCTR := #0
```

```
3646  012274  005037  016116                                                    CLR     CHCTR
3647  012300                                                          LET CHMSK := $DEVM
3648  012300  013737  001252  016114                                            MOV     $DEVM,CHMSK
3649  012306                                                        REPEAT
3650  012306                                                                    $50227:
3651  012306                                                          IF #BIT0 SETIN CHMSK THEN
3652  012306  032737  000001  016114                                            BIT     #BIT0,CHMSK
3653  012314  001002                                                           BNE     .+6
3654  012316  000137  012432                                                   JMP     $50230
3655  012322                                                                LET R0 := CHCTR
3656  012322  013700  016116                                                   MOV     CHCTR,R0
3657                                                                          ;OFFSET TO INDEX (2X)
3658  012326                                                                  LET R0 := R0 SHIFT +1
3659  012326  006300                                                           ASL     R0
3660                                                                          ;GET TABLE LOC FOR THAT CH.
3661  012330                                                                  LET R2 := TABL6(R0)
3662  012330  016002  016172                                                   MOV     TABL6(R0),R2
3663                                                                          ;OFFSET TO INDEX (4X)
3664  012334                                                                  LET R0 := R0 SHIFT +1
3665  012334  006300                                                           ASL     R0
3666                                                                          ;GET EXPECTED STARTING WORD
3667  012336                                                                  LET R1 := TMP0(R0)
3668  012336  016001  016126                                                   MOV     TMP0(R0),R1
3669  012342                                                                  LET R0 := R0 + #2
3670  012342  062700  000002                                                   ADD     #2,R0
3671                                                                          ;GET MAX WORD FOR THAT CH.
3672  012346                                                                  LET R3 := TMP0(R0)
3673  012346  016003  016126                                                   MOV     TMP0(R0),R3
3674
3675  012352                                                        REPEAT
3676  012352                                                                    $50231:
3677  012352                                                            IF #BIT15 SETIN (R2) THEN
3678  012352  032712  100000                                                   BIT     #BIT15,(R2)
3679  012356  001002                                                           BNE     .+6
3680  012360  000137  012372                                                   JMP     $50232
3681  012364                                                                LET ERWRD := ERWRD SET.BY #BIT8
3682  012364  052737  000400  016122                                           BIS     #BIT8,ERWRD
3683  012372                                                            ENDIF
3684  012372                                                                    $50232:
3685                                                                      ;CHECK FOR DATA COMPARE ERROR
3686  012372                                                            IFB (R2) NE R1 THEN
3687  012372  121201                                                           CMPB    (R2),R1
3688  012374  001002                                                           BNE     .+6
3689  012376  000137  012406                                                   JMP     $50233
3690  012402                                                                LET ERWRD := ERWRD + #1
3691  012402  005237  016122                                                   INC     ERWRD
3692  012406                                                            ENDIF
3693  012406                                                                    $50233:
3694                                                                      ;BUMP EXPECTED DATA WORD
3695  012406                                                                  LET R1 := R1 + #1
3696  012406  005201                                                           INC     R1
3697                                                                          ;BUMP TABLE PTR
3698  012410                                                                  LET R2 := R2 + #2
3699  012410  062702  000002                                                   ADD     #2,R2
3700
3701  012414                                                        UNTIL R1 EQ R3 OR ERWRD NE #0
```

```
3702  012414  020103                                                                  CMP     R1,R3
3703  012416  001403                                                                  BEQ     $50234
3704  012420  005737  016122                                                         TST     ERWRD
3705  012424  001752                                                                  BEQ     $50231
3706  012426                                                                        $50234:
3707  012426                                                          LET CHCTR := CHCTR + #1
3708  012426  005237  016116                                                         INC     CHCTR
3709  012432                                                           ENDIF
3710  012432                                                                        $50230:
3711  012432                                                            LET CHMSK := CHMSK SHIFT -1
3712  012432  006237  016114                                                         ASR     CHMSK
3713  012436                        IF #BIT00 SETIN KDDLV THEN
3714  012436  032737  000001  001306                                                 BIT     #BIT00,KDDLV
3715  012444  001002                                                                 BNE     .+6
3716  012446  000137  012460                                                         JMP     $50235
3717  012452                            LET .+12 := #4                      ;TEST 4 MODULES
3718  012452  012737  000004  012464                                                 MOV     #4,..+12
3719  012460                        ENDIF
3720  012460                                                                        $50235:
3721  012460                                                UNTIL CHCTR EQ #10 OR ERWRD NE #0
3722  012460  023727  016116  000010                                                 CMP     CHCTR,#10
3723  012466  001403                                                                 BEQ     $50236
3724  012470  005737  016122                                                         TST     ERWRD
3725  012474  001704                                                                 BEQ     $50227
3726  012476                                                                        $50236:
3727
3728  012476                                                              IF #BIT8 SETIN ERWRD THEN
3729  012476  032737  000400  016122                                                 BIT     #BIT8,ERWRD
3730  012504  001002                                                                 BNE     .+6
3731  012506  000137  012514                                                         JMP     $50237
3732                                                                        ;ERROR FLAG UP AFTER TRANSFER
3733  012512                                                                 ERRHRD  270
3734  012512  104270                ERROR   270
3735  012514                                                              ENDIF
3736  012514                                                                        $50237:
3737  012514                                                              IFB ERWRD NE #0 THEN
3738  012514  105737  016122                                                         TSTB    ERWRD
3739  012520  001002                                                                 BNE     .+6
3740  012522  000137  012530                                                         JMP     $50240
3741                                                                        ;DATA COMPARE ERROR
3742  012526                                                                 ERRHRD  271
3743  012526  104271                ERROR   271
3744  012530                                                              ENDIF
3745  012530                                                                        $50240:
3746  012530  000137  021324        JMP     ENDDER                              ;EXIT TEST
3747
3748                          ;;*****************************************************************
3749                          ;*              START OF SERVICE ROUTINES
3750                          ;;*****************************************************************
3751
3752  012534                                                              BGNSRV ROSRV
3753  012534                  ROSRV:
3754  012534                                                                  LET TMP := RCSR
3755  012534  013737  001262  016124                                                 MOV     RCSR,TMP
3756  012542                                                                  LET TMP := TMP + #2
3757  012542  062737  000002  016124                                                 ADD     #2,TMP
```

```
3758  012550
3759  012550  017777  003350  003474                                        LET @X0 := @TMP
3760  012556                                                                            MOV     @TMP,@X0
3761  012556  062737  000002  016252                                    LET X0 := X0 + #2
3762  012564                                                                            ADD     #2,X0
3763  012564  023737  016126  016130                                        IF TMP0 NE TMP0E THEN
3764  012572  001002                                                                   CMP     TMP0,TMP0E
3765  012574  000137  012620                                                           BNE     .+6
3766                                                                                    JMP     $50241
3767  012600                                                                        ;GO TO XCSR
3768  012600  062737  000002  016124                                            LET TMP := TMP + #2
3769                                                                                    ADD     #2,TMP
3770  012606                                                                        ;ENABLE XMIT INTERR
3771  012606  052777  000100  003310                                            LET @TMP := @TMP SET.BY #BIT6
3772  012614                                                                            BIS     #BIT6,@TMP
3773  012614  000137  012624                                                    ELSE
3774  012620                                                                            JMP     $50242
3775                                                                                    $50241:
3776  012620                                                                        ;ALL DONE
3777  012620  005337  016170                                                    LET TMP10 := TMP10 - #1
3778  012624                                                                            DEC     TMP10
3779  012624                                                                    ENDIF
3780  012624                                                                            $50242:
3781  012624  000002                  RTI                                      ENDSRV
3782
3783  012626                                                                BGNSRV X0SRV
3784  012626                  X0SRV:
3785  012626                                                                    LET TMP := RCSR
3786  012626  013737  001262  016124                                                   MOV     RCSR,TMP
3787  012634                                                                    LET TMP := TMP + #6
3788  012634  062737  000006  016124                                                   ADD     #6,TMP
3789  012642                                                                    LET @TMP := TMP0
3790  012642  013777  016126  003254                                                   MOV     TMP0,@TMP
3791  012650                                                                    LET TMP0 := TMP0 + #1
3792  012650  005237  016126                                                           INC     TMP0
3793                                                                                ;GO BACK TO XCSR
3794  012654                                                                    LET TMP := TMP - #2
3795  012654  162737  000002  016124                                                   SUB     #2,TMP
3796                                                                                ;DISABLE XMIT INTERRUPTS,
3797                                                                                ;NOT EXER DOUBLE BUFFERING
3798  012662                                                                    LET @TMP := @TMP CLR.BY #BIT6
3799  012662  042777  000100  003234                                                   BIC     #BIT6,@TMP
3800  012670                                                                ENDSRV
3801  012670  000002                  RTI
3802
3803  012672                                                                BGNSRV R1SRV
3804  012672                  R1SRV:
3805  012672                                                                    LET TMP := RCSR
3806  012672  013737  001262  016124                                                   MOV     RCSR,TMP
3807  012700                                                                    LET TMP := TMP + #12
3808  012700  062737  000012  016124                                                   ADD     #12,TMP
3809  012706                                                                    LET @X1 := @TMP
3810  012706  017777  003212  003340                                                   MOV     @TMP,@X1
3811  012714                                                                    LET X1 := X1 + #2
3812  012714  062737  000002  016254                                                   ADD     #2,X1
3813  012722                                                                    IF TMP1 NE TMP1E THEN
```

```
3814  012722  023737  016132  016134                                                  CMP     TMP1,TMP1E
3815  012730  001002                                                                  BNE     .+6
3816  012732  000137  012756                                                          JMP     $50243
3817
3818  012736                                                                      ;GO TO XCSR
3819  012736  062737  000002  016124                                              LET TMP := TMP + #2
3820                                                                                ADD     #2,TMP
3821  012744                                                                      ;ENABLE XMIT INTERR
3822  012744  052777  000100  003152                                              LET @TMP .= @TMP SET.BY #BIT6
3823  012752                                                                      BIS     #BIT6,@TMP
3824  012752  000137  012762                                                  ELSE
3825  012756                                                                      JMP     $50244
3826                                                                            $50243:
3827  012756                                                                  ;ALL DONE
3828  012756  005337  016170                                                  LET TMP10 := TMP10 - #1
3829  012762                                                                      DEC     TMP10
3830  012762                                                                  ENDIF
3831  012762                                                                      $50244:
3832  012762  000002                  RTI                                    ENDSRV
3833
3834  012764                                                                  BGNSRV X1SRV
3835  012764                  X1SRV:
3836  012764                                                                  LET TMP := RCSR
3837  012764  013737  001262  016124                                              MOV     RCSR,TMP
3838  012772                                                                  LET TMP := TMP + #16
3839  012772  062737  000016  016124                                              ADD     #16,TMP
3840  013000                                                                  LET @TMP := TMP1
3841  013000  013777  016132  003116                                              MOV     TMP1,@TMP
3842  013006                                                                  LET TMP1 := TMP1 + #1
3843  013006  005237  016132                                                      INC     TMP1
3844                                                                            ;GO BACK TO XCSR
3845  013012                                                                  LET TMP := TMP - #2
3846  013012  162737  000002  016124                                              SUB     #2,TMP
3847                                                                            ;DISABLE XMIT INTERRUPTS,
3848                                                                            ;NOT EXER DOUBLE BUFF
3849  013020                                                                  LET @TMP := @TMP CLR.BY #BIT6
3850  013020  042777  000100  003076                                              BIC     #BIT6,@TMP
3851  013026                                                                  ENDSRV
3852  013026  000002                  RTI
3853
3854  013030                                                                  BGNSRV R2SRV
3855  013030                  R2SRV:
3856  013030                                                                  LET TMP := RCSR
3857  013030  013737  001262  016124                                              MOV     RCSR,TMP
3858  013036                                                                  LET TMP := TMP + #22
3859  013036  062737  000022  016124                                              ADD     #22,TMP
3860  013044                                                                  LET @X2 := @TMP
3861  013044  017777  003054  003204                                              MOV     @TMP,@X2
3862  013052                                                                  LET X2 := X2 + #2
3863  013052  062737  000002  016256                                              ADD     #2,X2
3864  013060                                                                  IF TMP2 NE TMP2E THEN
3865  013060  023737  016136  016140                                              CMP     TMP2,TMP2E
3866  013066  001002                                                              BNE     .+6
3867  013070  000137  013114                                                      JMP     $50245
3868                                                                            ;GO TO XCSR
3869  013074                                                                  LET TMP := TMP + #2
```

```
3870  013074  062737  000002  016124                                                    ADD     #2,TMP
3871                                                                                  ;ENABLE XMIT INTERR
3872  013102                                                                          LET @TMP := @TMP SET.BY #BIT6
3873  013102  052777  000100  003014                                                    BIS     #BIT6,@TMP
3874  013110                                                                        ELSE
3875  013110  000137  013120                                                    JMP     $50246
3876  013114                                                                    $50245:
3877                                                                                  ;ALL DONE
3878  013114                                                                          LET TMP10 := TMP10 - #1
3879  013114  005337  016170                                                    DEC     TMP10
3880  013120                                                                        ENDIF
3881  013120                                                                    $50246.
3882  013120                                                                  ENDSRV
3883  013120  000002                  RTI
3884
3885  013122                                                                  BGNSRV X2SRV
3886  013122                  X2SRV:
3887  013122                                                                      LET TMP := RCSR
3888  013122  013737  001262  016124                                                MOV     RCSR,TMP
3889  013130                                                                      LET TMP := TMP + #26
3890  013130  062737  000026  016124                                                ADD     #26,TMP
3891  013136                                                                      LET @TMP := TMP2
3892  013136  013777  016136  002760                                                MOV     TMP2,@TMP
3893  013144                                                                      LET TMP2 := TMP2 + #1
3894  013144  005237  016136                                                    INC     TMP2
3895                                                                                ;GO BACK TO XCSR
3896  013150                                                                      LET TMP := TMP - #2
3897  013150  162737  000002  016124                                                SUB     #2,TMP
3898                                                                                ;DISABLE XMIT INTERRUPTS,
3899                                                                                ;NOT EXER DOUBLE BUFF.
3900  013156                                                                      LET @TMP := @TMP CLR.BY #BIT6
3901  013156  042777  000100  002740                                                BIC     #BIT6,@TMP
3902  013164                                                                  ENDSRV
3903  013164  000002                  RTI
3904
3905  013166                                                                  BGNSRV R3SRV
3906  013166                  R3SRV:
3907  013166                                                                      LET TMP := RCSR
3908  013166  013737  001262  016124                                                MOV     RCSR,TMP
3909  013174                                                                      LET TMP := TMP + #32
3910  013174  062737  000032  016124                                                ADD     #32,TMP
3911  013202                                                                      LET @X3 := @TMP
3912  013202  017777  002716  003050                                                MOV     @TMP,@X3
3913  013210                                                                      LET X3 := X3 + #2
3914  013210  062737  000002  016260                                                ADD     #2,X3
3915  013216                                                                      IF TMP3 NE TMP3E THEN
3916  013216  023737  016142  016144                                                CMP     TMP3,TMP3E
3917  013224  001002                                                               BNE     .+6
3918  013226  000137  013252                                                    JMP     $50247
3919
3920  013232                                                                        ;GO TO XCSR
3921  013232  062737  000002  016124                                              LET TMP := TMP + #2
3922                                                                                ADD     #2,TMP
3923  013240                                                                        ;ENABLE XMIT INTERR
3924  013240  052777  000100  002656                                              LET @TMP := @TMP SET.BY #BIT6
3925  013246                                                                        BIS     #BIT6,@TMP
                                                                              ELSE
```

