VS60

VS60 INST TST PT1

CZVSAB0

AH-9483B-MC

FICHE 1 OF 1

JAN 1980

MADE IN USA

IDENTIFICATION

PRODUCT CODE: AC-9482B-MC

PRODUCT NAME: CZVSAB0 VS60 INST TST PT1

DATE: JUNE 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHORS: R. SHOOP & R. MOORE

TABLE OF CONTENTS

1.0 ABSTRACT

THIS PROGRAM IS THE FIRST OF A SERIES DESIGNED TO TEST STRICTLY LOGIC FUNCTIONS OF THE VS60 DISPLAY PROCESSOR. THIS DIAGNOSTIC TESTS, READING AND WRITING OF VS60 BUS REGISTERS, NON GRAPHIC TYPE INSTRUCTIONS AND SEVERAL INTERRUPT CONDITIONS. THERE IS NO FULL SPEED VS60 TESTING IN THIS TEST. MAINTENANCE MODES 1, 2 OR 3 ARE USED.

*** REV B -- FIXES TIMING PROBLEM IN TEST 136 ***

2.0 REQUIREMENTS

2.1 EQUIPMENT

A. PDP-11 COMPUTER WITH AT LEAST 8K OF MEMORY
B. I/O TERMINAL (I.E. ASR33 TTY)
C. VS60 DISPLAY PROCESSOR

2.2 STORAGE

THIS PROGRAM OCCUPIES LOWER 8K OF MEMORY.

3.0 LOADING PROCEDURE

NORMAL PROCEDURE FOR LOADING A BINARY PROGRAM INTO MEMORY SHOULD BE FOLLOWED.

4.0 STARTING PROCEDURE

4.1 STARTING ADDRESS 200

LOADING ADDRESS 200 AND STARTING WILL IDENTIFY THE TEST, INITIALIZE THE SYSTEM, AND BEGIN TESTING.

4.2 RESTART ADDRESS 204

LOADING ADDRESS 204 AND STARTING WILL INITIALIZE THE SYSTEM AND BEGIN TESTING.

5.0 SWITCH REGISTER SETTINGS

SWITCH	FUNCTION
Sw15=1	HALT ON ERROR
SW14=1	LOOP ON TEST
SW13=1	INHIBIT ERROR TYPEOUTS
SW12=1	HALT AT END OF DIAGNOSTIC
SW11=1	INHIBIT ITERATIONS - NORMALLY 2000(8) AFTER 1ST PASS
SW10=1	BELL ON ERROR
SW09=1	LOOP ON ERROR
SW08=1	LOOP ON TEST IN SWR<7:0>

6.0 ERROR REPORTING

6.1 ERROR COMMENT

ALL ERRORS ARE ACCOMPANIED WITH AN ENGLISH LANGUAGE DESCRIPTIVE COMMENT AS TO THE TYPE OF FAILURE. FURTHER QUALIFICATION OF THE ERROR CAN BE OBTAINED IF NEEDED FROM THE COMMENT AT THE ERROR PC OF THE LISTING.

6.2 ERROR DATA

PC	LISTING ADDRESS WHERE THE ERROR WAS DETECTED.
TSTNUM	TEST NUMBER WHERE THE ERROR OCCURRED.
BUSADRS	DISPLAY BUS ADDRESS WHERE THE RESULTANT DATA WAS EXPECTED
EXPCT	DATA THAT WAS EXPECTED.
RCVD	DATA THAT WAS RECEIVED.

7.0 MISCELLANEOUS

SEQ 0005

7.1 VS60 BUS/VECTOR ADDRESS MODIFICATION:
MODIFY LOCATION '$VECT1' IF BASE VECTOR ADDRESS IS NOT 320.
MODIFY LOCATION '$BASE' IF BASE BUS ADDRESS IS NOT 172000.

NOTE: A RESTART IS REQUIRED AFTER THE ABOVE ADDRESS MODIFICATION.
THE ABOVE LOCATIONS ARE LOCATED IN THE 'APT MAILBOX-ETABLE'

7.2 XXDP/APT NOTES:
THIS DIAGNOSTIC IS CHAINABLE UNDER XXDP.
THIS DIAGNOSTIN INCLUDES THE 'APT' SOFTWARE HOOKS HOWEVER, THEY
HAVE NOT BEEN TESTED.

7.3 POWER FAIL:
A POWER FAILURE WILL CAUSE A RESTART MESSAGE ON POWER UP AT
WHICH TIME THE PROGRAM IS RESTARTED AT THE BEGINNING.

7.4 SINGLE VS60 TESTING:
THIS DIAGNOSTIC DOES NOT TEST MULTIPLE VS60'S.

8.0 EXECUTION TIME

EXECUTION TIME RANGES FROM ABOUT 3 SECONDS WITH NO ITERATIONS
TO ABOUT 2 MINUTES WITH ITERATIONS ENABLED. AN 'END PASS' MESSAGE
IS TYPED EACH PASS THRU THE DIAGNOSTIC.

9.0 PROGRAM TEST DESCRIPTIONS

9.1 GENERAL

THE TABLE OF CONTENTS AT THE BEGINNING OF THE LISTING CONTAINS
A BRIEF DESCIPTION OF EACH TEST. IT WILL ALSO IDENTIFY WHAT MAINTENANCE
MODE A PARTICULAR TEST USES. THE COMMENT FIELD IN EACH TEST
ALSO CAN BE HELPFUL IN UNDERSTANDING A TEST.

9.2 TYPICAL TEST SEQUENCE

MOST TESTS ARE ORGANIZED AROUND THE FOLLOWING FORMAT:
A MAINTENANCE MODE SWITCH IS SET (SW1, 2 OR 3). THEN A VS60 REGISTER
BUS REGISTER ADDRESS IS LOADED INTO LOCATION '$BDADR' WHICH
IS WHERE THE EXPECTED DATA WILL BE READ. NEXT STEP WILL BE TO LOAD THE
EXPECTED DATA INTO LOCATION '$GDDAT'. THEN ONE OR MORE INSTRUCTIONS
ARE LOADED STARTING AT LOCATION 'BUFFER'. THE DISPLAY PC IS THEN LOADED
WITH ADDRESS 'BUFFER' WHICH CAUSES AN NPR TO 'BUFFER'. AT THE COMPLETION
OF THE 'NPR' A NUMBER OF RESUMES COULD BE ISSUED IF ADDITIONAL 'NPRS' ARE
REQUIRED FOR INSTRUCTIONS OR DATA. AT THIS TIME THE CONDITION
UNDER TEST SHOULD HAVE BEEN SET UP AND AVAILABLE AT THE BUS ADDRESS
SET UP IN '$BDADR'. THIS CONDITION IS USUALLY LOADED INTO LOCATION
'$BDDAT' AND COMPARED TO THE EXPECTED RESULTS PREVIOUSLY SET UP
IN '$GDDAT'. IF AN ERROR IS DETECTED THEN AN ERROR TRAP (EMT) IS PREFORMED
ACCORDING TO THE ITEM NUMBER IN THE '$ERRTB' (ERROR TABLE).
THIS TABLE DIRECTS OR POINTS TO THE PERTINENT 'TYPE OUT' INFORMATION.