```
3926  013246  000137  013256                                                          JMP     $50250
3927  013252                                                                          $50247:
3928                                                                              ;ALL DONE
3929  013252                                                                      LET TMP10 := TMP10 - #1
3930  013252  005337  016170                                                          DEC     TMP10
3931  013256                                                                      ENDIF
3932  013256                                                                          $50250:
3933  013256                                                                  ENDSRV
3934  013256  000002                  RTI
3935
3936  013260                                                                  BGNSRV X3SRV
3937  013260                   X3SRV:
3938  013260                                                                      LET TMP := RCSR
3939  013260  013737  001262  016124                                                  MOV     RCSR,TMP
3940  013266                                                                      LET TMP := TMP + #36
3941  013266  062737  000036  016124                                                  ADD     #36,TMP
3942  013274                                                                      LET @TMP := TMP3
3943  013274  013777  016142  002622                                                  MOV     TMP3,@TMP
3944  013302                                                                      LET TMP3 := TMP3 + #1
3945  013302  005237  016142                                                          INC     TMP3
3946                                                                              ;GO BACK TO XCSR
3947  013306                                                                      LET TMP := TMP - #2
3948  013306  162737  000002  016124                                                  SUB     #2,TMP
3949                                                                              ;DISABLE XMIT INTERRUPTS,
3950                                                                              ;NOT EXER DOUBLE BUFF
3951  013314                                                                      LET @TMP := @TMP CLR.BY #BIT6
3952  013314  042777  000100  002602                                                  BIC     #BIT6,@TMP
3953  013322                                                                  ENDSRV
3954  013322  000002                  RTI
3955
3956  013324                                                                  BGNSRV R4SRV
3957  013324                   R4SRV:
3958  013324                                                                      LET TMP := RCSR
3959  013324  013737  001262  016124                                                  MOV     RCSR,TMP
3960  013332                                                                      LET TMP := TMP + #42
3961  013332  062737  000042  016124                                                  ADD     #42,TMP
3962  013340                                                                      LET @X4 := @TMP
3963  013340  017777  002560  002714                                                  MOV     @TMP,@X4
3964  013346                                                                      LET X4 := X4 + #2
3965  013346  062737  000002  016262                                                  ADD     #2,X4
3966  013354                                                                      IF TMP4 NE TMP4E THEN
3967  013354  023737  016146  016150                                                  CMP     TMP4,TMP4E
3968  013362  001002                                                                  BNE     .+6
3969  013364  000137  013410                                                          JMP     $50251
3970                                                                              ;GO TO XCSR
3971  013370                                                                      LET TMP := TMP + #2
3972  013370  062737  000002  016124                                                  ADD     #2,TMP
3973                                                                              ;ENABLE XMIT INTERR
3974  013376                                                                      LET @TMP := @TMP SET.BY #BIT6
3975  013376  052777  000100  002520                                                  BIS     #BIT6,@TMP
3976  013404                                                                      ELSE
3977  013404  000137  013414                                                          JMP     $50252
3978  013410                                                                          $50251:
3979                                                                              ;ALL DONE
3980  013410                                                                      LET TMP10 := TMP10 - #1
3981  013410  005337  016170                                                          DEC     TMP10
```

```
3982  013414                                                        ENDIF
3983  013414                                                                                $50252:
3984  013414                                                    ENDSRV
3985  013414  000002                    RTI
3986
3987  013416                                                    BGNSRV X4SRV
3988  013416                    X4SRV:
3989  013416                                                        LET TMP := RCSR
3990  013416  013737  001262  016124                                                MOV     RCSR,TMP
3991  013424                                                        LET TMP := TMP + #46
3992  013424  062737  000046  016124                                                ADD     #46,TMP
3993  013432                                                        LET @TMP := TMP4
3994  013432  013777  016146  002464                                                MOV     TMP4,@TMP
3995  013440                                                        LET TMP4 := TMP4 + #1
3996  013440  005237  016146                                                        INC     TMP4
3997                                                                ;GO BACK TO XCSR
3998  013444                                                        LET TMP := TMP - #2
3999  013444  162737  000002  016124                                                SUB     #2,TMP
4000                                                                ;DISABLE XMIT INTERRUPTS,
4001                                                                ;NOT EXER DOUBLE BUFF.
4002  013452                                                        LET @TMP := @TMP CLR.BY #BIT6
4003  013452  042777  000100  002444                                                BIC     #BIT6,@TMP
4004  013460                                                    ENDSRV
4005  013460  000002                    RTI
4006
4007  013462                                                    BGNSRV R5SRV
4008  013462                    R5SRV:
4009  013462                                                        LET TMP := RCSR
4010  013462  013737  001262  016124                                                MOV     RCSR,TMP
4011  013470                                                        LET TMP := TMP + #52
4012  013470  062737  000052  016124                                                ADD     #52,TMP
4013  013476                                                        LET @X5 := @TMP
4014  013476  017777  002422  002560                                                MOV     @TMP,@X5
4015  013504                                                        LET X5 := X5 + #2
4016  013504  062737  000002  016264                                                ADD     #2,X5
4017  013512                                                        IF TMP5 NE TMP5E THEN
4018  013512  023737  016152  016154                                                CMP     TMP5,TMP5E
4019  013520  001002                                                                BNE     .+6
4020  013522  000137  013546                                                        JMP     $50253
4021                                                                    ;GO TO XCSR
4022  013526                                                            LET TMP := TMP + #2
4023  013526  062737  000002  016124                                                ADD     #2,TMP
4024                                                                    ;ENABLE XMIT INTERR
4025  013534                                                            LET @TMP := @TMP SET.BY #BIT6
4026  013534  052777  000100  002362                                                BIS     #BIT6,@TMP
4027  013542                                                        ELSE
4028  013542  000137  013552                                                        JMP     $50254
4029  013546                                                                        $50253:
4030                                                                    ;ALL DONE
4031  013546                                                            LET TMP10 := TMP10 - #1
4032  013546  005337  016170                                                        DEC     TMP10
4033  013552                                                        ENDIF
4034  013552                                                                        $50254:
4035  013552                                                    ENDSRV
4036  013552  000002                    RTI
4037
```

```
4038  013554                                  BGNSRV X5SRV
4039  013554                        X5SRV:
4040  013554                            LET TMP := RCSR
4041  013554  013737  001262  016124                    MOV     RCSR,TMP
4042  013562                            LET TMP := TMP + #56
4043  013562  062737  000056  016124                    ADD     #56,TMP
4044  013570                            LET @TMP := TMP5
4045  013570  013777  016152  002326                    MOV     TMP5,@TMP
4046  013576                            LET TMP5 := TMP5 + #1
4047  013576  005237  016152                            INC     TMP5
4048                                        ;GO BACK TO XCSR
4049  013602                            LET TMP := TMP - #2
4050  013602  162737  000002  016124                    SUB     #2,TMP
4051                                        ;DISABLE XMIT INTERRUPTS,
4052                                        ;NOT EXER DOUBLE BUFF.
4053  013610                            LET @TMP := @TMP CLR.BY #BIT6
4054  013610  042777  000100  002306                    BIC     #BIT6,@TMP
4055  013616                            ENDSRV
4056  013616  000002                RTI
4057
4058  013620                            BGNSRV R6SRV
4059  013620                        R6SRV:
4060  013620                            LET TMP := RCSR
4061  013620  013737  001262  016124                    MOV     RCSR,TMP
4062  013626                            LET TMP := TMP + #62
4063  013626  062737  000062  016124                    ADD     #62,TMP
4064  013634                            LET @X6 := @TMP
4065  013634  017777  002264  002424                    MOV     @TMP,@X6
4066  013642                            LET X6 := X6 + #2
4067  013642  062737  000002  016266                    ADD     #2,X6
4068  013650                            IF TMP6 NE TMP6E THEN
4069  013650  023737  016156  016160                    CMP     TMP6,TMP6E
4070  013656  001002                                    BNE     .+6
4071  013660  000137  013704                            JMP     $50255
4072                                        ;GO TO XCSR
4073  013664                            LET TMP := TMP + #2
4074  013664  062737  000002  016124                    ADD     #2,TMP
4075                                        ;ENABLE XMIT INTERR
4076  013672                            LET @TMP := @TMP SET.BY #BIT6
4077  013672  052777  000100  002224                    BIS     #BIT6,@TMP
4078  013700                            ELSE
4079  013700  000137  013710                            JMP     $50256
4080  013704                                        $50255:
4081                                        ;ALL DONE
4082  013704                            LET TMP10 := TMP10 - #1
4083  013704  005337  016170                            DEC     TMP10
4084  013710                            ENDIF
4085  013710                                        $50256:
4086  013710                            ENDSRV
4087  013710  000002                RTI
4088
4089  013712                            BGNSRV X6SRV
4090  013712                        X6SRV:
4091  013712                            LET TMP := RCSR
4092  013712  013737  001262  016124                    MOV     RCSR,TMP
4093  013720                            LET TMP := TMP + #66
```

```
4094  013720  062737  000066  016124                                                          ADD     #66,TMP
4095  013726                                                                          LET @TMP := TMP6
4096  013726  013777  016156  002170                                                          MOV     TMP6,@TMP
4097  013734                                                                          LET TMP6 := TMP6 + #1
4098  013734  005237  016156                                                                 INC     TMP6
4099                                                                                      ;GO BACK TO XCSR
4100  013740                                                                          LET TMP := TMP - #2
4101  013740  162737  000002  016124                                                          SUB     #2,TMP
4102                                                                                      ;DISABLE XMIT INTERRUPTS,
4103                                                                                      ;NOT EXER DOUBLE BUFF.
4104  013746                                                                          LET @TMP := @TMP CLR.BY #BIT6
4105  013746  042777  000100  002150                                                          BIC     #BIT6,@TMP
4106  013754                                                                      ENDSRV
4107  013754  000002                          RTI
4108
4109  013756                                                                      BGNSRV R7SRV
4110  013756                          R7SRV:
4111  013756                                                                          LET TMP := RCSR
4112  013756  013737  001262  016124                                                          MOV     RCSR,TMP
4113  013764                                                                          LET TMP := TMP + #72
4114  013764  062737  000072  016124                                                          ADD     #72,TMP
4115  013772                                                                          LET @X7 := @TMP
4116  013772  017777  002126  002270                                                          MOV     @TMP,@X7
4117  014000                                                                          LET X7 := X7 + #2
4118  014000  062737  000002  016270                                                          ADD     #2,X7
4119  014006                                                                          IF TMP7 NE TMP7E THEN
4120  014006  023737  016162  016164                                                          CMP     TMP7,TMP7E
4121  014014  001002                                                                          BNE     .+6
4122  014016  000137  014042                                                                 JMP     $50257
4123                                                                                      ;GO TO XCSR
4124  014022                                                                              LET TMP := TMP + #2
4125  014022  062737  000002  016124                                                          ADD     #2,TMP
4126                                                                                      ;ENABLE XMIT INTERR
4127  014030                                                                              LET @TMP := @TMP SET.BY #BIT6
4128  014030  052777  000100  002066                                                          BIS     #BIT6,@TMP
4129  014036                                                                          ELSE
4130  014036  000137  014046                                                                 JMP     $50260
4131  014042                                                                              $50257:
4132                                                                                      ;ALL DONE
4133  014042                                                                              LET TMP10 := TMP10 - #1
4134  014042  005337  016170                                                                 DEC     TMP10
4135  014046                                                                          ENDIF
4136  014046                                                                              $50260:
4137  014046                                                                      ENDSRV
4138  014046  000002                              RTI
4139
4140  014050                                                                      BGNSRV X7SRV
4141  014050                              X7SRV:
4142  014050                                                                          LET TMP := RCSR
4143  014050  013737  001262  016124                                                          MOV     RCSR,TMP
4144  014056                                                                          LET TMP := TMP + #76
4145  014056  062737  000076  016124                                                          ADD     #76,TMP
4146  014064                                                                          LET @TMP := TMP7
4147  014064  013777  016162  002032                                                          MOV     TMP7,@TMP
4148  014072                                                                          LET TMP7 := TMP7 + #1
4149  014072  005237  016162                                                                 INC     TMP7
```

```
                                                        ;GO BACK TO XCSR
                                                    LET TMP := TMP - #2
014076                                                                  SUB     #2,TMP
014076  162737  000002  016124                          ;DISABLE XMIT INTERRUPTS.
                                                        ;NOT EXER DOUBLE BUFF.
014104                                                  LET @TMP :- @TMP CLR.BY #BIT6
014104  042777  000100  002012                                          BIC     #BIT6,@TMP
014112                                                ENDSRV
014112  000002                  RTI
014114                  SEREND:

                        ;** RECEIVER STATUS TABLES **
                                                        ;7 BIT WDS      8 BIT WDS
014114  000100          CH0TAB: .BLKW 100               ;  0- 37          0- 77
014314  000100          CH1TAB: .BLKW 100               ; 40- 77        100-177
014514  000100          CH2TAB: .BLKW 100               ;100-137        200-277
014714  000100          CH3TAB: .BLKW 100               ;140-177        300-377
015114  000100          CH4TAB: .BLKW 100
015314  000100          CH5TAB: .BLKW 100
015514  000100          CH6TAB: .BLKW   100
015714  000100          CH7TAB: .BLKW 100
016114                  STATEND:
016114  000000          CHMSK:  .WORD 0                 ;MASK OF ACTIVE CHANNELS
016116  000000          CHCTR:  .WORD 0                 ;CHANNEL CTR
016120  000000          ACTCH:  .WORD 0                 ;ACTIVE CHANNEL CTR
016122  000000          ERWRD:  .WORD 0                 ;ERROR WORD, LOW BYTE IS NO. OF DATA ERRORS
                                                        ;HIGH BYTE INDICATES ERROR FLAG DETECTED