10.0 LISTING

SEQ 0006

PROGRAM LISTING FOLLOWS.

```
                                                H 1
    13                          .TITLE  CZVSA-B  VS60 INSTRUCTION TEST PART 1
   (1)                          ;*COPYRIGHT (C) 1976
   (1)                          ;*DIGITAL EQUIPMENT CORP.
   (1)                          ;*MAYNARD, MASS. 01754
   (1)                          ;*
   (1)                          ;*PROGRAM BY R. MOORE
   (1)                          ;*
   (1)                          ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
   (1)                          ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
   (1)                          ;*
    14                          ;
    15                          ; REV B -- FIXES TIMING HACK IN TEST 136  (G.P. JUNE '79).
    16                          ;
    17                          .SBTTL  BASIC DEFINITIONS
   (1)
   (1)                          ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
   (1)          001100          STACK=  1100
   (1)                          .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
   (1)                          .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
   (1)
   (1)                          ;*MISCELLANEOUS DEFINITIONS
   (1)          000011          HT=     11              ;;CODE FOR HORIZONTAL TAB
   (1)          000012          LF=     12              ;;CODE FOR LINE FEED
   (1)          000015          CR=     15              ;;CODE FOR CARRIAGE RETURN
   (1)          000200          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
   (1)          177776          PS=     177776          ;;PROCESSOR STATUS WORD
   (1)                          .EQUIV  PS,PSW
   (1)          177774          STKLMT= 177774          ;;STACK LIMIT REGISTER
   (1)          177772          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
   (1)          177570          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
   (1)          177570          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
   (1)
   (1)                          ;*GENERAL PURPOSE REGISTER DEFINITIONS
   (1)          000000          R0=     %0              ;;GENERAL REGISTER
   (1)          000001          R1=     %1              ;;GENERAL REGISTER
   (1)          000002          R2=     %2              ;;GENERAL REGISTER
   (1)          000003          R3=     %3              ;;GENERAL REGISTER
   (1)          000004          R4=     %4              ;;GENERAL REGISTER
   (1)          000005          R5=     %5              ;;GENERAL REGISTER
   (1)          000006          R6=     %6              ;;GENERAL REGISTER
   (1)          000007          R7=     %7              ;;GENERAL REGISTER
   (1)          000006          SP=     %6              ;;STACK POINTER
   (1)          000007          PC=     %7              ;;PROGRAM COUNTER
   (1)
   (1)                          ;*PRIORITY LEVEL DEFINITIONS
   (1)          000000          PR0=    0               ;;PRIORITY LEVEL 0
   (1)          000040          PR1=    40              ;;PRIORITY LEVEL 1
   (1)          000100          PR2=    100             ;;PRIORITY LEVEL 2
   (1)          000140          PR3=    140             ;;PRIORITY LEVEL 3
   (1)          000200          PR4     200             ;;PRIORITY LEVEL 4
   (1)          000240          PR5=    240             ;;PRIORITY LEVEL 5
   (1)          000300          PR6=    300             ;;PRIORITY LEVEL 6
   (1)          000340          PR7=    340             ;;PRIORITY LEVEL 7
   (1)
   (1)                          ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
   (1)          100000          SW15=   100000
```

```
(1)             040000                  SW14=   40000
(1)             020000                  SW13=   20000
(1)             010000                  SW12=   10000
(1)             004000                  SW11=   4000
(1)             002000                  SW10=   2000
(1)             001000                  SW09=   1000
(1)             000400                  SW08=   400
(1)             000200                  SW07=   200
(1)             000100                  SW06=   100
(1)             000040                  SW05=   40
(1)             000020                  SW04=   20
(1)             000010                  SW03=   10
(1)             000004                  SW02=   4
(1)             000002                  SW01=   2
(1)             000001                  SW00=   1
(1)                                     .EQUIV  SW09,SW9
(1)                                     .EQUIV  SW08,SW8
(1)                                     .EQUIV  SW07,SW7
(1)                                     .EQUIV  SW06,SW6
(1)                                     .EQUIV  SW05,SW5
(1)                                     .EQUIV  SW04,SW4
(1)                                     .EQUIV  SW03,SW3
(1)                                     .EQUIV  SW02,SW2
(1)                                     .EQUIV  SW01,SW1
(1)                                     .EQUIV  SW00,SW0
(1)
(1)                                     ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)             100000                  BIT15=  100000
(1)             040000                  BIT14=  40000
(1)             020000                  BIT13=  20000
(1)             010000                  BIT12=  10000
(1)             004000                  BIT11=  4000
(1)             002000                  BIT10=  2000
(1)             001000                  BIT09=  1000
(1)             000400                  BIT08=  400
(1)             000200                  BIT07=  200
(1)             000100                  BIT06=  100
(1)             000040                  BIT05=  40
(1)             000020                  BIT04=  20
(1)             000010                  BIT03=  10
(1)             000004                  BIT02=  4
(1)             000002                  BIT01=  2
(1)             000001                  BIT00=  1
(1)                                     .EQUIV  BIT09,BIT9
(1)                                     .EQUIV  BIT08,BIT8
(1)                                     .EQUIV  BIT07,BIT7
(1)                                     .EQUIV  BIT06,BIT6
(1)                                     .EQUIV  BIT05,BIT5
(1)                                     .EQUIV  BIT04,BIT4
(1)                                     .EQUIV  BIT03,BIT3
(1)                                     .EQUIV  BIT02,BIT2
(1)                                     .EQUIV  BIT01,BIT1
(1)                                     .EQUIV  BIT00,BIT0
(1)
(1)                                     ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)             000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
```

```
(1)  000010  RESVEC= 10          ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)  000014  TBITVEC=14          ;;"T" BIT
(1)  000014  TRTVEC= 14          ;;TRACE TRAP
(1)  000014  BPTVEC= 14          ;;BREAKPOINT TRAP (BPT)
(1)  000020  IOTVEC= 20          ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)  000024  PWRVEC= 24          ;;POWER FAIL
(1)  000030  EMTVEC= 30          ;;EMULATOR TRAP (EMT) **ERROR**
(1)  000034  TRAPVEC=34          ;;"TRAP" TRAP
(1)  000060  TKVEC=  60          ;;TTY KEYBOARD VECTOR
(1)  000064  TPVEC=  64          ;;TTY PRINTER VECTOR
(1)  000240  PIRQVEC=240         ;;PROGRAM INTERRUPT REQUEST VECTOR
  18
  19  172000  ABASE=  172000      ; 1ST REGISTER ADDRESS (DISPLAY PC).
  20  100320  AVECT1= 100320      ; PRI (HI BYTE) AND VECTOR ( LO BYTE).
```

```
 22
(1)
(1)                                     .SBTTL  OPERATIONAL SWITCH SETTINGS
(1)                                     ;*
(1)                                     ;*      SWITCH                  USE
(1)                                     ;*      ------          --------------------
(1)                                     ;*        15            HALT ON ERROR
(1)                                     ;*        14            LOOP ON TEST
(1)                                     ;*        13            INHIBIT ERROR TYPEOUTS
(1)                                     ;*        12            HALT AT END OF DIAGNOSTIC
(1)                                     ;*        11            INHIBIT ITERATIONS
(1)                                     ;*        10            BELL ON ERROR
(1)                                     ;*         9            LOOP ON ERROR
(1)                                     ;*         8            LOOP ON TEST IN SWR<7:0>
 23
 24
(1)                                     .SBTTL  TRAP CATCHER
(1)
(1)             000000                          .=0
(1)                                     ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1)                                     ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)                                     ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)             000174                          .=174
(1)   000174  000000                   DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
(1)   000176  000000                   SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER
(1)                                     .SBTTL  STARTING ADDRESS(ES)
(1)   000200  000137  002110                   JMP     @#BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
 25   000204  000137  002472                   JMP     RESTRT  ; RESTART ADDRESS
 26
 27                                     .SBTTL  ACT11 HOOKS
(1)
(2)                                     ;;**************************************************************************
(1)                                     ;HOOKS REQUIRED BY ACT11
(1)             000210                          $SVPC=.                 ;SAVE PC
(1)             000046                          .=46
(1)   000046  024420                           $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
(1)             000052                          .=52
(1)   000052  000000                           .WORD   0               ;;2)SET LOC.52 TO ZERO
(1)             000210                          .-$SVPC                 ;; RESTORE PC
 28
 29             001000                          .=1000
 30                                     .SBTTL  APT PARAMETER BLOCK
(1)
(2)                                     ;;**************************************************************************
(1)                                     ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(2)                                     ;;**************************************************************************
(1)             001000                          .$X=.   ;;SAVE CURRENT LOCATION
(1)             000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
(1)   000024  000200                           200     ;;FOR APT START UP
(1)             000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
(1)   000044  001000                           $APTHDR ;;POINT TO APT HEADER BLOCK
(1)             001000                          .=.$X   ;;RESET LOCATION COUNTER
(2)                                     ;;**************************************************************************
(1)                                     ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(1)                                     ;INTERFACE SPEC.
(1)
(1)   001000                           $APTHD:
(1)   001000  000000                   $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1)   001002  001202                   $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
```

```
   (1)  001004  000012                  $TSTM:  .WORD   10.      ;;RUN TIM OF LONGEST TEST
   (1)  001006  000036                  $PASTM: .WORD   30.      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
   (1)  001010  000000                  $UNITM: .WORD   0        ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
   (1)  001012  000033                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
    31
```

```
 32                                     .SBTTL  COMMON TAGS
(1)
(2)                                     ;;**********************************************************************
(1)                                     ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1)                                     ;*USED IN THE PROGRAM.
(1)
(1)          001100                             .=1100
(1)  001100                             $CMTAG:                         ;;START OF COMMON TAGS
(1)  001100  000000                             .WORD   0
(1)  001102     000                     $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
(1)  001103     000                     $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
(1)  001104  000000                     $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
(1)  001106  000000                     $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
(1)  001110  000000                     $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
(1)  001112  000000                     $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
(1)  001114     000                     $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
(1)  001115     001                     $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
(1)  001116  000000                     $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
(1)  001120  000000                     $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
(1)  001122  000000                     $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
(1)  001124  000000                     $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
(1)  001126  000000                     $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
(1)  001130  000000                             .WORD   0               ;;RESERVED--NOT TO BE USED
(1)  001132  000000                             .WORD   0
(1)  001134     000                     $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