016124  000000          TMP:    .WORD 0                 ;TEMP STORAGE FOR SERVICE ROUTINES

016126  000000          TMP0:   0                       ;START WORD FOR CH 0
016130  000000          TMP0E:  0                       ;END   WORD FOR CH 0
016132  000000          TMP1:   0                       ;START              1
016134  000000          TMP1E:  0                       ;END                1
016136  000000          TMP2:   0                       ;START              2
016140  000000          TMP2E:  0                       ;END                2
016142  000000          TMP3:   0                       ;START              3
016144  000000          TMP3E:  0                       ;END                3
016146  000000          TMP4:   0
016150  000000          TMP4E:  0
016152  000000          TMP5:   0
016154  000000          TMP5E:  0
016156  000000          TMP6:   0
016160  000000          TMP6E:  0
016162  000000          TMP7:   0
016164  000002          TMP7E:  .BLKW   2

016170  000000          TMP10:  .WORD

016172  014114          TABL6:  CH0TAB
016174  014314                  CH1TAB
016176  014514                  CH2TAB
016200  014714                  CH3TAB
016202  015114                  CH4TAB
016204  015314                  CH5TAB
016206  015514                  CH6TAB
```

```
                                        CH7TAB

TABL7:  40                              ;7 BIT WORD TABLE
        100
        140
        200
        240
        300
        340
        400

TABL8:  100                             ;8 BIT WORD TABLE
        200
        300
        400
        500
        600
        700
        1000
X0:     .WORD 0
X1:     .WORD 0
X2:     .WORD 0
X3:     .WORD 0
X4:     .WORD 0
X5:     .WORD 0
X6:     .WORD 0
X7:     .WORD 0
```

```
4235
4236                                    ;*****************************************************************
4237                                    ;* ROUTINE:TIMER
4238                                    ;*       THIS ROUTINE IS USED TO TEST THE STATUS OF ANY BIT
4239                                    ;*       IN ANY REGISTER.
4240                                    ;*       INPUTS:
4241                                    ;*             HOWLONG    THE MAXIMUM AMOUNT OF TIME TO SPEND IN
4242                                    ;*                        THIS ROUTINE.
4243                                    ;*             WHICHBIT     A MASK WITH THE BIT(S) SET THAT ARE
4244                                    ;*                          TO BE CHECKED.
4245                                    ;*             REG      A POINTER TO THE REGISTER TO BE CHECKED
4246                                    ;*             SETCLR   THE DESIRED RESULTS
4247                                    ;*                      EITHER #-1 OR #CLEAR
4248                                    ;*       OUTPUT:
4249                                    ;*             THE 'C' BIT IS SET TO INDICATE AN ERROR
4250                                    ;*             BUT IT IS TESTED BY THE IF.ERROR STATEMENT
4251                                    ;*
4252                                    ;*
4253                                    ;*             NOTE::  THE USE OF (R5) IS PART OF THE LINKAGE
4254                                    ;*                     MECHANISM BETWEEN THE CALLER AND THE CALLED
4255                                    ;*****************************************************************
4256
4257
4258  016272                                        ROUTINE TIMER <HOWLONG,WHICHBIT,REG,SETCLR>
4259  016272                                                                              TIMER:
4260  016272                                LET     REGSAV := REG(R5) ; GET POINTER TO REGISTER
4261  016272  016537  000004  016512                                                    MOV     REG(R5),REGSAV
4262  016300                                LET     TIMSAV := HOWLONG(R5) ; SAVE HOWLONG FOR QUICKER RETRIEVAL
4263  016300  016537  000000  016514                                                    MOV     HOWLONG(R5),TIMSAV
4264  016306                                                        LET     FLAG :B= #0 ; INITIALIZE THE EXIT FLAG
4265  016306  105037  016516                                                          CLRB    FLAG
4266
4267
4268                                    ;       START OF AN INFINITE LOOP
4269
4270  016312                            LOOP
4271  016312                                                      $50263:
4272                                          ; TEST TO SEE IF WHICHBIT IS SET
4273  016312                                IF      WHICHBIT(R5)  NOTSETIN @REGSAV THEN
4274  016312  036577  000002  000172                                                  BIT     WHICHBIT(R5),@REGSAV
4275  016320  001402                                                                 BEQ     .+6
4276  016322  000137  016336                                                         JMP     $50265
4277  016326                                    LET     HOLDSC :B= #0
4278  016326  105037  016517                                                        CLRB    HOLDSC
4279  016332                                ELSE
4280  016332  000137  016344                                                         JMP     $50266
4281  016336                                                      $50265:
4282  016336                                    LET     HOLDSC :B- #-1 ; REMEMBER THIS
4283  016336  112737  177777  016517                                                MOVB    #-1,HOLDSC
4284  016344                                ENDIF
4285  016344                                                      $50266:
4286
4287                                    ; NOW SEE IF THAT WAS WHAT WE WANTED
4288  016344                                IFB     HOLDSC EQ SETCLR(R5) THEN
4289  016344  123765  016517  000006                                                  CMPB    HOLDSC,SETCLR(R5)
4290  016352  001402                                                                 BEQ     .+6
```

```
4291  016354  000137  016366                                                        JMP     $50267
4292                                                                        ; JUST THE THING WE NEEDED
4293  016360                                                                      LET     FLAG :B= #BIT00
4294  016360  112737  000001  016516                                              MOVB    #BIT00,FLAG
4295  016366                                                                    ENDIF
4296  016366                                                                    $50267:
4297
4298  016366                                                                  EXIFB   FLAG EQ #BIT00
4299  016366  123727  016516  000001                                            CMPB    FLAG,#BIT00
4300  016374  001002                                                          BNE     $50270
4301  016376  000137  016452                                                  JMP     $50264
4302  016402                                                                  $50270:
4303  016402                                                                EXIF TIMSAV LE #0
4304  016402  005737  016514                                                  TST     TIMSAV
4305  016406  003002                                                          BGT     $50271
4306  016410  000137  016452                                                  JMP     $50264
4307  016414                                                                  $50271:
4308                                                                      ; ONE WAY OR THE OTHER, WE ARE DONE
4309                                                                    ; IF WE ARE STILL HERE THEN HANG AROUND A WHILE
4310
4311  016414                                                                CALL WAIT IN <#1>
4312  016414  010546                                                          MOV     R5,-(SP)
4313  016416  010605                                                          MOV     SP,R5
4314  016420  162706  000002                                                  SUB     #2,SP
4315  016424  010546                                                          MOV     R5,-(SP)
4316  016426  012745  000001                                                  MOV     #1,-(R5)
4317  016432  004737  016524                                                  JSR     PC,WAIT
4318  016436  012605                                                          MOV     (SP)+,R5
4319  016440  010506                                                          MOV     R5,SP
4320  016442  012605                                                          MOV     (SP)+,R5
4321  016444                                                                LET     TIMSAV := TIMSAV - #1 ; COUNTING DOWN
4322  016444  005337  016514                                                  DEC     TIMSAV
4323  016450                                                              ENDLOOP                 ; CONTINUED AT THE TOP
4324  016450  000720                                                          BR      $50263
4325  016452                                                                  $50264:
4326
4327                                                                      ; ONLY 2 WAYS TO GET HERE
4328                                                                      ;       1). WE RAN OUT OF TIME---ERROR ! !
4329                                                                      ;       2). THE BIT IS IN THE CORRECT CONDITION--GOOD ! !
4330  016452                                                            IF #BIT00 SETIN FLAG  AND TIMSAV GT #0 THEN
4331  016452  032737  000001  016516                                            BIT     #BIT00,FLAG
4332  016460  001002                                                          BNE     .+6
4333  016462  000137  016504                                                  JMP     $50272
4334  016466  005737  016514                                                  TST     TIMSAV
4335  016472  003002                                                          BGT     .+6
4336  016474  000137  016504                                                  JMP     $50272
4337  016500                                                                RETURN NO.ERROR
4338  016500  000137  016520                                                  JMP     $50261
4339  016504                                                            ENDIF
4340  016504                                                                  $50272:
4341
4342  016504                                                            RETURN ERROR
4343  016504  000261                                                          SEC
4344  016506  000137  016522                                                  JMP     $50262
4345
4346
```

```
 4347  016512  000000                  REGSAV: .WORD 0
 4348  016514  000000                  TIMSAV: .WORD 0
 4349  016516     000                  FLAG:   .BYTE 0
 4350  016517     000                  HOLDSC: .BYTE 0
 4351                                          ; WE ARE DONE  GO BACK HOME
 4352  016520                                        ENDRTN
 4353  016520                                                          $50261:
 4354  016520  000241                                                 CLC
 4355  016522                                                          $50262:
 4356  016522  000207                                                 RTS     PC
```

```
                                ;;*****************************************************************
                                ;* ROUTINE:WAIT
                                ;*       THIS ROUTINE IS USED TO DELAY EXECUTION OF THE
                                ;*       MAIN  PROGRAM  FOR A SPECIFIED AMOUNT OF TIME.
                                ;*       THIS   IS   ACCOMPLISHED   BY  INCREMENTING  A
                                ;*       REGISTER UP TO A LIMIT.  THE INNER LOOP IS SET
                                ;*       TO APPROXIMATE 1 MILLI SEC.
                                ;;*****************************************************************
                                            ROUTINE WAIT    <TIME>
                                                                        WAIT:
                                                    PUSH    <R1,R2,R3>
                                                    LET R1 := TIME(R5)
016524                                                                  MOV     TIME(R5),R1
                                                    INCRU R2 FROM #1 TO R1 BY #1
016536  012702  000001                                                  MOV     #1,R2
016542  000402                                                         BR      $50276
                                                                        $50275:
016544  062702  000001                                                  ADD     #01,R2
                                                                        $50276:
016550  020201                                                         CMP     R2,R1
016552  101402                                                         BLOS    $50277
016554  000137  016604                                                 JMP     $50300
                                                                        $50277:
                                                        INCR R3 FROM #0 TO #100 BY #1
016560  005003                                                         CLR     R3
016562  000401                                                         BR      $50302
                                                                        $50301:
016564  005203                                                         INC     R3
                                                                        $50302:
016566  020327  000100                                                 CMP     R3,#100
016572  003402                                                         BLE     $50303
016574  000137  016602                                                 JMP     $50304
                                                                        $50303:
                                                        ENDINC
016600  000771                                                         BR      $50301
                                                                        $50304:
                                                    ENDINC
016602  000760                                                         BR      $50275
                                                                        $50300:
                                                    POP     <R3,R2,R1>
                                        ENDRTN
                                                                        $50273:
                                                                        $50274:
016612  000207                                                         RTS     PC
```

```
                                        .SBTTL  ROUTINE TO SIZE THE BUS


                                        ;*************************************************************
                                        ;       THIS ROUTINE WILL LOOK AT THE BUS AND DETERMINE
                                        ;       WHAT DEVICES ARE THERE BY LOOKING AT $USWR BITS
                                        ;       10 AND 9.
                                        ;* *************************************************************

016614                                  ROUTINE SIZE
016614                                                                                  SIZE:

016614                                  LET ERRVEC := #INTSRV                   ;MY INTR SERVICE ROUTINE
016614  012737  017632  000004                                                          MOV     #INTSRV,ERRVEC
016622                                  LET ERRVEC+2 := #340
016622  012737  000340  000006                                                          MOV     #340,ERRVEC+2
016630                                  LET P1CNT := $USWR                      ;A TEMP STORE
016630  013737  001220  001300                                                          MOV     $USWR,P1CNT
016636                                  LET P1CNT := P1CNT SHIFT -11            ;SHIFT OVER TO ZERO
016636  006237  001300                                                                  ASR     P1CNT
016642  006237  001300                                                                  ASR     P1CNT
016646  006237  001300                                                                  ASR     P1CNT
016652  006237  001300                                                                  ASR     P1CNT
016656  006237  001300                                                                  ASR     P1CNT
016662  006237  001300                                                                  ASR     P1CNT
016666  006237  001300                                                                  ASR     P1CNT
016672  006237  001300                                                                  ASR     P1CNT
016676  006237  001300                                                                  ASR     P1CNT

016702                                  SELECT P1CNT OF 4 VERIFY
016702  013746  001300                                                          MOV     P1CNT,-(SP)
016706  002002                                                                  BGE     .+6
016710  000137  017110                                                          JMP     $50315
016714  023727  001300  000004                                                  CMP     P1CNT,#4
016722  003402                                                                  BLE     .+6
016724  000137  017110                                                          JMP     $50315
016730  006316                                                                  ASL     (SP)
016732  062716  016740                                                          ADD     #$50307,(SP)
016736  013607                                                                  MOV     @(SP)+,PC
016740                                                                          $50307:
016740  016752                                                                  .WORD   $50314
016742  016770                                                                  .WORD   $50313
016744  017006                                                                  .WORD   $50312
016746  017024                                                                  .WORD   $50311
016750  017024                                                                  .WORD   $50310
016752                                      CASE 0
016752                                                                          $50314:
016752                                          LET NODLV := #1
016752  012737  000001  001304                                                  MOV     #1,NODLV
016760                                          INLINE <JSR     PC,ONKDF>       ;NO DLV'S
016760  004737  017764                                                          JSR     PC,ONKDF
016764                                      CASE 1
016764  000137  017114                                                          JMP     $50316
016770                                                                          $50313:
```

```
 4458  016770                                  LET KDDLV := #1                                          MOV     #1,KDDLV
 4459  016770  012737  000001  001306          INLINE <JSR     PC,KDDLVJ>      ;ONE DLV                 JSR     PC,KDDLVJ
 4460  016776                                CASE 2                                                     JMP     $50316
 4461  016776  004737  020276                                                                         $50312:
 4462  017002                                  LET DLV211 := #1                                         MOV     #1,DLV211
 4463  017002  000137  017114                  INLINE <JSR     PC,KD2DLV>      ;TWO DLV'S               JSR     PC,KD2DLV
 4464  017006                                DEFAULT                                                    JMP     $50316
 4465  017006                                                                                           $50311:
 4466  017006  012737  000001  001310                                                                   $50310:
 4467  017014                                  IFB $AUTOB EQ #0 THEN                                    TSTB    $AUTOB
 4468  017014  004737  017642                                                                         BEQ     .+6
 4469  017020                                          TYPTXT  <<CRLF>!CANNOT TEST THIS CONFIGURATION!> JMP     $50317
 4470  017020  000137  017114                  ENDIF                                                    $50317:
 4471  017024                                  INLINE <BR      .>                                       BR      .
 4472  017024                                ENDSELECT                                                  BR      $50316
 4473  017024                                                                                           $50315:
 4474  017024  105737  001134                                                                         ADD     #2,SP
 4475  017030  001402                                                                                  $50316:
 4476  017032  000137  017104
 4477  017036
 4478  017104
 4479  017104
 4480  017104
 4481  017104  000777
 4482  017106
 4483  017106  000402
 4484  017110
 4485  017110  062706  000002
 4486  017114
 4487
 4488
 4489  017114                                IF DLV211 EQ #1 THEN                                       CMP     DLV211,#1
 4490  017114  023727  001310  000001                                                                   BEQ     .+6
 4491  017122  001402                                                                                  JMP     $50320
 4492  017124  000137  017230
 4493  017130                                IFB $AUTOB EQ #0 THEN                                      TSTB    $AUTOB
 4494  017130  105737  001134                                                                         BEQ     .+6
 4495  017134  001402                                                                                  JMP     $50321
 4496  017136  000137  017230
 4497  017142                                    TYPTXT      <<CRLF>!WILL TEST KDF11-B!>
 4498  017174                                    TYPTXT      <<CRLF>!CONSOL ADRS !>
 4499  017220                                    LET -(SP) := CONADR                                    MOV     CONADR,-(SP)
 4500  017220  013746  001274
 4501  017224  104403                            TYPOS
 4502  017226     006                            .BYTE       6
 4503  017227     000                            .BYTE       0
 4504  017230                                ENDIF
 4505  017230                                                                                           $50321:
 4506  017230                                ENDIF
 4507  017230                                                                                           $50320:
 4508
 4509  017230                                IF KDDLV EQ #1 THEN                                        CMP     KDDLV,#1
 4510  017230  023727  001306  000001                                                                   BEQ     .+6
 4511  017236  001402                                                                                  JMP     $50322
 4512  017240  000137  017316
 4513  017244                                    LET DLV211 := #0
```

```
 4514  017244  005037  001310                                                    CLR     DLV211
 4515  017250
 4516  017250  105737  001134          IFB $AUTOB EQ #0 THEN                     TSTB    $AUTOB
 4517  017254  001402                                                            BEQ     .+6
 4518  017256  000137  017316                                                    JMP     $50323
 4519  017262                                TYPTXT      <<CRLF>!SLU 2 ADRS !>
 4520  017306                          LET -(SP) := SECSLU                       MOV     SECSLU,-(SP)
 4521  017306  013746  001302
 4522  017312  104403                          TYPOS
 4523  017314     006                          .BYTE       6
 4524  017315     000                          .BYTE       0
 4525  017316                                   ENDIF
 4526  017316                                                                    $50323:
 4527  017316                      ENDIF                                         $50322:
 4528  017316
 4529
 4530  017316                      IF NODLV EQ #1 THEN                           CMP     NODLV,#1
 4531  017316  023727  001304  000001                                            BEQ     .+6
 4532  017324  001402                                                            JMP     $50324
 4533  017326  000137  017342
 4534  017332                          LET DLV211 := #0                          CLR     DLV211
 4535  017332  005037  001310
 4536  017336                          LET KDDLV := #0                           CLR     KDDLV
 4537  017336  005037  001306
 4538  017342                      ENDIF                                         $50324:
 4539  017342
 4540
 4541  017342                      LET ERRVEC := #6                ;RESTORE
 4542  017342  012737  000006  000004                                            MOV     #6,ERRVEC
 4543  017350                      LET ERRVEC+2 := #0              ;VECTORS
 4544  017350  005037  000006                                                    CLR     ERRVEC+2
 4545                              ;*****************************************************************
 4546
 4547  017354                      IF $DEVM EQ #0 THEN             ;ANY DLV'S
 4548  017354  005737  001252                                                    TST     $DEVM
 4549  017360  001402                                                            BEQ     .+6
 4550  017362  000137  017442                                                    JMP     $50325
 4551  017366                          IFB $AUTOB EQ #0 THEN
 4552  017366  105737  001134                                                    TSTB    $AUTOB
 4553  017372  001402                                                            BEQ     .+6
 4554  017374  000137  017436                                                    JMP     $50326
 4555  017400                                TYPTXT      <<CRLF>!THERE ARE NO DLV11'S !>
 4556  017436                          ENDIF
 4557  017436                                                                    $50326:
 4558  017436
 4559  017436  000137  017630                                                    JMP     $50327
 4560  017442                                                                    $50325:
 4561  017442
 4562  017442  013737  001250  001312          DLBASE := $BASE                   MOV     $BASE,DLBASE
 4563  017450
 4564  017450  052737  000001  001320      LET MASK := MASK SET.BY #BIT00        BIS     #BIT00,MASK
 4565  017456
 4566  017456                              WHILE DLBASE NE #176700 DO            $50330:
 4567  017456  023727  001312  176700                                            CMP     DLBASE,#176700
 4568  017464  001002                                                            BNE     .+6
 4569  017466  000137  017600                                                    JMP     $50331
```

```
4570  017472
4571  017472  033737  001320  001252          IF MASK SETIN $DEVM THEN                                BIT     MASK,$DEVM
4572  017500  001002                                                                                  BNE     .+6
4573  017502  000137  017564                                                                          JMP     $50332
4574  017506
4575  017506  105737  001134                      IFB $AUTOB EQ #0 THEN                               TSTB    $AUTOB
4576  017512  001402                                                                                  BEQ     .+6
4577  017514  000137  017554                                                                          JMP     $50333
4578  017520                                              TYPTXT      <<CRLF>!DLV11-J MODULE ADRS !>
4579  017554                                          ENDIF
4580  017554                                                                                          $50333:
4581  017554
4582  017554  013746  001312                      LET -(SP) := DLBASE                                 MOV     DLBASE,-(SP)
4583  017560  104403                              TYPOS
4584  017562     006                              .BYTE       6
4585  017563     000                              .BYTE       0
4586  017564                                  ENDIF
4587  017564                                                                                          $50332:
4588  017564
4589  017564  062737  000010  001312          LET DLBASE := DLBASE + #10                              ADD     #10,DLBASE
4590  017572
4591  017572  006337  001320                  LET MASK := MASK SHIFT 1                                ASL     MASK
4592  017576                              ENDDO
4593  017576  000727                                                                                  BR      $50330
4594  017600                                                                                          $50331:
4595  017600
4596  017600  010046                      LET -(SP) := R0                             ;SAVE R0        MOV     R0,-(SP)
4597  017602
4598  017602  005000                      INCRU R0 FROM #0 TO #10000 BY #1                            CLR     R0
4599  017604  000402                                                                                  BR      $50335
4600  017606                                                                                          $50334:
4601  017606  062700  000001                                                                          ADD     #01,R0
4602  017612                                                                                          $50335:
4603  017612  020027  010000                                                                          CMP     R0,#10000
4604  017616  101402                                                                                  BLOS    $50336
4605  017620  000137  017626                                                                          JMP     $50337
4606  017624                                                                                          $50336:
4607  017624                                  ENDINC                              ;WAIT FOR TERMINAL TO STOP
4608  017624  000770                                                                                  BR      $50334
4609  017626                                                                                          $50337:
4610  017626
4611  017626  012600                      LET R0 := (SP)+                         ;RESTORE R0         MOV     (SP)+,R0
4612  017630                              ENDIF .
4613  017630                                                                                          $50327:
4614
4615  017630                              ENDRTN
4616  017630                                                                                          $50305:
4617  017630                                                                                          $50306:
4618  017630  000207                                                                                  RTS     PC
4619
4620
4621
4622                                    .SBTTL  INTSRV  INTERRUPT SERVICE RUUTINE
4623
4624                                    ;;*****************************************************************
4625                                    ;*       THIS GLOBAL ROUTINE DOES NUTHING BUT INCREMENT
```

```
                        ;*      'INTFLG' EACH TIME IT IS CALLED.  IT ASSUMES
                        ;*      THAT THE MAIN CALLING ROUTINE WILL  KNOW  WHAT
                        ;*      TO LOOK FOR.
                        ;;*****************************************************************
017632                  BGNSRV INTSRV                   ;GLOBAL INTERRUPT SERVICE ROUTINE
017632                  INTSRV:
                                                ;ADD 1 TO 'INTERRUPT OCCURED' FLAG
017632                                                      LET INTFLG := INTFLG + #1
017632  005237  017640                                              INC     INTFLG
017636                          ENDSRV                  ;THAT'S ALL
017636  000002                RTI
017640  000000          INTFLG:  0

                        .SBTTL ROUTINE FOR KDF11 AND 2 DLV11'S

                        ;;*****************************************************************
                        ;*      THIS ROUTINE WILL MAKE SURE THAT THE
                        ;*      PROPER DEVICES ARE THERE.
                        ;;*****************************************************************

017642                  ROUTINE KD2DLV                  ;KDF11-B AS CONSOLE AND 2
017642                                                          KD2DLV:
                                                        ;DLV11-J'S AT 176500
017642                  LET SFTREG := #1
017642  012737  000001  001316                                  MOV     #1,SFTREG
017650                  LET TMP9 := $BASE               ;BASE ADRS
017650  013737  001250  001326                                  MOV     $BASE,TMP9

017656                  INLINE <JSR     PC,REPEAT>              ;LOOK FOR DLV'S
017656  004737  020222                                          JSR     PC,REPEAT

017662                  IF $DEVM NE #377 THEN
017662  023727  001252  000377                                  CMP     $DEVM,#377
017670  001002                                                  BNE     .+6
017672  000137  017754                                          JMP     $50342
017676                      IFB $AUTOB EQ #0 THEN
017676  105737  001134                                          TSTB    $AUTOB
017702  001402                                                  BEQ     .+6
017704  000137  017752                                          JMP     $50343
017710                          TYPTXT      <<CRLF>! DEVICE MAP DOES NOT AGREE!>
017752                      ENDIF
017752                                                          $50343:
017752                  INLINE <BR  .>
017752  000777                                                  BR      .
017754                  ENDIF
017754                                                          $50342:

017754                  LET CONADR := #177560
017754  012737  177560  001274                                  MOV     #177560,CONADR

017762                  ENDRTN
017762                                                          $50340:
017762                                                          $50341:
017762  000207                                                  RTS     PC
                                .SBTTL ROUTINE FOR NO DLV11-J AND A KDF11-B
```

```
                        ;;*****************************************************************
                        ;*      THIS ROUTINE WILL CHECH THE PROPER ADDRESSES
                        ;*      FOR THE 2 SLU'S ON THE KDF11-B. NO DLV11-J'S
                        ;;*****************************************************************

017764                  ROUTINE ONKDF                                           ONKDF:

017764
017764                  LET TMP9 := #177560                     ;CONSOLE ADRS
017764  012737  177560  001326                                                  MOV     #177560,TMP9
017772
017772  013737  001326  001274  LET CONADR := TMP9                              MOV     TMP9,CONADR
020000
020000  005037  017640          LET INTFLG := #0                                CLR     INTFLG
020004
020004  017705  161316          LET R5 := @TMP9                                 MOV     @TMP9,R5
020010
020010  005737  017640          IF INTFLG NE #0 THEN                            TST     INTFLG
020014  001002                                                                  BNE     .+6
020016  000137  020062                                                          JMP     $50346
020022
020022  105737  001134                  IFB $AUTOB EQ #0 THEN                   TSTB    $AUTOB
020026  001402                                                                  BEQ     .+6
020030  000137  020060                                                          JMP     $50347
020034                              TYPTXT      <<CRLF>& NO CONSOLE &>
020060                          ENDIF
020060                                                                          $50347:
020060                      INLINE <BR  .>                                      BR      .
020060  000777
020062                  ENDIF                                                   $50346:
020062
020062                  LET TMP9 := $BASE                       ;176500
020062  013737  001250  001326                                                  MOV     $BASE,TMP9
020070
020070  005037  017640          LET INTFLG := #0                                CLR     INTFLG
020074
020074  017705  161226          LET R5 := @TMP9                                 MOV     @TMP9,R5
020100
020100  005737  017640          IF INTFLG EQ #0 THEN                            TST     INTFLG
020104  001402                                                                  BEQ     .+6
020106  000137  020124                                                          JMP     $50350
020112
020112  013737  001250  001302      LET SECSLU := $BASE         ;NO INTR, THE DEVICE IS THERE
020120                          ELSE                                            MOV     $BASE,SECSLU
020120  000137  020200                                                          JMP     $50351
020124                                                                          $50350:
020124
020124  105737  001134                  IFB $AUTOB EQ #0 THEN                   TSTB    $AUTOB
020130  001402                                                                  BEQ     .+6
020132  000137  020176                                                          JMP     $50352
020136                              TYPTXT      <<CRLF>& SECOND SLU   WRONG ADRS&>
020176                          ENDIF
020176                                                                          $50352:
020176                      INLINE <BR  .>                                      BR      .
020176  000777
020200                  ENDIF
```

```
                                                                                $50351:
                        LET $DEVM := #0                                         CLR     $DEVM
                        LET DLV211 := #1                ;SET THESE FOR PROPER   MOV     #1,DLV211
                        LET KDDLV := #1                 ;PRINT OUTS             MOV     #1,KDDLV

                        ENDRTN
                                                                                $50344:
                                                                                $50345:
                                                                                RTS     PC

                              .SBTTL  ROUTINE TO FIND DLV11-J'S

                        ;;*************************************************************
                        ;*      THIS ROUTINE WILL LOOK FOR THE DLV11-J BEING
                        ;*      IN THE PROPER ADRS LOCATIONS.
                        ;;*************************************************************

                        ROUTINE REPEAT
                                                                                REPEAT:

                        REPEAT
                                                                                $50355:
                            LET INTFLG := #0                                    CLR     INTFLG
                            LET R5 := @TMP9                                     MOV     @TMP9,R5
                            IF INTFLG EQ #0 THEN                                TST     INTFLG
                                                                                BEQ     .+6
                                                                                JMP     $50356
                                LET $DEVM := $DEVM SET.BY SFTREG    ;SET BIT IN DEVM
                                                                                BIS     SFTREG,$DEVM
                            ENDIF
                                                                                $50356:
                            LET SFTREG := SFTREG SHIFT 1            ;ELSE A ZERO
                                                                                ASL     SFTREG
                            LET TMP9 := TMP9 + #10
                                                                                ADD     #10,TMP9
                        UNTIL TMP9 EQ #176600
                                                                                CMP     TMP9,#176600
                                                                                BNE     $50355

                        ENDRTN
                                                                                $50353:
                                                                                $50354:
                                                                                RTS     PC