(1)  001135     000                     $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
(1)  001136  000000                             .WORD   0
(1)  001140  177570                     SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
(1)  001142  177570                     DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
(1)  001144  177560                     $TKS:   177560                  ;;TTY KBD STATUS
(1)  001146  177562                     $TKB:   177562                  ;;TTY KBD BUFFER
(1)  001150  177564                     $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
(1)  001152  177566                     $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
(1)  001154     000                     $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
(1)  001155     002                     $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1)  001156     012                     $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
(1)  001157     000                     $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
(1)  001160  000000                     $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
(1)                                                                     ;;WHICH  ($REG0) WAS OBTAINED
(3)  001162  000000                     $REG0:  .WORD   0               ;;CONTAINS (($REGAD)+0)
(3)  001164  000000                     $REG1:  .WORD   0               ;;CONTAINS (($REGAD)+2)
(1)  001166  000000                     $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
(1)  001170  000000                     $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
(1)  001172  177607  000377             $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
(1)  001176     077                     $QUES:  .ASCII  /?/             ;;QUESTION MARK
(1)  001177     015                     $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
(1)  001200  000012                     $LF:    .ASCIZ  <12>            ;;LINE FEED
(2)                                     ;;**********************************************************************
(2)                                     .SBTTL  APT MAILBOX-ETABLE
(2)
(3)                                     ;;**********************************************************************
(2)                                     .EVEN
(2)  001202                             $MAIL:                          ;;APT MAILBOX
(2)  001202  000000                     $MSGTY: .WORD   AMSGTY           ;;MESSAGE TYPE CODE
(2)  001204  000000                     $FATAL: .WORD   AFATAL           ;;FATAL ERROR NUMBER
(2)  001206  000000                     $TESTN: .WORD   ATESTN           ;;TEST NUMBER
```

```
(2)  001210  000000          $PASS:  .WORD   APASS   ;;PASS COUNT
(2)  001212  000000          $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
(2)  001214  000000          $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
(2)  001216  000000          $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
(2)  001220  000000          $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
(2)  001222                  $ETABLE:                ;;APT ENVIRONMENT TABLE
(2)  001222      000         $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
(2)  001223      000         $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
(2)  001224  000000          $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
(2)  001226  000000          $USWR:  .WORD   AUSWR   ;;USER SWITCHES
(2)  001230  000000          $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
(2)                          ;*                      BITS 15-11=CPU TYPE
(2)                          ;*                              11/04=01,11/05=02,11/20=03,11/40-04,11/45-05
(2)                          ;*                              11/70=06,PDQ=07,Q=10
(2)                          ;*                      BIT 10=REAL TIME CLOCK
(2)                          ;*                      BIT  9=FLOATING POINT PROCESSOR
(2)                          ;*                      BIT  8=MEMORY MANAGEMENT
(2)  001232      000         $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(2)  001233      000         $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
(2)                          ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
(2)                          ;*                              900 NSEC CORE=001
(2)                          ;*                              300 NSEC BIPOLAR=002
(2)                          ;*                              500 NSEC MOS=003
(2)  001234  000000          $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
(2)                          ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE'' ABOVE
(2)  001236      000         $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(2)  001237      000         $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
(2)  001240  000000          $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(2)  001242      000         $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(2)  001243      000         $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
(2)  001244  000000          $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(2)  001246      000         $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(2)  001247      000         $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
(2)  001250  000000          $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
(2)  001252  100320          $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(2)  001254  000000          $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(2)  001256  172000          $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)  001260  000000          $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
(2)  001262  000000          $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
(2)  001264  000000          $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
(2)  001266  000000          $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
(2)  001270                  $ETEND:
(2)                          .MEXIT
```

```
(1)                                     .SBTTL  ERROR POINTER TABLE
(1)
(1)                                     ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1)                                     ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1)                                     ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1)                                     ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
(1)                                     ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1)
(1)                                     ;*      EM              ;;POINTS TO THE ERROR MESSAGE
(1)                                     ;*      DH              ;;POINTS TO THE DATA HEADER
(1)                                     ;*      DT              ;;POINTS TO THE DATA
(1)                                     ;*      DF              ;;POINTS TO THE DATA FORMAT
(1)
(1)
(1)  001270                             $ERRTB:
 33
 34                                     ;ITEM   1
 35
 36  001270  027216                             EM1                     ;MAINT. SWITCH REGISTER
 37  001272  031014                             DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
 38  001274  031062                             DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
 39  001276  000000                             0
 40
 41                                     ;ITEM   2
 42
 43  001300  027240                             EM2                     ;LOADING D.P.C. REGISTER <STATIC>
 44  001302  031014                             DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
 45  001304  031062                             DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
 46  001306  000000                             0
 47
 48                                     ;ITEM   3
 49
 50  001310  027274                             EM3                     ;LOADING MODE REGISTER
 51  001312  031014                             DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
 52  001314  031062                             DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
 53  001316  000000                             0
 54
 55                                     ;ITEM   4
 56
 57  001320  027315                             EM4                     ;SEQUENCING DPC REGISTER
 58  001322  031014                             DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
 59  001324  031062                             DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
 60  001326  000000                             0
 61
 62                                     ;ITEM   5
 63
 64  001330  027340                             EM5                     ;LOADING LINE TYPE REGISTER
 65  001332  031014                             DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
 66  001334  031062                             DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
 67  001336  000000                             0
```

```
                                ;ITEM   6
                                
001340  027366                          EM6                     ;LOADING INTENSITY LEVEL REGISTER
001342  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001344  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001346  000000                          0
                                
                                ;ITEM   7
                                
001350  027422                          EM7                     ;LOADING GRAPHPLOT INCREMENT REGISTER
001352  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001354  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001356  000000                          0
                                
                                ;ITEM   10
                                
001360  027454                          EM10                    ;DISPLAY JUMP ABSOLUTE TO AN ADDRESS
001362  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001364  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001366  000000                          0
                                
                                ;ITEM   11
                                
001370  027510                          EM11                    ;DISPLAY JUMP RELATIVE TO AN ADDRESS
001372  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001374  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001376  000000                          0
                                
                                ;ITEM   12
                                
001400  027544                          EM12                    ;DISPLAY JSR ABSOLUTE TO AN ADDRESS
001402  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001404  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001406  000000                          0
                                
                                ;ITEM   13
                                
001410  027577                          EM13                    ;DISPLAY JSR RELATIVE TO AN ADDRESS
001412  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001414  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001416  000000                          0
                                
                                ;ITEM   14
                                
001420  027632                          EM14                    ;LOADING X POSITION REGISTER
001422  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001424  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001426  000000                          0
```

```
                                        ;ITEM   15
                                        EM15                    ;LOADING Y POSITION REGISTER
001430  027654                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001432  031014                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001434  031062                          0
001436  000000
                                        ;ITEM   16
                                        EM16                    ;BUS ERROR TIME OUT
001440  027676                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001442  031014                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001444  031062                          0
001446  000000
                                        ;ITEM   17
                                        EM17                    ;CHARACTER TERMINATOR ERROR
001450  027716                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001452  031014                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001454  031062                          0
001456  000000
                                        ;ITEM   20
                                        EM20                    ;Y OFFSET REGISTER ERROR
001460  027740                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001462  031014                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001464  031062                          0
001466  000000
                                        ;ITEM   21
                                        EM21                    ;X OFFSET REGISTER ERROR
001470  027764                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001472  031014                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001474  031062                          0
001476  000000
                                        ;ITEM   22
                                        EM22                    ;RELOCATE REGISTER ERROR
001500  030010                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001502  031014                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001504  031062                          0
001506  000000
                                        ;ITEM   23
                                        EM23                    ;D.P.U. NAME REGISTER ERROR
001510  030030                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001512  031014                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001514  031062                          0
001516  000000
```

```
                                ;ITEM   24
                                
001520  030053                          EM24                    ;SEARCH OR ASSOCITIVE NAME MATCH ERROR
001522  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001524  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001526  000000                          0
                                
                                ;ITEM   25
                                
001530  030115                          EM25                    ;VECTOR SCAL ERROR
001532  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001534  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001536  000000                          0
                                
                                ;ITEM   26
                                
001540  030135                          EM26                    ;CHARACTER SCALE ROTATE ERROR
001542  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001544  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001546  000000                          0
                                
                                ;ITEM   27
                                
001550  030153                          EM27                    ;BLINK LOGIC ERROR
001552  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001554  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001556  000000                          0
                                
                                ;ITEM   30
                                
001560  030172                          EM30                    ;ITALIC LOGIC ERROR
001562  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001564  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001566  000000                          0
                                
                                ;ITEM   31
                                
001570  030212                          EM31                    ;MENU LOGIC ERROR
001572  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001574  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001576  000000                          0
                                
                                ;ITEM   32
                                
001600  030230                          EM32                    ;RESET FAILED TO CLEAR A REGISTER
001602  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001604  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001606  000000                          0
```

```
                                ;ITEM   33

                                        EM33                    ;STOP INTERRUPT ERROR
                                        DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
                                        DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
                                        0

                                ;ITEM   34

                                        EM34                    ;L.P. FLAG ERROR
                                        DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
                                        DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
                                        0

                                ;ITEM   35

                                        EM35                    ;COLOR LEVEL ERROR
                                        DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
                                        DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
                                        0

                                ;ITEM   36

                                        EM36                    ;INTENSITY ENABLE - CONSOLE 0/1 ERROR
                                        DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
                                        DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
                                        0

                                ;ITEM   37

                                        EM37                    ;INTERNAL STOP FLAG FAILED TO SET
                                        DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
                                        DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
                                        0

                                ;ITEM   40

                                        EM40                    ;INTERNAL STOP FLAG FAILED TO CLEAR
                                        DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
                                        DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
                                        0

                                ;ITEM   41

                                        EM41                    ;WORD 0-1-2 INDICATOR ER
                                        DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
                                        DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
                                        0
```

```
                                        ;ITEM   42

001700  030531                          EM42                    ;L.P. INTR ENABLE - CONSOLE 0/1 ERROR
001702  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001704  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001706  000000                          0

                                        ;ITEM   43

001710  030572                          EM43                    ;L.P. SWITCH INTR ENABLE - CONSOLE 0/1 ERROR
001712  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001714  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001716  000000                          0

                                        ;ITEM   44

001720  030642                          EM44                    ;DISPLAY BUSY ERROR
001722  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001724  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001726  000000                          0

                                        ;ITEM   45
001730  030662                          EM45                    ;EXTERNAL STOP LOGIC ERROR
001732  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001734  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001736  000000                          0

                                        ;ITEM   46

001740  030711                          EM46                    ;TIME-OUT INTERRUPT ERROR
001742  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001744  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001746  000000                          0

                                        ;ITEM   47

001750  030732                          EM47                    ;NAME MATCH INTERRUPT ERROR
001752  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001754  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001756  000000                          0

                                        ;ITEM   50

001760  030755                          EM50                    ;L.P. FLAG INTERRUPT ERROR
001762  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001764  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001766  000000                          0

                                        ;ITEM   51

001770  030777                          EM51                    ;STACK PTR ER
001772  031014                          DH1                     ;ERRPC  TSTNUM  BUSADR  EXPCT   RCVD
001774  031062                          DT1                     ;$ERRPC TSTNUM  $BDADR  $GDDAT  $BDDAT
001776  000000                          0
```

```
                                ;COMMON PROGRAM TAGS AND STORAGE LOCATIONS

002000  031076          DBUF:   BUFFER                  ;FIRST WORD IN THE DISPLAY BUFFER
002002  031100          DBUF1:  BUFFER+2                ;SECOND WORD
002004  031102          DBUF2:  BUFFER+4                ;THIRD WORD
002006  031104          DBUF3:  BUFFER+6                ;FOURTH WORD
002010  031106          DBUF4:  BUFFER+10               ;FIFTH WORD
002012  031110          DBUF5:  BUFFER+12               ;SIXTH WORD
002014  000000          DSAVE:  0                       ;TEMP REG.
002016  000000          DSAVE1: 0
002020  000000          SIZE:   0                       ;BUFFER SIZE
002022  000000          CNTR:   0
002024  000000          TSTNUM: 0                       ;CONTAINS TEST NO. ON ER
002026  000020          DELAY:  20                      ;COUNT USED FOR GENERAL PURPOSE DELAYS

                                ;VS60 ADDRESSES AND VECTORS POINTERS

002030  172000          DPC:    172000                  ;DISPLAY PC REGISTER
002032  172002          SREG0:  172002                  ;DISPLAY STATUS REGISTER
002034  172004          XPOS:   172004                  ;X AXIS REGISTER AND GRAPHPLOT REG <READ ONLY>
002036  172006          YPOS:   172006                  ;Y AXIS REGISTER AND CHARACTER REG <READ ONLY>
002040  172010          RLO:    172010                  ;RELOCATION REGISTER
002042  172012          SREG1:  172012                  ;MISC STATUS REGISTER #1
002044  172014          XDOFF:  172014                  ;X DYNAMIC OFFSET REGISTER
002046  172016          YDOFF:  172016                  ;Y DYNAMIC OFFSET REGISTER
002050  172020          ANAME:  172020                  ;ASSOCIATIVE NAME REGISTER
002052  172022          CONS:   172022                  ;CONSOLE STATUS REGISTER
002054  172024          DNAME:  172024                  ;DISPLAY NAME REGISTER
002056  172026          STKVAL: 172026                  ;VALUE OF CURRENT STACK WORD
002060  172030          TERMCH: 172030                  ;TERMINATE CHARACTER REGISTER
002062  172032          STKPT:  172032                  ;STACK POINTER REGISTER
002064  172034          ZPOS:   172034                  ;Z POSITION REGISTER <READ ONLY>
002066  172036          ZDOFF:  172036                  ;Z DYNAMIC OFFSET REGISTER

002070  000320          DDONE:  320                     ;DISPLAY STOP <DONE> VECTOR
002072  000322          DDONE1: 322                     ;

002074  000324          LPVCT:  324                     ;DISPLAY LIGHT PEN VECTOR
002076  000326          LPVCT1: 326                     ;

002100  000330          TIMEVT: 330                     ;DISPLAY TIME-OUT <NXM.> ERROR VECTOR
002102  000332          TMEVT1: 332                     ; OR ''SHIFT-OUT'' VECTOR

002104  000334          NAMEV:  334                     ;NAME MATCH VECTOR
002106  000336          NAMEV1: 336
```

```
  385                                   ;SOFTWARE INITALIZATION ROUTINE
  386
  388  002110                           BEGIN:
  (1)                                   .SBTTL  INITIALIZE THE COMMON TAGS
  (1)                                   ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
  (1)  002110  012706  001100                   MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
  (1)  002114  005026                           CLR     (R6)+           ;;CLEAR MEMORY LOCATION
  (1)  002116  022706  001140                   CMP     #SWR,R6 ;;DONE?
  (1)  002122  001374                           BNE     .-6             ;;LOOP BACK IF NO
  (1)  002124  012706  001100                   MOV     #STACK,SP       ;;SETUP THE STACK POINTER
  (1)                                   ;;INITIALIZE A FEW VECTORS
  (1)  002130  012737  024526  000020           MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
  (1)  002136  012737  000340  000022           MOV     #340,@#IOTVEC+2 ;;LEVEL 7
  (1)  002144  012737  025252  000030           MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
  (1)  002152  012737  000340  000032           MOV     #340,@#EMTVEC+2 ;;LEVEL 7
  (1)  002160  012737  027062  000034           MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
  (1)  002166  012737  000340  000036           MOV     #340,@#TRAPVEC+2;LEVEL 7
  (1)  002174  012737  026262  000024           MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
  (1)  002202  012737  000340  000026           MOV     #340,@#PWRVEC+2 ;;LEVEL 7
  (1)  002210  013737  024366  024360           MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
  (1)  002216  005037  001166                   CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
  (1)  002222  005037  001170                   CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
  (1)  002226  112737  000001  001115           MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
  (1)  002234  012737  002234  001106           MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
  (1)  002242  012737  002242  001110           MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
  (2)                                   ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
  (2)                                   ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
  (2)  002250  013746  000004                   MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
  (2)  002254  012737  002310  000004           MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
  (2)  002262  012737  177570  001140           MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
  (2)  002270  012737  177570  001142           MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
  (2)  002276  022777  177777  176634           CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
  (2)  002304  001012                           BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
  (2)                                                                   ;;AND  THE HARDWARE SWR IS NOT - -1
  (2)  002306  000403                           BR      65$             ;;BRANCH IF NO TIMEOUT