                              .SBTTL ROUTINE FOR KDF11 AND ONE DLV11-J

                        ;;*************************************************************
                        ;*      THIS ROUTINE IS USED WHEN THE CONSOLE IS ON
                        ;*      THE KDF11-B AND WE HAVE ONE DLV11-J MODULE.
                        ;;*************************************************************
```

```
4794
4795  020276                          ROUTINE KDDLVJ                                  KDDLVJ:
4796  020276
4797
4798  020276                          LET TMP9 := $BASE                               MOV     $BASE,TMP9
4799  020276  013737  001250  001326
4800  020304                          LET TMP9 := TMP9 + #40          ;17o540         ADD     #40,TMP9
4801  020304  062737  000040  001326
4802  020312                          LET SECSLU := TMP9                              MOV     TMP9,SECSLU
4803  020312  013737  001326  001302
4804  020320                          LET INTFLG := #0                ;MAKE SURE CH1 IS THERE
4805  020320  005037  017640                                                         CLR     INTFLG
4806  020324                          LET R5 := @TMP9                                 MOV     @TMP9,R5
4807  020324  017705  160776
4808
4809  020330                          IF INTFLG NE #0 THEN            ;IF 0 THEN NO INTERRUPT
4810  020330  005737  017640                                                         TST     INTFLG
4811  020334  001002                                                                 BNE     .+6
4812  020336  000137  020402                                                         JMP     $50361
4813  020342                                  IFB $AUTOB EQ #0 THEN
4814  020342  105737  001134                                                       TSTB    $AUTOB
4815  020346  001402                                                               BEQ     .+6
4816  020350  000137  020400                                                       JMP     $50362
4817  020354                                      TYPTXT  <<CRLF>& NO KDF11-B&>
4818  020400                                  ENDIF
4819  020400                                                                      $50362:
4820  020400                              INLINE <BR  .>
4821  020400  000777                                                             BR      .
4822  020402                          ENDIF                           ;SHOULD BE SOMETHING THERE
4823  020402                                                                      $50361:
4824
4825  020402                          LET SFTREG := #1
4826  020402  012737  000001  001316                                             MOV     #1,SFTREG
4827  020410                          LET TMP9 := $BASE               ;SET UP TMP9 TO 176500
4828  020410  013737  001250  001326                                             MOV     $BASE,TMP9
4829  020416                          INLINE <JSR     PC,REPEAT>      ;LOOK FOR OTHERS
4830  020416  004737  020222                                                     JSR     PC,REPEAT
4831
4832  020422                          LET $DEVM := $DEVM CLR.BY #BIT04
4833  020422  042737  000020  001252                                             BIC     #BIT04,$DEVM
4834  020430                          IF $DEVM NE #17 THEN            ;$DEVM MUST BE 17
4835  020430  023727  001252  000017                                             CMP     $DEVM,#17
4836  020436  001002                                                             BNE     .+6
4837  020440  000137  020522                                                     JMP     $50363
4838  020444                                  IFB $AUTOB EQ #0 THEN
4839  020444  105737  001134                                                   TSTB    $AUTOB
4840  020450  001402                                                           BEQ     .+6
4841  020452  000137  020520                                                   JMP     $50364
4842  020456                                      TYPTXT      <<CRLF>!DEVICE MAP DOES NOT AGREE!>
4843  020520                                  ENDIF
4844  020520                                                                  $50364:
4845  020520                              INLINE <BR  .>
4846  020520  000777                                                         BR      .
4847  020522                          ENDIF                                   $50363:
4848  020522
4849
```

```
                        LET CONADR := #177560
                                                                        MOV     #177560,CONADR
                        LET DLV211 := #1                        ;SET FOR PROPER PRINTING
                                                                        MOV     #1,DLV211

                        ENDRTN
                                                                $50357:
                                                                $50360:
                                                                        RTS     PC
                .SBTTL ROUTINE TO SET UP ADDRESS FOR TESTING


                ;;*****************************************************************
                ;*      THIS  ROUTINE  FINDS THE NEXT CORRECT
                ;*      ADDRESS  AND VECTOR OF A DEVICE TO BE TESTED.
                ;;*****************************************************************

                ROUTINE CYCLE
                                                                CYCLE:

                IF DLADD EQ #0 THEN
                                                                        TST     DLADD
                                                                        BEQ     .+6
                                                                        JMP     $50367
                    LET CONSOL := #1                            ;CONSOLE TESTING BIT
                                                                        MOV     #1,CONSOL
                    LET DLVEC := #TKVEC                 ;BEFORE TESTING SET IT
                                                                        MOV     #TKVEC,DLVEC
                    LET DLADD := $TKS                   ;DLADD EQ CONSOLE
                                                                        MOV     $TKS,DLADD
                    LET CONTST := #1                    ;TESTED THE CONSOLE
                                                                        MOV     #1,CONTST
                ELSE
                                                                        JMP     $50370
                                                                $50367:
                    LET CONSOL := #0                    ;CLR CONSOLE TESTING BIT
                                                                        CLR     CONSOL
                    IF DLADD EQ CONADR THEN
                                                                        CMP     DLADD,CONADR
                                                                        BEQ     .+6
                                                                        JMP     $50371
                        LET DLVEC := $VECT1             ;DLVEC EQ 300
                                                                        MOV     $VECT1,DLVEC
                        IF P1CNT EQ #2 THEN             ;SECSLU IS DESELECTED
                                                                        CMP     P1CNT,#2
                                                                        BEQ     .+6
                                                                        JMP     $50372
                            LET DLADD := $BASE          ;176500
                                                                        MOV     $BASE,DLADD
                            LET KDFTST := #1
                                                                        MOV     #1,KDFTST
                        ELSE
                                                                        JMP     $50373
                                                                $50372:
                            LET DLADD := SECSLU
                                                                        MOV     SECSLU,DLADD
```

```
020676                                      ENDIF
020676                                                                                 $50373:
020676                                      IF SECSLU EQ #176540 THEN
020676  023727  001302  176540                                                         CMP     SECSLU,#176540
020704  001402                                                                         BEQ     .+6
020706  000137  020720                                                                 JMP     $50374
020712                                          LET DLVEC := DLVEC + #40
020712  062737  000040  001260                                                         ADD     #40,DLVEC
020720                                      ENDIF
020720                                                                                 $50374:
020720                                      LET KDF11B := #1                           ;SECOND SLU BIT
020720  012737  000001  001322                                                         MOV     #1,KDF11B
020726                                  ELSE
020726  000137  020740                                                                 JMP     $50375
020732                                                                                 $50371:
020732                                      LET KDFTST := #1               ;THIS BIT IS
020732  012737  000001  001314                                                         MOV     #1,KDFTST
020740                                  ENDIF                          ;SET AFTER TESTING
020740                                                                                 $50375:

020740                              ENDIF
020740                                                                                 $50370:

020740                              IF CONTST EQ #1 AND KDFTST EQ #1 THEN          ;BOTH SLU'S ON KDF11B
020740  023727  001272  000001                                                         CMP     CONTST,#1
020746  001402                                                                         BEQ     .+6
020750  000137  021322                                                                 JMP     $50376
020754  023727  001314  000001                                                         CMP     KDFTST,#1
020762  001402                                                                         BEQ     .+6
020764  000137  021322                                                                 JMP     $50376
020770                                  IF $DEVM EQ #0 THEN                        ;HAVE BEEN TESTED
020770  005737  001252                                                                 TST     $DEVM
020774  001402                                                                         BEQ     .+6
020776  000137  021004                                                                 JMP     $50377
021002                                      INLINE <BR      ENDDER>
021002  000550                                                                         BR      ENDDER
021004                                  ENDIF
021004                                                                                 $50377:
021004                                  IF SECSLU EQ #176500 THEN
021004  023727  001302  176500                                                         CMP     SECSLU,#176500
021012  001402                                                                         BEQ     .+6
021014  000137  021140                                                                 JMP     $50400
021020                                      IF MASK EQ #0 THEN
021020  005737  001320                                                                 TST     MASK
021024  001402                                                                         BEQ     .+6
021026  000137  021074                                                                 JMP     $50401
021032                                          LET MASK := MASK SET.BY #BIT04
021032  052737  000020  001320                                                         BIS     #BIT04,MASK
021040                                          LET DLVEC := $VECT1 + #40
021040  013737  001244  001260                                                         MOV     $VECT1,DLVEC
021046  062737  000040  001260                                                         ADD     #40,DLVEC
021054                                          LET DLADD := SECSLU + #40
021054  013737  001302  001256                                                         MOV     SECSLU,DLADD
021062  062737  000040  001256                                                         ADD     #40,DLADD
021070                                      ELSE
021070  000137  021134                                                                 JMP     $50402
```

```
4962  021074
4963  021074
4964  021074  006337  001320                    LET MASK := MASK SHIFT 1
4965  021100
4966  021100  062737  000010  001260            LET DLVEC := DLVEC + #10
4967  021106
4968  021106  062737  000010  001256            LET DLADD := DLADD + #10
4969  021114
4970  021114  033737  001320  001252            IF MASK NOTSETIN $DEVM THEN
4971  021122  001402
4972  021124  000137  021134
4973  021130                                        INLINE <JMP     TEST20>
4974  021130  000137  010102
4975  021134                                    FNDIF
4976  021134
4977  021134                                ENDIF
4978  021134
4979  021134                            ELSE
4980  021134  000137  021322
4981  021140
4982  021140                                IF MASK EQ #0 THEN
4983  021140  005737  001320
4984  021144  001402
4985  021146  000137  021200                       LET MASK := MASK SET.BY #BIT00
4986  021152
4987  021152  052737  000001  001320               LET DLVEC := $VECT1
4988  021160
4989  021160  013737  001244  001260               LET DLADD := $BASE
4990  021166
4991  021166  013737  001250  001256           ELSE
4992  021174
4993  021174  000137  021322                       LET MASK := MASK SHIFT 1
4994  021200
4995  021200                                       LET DLVEC := DLVEC + #10
4996  021200  006337  001320
4997  021204                                       LET DLADD := DLADD + #10
4998  021204  062737  000010  001260
4999  021212                                       IF MASK EQ #20 AND SECSLU NE #0 THEN
5000  021212  062737  000010  001256
5001  021220
5002  021220  023727  001320  000020                   LET MASK := MASK SHIFT 4
5003  021226  001402
5004  021230  000137  021302                           LET DLVEC := DLVEC + #40
5005  021234  005737  001302
5006  021240  001002                                   LET DLADD := DLADD + #40
5007  021242  000137  021302
5008  021246                                       ENDIF
5009  021246  006337  001320
5010  021252  006337  001320
5011  021256  006337  001320
5012  021262  006337  001320
5013  021266
5014  021266  062737  000040  001260
5015  021274
5016  021274  062737  000040  001256
5017  021302
```

```
                        $50401:
                ;NEXT DEVM AND ADDRESS
                        ASL     MASK
                        ADD     #10,DLVEC
                        ADD     #10,DLADD
                        BIT     MASK,$DEVM
                        BEQ     .+6
                        JMP     $50403
                        JMP     TEST20
                        $50403:
                        $50402:
                ;SECSLU MUST BE 176540 OR ZERO
                        JMP     $50404
                        $50400:
                        TST     MASK
                        BEQ     .+6
                        JMP     $50405
                        BIS     #BIT00,MASK
                ;START AT 300
                        MOV     $VECT1,DLVEC
                ;START AT 176500
                        MOV     $BASE,DLADD
                        JMP     $50406
                        $50405:
                        ASL     MASK
                        ADD     #10,DLVEC
                ;NXT DEVM AND ADDRESS
                        ADD     #10,DLADD
                        CMP     MASK,#20
                        BEQ     .+6
                        JMP     $50407
                        TST     SECSLU
                        BNE     .+6
                        JMP     $50407
                ;BYPASS CONSOLE
                        ASL     MASK
                        ASL     MASK
                        ASL     MASK
                        ASL     MASK
                        ADD     #40,DLVEC
                        ADD     #40,DLADD
```

```
5018  021302                                                                    $50407:
5019  021302                        IF MASK NOTSETIN $DEVM THEN                 ;ANY MORE DEVICES ?
5020  021302  033737  001320  001252                                            BIT     MASK,$DEVM
5021  021310  001402                                                            BEQ     .+6
5022  021312  000137  021322                                                    JMP     $50410
5023  021316                            INLINE <JMP     TEST20>
5024  021316  000137  010102                                                    JMP     TEST20
5025  021322                        ENDIF
5026  021322                                                                    $50410:
5027  021322                    ENDIF                                           $50406:
5028  021322                                                                    $50404:
5029  021322                ENDIF
5030  021322                                                                    $50376:
5031  021322            ENDIF
5032  021322
5033
5034  021322            ENDRTN                                                  $50365:
5035  021322                                                                    $50366:
5036  021322
5037  021322  000207                                                            RTS     PC
5038
5039                    ;*************************************************************
5040                    ;       THIS IS THE ROUTINE CALLED ENDDER FROM ABOVE
5041                    ;*************************************************************
5042
5043  021324            ROUTINE ENDDER                                          ENDDER:
5044  021324
5045
5046  021324            LET KDFTST := #0                                        CLR     KDFTST
5047  021324  005037  001314
5048  021330            LET KDF11B := #0                                        CLR     KDF11B
5049  021330  005037  001322
5050  021334            LET DLADD := CONADR                                     MOV     CONADR,DLADD
5051  021334  013737  001274  001256
5052  021342            IF #BIT12 SETIN @SWR THEN                               BIT     #BIT12,@SWR
5053  021342  032777  010000  157570                                            BNE     .+6
5054  021350  001002                                                           JMP     $50413
5055  021352  000137  021416
5056  021356                        IFB $AUTOB EQ #0 THEN                       TSTB    $AUTOB
5057  021356  105737  001134                                                   BEQ     .+6
5058  021362  001402                                                           JMP     $50414
5059  021364  000137  021416
5060  021370                            TYPTXT      <<CRLF>*ERRORS: *>
5061  021410                            TYPDEC      $ERTTL
5062  021416                        ENDIF
5063  021416                                                                    $50414:
5064  021416            ENDIF                                                   $50413:
5065  021416
5066
5067  021416            LET MASK := #0                                          CLR     MASK
5068  021416  005037  001320
5069  021422            LET $ERTTL := #0                                        CLR     $ERTTL
5070  021422  005037  001112
5071  021426            INLINE <BR      $EOP>                   ;SET UP FOR MORE PASSES
5072  021426  000476                                                           BR      $EOP
5073                                                  ;GO TO END OF PASS ROUTINE
```

ROUTINE TO SET UP ADDRESS FOR TESTING SEQ 0100