  (2)  002310  012716  002316          64$:     MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
  (2)  002314  000002                           RTI
  (2)  002316  012737  000176  001140  65$:     MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
  (2)  002324  012737  000174  001142           MOV     #DISPREG,DISPLAY
  (2)  002332  012637  000004          66$:     MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
  (1)
  (2)  002336  005037  001210                   CLR     $PASS           ;;CLEAR PASS COUNT
  (2)  002342  132737  000200  001223           BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
  (2)  002350  001403                           BEQ     67$             ;;YES,USE NON-APT SWITCH
  (2)  002352  012737  001224  001140           MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
  (2)  002360                           67$:
  389  002360  012700  002030          SETUP1: MOV     #DPC,R0         ;SET UP VS60 REG ADRS POINTERS
  390  002364  013701  001256                   MOV     $BASE,R1        ;GET VS60 BASE ADRS FROM APT MAILBOX-ETABLE
  391  002370  010120                  SETUP2: MOV     R1,(0)+
  392  002372  062701  000002                   ADD     #2,R1
  393  002376  022700  002070                   CMP     #DDONE,R0       ;TEST FOR LAST ADDRESS
  394  002402  001372                           BNE     SETUP2
  395  002404  012700  002070                   MOV     #DDONE,R0       ;SET UP VS60 VECTOR ADRS POINTERS
  396  002410  013701  001252                   MOV     $VECT1,R1       ;GET VS60 BASE VECTOR ADRS FROM APT MAILBOX-ETABLE
  397  002414  042701  160000                   BIC     #160000,R1      ; CLEAR PSW BITS        ** B **
  398  002420  010120                  SETUP3: MOV     R1,(0)+
```

```
 399  002422  062701  000002                    ADD     #2,R1
 400  002426  022700  002110                    CMP     #BEGIN,R0       ;TEST FOR LAST VECTOR
 401  002432  001372                           BNE     SETUP3
 402  002434  004737  024454                    JSR     PC,RSTVEC       ;GO SET UP VS60 VECTOR ADR'S WITH HLTS
 403  002440  004737  026502                    JSR     PC,$SIZE        ;DETERMINE LAST MEM LOCATION
 404  002444  013737  026576  002020            MOV     $LSTAD,SIZE     ;LOAD LAST MEM ADDRESS
 405  002452  162737  006000  002020            SUB     #6000,SIZE      ;SAVE LOADERS AND XXDP
 406  002460  005737  000042                    TST     @#42            ;TEST IF RUNNING CHAIN MODE UNDER XXDP
 407  002464  001002                           BNE     RESTRT          ;BR IF SO
 408  002466  104401                           TYPE
 409  002470  027132                           HEADER
 410  002472  012737  000340  177776  RESTRT:  MOV     #340,PSW        ;SET PRIORTY TO HIGHEST LEVEL
 411  002500  012706  001100                    MOV     #STACK,SP       ;ALWAYS SET UP STACK POINTER
 412  002504  000005                           RESET                   ;START TESTS WITH VS60 INITIALIZED
 413  002506  105037  001102                    CLRB    $TSTNM          ;RESET TEST NUMBER
```

```
                                ;;*****************************************************************
                                ;*TEST 1         SEE IF ALL VS60 REGS ARE THERE
                                ;;*****************************************************************
                                ;
002512  000004                  TST1:   SCOPE
002514  012737  000001  001206          MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002522  012737  002556  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
002530  012737  002576  000004          MOV     #2$,@#ERRVEC    ;SET UP TIME OUT TRAP
002536  012737  000340  000006          MOV     #340,@#ERRVEC+2 ;SET UP TIME OUT PRIORITY
002544  013737  001256  001122          MOV     $BASE,$BDADR    ;PUT 1ST VS60 BUS ADRS IN LOC $BDADR
002552  012700  000020                  MOV     #20,R0          ;SET UP REG COUNT
002556  005777  176340          1$:     TST     @$BDADR         ;LOOK FOR VS60 REG
002562  062737  000002  001122          ADD     #2,$BDADR       ;OK-ADVANCE TO NEXT
002570  005300                          DEC     R0              ;HAVE WE TRIED ALL REGS?
002572  001371                          BNE     1$              ;BR IF NOT
002574  000402                          BR      3$              ;GO RESTORE LOCS 4 & 6
002576  022626                  2$:     CMP     (R6)+,(R6)+     ;FIX STK SINCE NO RTI
002600  104016                          ERROR   16              ;REG ADRS TIME OUT ER
002602  012737  000006  000004  3$:     MOV     #ERRVEC+2,@#ERRVEC      ;POINT TO HALT IN ADRS 6
002610  005037  000006                  CLR     @#ERRVEC+2      ;SET UP HALT
                                ;;*****************************************************************
                                ;*TEST 2         FLOAT A ONE THRU TERMINATE CHARACTER REG
                                ;;*****************************************************************
002614  000004                  TST2:   SCOPE
002616  012737  000002  001206          MOV     #2,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002624  012737  002646  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
002632  013737  002060  001122          MOV     TERMCH,$BDADR   ;SET UP TERMCH ADRS
002640  012737  000100  001124          MOV     #100,$GDDAT     ;EXPECT 100 INITIALLY
002646  013700  001124          1$:     MOV     $GDDAT,R0       ;PUT PATTERN IN R0
002652  052700  000200                  BIS     #200,R0         ;SET THE ENABLE BIT
002656  010077  177176                  MOV     R0,@TERMCH      ;LOAD UP TERMCH REG
002662  017737  177172  001126          MOV     @TERMCH,$BDDAT  ;READ IT BACK
002670  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
002676  001401                          BEQ     2$              ;BR IF OK
002700  104017                          ERROR   17              ;TERMINATE CHARACTER READ/WRITE EP
002702  006237  001124          2$:     ASR     $GDDAT          ;SHIFT ONE BIT RIGHT
002706  001357                          BNE     1$              ;BR IF NOT DONE
                                ;;*****************************************************************
                                ;*TEST 3         FLOAT A ZERO THRU TERMINATE CHARACTER REG
                                ;;*****************************************************************
002710  000004                  TST3:   SCOPE
002712  012737  000003  001206          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
002720  012737  002746  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
002726  013737  002060  001122          MOV     TERMCH,$BDADR   ;SET UP TERMCH ADRS
002734  012737  000077  001124          MOV     #77,$GDDAT      ;EXPECT 77 INITIALLY
002742  012700  177677                  MOV     #177677,R0      ;PUT PATTERN IN R0
002746  010077  177106          1$:     MOV     R0,@TERMCH      ;LOAD UP TERMCH REG - SEND HIGH BITS FOR NOISE
002752  017737  177102  001126          MOV     @TERMCH,$BDDAT  ;READ IT BACK
002760  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
002766  001401                          BEQ     2$              ;BR IF OK
002770  104017                          ERROR   17              ;TERMINATE CHARACTER READ/WRITE ER
002772  006200                  2$:     ASR     R0              ;SHIFT ZERO BIT RIGHT
002774  006237  001124                  ASR     $GDDAT          ;SHIFT PATTERN IN COMPARE LOC
003000  052737  000100  001124          BIS     #100,$GDDAT     ;SET MSB
003006  103757                          BCS     1$              ;BR IF NOT DONE
```

```
                                        ;;*****************************************************************
                                        ;*TEST 4         TEST TERMINATE CHARACTER REG IS NOT WRITEABLE WHEN CHANGE ENA=0
                                        ;;*****************************************************************
003010  000004                  TST4:   SCOPE
003012  012737  000004  001206          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
003020  013737  002060  001122          MOV     TERMCH,$BDADR   ;SET UP TERMCH ADRS
003026  012737  000177  001124          MOV     #177,$GDDAT     ;SET UP DATA PATTERN
003034  012777  177777  177016          MOV     #177777,@TERMCH ;LOAD UP ALL WRITEABLE BITS
003042  005077  177012                  CLR     @TERMCH         ;TRY TO CLEAR ALL BITS
003046  017737  177006  001126          MOV     @TERMCH,$BDDAT  ;READ IT BACK
003054  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SEE IF ANY BITS WERE CLEARED
003062  001401                          BEQ     TST5            ;;ADVANCE TO NEXT TEST IF OK
003064  104017                          ERROR   17              ;TERMINATE CHARACTER ENABLE CHANGE ER

                                        ;;*****************************************************************
                                        ;*TEST 5         FLOAT A ONE THRU Y DYNAMIC OFFSET REG
                                        ;;*****************************************************************
003066  000004                  TST5:   SCOPE
003070  012737  000005  001206          MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
003076  012737  003120  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
003104  013737  002046  001122          MOV     YDOFF,$BDADR    ;SET UP YDOFF ADRS
003112  012737  004000  001124          MOV     #4000,$GDDAT    ;SET UP MSB TO A ONE
003120  013777  001124  176720  1$:     MOV     $GDDAT,@YDOFF   ;LOAD UP YDOFF REG
003126  017737  176714  001126          MOV     @YDOFF,$BDDAT   ;READ IT BACK
003134  042737  170000  001126          BIC     #170000,$BDDAT  ;SAVE ONLY OFFSET VALUE
003142  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
003150  001401                          BEQ     2$              ;BR IF OK
003152  104020                          ERROR   20              ;Y OFFSET READ/WRITE ER
003154  006237  001124          2$:     ASR     $GDDAT          ;SHIFT ONE BIT RIGHT
003160  001357                          BNE     1$              ;BR IF NOT DONE

                                        ;;*****************************************************************
                                        ;*TEST 6         FLOAT A ZERO THRU Y DYNAMIC OFFSET REG
                                        ;;*****************************************************************
003162  000004                  TST6:   SCOPE
003164  012737  000006  001206          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
003172  012737  003212  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
003200  013737  002046  001122          MOV     YDOFF,$BDADR    ;SET UP YDOFF ADRS
003206  012700  173777                  MOV     #173777,R0      ;SET UP MSB TO A ZERO
003212  010037  001124          1$:     MOV     R0,$GDDAT       ;SAVE PATTERN FOR COMPARE
003216  042737  170000  001124          BIC     #170000,$GDDAT  ;GET RID OF HIGH BITS THAT ARE NOT WRITEABLE
003224  010077  176616                  MOV     R0,@YDOFF       ;LOAD UP YDOFF REG - SEND HIGH BITS FOR NOISE
003230  017737  176612  001126          MOV     @YDOFF,$BDDAT   ;READ IT BACK
003236  042737  170000  001126          BIC     #170000,$BDDAT  ;SAVE ONLY OFFSET VALUE
003244  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
003252  001401                          BEQ     2$              ;BR IF OK
003254  104020                          ERROR   20              ;Y OFFSET READ/WRITE ER
003256  006200                  2$:     ASR     R0              ;SHIFT ZERO BIT RIGHT
003260  103754                          BCS     1$              ;BR IF NOT DONE
003262  005077  176560                  CLR     @YDOFF          ;INSURE 0 Y OFFSET BEFORE ADVANCING

                                        ;;*****************************************************************
                                        ;*TEST 7         FLOAT A ONE THRU X DYNAMIC OFFSET REG
                                        ;;*****************************************************************
003266  000004                  TST7:   SCOPE
```

```
   (2)  003270  012737  000007  001206          MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
   500  003276  012737  003320  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
   501  003304  013737  002044  001122          MOV     XDOFF,$BDADR    ;SET UP XDOFF ADRS
   502  003312  012737  004000  001124          MOV     #4000,$GDDAT    ;SET UP MSB TO A ONE
   503  003320  013777  001124  176516  1$:     MOV     $GDDAT,@XDOFF   ;LOAD UP XDOFF REG
   504  003326  017737  176512  001126          MOV     @XDOFF,$BDDAT   ;READ IT BACK
   505  003334  042737  170000  001126          BIC     #170000,$BDDAT  ;SAVE ONLY OFFSET VALUE
   506  003342  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
   507  003350  001401                          BEQ     2$              ;BR IF OK
   508  003352  104021                          ERROR   21              ;X OFFSET READ/WRITE ER
   509  003354  006237  001124          2$:     ASR     $GDDAT          ;SHIFT ONE BIT RIGHT
   510  003360  001357                          BNE     1$              ;BR IF NOT DONE
   511
   512                                  ;;*******************************************************************
   (3)                                  ;*TEST 10        FLOAT A ZERO THRU X DYNAMIC OFFSET REG
   (3)                                  ;;*******************************************************************
   (2)  003362  000004                  TST10:  SCOPE
   (2)  003364  012737  000010  001206          MOV     #10,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   513  003372  012737  003412  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
   514  003400  013737  002044  001122          MOV     XDOFF,$BDADR    ;SET UP XDOFF ADRS
   515  003406  012700  173777                  MOV     #173777,R0      ;SET MBB TO A ZERO
   516  003412  010037  001124          1$:     MOV     R0,$GDDAT       ;SAVE PATTERN FOR COMPARE
   517  003416  042737  170000  001124          BIC     #170000,$GDDAT  ;GET RID IF HIGH BITS THAT ARE NOT WRITEABLE
   518  003424  010077  176414                  MOV     R0,@XDOFF       ;LOAD UP XDOFF REG - SEND HIGH BITS FOR NOISE
   519  003430  017737  176410  001126          MOV     @XDOFF,$BDDAT   ;READ IT BACK
   520  003436  042737  170000  001126          BIC     #170000,$BDDAT  ;SAVE ONLY OFFSET VALUE
   521  003444  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
   522  003452  001401                          BEQ     2$              ;BR IF OK
   523  003454  104021                          ERROR   21              ;X OFFSET READ/WRITE ER
   524  003456  006200                  2$:     ASR     R0              ;SHIFT ZERO BIT RIGHT
   525  003460  103754                          BCS     1$              ;BR IF NOT DONE
   526  003462  005077  176356                  CLR     @XDOFF          ;INSURE 0 X OFFSET BEFORE ADVANCING
   527
   528                                  ;;*******************************************************************
   (3)                                  ;*TEST 11        TEST THAT THE Y OFFSET POLARITY BIT CAN SET & RESET
   (3)                                  ;;*******************************************************************
   (2)  003466  000004                  TST11:  SCOPE
   (2)  003470  012737  000011  001206          MOV     #11,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   529  003476  012737  003524  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
   530  003504  013737  002066  001122          MOV     ZDOFF,$BDADR    ;SET UP REG 36 ADRS
   531  003512  012700  020000                  MOV     #20000,R0       ;R0 HAS BIT
   532  003516  012737  040000  001124          MOV     #40000,$GDDAT   ;LD EXPCTED
   533  003524  010077  176316          1$:     MOV     R0,@YDOFF       ;SEND IT OUT
   534  003530  017737  176332  001126          MOV     @ZDOFF,$BDDAT   ;READ AT REG 36
   535  003536  042737  037777  001126          BIC     #37777,$BDDAT   ;SAVE X & Y POLARITY BITS
   536  003544  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CORRECT?
   537  003552  001401                          BEQ     2$              ;BR IF SO
   538  003554  104020                          ERROR   20              ;Y OFFSET POLARITY READ/WRITE ER
   539  003556  005037  001124          2$:     CLR     $GDDAT          ;EXPECT ZERO
   540  003562  162700  020000                  SUB     #20000,R0       ;NOW RESET STATE
   541  003566  100356                          BPL     1$              ;BR IF NOT YET TESTED
   542
   543                                  ;;*******************************************************************
   (3)                                  ;*TEST 12        TEST THAT THE X OFFSET POLARITY BIT CAN SET & RESET
   (3)                                  ;;*******************************************************************
   (2)  003570  000004                  TST12:  SCOPE
```

```
(2)  003572  012737  000012  001206          MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
544  003600  012737  003626  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
545  003606  013737  002066  001122          MOV     ZDOFF,$BDADR    ;SET UP REG 34 ADRS
546  003614  012700  020000                  MOV     #20000,R0       ;R0 HAS BIT
547  003620  012737  100000  001124          MOV     #100000,$GDDAT  ;LD EXPECTED
548  003626  010077  176212          1$:     MOV     R0,@XDOFF       ;SEND IT OUT
549  003632  017737  176230  001126          MOV     @ZDOFF,$BDDAT   ;READ IT BACK
550  003640  042737  037777  001126          BIC     #37777,$BDDAT   ;SAVE X & Y POLARITY BITS
551  003646  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
552  003654  001401                          BEQ     2$              ;BR IF SO
553  003656  104021                          ERROR   21              ;X OFFSET POLARITY READ/WRITE ER
554  003660  005037  001124          2$:     CLR     $GDDAT          ;EXPECT ZERO
555  003664  162737  020000  001124          SUB     #20000,$GDDAT   ;ADVANCE TO RESET STATE
556  003672  100355                          BPL     1$              ;BR IF NOT YET TESTED
557
558                                  ;;*******************************************************************
(3)                                  ;*TEST 13        FLOAT A ONE THRU RELOCATE REG
(3)                                  ;;*******************************************************************
(2)  003674  000004                  TST13:  SCOPE
(2)  003676  012737  000013  001206          MOV     #13,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
559  003704  012737  003726  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
560  003712  013737  002040  001122          MOV     RLO,$BDADR      ;SET UP RLO ADRS
561  003720  012737  004000  001124          MOV     #4000,$GDDAT    ;SET UP MSB TO A ONE
562  003726  013777  001124  176104  1$:     MOV     $GDDAT,@RLO     ;LOAD UP RLO REG
563  003734  017737  176100  001126          MOV     @RLO,$BDDAT     ;READ IT BACK
564  003742  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
565  003750  001401                          BEQ     2$              ;BR IF OK
566  003752  104022                          ERROR   22              ;RELOCATE REG READ/WRITE ER
567  003754  006237  001124          2$:     ASR     $GDDAT          ;SHIFT ONE BIT RIGHT
568  003760  001362                          BNE     1$              ;BR IF NOT DONE
569
570                                  ;;*******************************************************************
(3)                                  ;*TEST 14        FLOAT A ZERO THRU RELOCATE REG
(3)                                  ;;*******************************************************************
(2)  003762  000004                  TST14:  SCOPE
(2)  003764  012737  000014  001206          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
571  003772  012737  004012  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
572  004000  013737  002040  001122          MOV     RLO,$BDADR      ;SET UP RLO ADRS
573  004006  012700  173777                  MOV     #173777,R0      ;SET MSB TO A ZERO
574  004012  010037  001124          1$:     MOV     R0,$GDDAT       ;SAVE PATTERN FOR COMPARE
575  004016  042737  170000  001124          BIC     #170000,$GDDAT  ;GET RID OF HIGH BITS THAT DO NOT READ/WRITE
576  004024  010077  176010                  MOV     R0,@RLO         ;LOAD UP RLO REG - SEND HIGH BITS FOR NOISE
577  004030  017737  176004  001126          MOV     @RLO,$BDDAT     ;READ IT BACK
578  004036  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
579  004044  001401                          BEQ     2$              ;BR IF OK
580  004046  104022                          ERROR   22              ;RELOCATE REG READ/WRITE ER
581  004050  006000                  2$:     ASR     R0              ;SHIFT ZERO BIT RIGHT
582  004052  103757                          BCS     1$              ;BR IF NOT DONE
583  004054  005077  175760                  CLR     @RLO            ;CLR RELOCATE REG BEFORE ADVANCING
584
585                                  ;;*******************************************************************
(3)                                  ;*TEST 15        TEST THAT ALL MAINT TYPE BITS ARE READ/WRITEABLE
(3)                                  ;;*******************************************************************
(2)  004060  000004                  TST15:  SCOPE
(2)  004062  012737  000015  001206          MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
586  004070  013737  002062  001122          MOV     STKPT,$BDADR    ;SET UP REG 32 ADRS
```

```
   587  004076  012737  004112  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
   588  004104  012737  170000  001124          MOV     #170000,$GDDAT  ;EXPECT ALL BITS INITIALLY
   589  004112  013777  001124  175742  1$:     MOV     $GDDAT,@STKPT   ;LOAD UP MAINT BITS
   590  004120  017737  175736  001126          MOV     @STKPT,$BDDAT   ;READ THEM BACK
   591  004126  042737  007777  001126          BIC     #7777,$BDDAT    ;SAVE ONLY MAINT SWITCHES
   592  004134  023737  001124  00[illegible]    CMP     $GDDAT,$BDDAT   ;ARE THEY CORRECT?
   593  004142  001401                          BEQ     2$              ;BR IF SO
   594  004144  104001                          ERROR   1               ;MAINT SWITCH READ/WRITE ERROR
   595  004146  162737  010000  00[illegible]  2$:     SUB     #10000,$GDDAT   ;GO TO NEXT PATTERN
   596  004154  100356                          BPL     1$              ;BR IF ALL STATES NOT TESTED
   597
   598                                  ;;*****************************************************************
   (3)                                  ;*TEST 16       TEST THAT RESET WILL CLEAR THE RELOCATE REG
   (3)                                  ;;*****************************************************************
   (2)  004156  000004                  TST16:  SCOPE
   (1)  004160  012737  000010  001166          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
   (2)  004166  012737  000016  001206          MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   599  004174  013737  002040  001122          MOV     RLO,$BDADR      ;SET UP REG 10 ADRS
   600  004202  005037  001124                  CLR     $GDDAT          ;EXPECT ALL ZEROS
   601  004206  012777  007777  175624          MOV     #7777,@RLO      ;LOAD IT UP
   602  004214  000005                          RESET                   ;CLR IT OUT
   603  004216  017737  175616  001126          MOV     @RLO,$BDDAT     ;READ IT
   604  004224  001401                          BEQ     TST17           ;;GO TO NEXT TEST IF CLEARED
   605  004226  104022                          ERROR   22              ;RESET FAILED TO CLR RELOCATE REG
   606
   607                                  ;;*****************************************************************
   (3)                                  ;*TEST 17       TEST THAT RESET WILL CLEAR THE X DYNAMIC OFFSET REG
   (3)                                  ;;*****************************************************************
   (2)  004230  000004                  TST17:  SCOPE
   (1)  004232  012737  000010  001166          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
   (2)  004240  012737  000017  001206          MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   608  004246  013737  002044  001122          MOV     XDOFF,$BDADR    ;SET UP REG 14 ADRS
   609  004254  005037  001124                  CLR     $GDDAT          ;EXPECT ALL ZEROS
   610  004260  012777  007777  175556          MOV     #7777,@XDOFF    ;LOAD IT UP
   611  004266  000005                          RESET                   ;CLR IT OUT
   612  004270  017737  175550  001126          MOV     @XDOFF,$BDDAT   ;READ JT
   613  004276  001401                          BEQ     TST20           ;;GO TO NEXT TEST IF CLEARED
   614  004300  104021                          ERROR   21              ;RESET FAILED TO CLR X DYNAMIC OFFSET REG
   615
   616                                  ;;*****************************************************************
   (3)                                  ;*TEST 20       TEST THAT RESET WILL CLEAR THE Y DYNAMIC OFFSET REG
   (3)                                  ;;*****************************************************************
   (2)  004302  000004                  TST20:  SCOPE
   (1)  004304  012737  000010  001166          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
   (2)  004312  012737  000020  001206          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   617  004320  013737  002046  001122          MOV     YDOFF,$BDADR    ;SET UP REG 16 ADRS
   618  004326  005037  001124                  CLR     $GDDAT          ;EXPECT ALL ZEROS
   619  004332  012777  007777  175506          MOV     #7777,@YDOFF    ;LOAD IT UP
   620  004340  000005                          RESET                   ;CLR IT OUT
   621  004342  017737  175500  001126          MOV     @YDOFF,$BDDAT   ;READ IT
   622  004350  001401                          BEQ     TST21           ;;GO TO NEXT TEST IF CLEARED
   623  004352  104020                          ERROR   20              ;RESET FAILED TO CLR Y DYNAMIC OFF SET REG
   624
   625                                  ;;*****************************************************************
   (3)                                  ;*TEST 21       TEST THAT RESET WILL CLEAR X & Y OFFSET POLARITY BITS
   (3)                                  ;;*****************************************************************
```

```
   (2)  004354  000004                        TST21:  SCOPE
   (1)  004356  012737  000040  001166                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
   (2)  004364  012737  000021  001206                MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   626  004372  013737  002066  001122                MOV     ZDOFF,$BDADR    ;SET UP REG 36 ADRS
   627  004400  005037  001124                        CLR     $GDDAT          ;EXPECT ZERO
   628  004404  012777  020000  175432                MOV     #20000,@XDOFF   ;SET X POLARIYT BIT
   629  004412  012777  020000  175426                MOV     #20000,@YDOFF   ;SET Y POLARITY BIT
   630  004420  000005                                RESET                   ;CLR THEM
   631  004422  017737  175440  001126                MOV     @ZDOFF,$BDDAT   ;READ REG 36
   632  004430  042737  037777  001126                BIC     #37777,$BDDAT   ;SAVE ONLY POLARITY BITS
   633  004436  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;CORRECT?
   634  004444  001401                                BEQ     TST22           ;;NEXT TEST IF SO
   635  004446  104032                                ERROR   32              ;RESET FAILED TO CLR OFFSET POLARIY BITS