```
5074                                                    ;AND THEN TO LOOP. WE WILL ONLY
5075                                                    ;TEST THE CONSOLE ON THE FIRST PASS
5076
5077  021430                    INLINE <HALT>
5078  021430  000000                                                HALT
5079  021432                    ENDRTN
5080  021432                                                        $50411:
5081  021432                                                        $50412:
5082  021432  000207                                                RTS     PC
```

```
5083
5084  021434                                                        ROUTINE MYTYPE
5085  021434                                                                        MYTYPE:
5086  021434                                    TYPTXT  <<CRLF>*TEST # *>
5087  021454                                    TYPOCT  $TESTN
5088  021462                                    TYPTXT  <*,ERROR # *>
5089  021502  113737  001114  001176            MOVB    $ITEMB,$FATAL   ; APT FATAL ERROR NUMBER
5090  021510                                    TYPOCS  $FATAL
5091  021520                                    TYPTXT  <*,PC = *>
5092  021536                                    TYPOCT  $ERRPC
5093  021544                                    TYPTXT  <*,CSR: *>
5094  021562                                    TYPOCT  DLADD
5095  021570                                    TYPTXT  <*,VECTOR: *>
5096  021610                                    TYPOCT  DLVEC
5097  021616  104401  001171                    TYPE    ,$CRLF
5098  021622                                                        ENDRTN
5099  021622                                                                        $50415:
5100  021622                                                                        $50416:
5101  021622  000207                                                                RTS     PC
5102
```

```
                                .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 LOOP

021624                          $EOP:
021624  000004                          SCOPE
021626  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
021632  005037  001160                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
021636  005237  001202                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
021642  042737  100000  001202          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
021650  005327                          DEC     (PC)+           ;;LOOP?
021652  000001          $EOPCT: .WORD   1
021654  003022                  BGT     $DOAGN          ;;YES
021656  012737                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
021660  000001          $ENDCT: .WORD   1
021662  021652                  $EOPCT
021664  104401  021731          TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
021670  013746  001202          MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
021674  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
021676  104401  021726          TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
021702  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
021706  001405                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
021710  000005                  RESET                   ;;CLEAR THE WORLD
021712  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
021714  000240                  NOP                     ;;SAVE ROOM
021716  000240                  NOP                     ;;FOR
021720  000240                  NOP                     ;;ACT11
021722                          $DOAGN:
021722  000137                  JMP     @(PC)+          ;;RETURN
021724  002000          $RTNAD: .WORD   LOOP
021726     377     377     000  $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
021731     015  042412  042116  $ENDMG: .ASCIZ  <15><12>/END PASS #/
021736  050040  051501  020123
021744  000043
```

```
                                        .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
                                        ;*

021746  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
021752  100002                          BPL     1$              ;;BR IF YES
021754  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
021756  000430                          BR      3$              ;;LEAVE
021760  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
021762  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
021766  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
021774  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
021776  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
022004  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
022006  010037  022016                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
022012  004737  023074                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
022016  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
022020  132737  000040  001215  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
022026  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
022030  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
022032  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
022034  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
022036  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
022040  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
022044  000002                          RTI                     ;;RETURN
022046  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
022052  001430                          BEQ     8$
022054  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
022060  001006                          BNE     5$
022062  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
022064  104401                          TYPE                    ;;TYPE A CR AND LF
022066  001171                          $CRLF
022070  105037  022276                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
022074  000755                          BR      2$              ;;GET NEXT CHARACTER
022076  004737  022160          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
022102  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
022106  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
022110  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                        ;;AND THE NULL CHAR.
022114  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
022120  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
022122  004737  022160                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
022126  105337  022276                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
5198  022132  000770                        BR      7$              ;;LOOP
5199
5200                                  ;HORIZONTAL TAB PROCESSOR
5201
5202  022134  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
5203  022140  004737  022160          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
5204  022144  132737  000007  022276        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
5205  022152  001372                        BNE     9$              ;;TAB STOP
5206  022154  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
5207  022156  000724                        BR      2$              ;;GET NEXT CHARACTER
5208  022160                        $TYPEC:
5209  022160  105777  156760                TSTB    @$TKS           ;;CHAR IN KYBD BUFFER?          ;MJD001
5210  022164  100022                        BPL     10$             ;;BR IF NOT                     ;MJD001
5211  022166  017746  156754                MOV     @$TKB,-(SP)     ;;GET CHAR                      ;MJD001
5212  022172  042716  177600                BIC     #177600,(SP)    ;;STRIP EXTRANEOUS BITS         ;MJD001
5213  022176  122716  000023                CMPB    #$XOFF,(SP)     ;;WAS CHAR XOFF                 ;MJD001
5214  022202  001012                        BNE     102$            ;;BR IF NOT                     ;MJD001
5215  022204                        101$:                                                           ;MJD001
5216  022204  105777  156734                TSTB    @$TKS           ;;WAIT FOR CHAR                 ;MJD001
5217  022210  100375                        BPL     101$                                            ;MJD001
5218  022212  117716  156730                MOVB    @$TKB,(SP)      ;;GET CHAR                      ;MJD001
5219  022216  042716  177600                BIC     #177600,(SP)    ;;STRIP IT                      ;MJD001
5220  022222  122716  000021                CMPB    #$XON,(SP)      ;;WAS IT XON?                   ;MJD001
5221  022226  001366                        BNE     101$            ;;BR IF NOT                     ;MJD001
5222  022230                        102$:                                                           ;MJD001
5223  022230  005726                        TST     (SP)+           ;;FIX STACK                     ;MJD001
5224  022232                        10$:                                                            ;MJD001
5225  022232  105777  156712                TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
5226  022236  100375                        BPL     10$                                             ;MJD001
5227  022240  116677  000002  156704        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
5228  022246  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
5229  022254  001003                        BNE     1$              ;;BRANCH IF NO
5230  022256  105037  022276                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
5231  022262  000406                        BR      $TYPEX          ;;EXIT
5232  022264  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
5233  022272  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
5234  022274  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
5235  022276  000000          $CHARCNT:.WORD  0                       ;;CHARACTER COUNT STORAGE
5236  022300  000207          $TYPEX: RTS     PC
5237
```

```
                                        .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.
022302  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
022310  001074                         BNE     15$             ;;BRANCH IF NO
022312  105777  156626                 TSTB    @$TKS           ;;CHAR THERE?
022316  100071                         BPL     15$             ;;IF NO, DON'T WAIT AROUND
022320  117746  156622                 MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
022324  042716  177600                 BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
022330  022726  000007                 CMP     #7,(SP)+        ;;IS IT A CONTROL G?
022334  001062                         BNE     15$             ;;NO, RETURN TO USER
022336  123727  001134  000001         CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
022344  001456                         BEQ     15$             ;;BRANCH IF YES

022346  104401  023037                 TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
022352  104401  023044         $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
022356  013746  000176                 MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
022362  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
022364  104401  023055                 TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
022370  005046                 19$:    CLR     -(SP)           ;;CLEAR COUNTER
022372  005046                         CLR     -(SP)           ;;THE NEW SWR
022374  105777  156544         7$:     TSTB    @$TKS           ;;CHAR THERE?
022400  100375                         BPL     7$              ;;IF NOT TRY AGAIN

022402  117746  156540                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
022406  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII



022412  021627  000025         9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
022416  001005                         BNE     10$             ;;BRANCH IF NOT
022420  104401  023032                 TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
022424  062706  000006         20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
022430  000757                         BR      19$             ;;LET'S TRY IT AGAIN

022432  021627  000015         10$:    CMP     (SP),#15        ;;IS IT A <CR>?
022436  001022                         BNE     16$             ;;BRANCH IF NO
022440  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
022444  001403                         BEQ     11$             ;;BRANCH IF YES
022446  016677  000002  156464         MOV     2(SP),@SWR      ;;SAVE NEW SWR
022454  062706  000006         11$:    ADD     #6,SP           ;;CLEAR UP STACK
022460  104401  001171         14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
022464  123727  001135  000001         CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
022472  001003                         BNE     15$             ;;BRANCH IF NOT
022474  012777  000100  156442         MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
022502  000002                 15$:    RTI                     ;;RETURN
022504  004737  022160         16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
022510  021627  000060                 CMP     (SP),#60        ;;CHAR < 0?
```

```
5294  022514  002420                        BLT     18$             ;;BRANCH IF YES
5295  022516  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
5296  022522  003015                        BGT     18$             ;;BRANCH IF YES
5297  022524  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
5298  022530  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
5299  022534  001403                        BEQ     17$             ;;BRANCH IF YES
5300  022536  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
5301  022540  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
5302  022542  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
5303  022544  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
5304  022550  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
5305  022554  000707                        BR      7$              ;;GET THE NEXT ONE
5306  022556  104401  001170        18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
5307  022562  000720                        BR      20$             ;;SIMULATE CONTROL-U
5308                                .DSABL  LSB
5309
5310
5311                                ;;*****************************************************************
5312                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
5313                                ;*CALL:
5314                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
5315                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
5316                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
5317                                ;
5318
5319  022564  011646                $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
5320  022566  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
5321  022574  105777  156344        1$:     TSTB    @$TKS           ;;WAIT FOR
5322  022600  100375                        BPL     1$              ;;A CHARACTER
5323  022602  117766  156340  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
5324  022610  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
5325  022616  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
5326  022624  001013                        BNE     3$              ;;BRANCH IF NO
5327  022626  105777  156312        2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
5328  022632  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
5329  022634  117746  156306                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
5330  022640  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
5331  022644  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
5332  022650  001366                        BNE     2$              ;;IF NOT DISCARD IT
5333  022652  000750                        BR      1$              ;;YES, RESUME
5334  022654  026627  000004  000021  3$:   CMP     4(SP),#$XON     ;;IS IT A RANDOM XON?                   ;RAN001
5335  022662  001744                        BEQ     1$              ;;BRANCH IF YES                         ;RAN001
5336  022664  026627  000004  000140        CMP     4(SP),#140      ;;IS IT UPPER CASE?
5337  022672  002407                        BLT     4$              ;;BRANCH IF YES
5338  022674  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
5339  022702  003003                        BGT     4$              ;;BRANCH IF YES
5340  022704  042766  000040  000004        BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
5341  022712  000002                4$:     RTI                     ;;GO BACK TO USER
5342                                ;;*****************************************************************
5343                                ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
5344                                ;*CALL:
5345                                ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
5346                                ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
5347                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
5348
5349  022714  010346                $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
```

```
 5350  022716  012703  023022            1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
 5351  022722  022703  023032            2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
 5352  022726  101405                            BLOS    4$              ;;BR IF YES
 5353  022730  104410                            RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
 5354  022732  112613                            MOVB    (SP)+,(R3)      ;;GET CHARACTER
 5355  022734  122713  000177            10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
 5356  022740  001003                            BNE     3$              ;;SKIP IF NOT
 5357  022742  104401  001170            4$:     TYPE    ,$QUES          ;;TYPE A '?'
 5358  022746  000763                            BR      1$              ;;CLEAR THE BUFFER AND LOOP
 5359  022750  111337  023020            3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
 5360  022754  104401  023020                    TYPE    ,9$
 5361  022760  122723  000015                    CMPB    #15,(R3)+       ;;CHECK FOR RETURN
 5362  022764  001356                            BNE     2$              ;;LOOP IF NOT RETURN
 5363  022766  105063  177777                    CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 5364  022772  104401  001172                    TYPE    ,$LF            ;;TYPE A LINE FEED
 5365  022776  012603                            MOV     (SP)+,R3        ;;RESTORE R3
 5366  023000  011646                            MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 5367  023002  016666  000004  000002            MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 5368  023010  012766  023022  000004            MOV     #$TTYIN,4(SP)
 5369  023016  000002                            RTI                     ;;RETURN
 5370  023020     000            9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
 5371  023021     000                    .BYTE   0                       ;;TERMINATOR
 5372  023022  000010            $TTYIN: .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
 5373  023032  052536  005015     000    $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
 5374  023037     136  006507  000012    $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
 5375  023044  005015  053523  020122    $MSWR:  .ASCIZ  <15><12>/SWR = /
 5376  023052  020075     000
 5377  023055     040  047040  053505    $MNEW:  .ASCIZ  /  NEW = /
 5378  023062  036440  000040
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;;*****************************************************************
023066  112737  000001  023332  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
023074  112737  000001  023330  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
023102  000403                  $ATY4:  BR      $ATYC
023104  112737  000001  023332          MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
023112                          $ATYC:
023112  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
023114  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
023116  105737  023330                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
023122  001450                          BEQ     5$              ;;IF NOT:  BR
023124  122737  000001  001214          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
023132  001031                          BNE     3$              ;;IF NOT:  BR
023134  132737  000100  001215          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
023142  001425                          BEQ     3$              ;;IF NOT:  BR
023144  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
023150  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
023156  005737  001174          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
023162  001375                          BNE     1$              ;;IF NOT:  WAIT
023164  010037  001210                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
023170  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
023172  001376                          BNE     2$
023174  163700  001210                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
023200  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
023202  010037  001212                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
023206  012737  000004  001174          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
023214  000413                          BR      5$
023216  017637  000004  023242  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
023224  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
023232  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
023236  004737  021746                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
023242  000000                  4$:     .WORD   0
023244                          5$:
023244  105737  023332          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
023250  001416                          BEQ     12$             ;;IF NOT:  BR
023252  005737  001214                  TST     $ENV            ;;RUNNING UNDER APT?
023256  001413                          BEQ     12$             ;;IF NOT:  BR
023260  005737  001174          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
023264  001375                          BNE     11$             ;;IF NOT:  WAIT
023266  017637  000004  001176          MOV     @4(SP),$FATAL   ;;GET ERROR #
023274  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
023302  005237  001174                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
023306  105037  023332          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
023312  105037  023331                  CLRB    $LFLG           ;;CLEAR LOG FLAG
023316  105037  023330                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
023322  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
023324  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
023326  000207                          RTS     PC              ;;RETURN
023330     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
023331     000          $LFLG:  .BYTE   0               ;;LOG FLAG
023332     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
023334                          .EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
  5435          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