   636
   637                                        ;;*****************************************************************
   (3)                                        ;*TEST 22       TEST THAT RESET WILL CLEAR TERMINATE CHARACTER REG
   (3)                                        ;;*****************************************************************
   (2)  004450  000004                        TST22:  SCOPE
   (1)  004452  012737  000010  001166                MOV     #10,$TIMES      ;;DO 10 ITERATIONS
   (2)  004460  012737  000022  001206                MOV     #22,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   638  004466  013737  002060  001122                MOV     TERMCH,$BDADR   ;SET UP REG 30 ADRS
   639  004474  005037  001124                        CLR     $GDDAT          ;EXPECT ALL ZEROS
   640  004500  012777  000377  175352                MOV     #377,@TERMCH    ;LOAD IT UP
   641  004506  000005                                RESET                   ;CLR IT OUT
   642  004510  017737  175344  001126                MOV     @TERMCH,$BDDAT  ;READ IT
   643  004516  001401                                BEQ     TST23           ;;GO TO NEXT TEST IF CLEARED
   644  004520  104017                                ERROR   17              ;RESET FAILED TO CLR TERMINATE CHAR REG
   645
   646                                        ;;*****************************************************************
   (3)                                        ;*TEST 23       TEST THAT RESET CLEARS ALL MAINT BITS
   (3)                                        ;;*****************************************************************
   (2)  004522  000004                        TST23:  SCOPE
   (1)  004524  012737  000100  001166                MOV     #100,$TIMES     ;;DO 100 ITERATIONS
   (2)  004532  012737  000023  001206                MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   647  004540  013737  002062  001122                MOV     STKPT,$BDADR    ;SET UP REG ADRS 32
   648  004546  005037  001124                        CLR     $GDDAT          ;LOAD EXPECTED
   649  004552  012777  170000  175302                MOV     #170000,@STKPT  ;LOAD REGISTER
   650  004560  000005                                RESET                   ;INITILIZE
   651  004562  017737  175274  001126                MOV     @STKPT,$BDDAT   ;READ REG.
   652  004570  042737  003777  001126                BIC     #3777,$BDDAT    ;MASK OUT OTHER BITS
   653  004576  001401                                BEQ     TST24           ;;BR IF ALL CLEARED
   654  004600  104032                                ERROR   32              ;RESET FAILED TO CLEAR MAINT. BITS
   655
   656                                        ;;*****************************************************************
   (3)                                        ;*TEST 24       TEST THAT THE STACK SELECTION BITS ARE READ/WRITEABLE
   (3)                                        ;;*****************************************************************
   (2)  004602  000004                        TST24:  SCOPE
   (2)  004604  012737  000024  001206                MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
   657  004612  012737  004634  001110                MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
   658  004620  013737  002062  001122                MOV     STKPT,$BDADR    ;SET UP REG ADRS 32
   659  004626  012737  000037  001124                MOV     #37,$GDDAT      ;EXPECT ALL BITS INITIALLY
   660  004634  013777  001124  175220  1$:           MOV     $GDDAT,@STKPT   ;SEND PATTERN OUT
   661  004642  017737  175214  001126                MOV     @STKPT,$BDDAT   ;READ IT BACK
   662  004650  042737  177740  001126                BIC     #177740,$BDDAT  ;SAVE ONLY STACK SELECTION BITS
   663  004656  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;CORRECT?
   664  004664  001401                                BEQ     2$              ;BR IF SO
```

```
                                        ERROR   51              ;STACK POINTER READ/WRITE ER
004670  005337  001124        2$:       DEC     $GDDAT          ;ADVANCE PATTERN
004674  100357                          BPL     1$              ;BR IF ALL PATTERNS NOT TESTED

                                ;;*******************************************************************
                                ;*TEST 25        TEST THAT 'TOP OF STACK' AND RESET CLEARS STACK SELECTION BITS
                                ;*******************************************************************
004676  000004                  TST25:  SCOPE
004700  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
004706  012737  000025  001206          MOV     #25,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
004714  013737  002062  001122          MOV     STKPT,$BDADR    ;SET UP REG ADRS 32
004722  012737  000041  001124          MOV     #41,$GDDAT      ;LOAD EXPECTED
004730  012777  000037  175124          MOV     #37,@STKPT      ;SET ALL BITS
004736  012777  000077  175116          MOV     #77,@STKPT      ;CLR ALL BITS
004744  017737  175112  001126          MOV     @STKPT,$BDDAT   ;READ STACK SELECTION BITS
004752  042737  177700  001126          BIC     #177700,$BDDAT  ;SAVE SEL BITS ONLY
004760  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CORRECT?
004766  001401                          BEQ     1$              ;BR IF SO
004770  104051                          ERROR   51              ;'TOP OF STACK' FAILED TO CLR BITS
004772  012737  000040  001124  1$:     MOV     #40,$GDDAT      ;LD EXPECTED
005000  012777  000037  175054          MOV     #37,@STKPT      ;SET ALL BITS
005006  000005                          RESET                   ;RESET CLRS THEM & SET 'TOP OF STACK'
005010  017737  175046  001126          MOV     @STKPT,$BDDAT   ;READ THEM
005016  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CORRECT?
005024  001401                          BEQ     TST26           ;;NEXT TEST IF SO
005026  104051                          ERROR   51              ;RESET FAILED TO SET UP 'TOP OF STACK'

                                        .SBTTL
                                        .SBTTL  THESE TESTS USE MAINT SWITCH #1
                                        .SBTTL

                                ;;*******************************************************************
                                ;*TEST 26        FLOAT A 1 ACROSS THE D.P.C.
                                ;*******************************************************************
005030  000004                  TST26:  SCOPE
005032  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
005040  012737  000026  001206          MOV     #26,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
005046  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
005054  012737  000002  001124          MOV     #BIT1,$GDDAT    ;LOAD EXPECTED
005062  012777  010000  174772          MOV     #BIT12,@STKPT           ;SET MAINT SWITCH #1
005070  013777  001124  174732  1$:     MOV     $GDDAT,@DPC     ;LOAD D.P.C.
005076  017737  174726  001126          MOV     @DPC,$BDDAT     ;READ D.P.C. INTO $BDDAT
005104  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;COMPARE VALUE EXPECTED TO READ
005112  001402                          BEQ     2$              ;;BR IF SAME
005114  104002                          ERROR   2               ;FAILED TO LOAD DPC WITH A FLOATING 1
005116  000403                          BR      TST27           ;;
005120  006337  001124          2$:     ASL     $GDDAT          ;CHANGE DATA EXPECTED
005124  103361                          BCC     1$              ;BR IF NOT COMPLETED

                                ;;*******************************************************************
                                ;*TEST 27        TEST THAT THE RELOCATE REG WILL CORRECTLY SET UP THE DPC
                                ;*******************************************************************
005126  000004                  TST27:  SCOPE
005130  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
005136  012737  000027  001206          MOV     #27,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
005144  012737  005204  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
```

```
706  005152  012777  010000  174702          MOV     #BIT12,@STKPT           ;SET MAINT SWITCH #1
707  005160  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG 00 ADRS
708  005166  005077  174636                  CLR     @DPC            ;CLEAR DPC
709  005172  012737  000100  001124          MOV     #100,$GDDAT     ;EXPECT 100 INITIALLY
710  005200  012700  000001                  MOV     #1,R0           ;START WITH 1
711  005204  010077  174630          1$:     MOV     R0,@RLO         ;LOAD RELOCATE REG
712  005210  017737  174614  001126          MOV     @DPC,$BDDAT     ;READ RELOCATE REG - SHOULD BE TIMES 64
713  005216  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
714  005224  001401                          BEQ     2$              ;BR IF SO
715  005226  104022                          ERROR   22              ;RELOCATE FAILED TO SET UP DPC CORRECTLY
716  005230  062737  000100  001124  2$:     ADD     #100,$GDDAT     ;SET UP NEXT EXPECTED VALUE
717  005236  001402                          BEQ     TST30           ;;GO TO NEXT TEST IF ALL VALUES TESTED
718  005240  005200                          INC     R0              ;SET UP NEXT PATTERN
719  005242  000760                          BR      1$              ;GO TRY IT
720
721                                  ;;*******************************************************************
(3)                                  ;*TEST 30       TEST THAT RESET CLEARS THE D.P.C.
(3)                                  ;;*******************************************************************
(2)  005244  000004                  TST30:  SCOPE
(1)  005246  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
(2)  005254  012737  000030  001206          MOV     #30,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
722  005262  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
723  005270  012777  010000  174564          MOV     #BIT12,@STKPT           ;SET MAINT SWITCH #1
724  005276  005037  001124                  CLR     $GDDAT          ;LOAD EXPECTED
725  005302  012777  177776  174520          MOV     #-2,@DPC        ;LOAD REGISTER
726  005310  000005                          RESET                   ;INITILIZE
727  005312  017737  174512  001126          MOV     @DPC,$BDDAT     ;READ D.P.C. REG.
728  005320  001401                          BEQ     TST31           ;;BR IF ALL BITS CLEARED
729  005322  104032                          ERROR   32              ;RESET FAILED TO CLEAR THE D.P.C.
730
731                                          .SBTTL
732                                          .SBTTL  THESE TESTS USE MAINT. SWITCH 2
733                                          .SBTTL
734
735                                  ;;*******************************************************************
(3)                                  ;*TEST 31       LOAD #17 INTO THE 'MODE' REGISTERS
(3)                                  ;;*******************************************************************
(2)  005324  000004                  TST31:  SCOPE
(2)  005326  012737  000031  001206          MOV     #31,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
736  005334  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
737  005342  012777  020000  174512          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
738  005350  012737  174000  031076          MOV     #174000,BUFFER  ;LOAD MODE REGISTER=17
739  005356  013777  002000  174444          MOV     DBUF,@DPC       ;LOAD DISPLAY PC.
740  005364  012737  074000  001124          MOV     #74000,$GDDAT   ;LOAD EXPECTED
741  005372  017737  174434  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
742  005400  042737  103777  001126          BIC     #103777,$BDDAT  ;MASK TO BITS 14-11
743  005406  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
744  005414  001401                          BEQ     TST32           ;;BR IF EQUAL
745  005416  104003                          ERROR   3               ;MODE BITS (14-11) FAILED TO SET
746
747
748                                  ;;*******************************************************************
(3)                                  ;*TEST 32       LOAD #10 INTO THE 'MODE' REGISTERS
(3)                                  ;;*******************************************************************
(2)  005420  000004                  TST32:  SCOPE
(2)  005422  012737  000032  001206          MOV     #32,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
```

```
 749  005430  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 750  005436  012777  020000  174416          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 751  005444  012737  140000  031076          MOV     #140000,BUFFER  ;LOAD MODE REGISTER=10
 752  005452  013777  002000  174350          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 753  005460  012737  040000  001124          MOV     #40000,$GDDAT           ;LOAD EXPECTED
 754  005466  017737  174340  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 755  005474  042737  103777  001126          BIC     #103777,$BDDAT  ;MASK TO BITS 14-11
 756  005502  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 757  005510  001401                          BEQ     TST33           ;;BR IF EQUAL
 758  005512  104003                          ERROR   3               ;MODE BIT 14 FAILED TO SET
 759
 760                                  ;;*****************************************************************
 (3)                                  ;*TEST 33       LOAD #14 INTO THE 'MODE'' REGISTERS
 (3)                                  ;;*****************************************************************
 (2)  005514  000004                  TST33:  SCOPE