023334                                  $ERROR:
023334  104407                                  CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
023336  105237  001103                  7$:     INCB    $ERRFLG                 ;;SET THE ERROR FLAG
023342  001775                                  BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
023344  013777  001102  155570                  MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER AND ERROR FLAG
023352  032777  002000  155560                  BIT     #BIT10,@SWR             ;;BELL ON ERROR?
023360  001402                                  BEQ     1$                      ;;NO - SKIP
023362  104401  001164                          TYPE    ,$BELL                  ;;RING BELL
023366  005237  001112                  1$:     INC     $ERTTL                  ;;COUNT THE NUMBER OF ERRORS
023372  011637  001116                          MOV     (SP),$ERRPC             ;;GET ADDRESS OF ERROR INSTRUCTION
023376  162737  000002  001116                  SUB     #2,$ERRPC
023404  117737  155506  001114                  MOVB    @$ERRPC,$ITEMB          ;;STRIP AND SAVE THE ERROR ITEM CODE
023412  032777  020000  155520                  BIT     #BIT13,@SWR             ;;SKIP TYPEOUT IF SET
023420  001004                                  BNE     20$                     ;;SKIP TYPEOUTS
023422  004737  021434                          JSR     PC,MYTYPE               ;;GO TO USER ERROR ROUTINE
023426  104401  001171                          TYPE    ,$CRLF
023432                                  20$:
023432  122737  000001  001214                  CMPB    #APTENV,$ENV            ;;RUNNING IN APT MODE
023440  001007                                  BNE     2$                      ;;NO,SKIP APT ERROR REPORT
023442  113737  001114  023454                  MOVB    $ITEMB,21$              ;;SET ITEM NUMBER AS ERROR NUMBER
023450  004737  023104                          JSR     PC,$ATY4                ;;REPORT FATAL ERROR TO APT
023454     000                  21$:    .BYTE   0
023455     000                          .BYTE   0
023456  000777                  22$:    BR      22$                     ;;APT ERROR LOOP
023460  005777  155454          2$:     TST     @SWR                    ;;HALT ON ERROR
023464  100002                          BPL     3$                      ;;SKIP IF CONTINUE
023466  000000                          HALT                            ;;HALT ON ERROR!
023470  104407                          CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
023472  032777  001000  155440  3$:     BIT     #BIT09,@SWR             ;;LOOP ON ERROR SWITCH SET?
023500  001402                          BEQ     4$                      ;;BR IF NO
023502  013716  001110                  MOV     $LPERR,(SP)             ;;FUDGE RETURN FOR LOOPING
023506  005737  001162          4$:     TST     $ESCAPE                 ;;CHECK FOR AN ESCAPE ADDRESS
023512  001402                          BEQ     5$                      ;;BR IF NONE
023514  013716  001162                  MOV     $ESCAPE,(SP)            ;;FUDGE RETURN ADDRESS FOR ESCAPE
023520                                  5$:
023520  022737  021712  000042                  CMP     #$ENDAD,@#42            ;;ACT-11 AUTO-ACCEPT?
023526  001001                                  BNE     6$                      ;;BRANCH IF NO
023530  000000                                  HALT                            ;;YES
023532                                  6$:
023532  000002                                  RTI                             ;;RETURN
```

```
                                        .SBTTL  SCOPE HANDLER ROUTINE

                                        ;;*****************************************************************
                                        ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                        ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                        ;*AND LOAD THE ERROR FLAG ($ERRFLG) 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

023534                                  $SCOPE:
023534  104407                                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
023536  032777  040000  155374          1$:    BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
023544  001114                                 BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
023546  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
023550  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
023554  012737  023574  000004                 MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
023562  005737  177060                         TST     @#177060        ;;TIME OUT ON XOR?
023566  012637  000004                         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023572  000463                                 BR      $SVLAD          ;;GO TO THE NEXT TEST
023574  022626                          5$:    CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
023576  012637  000004                         MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
023602  000423                                 BR      7$              ;;LOOP ON THE PRESENT TEST
023604                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
023604  032777  000400  155326                 BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
023612  001404                                 BEQ     2$              ;;BR IF NO
023614  127737  155320  001102                 CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
023622  001465                                 BEQ     $OVER           ;;BR IF YES
023624  105737  001103          2$:    TSTB    $ERRFLG         ;;HAS AN ERROR OCCURRED?
023630  001421                                 BEQ     3$              ;;BR IF NO
023632  123737  001115  001103                 CMPB    $ERMAX,$ERRFLG  ;;MAX. ERRORS FOR THIS TEST OCCURRED?
023640  101015                                 BHI     3$              ;;BR IF NO
023642  032777  001000  155270                 BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
023650  001404                                 BEQ     4$              ;;BR IF NO
023652  013737  001110  001106          7$:    MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
023660  000446                                 BR      $OVER
023662  105037  001103          4$:    CLRB    $ERRFLG         ;;ZERO THE ERROR FLAG
023666  005037  001160                         CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
023672  000415                                 BR      1$              ;;ESCAPE TO THE NEXT TEST
023674  032777  004000  155236          3$:    BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
023702  001011                                 BNE     1$              ;;BR IF YES
023704  005737  001202                         TST     $PASS           ;;IF FIRST PASS OF PROGRAM
023710  001406                                 BEQ     1$              ;;      INHIBIT ITERATIONS
023712  005237  001104                         INC     $ICNT           ;;INCREMENT ITERATION COUNT
023716  023737  001160  001104                 CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
023724  002024                                 BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
023726  012737  000001  001104          1$:    MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
023734  013737  024012  001160                 MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
023742  105237  001102                  $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
023746  113737  001102  001200                 MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
```

```
 5546  023754  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
 5547  023760  011637  001110                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
 5548  023764  005037  001162                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
 5549  023770  112737  000001  001115          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
 5550  023776  013777  001102  155136 $OVER:   MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
 5551  024004  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
 5552  024010  000002                          RTI                     ;;FIXES PS
 5553  024012  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
                                        $TYPDS:
024014                                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
024014  010046                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024016  010146                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
024020  010246                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
024022  010346                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
024024  010546                                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
024026  012746  020200                         MOV     20(SP),R5       ;;GET THE INPUT NUMBER
024032  016605  000020                         BPL     1$              ;;BR IF INPUT IS POS.
024036  100004                                  NEG     R5              ;;MAKE THE BINARY NUMBER POS.
024040  005405                                  MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
024042  112766  000055  000001                 CLR     R0              ;;ZERO THE CONSTANTS INDEX
024050  005000                          1$:     MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
024052  012703  024230                         MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
024056  112723  000040                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
024062  005002                                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
024064  016001  024220                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
024070  160105                                  BLT     4$              ;;BR IF DONE
024072  002402                                  INC     R2              ;;INCREASE THE BCD DIGIT BY 1
024074  005202                                  BR      3$
024076  000774                          4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
024100  060105                                  TST     R2              ;;CHECK IF BCD DIGIT=0
024102  005702                                  BNE     5$              ;;FALL THROUGH IF 0
024104  001002                                  TSTB    (SP)            ;;STILL DOING LEADING 0'S?
024106  105716                                  BMI     7$              ;;BR IF YES
024110  100407                          5$:     ASLB    (SP)            ;;MSD?
024112  106316                                  BCC     6$              ;;BR IF NO
024114  103003                                  MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
024116  116663  000001  177777         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
024124  052702  000060                 7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
024130  052702  000040                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
024134  110223                                  TST     (R0)+           ;;JUST INCREMENTING
024136  005720                                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
024140  020027  000010                         BLT     2$              ;;GO DO THE NEXT DIGIT
024144  002746                                  BGT     8$              ;;GO TO EXIT
024146  003002                                  MOV     R5,R2           ;;GET THE LSD
024150  010502                                  BR      6$              ;;GO CHANGE TO ASCII
024152  000764                          8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
024154  105726                                  BPL     9$              ;;BR IF NO
024156  100003                                  MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
024160  116663  177777  177776                 CLRB    (R3)            ;;SET THE TERMINATOR
024166  105013                          9$:     MOV     (SP)+,R5        ;;POP STACK INTO R5
024170  012605                                  MOV     (SP)+,R3        ;;POP STACK INTO R3
024172  012603                                  MOV     (SP)+,R2        ;;POP STACK INTO R2
024174  012602
```

```
 5610  024176  012601                         MOV     (SP)+,R1        ;;POP STACK INTO R1
 5611  024200  012600                         MOV     (SP)+,R0        ;;POP STACK INTO R0
 5612  024202  104401  024230                 TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 5613  024206  016666  000002  000004         MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 5614  024214  012616                         MOV     (SP)+,(SP)
 5615  024216  000002                         RTI                     ;;RETURN TO USER
 5616  024220  023420                 $DTBL:  10000.
 5617  024222  001750                         1000.
 5618  024224  000144                         100.
 5619  024226  000012                         10.
 5620  024230  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

024240  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
024244  116637  000001  024463         MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
024252  112637  024465                 MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
024256  062716  000002                 ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
024262  000406                         BR      $TYPON
024264  112737  000001  024463 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
024272  112737  000006  024465         MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
024300  112737  000005  024462 $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
024306  010346                         MOV     R3,-(SP)        ;;SAVE R3
024310  010446                         MOV     R4,-(SP)        ;;SAVE R4
024312  010546                         MOV     R5,-(SP)        ;;SAVE R5
024314  113704  024465                 MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
024320  005404                         NEG     R4
024322  062704  000006                 ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
024326  110437  024464                 MOVB    R4,$OMODE       ;;SAVE IT FOR USE
024332  113704  024463                 MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
024336  016605  000012                 MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
024342  005003                         CLR     R3              ;;CLEAR THE OUTPUT WORD
024344  006105                 1$:     ROL     R5              ;;ROTATE MSB INTO "C"
024346  000404                         BR      3$              ;;GO DO MSB
024350  006105                 2$:     ROL     R5              ;;FORM THIS DIGIT
024352  006105                         ROL     R5
024354  006105                         ROL     R5
024356  010503                         MOV     R5,R3
024360  006103                 3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
024362  105337  024464                 DECB    $OMODE          ;;TYPE THIS DIGIT?
024366  100016                         BPL     7$              ;;BR IF NO
024370  042703  177770                 BIC     #177770,R3      ;;GET RID OF JUNK
024374  001002                         BNE     4$              ;;TEST FOR 0
024376  005704                         TST     R4              ;;SUPPRESS THIS 0?
024400  001403                         BEQ     5$              ;;BR IF YES
```

```
5677  024402  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5678  024404  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5679  024410  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5680  024414  110337  024460                  MOVB    R3,8$           ;;SAVE FOR TYPING
5681  024420  104401  024460                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
5682  024424  105337  024462          7$:     DECB    $OCNT           ;;COUNT BY 1
5683  024430  003347                          BGT     2$              ;;BR IF MORE TO DO
5684  024432  002402                          BLT     6$              ;;BR IF DONE
5685  024434  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5686  024436  000744                          BR      2$              ;;GO DO THE LAST DIGIT
5687  024440  012605          6$:             MOV     (SP)+,R5        ;;RESTORE R5
5688  024442  012604                          MOV     (SP)+,R4        ;;RESTORE R4
5689  024444  012603                          MOV     (SP)+,R3        ;;RESTORE R3
5690  024446  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5691  024454  012616                          MOV     (SP)+,(SP)
5692  024456  000002                          RTI                     ;;RETURN
5693  024460     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5694  024461     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5695  024462     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
5696  024463     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
5697  024464  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.

024466  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
024470  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
024474  005740                          TST     -(R0)           ;;BACKUP BY 2
024476  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
024500  006300                          ASL     R0              ;;POSITION FOR INDEXING
024502  016000  024522                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
024506  000200                          RTS     R0              ;;GO TO ROUTINE

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

024510  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
024512  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
024520  000002                          RTI                     ;;RESTORE THE PSW

                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
                                        ;       -------
024522  024510                  $TRPAD: .WORD   $TRAP2
024524  021746                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
024526  024264                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
024530  024240                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
024532  024300                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
024534  024014                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

024536  022352                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

024540  022302                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
024542  022564                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
024544  022714                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE

                                        ;MAXMEM FOR APT = 17400
                                        ;MAXMEM NON APT = 17500  (ABS LOADER)
        000001                  .END
```

```
ABASE = 176500
ACDW1 = 000000
ACDW2 = 000000
ACPUOP= 000000
ACTCH   016120
ADDW0 = 000000
ADDW1 = 000000
ADDW10= 000000
ADDW11= 000000
ADDW12= 000000
ADDW13= 000000
ADDW14= 000000
ADDW15= 000000
ADDW2 = 000000
ADDW3 = 000000
ADDW4 = 000000
ADDW5 = 000000
ADDW6 = 000000
ADDW7 = 000000
ADDW8 = 000000
ADDW9 = 000000
ADEVCT= 000000
ADEVM = 000000
ADRS  =%000001
AENV  = 000000
AENVM = 000000
AFATAL= 000000
AMADR1= 000000
AMADR2= 000000
AMADR3= 000000
AMADR4= 000000
AMAMS1= 000000
AMAMS2= 000000
AMAMS3= 000000
AMAMS4= 000000
AMSGAD= 000000
AMSGLG= 000000
AMSGTY= 000000
AMTYP1= 000000
AMTYP2= 000000
AMTYP3= 000000
AMTYP4= 000000
APASS = 000000
APRIOR= 000000
APTCSU= 000040
APTENV= 000001
APTSIZ= 000200
APTSPO= 000100
ASSEMB= 000010
ASWREG= 000000
ATESTN= 000000
AUNIT = 000000
AUSWR = 000031
AVECT1= 000300
AVECT2= 000000
BIT0  = 000001
BIT00 = 000001
BIT01 = 000002
BIT02 = 000004
BIT03 = 000010
BIT04 = 000020
BIT05 = 000040
BIT06 = 000100
BIT07 = 000200
BIT08 = 000400
BIT09 = 001000
BIT1  = 000002
BIT10 = 002000
BIT11 = 004000
BIT12 = 010000
BIT13 = 020000
BIT14 = 040000
BIT15 = 100000
BIT2  = 000004
BIT3  = 000010
BIT4  = 000020
BIT5  = 000040
BIT6  = 000100
BIT7  = 000200
BIT8  = 000400
BIT9  = 001000
BPTVEC= 000014
CHCNT   011514
CHCTR   016116
CHMASK  011512
CHMSK   016114
CHOTAB  014114
CH1TAB  014314
CH2TAB  014514
CH3TAB  014714
CH4TAB  015114
CH5TAB  015314
CH6TAB  015514
CH7TAB  015714
CKSWR = 104407
CONADR  001274
CONSOL  001324
CONTST  001272
CR    = 000015
CRLF  = 000200
CYCLE   020540
DATA    007042
DDISP = 177570
DIAGMC= 000000
DISPLA  001142
DISPRE  000174
DLADD   001256
DLBASE  001312
DLVEC   001260
DLV211  001310
DSWR  = 177570
EMTVEC= 000030
ENDDER  021324
ERFLG = 000400
ERRCHK  010100
ERRCNT  006772
ERRORF  003604
ERRVEC= 000004
ERWRD   016122
FSR     005010
ESR1    007552
EXITFL  003606
FLAG    016516
GTSWR = 104406
HOLDSC  016517
HOWLON= 000000
HT    = 000011
I       001254
INTFLG  017640
INTRTA  011452
INTSRV  017632
IOTVEC= 000020
KDDLV   001306
KDDLVJ  020276
KDFTST  001314
KDF11B  001322
KD2DLV  017642
LF    = 000012
LOOP    002000
MASK    001320
MODTST  010162
MYTYPE  021434
NODLV   001304
NUMBER  006774
ONKDF   017764
PASS    003602
PHASE2  001276
PIRQ  = 177772
PIRQVE= 000240
PRIERR  011516
PR0   = 000000
PR1   = 000040
PR2   = 000100
PR3   = 000140
PR4   = 000200
PR5   = 000240
PR6   = 000300
PR7   = 000340
PS    = 177776
PSW   = 177776
PWRVEC= 000024
P1CNT   001300
RBUF    001264
RCH0    011412
RCH1    011414
RCH2    011416
RCH3    011420
RCH4    011422
RCH5    011424
RCH6    011426
RCH7    011430
RCSR    001262
RCVR0S  010612
RCVR1S  010672
RCVR2S  010752
RCVR3S  011032
RCVR4S  011112
RCVR5S  011172
RCVR6S  011252
RCVR7S  011332
RDCHR = 104410
RDLIN = 104411
REC     007044
REG   = 000004
REGSAV  016512
REPEAT  020222
RESVEC= 000010
RHLD    007164
R0SRV   012534
R1SRV   012672
R2SRV   013030
R3SRV   013166
R4SRV   013324
R5SRV   013462
R6    =%000006
R6SRV   013620
R7    =%000007
R7SRV   013756
SECSLU  001302
SEREND  014114
SETCLR= 000006
SFTREG  001316
SIZE    016614
SRVEND  011412
STACK = 001100
START   001330
STATEN  016114
STKLMT= 177774
SWR     001140
SWREG   000176
SW0   = 000001
SW00  = 000001
SW01  = 000002
SW02  = 000004
SW03  = 000010
SW04  = 000020
SW05  = 000040
SW06  = 000100
SW07  = 000200
SW08  = 000400
SW09  = 001000
SW1   = 000002
SW10  = 002000
SW11  = 004000
SW12  = 010000
SW13  = 020000
SW14  = 040000
SW15  = 100000
SW2   = 000004
SW3   = 000010
SW4   = 000020
SW5   = 000040
SW6   = 000100
SW7   = 000200
SW8   = 000400
SW9   = 001000
SYM   = 000037
SYMD  = 000007
SYMS  = 000007
TABEND  011512
TABL6   016172
TABL7   016212
TABL8   016232
TBITVE= 000014
TBUF    001270
TCH0    011432
TCH1    011434
TCH2    011436
TCH3    011440
TCH4    011442
TCH5    011444
TCH6    011446
TCH7    011450
TCNT    011520
TCSR    001266
TEMP    011522
TEST20  010102
TIME  = 000000
TIMER   016272
TIMSAV  016514
TKVEC = 000060
TMP     016124
TMP0    016126
TMP0E   016130
TMP1    016132
```

```
TMP1E   016134          X0SRV   012626          $ERRTB  001254          $LSTTA= 000001          $SWREG  001216
TMP10   016170          X1      016254          $ERTTL  001112          $MADR1  001226          $SWRMK= 000000
TMP2    016136          X1SRV   012764          $ESCAP  001162          $MADR2  001232          $TAGLE= 177777
TMP2E   016140          X2      016256          $ETABL  001214          $MADR3  001236          $TAGNU= 050417
TMP3    016142          X2SRV   013122          $ETEND  001254          $MADR4  001242          $TEMP = 000300
TMP3E   016144          X3      016260          $FATAL  001176          $MAIL   001174          $TESTN  001200
TMP4    016146          X3SRV   013260          $FFLG   023332          $MAMS1  001224          $TIMES  001160
TMP4E   016150          X4      016262          $FILLC  001156          $MAMS2  001230          $TKB    001146
TMP5    016152          X4SRV   013416          $FILLS  001155          $MAMS3  001234          $TKS    001144
TMP5E   016154          X5      016264          $F$AND= 000310          $MAMS4  001240          $TN   = 000023
TMP6    016156          X5SRV   013554          $F$BAD= 000401          $MBADR  001002          $TPB    001152
TMP6E   016160          X6      016266          $F$BLA= 000170          $MFLG   023330          $TPFLG  001157
TMP7    016162          X6SRV   013712          $F$CAS= 000150          $MNEW   023055          $TPS    001150
TMP7E   016164          X7      016270          $F$DEC= 000220          $MSGAD  001210          $TRAP   024466
TMP9    001326          X7SRV   014050          $F$DO = 000340          $MSGLG  001212          $TRAP2  024510
TPVEC = 000064          $APTHD  001000          $F$FAL= 000405          $MSGTY  001174          $TRP  = 000012
TRAN    006776          $ATYC   023112          $F$GOO= 000400          $MSWR   023044          $TRPAD  024522
TRAPVE= 000034          $ATY1   023066          $F$IF = 000110          $MTYP1  001225          $TSK0 = 050413
TRTVEC= 000014          $ATY3   023074          $F$INC= 000210          $MTYP2  001231          $TSK1 = 050414
TST1    002066          $ATY4   023104          $F$LOO= 000200          $MTYP3  001235          $TSK2 = 050406
TST10   004012          $AUTOB  001134          $F$NAM= 000160          $MTYP4  001241          $TSK3 = 050410
TST11   004204          $BASE   001250          $F$NO = 000403          $MXCNT  024012          $TSK4 = 050333
TST12   004352          $BDADR  001122          $F$OR = 000320          $NESTL= 177777          $TSK5 = 177777
TST13   005000          $BDDAT  001126          $F$RTI= 000350          $NSK0 = 000300          $TSK6 = 050316
TST14   005356          $BELL   001164          $F$RTN= 000300          $NSK1 = 000110          $TSK7 = 050317
TST15   006034          $BGNLE= 177777          $F$SEL= 000140          $NSK2 = 000110          $TSTM   001004
TST16   006432          $BRJMP= 000000          $F$THE= 000330          $NSK3 = 000110          $TSTNM  001102
TST17   007166          $CHARC  022276          $F$TRU= 000404          $NSK4 = 000110          $TTYIN  023022
TST2    002262          $CKSWR  022302          $F$UNT= 000130          $NULL   001154          $TYPDS  024014
TST20   010102          $CMTAG  001100          $F$WHI= 000120          $NWTST= 000001          $TYPE   021746
TST21   010162          $CM3  = 000000          $F$YES= 000402          $OCNT   024462          $TYPEC  022160
TST22   011524          $CNTLG  023037          $GDADR  001120          $OMODE  024464          $TYPEX  022300
TST3    002516          $CNTLU  023032          $GDDAT  001124          $OVER   023776          $TYPOC  024264
TST4    002730          $CPUOP  001222          $GET42  021702          $PASS   001202          $TYPON  024300
TST5    003130          $CRLF   001171          $GTSWR  022352          $PASTM  001006          $TYPOS  024240
TST6    003230          $DBLK   024230          $HD   = 000000          $QUES   001170          $U    = 000402
TST7    003610          $DEVCT  001204          $HIBTS  001000          $RDCHR  022564          $UNIT   001206
TYPDS = 104405          $DEVM   001252          $ICNT   001104          $RDLIN  022714          $UNITM  001010
TYPE  = 104401          $DOAGN  021722          $IFLEV= 177777          $RDSZ = 000010          $USWR   001220
TYPOC = 104402          $DTBL   024220          $INTAG  001135          $RTNAD  021724          $VECT1  001244
TYPON = 104404          $ENDAD  021712          $ISK0 = 000001          $SAVE = 050314          $VECT2  001246
TYPOS = 104403          $ENDCT  021660          $ISK1 = 000001          $SAVE2= 050315          $XOFF = 000023
WAIT    016524          $ENDMG  021731          $ISK2 = 000001          $SAVLE= 177777          $XON  = 000021
WHICHB= 000002          $ENULL  021726          $ISK3 = 000001          $SCOPE  023534          $XTSTR  023546
XMIT0S  010642          $ENV    001214          $ITEMB  001114          $SELLE= 000000          $$ARGC= 000000
XMIT1S  010722          $ENVM   001215          $LF     001172          $SETUP= 000127          $$BYTE= 000402
XMIT2S  011002          $EOP    021624          $LFLG   023331          $SSK0 = 050337          $$CASE= 000404
XMIT3S  011062          $EOPCT  021652          $LO   = 177777          $SSK1 = 000402          $$DST = 000000
XMIT4S  011142          $ERFLG= 000400          $LOCTA= 000001          $SSK2 = 050315          $$ELOC= 000402
XMIT5S  011222          $ERMAX  001115          $LPADR  001106          $STUP = 177777          $$ERFL= 000000
XMIT6S  011302          $ERRFL  001103          $LPERR  001110          $SVLAD  023742          $$FLAG= 000001
XMIT7S  011362          $ERROR  023334          $LSK0 = 000000          $SVPC = 000430          $$FRMB= 000000
X0      016252          $ERRPC  001116          $LSTIN= 000001          $SWR  = 167400          $$FROM= 000000
```

```
$$GET4= 000000
$$IN  = 000002
$$INH = 000402
$$LOC = 020272
$$LOCN= 000000
$$OUT = 000000
$$REG = 177777
$$RETU= 000000
$$RTN1= 050415
$$RTN2= 050416
$$SRC = 000000
$$TGSV= 050316
$$TGS1= 050315
$$TGS2= 050316
$$TO  = 000001
$$TOB = 000000
$$TOTL= 000002
$$$TAG= 050000
$OFILL  024463
$50000  002122
$50001  002152
$50002  002310
$50003  002326
$50004  002352
$50005  002404
$50006  002436
$50007  002470
$50010  002516
$50011  002516
$50012  002564
$50013  002616
$50014  002650
$50015  002702
$50016  002730
$50017  002730
$50020  002764
$50021  003016
$50022  003050
$50023  003102
$50024  003130
$50025  003130
$50026  003164
$50027  003230
$50030  003204
$50031  003230
$50032  003256
$50033  003262
$50034  003574
$50035  003356
$50036  003424
$50037  003500
$50040  003500
$50041  003550
$50042  003544
$50043  003556
$50044  003572
$50045  003644
$50046  003646
$50047  003734
$50050  003766
$50051  003770
$50052  004006
$50053  004046
$50054  004050
$50055  004136
$50056  004154
$50057  004200
$50060  004240
$50061  004242
$50062  004330
$50063  004352
$50064  004424
$50065  004414
$50066  004420
$50067  004430
$50070  004516
$50071  004574
$50072  004620
$50073  004646
$50074  004734
$50075  004754
$50076  004772
$50077  005210
$50100  005256
$50101  005412
$50102  005416
$50103  005622
$50104  005670
$50105  005770
$50106  006032
$50107  006034
$50110  006070
$50111  006072
$50112  006116
$50113  006120
$50114  006124
$50115  006126
$50116  006140
$50117  006432
$50120  006216
$50121  006300
$50122  006320
$50123  006376
$50124  006366
$50125  006362
$50126  006372
$50127  006430
$50130  006430
$50131  006466
$50132  006472
$50133  006566
$50134  006574
$50135  006612
$50136  006630
$50137  006770
$50140  006766
$50141  006714
$50142  006762
$50143  006736
$50144  006762
$50145  006760
$50146  006762
$50147  006770
$50150  007032
$50151  007076
$50152  007122
$50153  007162
$50154  007154
$50155  007162
$50156  007222
$50157  007232
$50160  007252
$50161  007252
$50162  007354
$50163  007376
$50164  007416
$50165  007512
$50166  007472
$50167  007466
$50170  007512
$50171  007512
$50172  007536
$50173  007552
$50174  007570
$50175  007606
$50176  007624
$50177  007642
$50200  007732
$50201  010056
$50202  010074
$50203  010146
$50204  010162
$50205  010176
$50206  010230
$50207  010266
$50210  010406
$50211  010402
$50212  010440
$50213  010606
$50214  011544
$50215  011564
$50216  011652
$50217  011716
$50220  011750
$50221  012052
$50222  012102
$50223  012112
$50224  012140
$50225  012166
$50226  012202
$50227  012306
$50230  012432
$50231  012352
$50232  012372
$50233  012406
$50234  012426
$50235  012460
$50236  012476
$50237  012514
$50240  012530
$50241  012620
$50242  012624
$50243  012756
$50244  012762
$50245  013114
$50246  013120
$50247  013252
$50250  013256
$50251  013410
$50252  013414
$50253  013546
$50254  013552
$50255  013704
$50256  013710
$50257  014042
$50260  014046
$50261  016520
$50262  016522
$50263  016312
$50264  016452
$50265  016336
$50266  016344
$50267  016366
$50270  016402
$50271  016414
$50272  016504
$50273  016612
$50274  016612
$50275  016544
$50276  016550
$50277  016560
$50300  016604
$50301  016564
$50302  016566
$50303  016600
$50304  016602
$50305  017630
$50306  017630
$50307  016740
$50310  017024
$50311  017024
$50312  017006
$50313  016770
$50314  016752
$50315  017110
$50316  017114
$50317  017104
$50320  017230
$50321  017230
$50322  017316
$50323  017316
$50324  017342
$50325  017442
$50326  017436
$50327  017630
$50330  017456
$50331  017600
$50332  017564
$50333  017554
$50334  017606
$50335  017612
$50336  017624
$50337  017626
$50340  017762
$50341  017762
$50342  017754
$50343  017752
$50344  020220
$50345  020220
$50346  020062
$50347  020060
$50350  020124
$50351  020200
$50352  020176
$50353  020274
$50354  020274
$50355  020222
$50356  020252
$50357  020536
$50360  020536
$50361  020402
$50362  020400
$50363  020522
$50364  020520
$50365  021322
```

```
$50366  021322          $50374  020720          $50402  021134          $50410  021322          $50416  021622
$50367  020606          $50375  020740          $50403  021134          $50411  021432          .       = 024546
$50370  020740          $50376  021322          $50404  021322          $50412  021432          .$X     = 001000
$50371  020732          $50377  021004          $50405  021200          $50413  021416
$50372  020670          $50400  021140          $50406  021322          $50414  021416
$50373  020676          $50401  021074          $50407  021302          $50415  021622

. ABS.  024546      000


 ERRORS DETECTED:  0

 CJDLAA,CJDLAA.PRT/SOL=SPMACJ/ML,CJDLAA.SML,CJDLAA.P11
 RUN-TIME:  121  120  .8 SECONDS
 RUN-TIME RATIO: 605/242=2.5
 CORE USED:  51K  (102 PAGES)
```