 (2)  005516  012737  000033  001206          MOV     #33,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 761  005524  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 762  005532  012777  020000  174322          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 763  005540  012737  160000  031076          MOV     #160000,BUFFER  ;LOAD MODE REGISTER=14
 764  005546  013777  002000  174254          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 765  005554  012737  060000  001124          MOV     #60000,$GDDAT   ;LOAD EXPECTED
 766  005562  017737  174244  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 767  005570  042737  103777  001126          BIC     #103777,$BDDAT  ;MASK TO BITS 14-11
 768  005576  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 769  005604  001401                          BEQ     TST34           ;;BR IF EQUAL
 770  005606  104003                          ERROR   3               ;MODE BIT 13 FAILED TO SET
 771
 772                                  ;;*****************************************************************
 (3)                                  ;*TEST 34       LOAD #16 INTO THE 'MODE'' REGISTERS
 (3)                                  ;;*****************************************************************
 (2)  005610  000004                  TST34:  SCOPE
 (2)  005612  012737  000034  001206          MOV     #34,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 773  005620  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 774  005626  012777  020000  174226          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 775  005634  012737  170000  031076          MOV     #170000,BUFFER  ;LOAD MODE REGISTER=16
 776  005642  013777  002000  174160          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 777  005650  012737  070000  001124          MOV     #70000,$GDDAT   ;LOAD EXPECTED
 778  005656  017737  174150  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS PEGISTER
 779  005664  042737  103777  001126          BIC     #103777,$BDDAT  ;MASK TO BITS 14-11
 780  005672  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 781  005700  001401                          BEQ     TST35           ;;BR IF EQUAL
 782  005702  104003                          ERROR   3               ;MODE BIT 12 FAILED TO SET
 783
 (3)                                  ;*TEST 35       SIMPLE INCREMENT D.P.C. TEST
 (3)                                  ;;*****************************************************************
 (2)  005704  000004                  TST35:  SCOPE
 (2)  005706  012737  000035  001206          MOV     #35,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 784  005714  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
 785  005722  012777  020000  174132          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 786  005730  012700  031076                  MOV     #BUFFER,R0      ;LOAD BUFFER POINTER
 787  005734  012720  164000                  MOV     #164000,(R0)+   ;LOAD DISPLAY NOP
 788  005740  012720  164000                  MOV     #164000,(R0)+
 789  005744  012720  164000                  MOV     #164000,(R0)+
 790  005750  012720  164000                  MOV     #164000,(R0)+
 791  005754  013777  002000  174046          MOV     DBUF,@DPC       ;START THE DISPLAY
 792  005762  012737  031100  001124          MOV     #BUFFER+2,$GDDAT        ;LOAD EXPECTED
```

```
793  005770  017737  174034  001126          MOV     @DPC,$BDDAT             ;READ THE DISPLAY P.C.
794  005776  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST FOR INCREMENT
795  006004  001402                          BEQ     1$              ;
796  006006  104004                          ERROR   4               ;DISPLAY P.C. FAILED TO INCREMENT
797  006010  000451                          BR      TST36           ;;
798  006012  005277  174012          1$:     INC     @DPC            ;STEP THE DISPLAY P.C.
799  006016  062737  000002  001124          ADD     #2,$GDDAT       ;UPDATE EXPECTED
800  006024  017737  174000  001126          MOV     @DPC,$BDDAT     ;READ THE DISPLAY P.C.
801  006032  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST FOR INCREMENT
802  006040  001402                          BEQ     2$              ;
803  006042  104004                          ERROR   4               ;DISPLAY P.C. FAILED TO INCREMENT
804  006044  000433                          BR      TST36           ;;
805  006046                          2$:
(1)  006046  005277  173756                  INC     @DPC            ;SINGLE STEP THE DISPLAY
806  006052  062737  000002  001124          ADD     #2,$GDDAT       ;UPDATE EXPECTED
807  006060  017737  173744  001126          MOV     @DPC,$BDDAT     ;READ THE DISPLAY P.C.
808  006066  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST FOR INCREMENT
809  006074  001402                          BEQ     3$
810  006076  104004                          ERROR   4               ;DISPLAY P.C. FAILED TO INCREMENT
811  006100  000415                          BR      TST36           ;;
812  006102                          3$:
(1)  006102  005277  173722                  INC     @DPC            ;SINGLE STEP THE DISPLAY
813  006106  062737  000002  001124          ADD     #2,$GDDAT       ;UPDATE EXPECTED
814  006114  017737  173710  001126          MOV     @DPC,$BDDAT     ;READ THE DISPLAY P.C.
815  006122  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST FOR INCREMENT
816  006130  001401                          BEQ     TST36           ;;BR IF EQUAL
817  006132  104004                          ERROR   4               ;DISPLAY P.C. FAILED TO INCREMENT
818
819                                  ;;*****************************************************************
(3)                                  ;*TEST 36       EXPANDED D.P.C. INCREMENT TEST
(3)                                  ;;*****************************************************************
(2)  006134  000004                  TST36:  SCOPE
(1)  006136  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
(2)  006144  012737  000036  001206          MOV     #36,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
820  006152  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
821  006160  012777  020000  173674          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
822  006166  013702  002000                  MOV     DBUF,R2         ;SET UP POINTER
823  006172  012722  164000          1$:     MOV     #164000,(R2)+   ;MOVE DNOP INTO THE BUFFER
824  006176  023702  002020                  CMP     SIZE,R2         ;FINISHED FILLING THE BUFFER?
825  006202  001373                          BNE     1$              ;NO
826  006204  013777  002000  173616          MOV     DBUF,@DPC       ;YES, START THE DISPLAY
827  006212  012737  031076  001124          MOV     #BUFFER,$GDDAT  ;LOAD EXPECTED
828  006220  013702  002020                  MOV     SIZE,R2         ;SETUP A COUNT
829  006224  024242                          CMP     -(R2),-(R2)     ;DEC BY 2
830  006226  062737  000002  001124  2$:     ADD     #2,$GDDAT       ;UPDATE EXPECTED
831  006234  017737  173570  001126          MOV     @DPC,$BDDAT     ;READ DISPLAY P.C.
832  006242  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID IT INCREMENT BY 2?
833  006250  001402                          BEQ     3$              ;YES
834  006252  104004                          ERROR   4               ;DISPLAY PC FAILED TO INCREMENT
835  006254  000406                          BR      TST37           ;;
836
837  006256  020237  001124          3$:     CMP     R2,$GDDAT       ;FINISHED THE BUFFER
838  006262  001403                          BEQ     TST37           ;;BR TO NEXT TEST
839  006264  005277  173540                  INC     @DPC            ;SINGLE STEP THE DISPLAY
840  006270  000756                          BR      2$              ;;TRY AGAIN
841
```

```
842                                     ;;*****************************************************************
(3)                                     ;*TEST 37        TEST THAT THE NAME INSTR CAN FLOAT A ONE THRU DPU NAME REG
(3)                                     ;;*****************************************************************
(2) 006272 000004                       TST37:  SCOPE
(2) 006274 012737 000037 001206                 MOV     #37,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
843 006302 012777 020000 173552                 MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
844 006310 012737 006324 001110                 MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
845 006316 012737 002000 001124                 MOV     #2000,$GDDAT    ;SET UP MSB
846 006324 012737 150000 031076 1$:             MOV     #150000,BUFFER  ;SET UP INSTR LOC
847 006332 053737 001124 031076                 BIS     $GDDAT,BUFFER   ;SET NAME BIT AT INSTR LOC
848 006340 012777 031076 173462                 MOV     #BUFFER,@DPC    ;START
849 006346 013737 002054 001122                 MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
850 006354 017737 173474 001126                 MOV     @DNAME,$BDDAT   ;READ NAME REG
851 006362 042737 174000 001126                 BIC     #174000,$BDDAT  ;ONLY INTERESTED IN THE NAME BITS
852 006370 023737 001124 001126                 CMP     $GDDAT,$BDDAT   ;DID IT GET LOADED PROPERLY?
853 006376 001401                               BEQ     2$              ;BR IF OK
854 006400 104023                               ERROR   23              ;DPU NAME INSTR ER
855 006402 006237 001124        2$:             ASR     $GDDAT          ;SHIFT 'ONE' BIT RIGHT
856 006406 103346                               BCC     1$              ;BR IF MORE BIT POSITIONS
857
858                                     ;;*****************************************************************
(3)                                     ;*TEST 40        TEST THAT THE NAME INSTR CAN FLOAT A ZERO THRU DPU NAME REG
(3)                                     ;;*****************************************************************
(2) 006410 000004                       TST40:  SCOPE
(2) 006412 012737 000040 001206                 MOV     #40,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
859 006420 012777 020000 173434                 MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
860 006426 012737 006440 001110                 MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
861 006434 012700 175777                        MOV     #175777,R0      ;SET MSB TO A ZERO
862 006440 010037 001124        1$:             MOV     R0,$GDDAT       ;PUT PATTERN IN COMPARE LOC
863 006444 042737 174000 001124                 BIC     #174000,$GDDAT  ;GET RID OF EXTRA BITS
864 006452 012737 150000 031076                 MOV     #150000,BUFFER  ;SET UP INSTR
865 006460 053737 001124 031076                 BIS     $GDDAT,BUFFER   ;SET UP PATTERN AT INSTR LOC
866 006466 012777 031076 173334                 MOV     #BUFFER,@DPC    ;START
867 006474 013737 002054 001122                 MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
868 006502 017737 173346 001126                 MOV     @DNAME,$BDDAT   ;READ NAME REG
869 006510 042737 170000 001126                 BIC     #170000,$BDDAT  ;ONLY INTERESTED IN THE NAME BITS
870 006516 023737 001124 001126                 CMP     $GDDAT,$BDDAT   ;DID IT GET LOADED PROPERLY?
871 006524 001401                               BEQ     2$              ;BR IF OK
872 006526 104023                               ERROR   23              ;DPU NAME INSTR ER
873 006530 006200                       2$:     ASR     R0              ;SHIFT ZERO BIT RIGHT
874 006532 103742                               BCS     1$              ;BR IF MORE BIT POSITIONS
875
876                                     ;;*****************************************************************
(3)                                     ;*TEST 41        TEST THAT RESET WILL CLEAR DPU NAME
(3)                                     ;;*****************************************************************
(2) 006534 000004                       TST41:  SCOPE
(1) 006536 012737 000040 001166                 MOV     #40,$TIMES      ;;DO 40 ITERATIONS
(2) 006544 012737 000041 001206                 MOV     #41,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
877 006552 012777 020000 173302                 MOV     #BIT13,@STKPT   ;SET MAINT SWITCH 2
878 006560 005037 001124                        CLR     $GDDAT          ;EXPECT ALL ZEROS
879 006564 012737 153777 031076                 MOV     #153777,BUFFER  ;SET UP NAME INSTR
880 006572 012777 031076 173230                 MOV     #BUFFER,@DPC    ;START
881 006600 013737 002054 001122                 MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
882 006606 000005                               RESET                   ;CLR DPU NAME REG
883 006610 017737 173240 001126                 MOV     @DNAME,$BDDAT   ;READ IT
884 006616 001401                               BEQ     TST42           ;;NEXT TEST IF ALL CLEARED
```

```
 885  006620  104023                          ERROR   23              ;RESET FAILED TO CLEAR DPU NAME
 886
 887
 (3)                                  ;;*****************************************************************
 (3)                                  ;*TEST 42       TEST THAT THE SEARCH CODE BITS CAN BE READ AT REG 24
 (3)                                  ;;*****************************************************************
 (2)  006622  000004                  TST42:  SCOPE
 (2)  006624  012737  000042  001206          MOV     #42,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 888  006632  012737  006660  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
 889  006640  013737  002054  001122          MOV     DNAME,$BDADR    ;SET UP REG 24 ADRS
 890  006646  012737  030000  001124          MOV     #30000,$GDDAT   ;EXPECT BOTH BITS INITIALLY
 891  006654  012700  070000                  MOV     #70000,R0       ;SET UP R0 WITH ENABLE
 892  006660  010077  173164          1$:     MOV     R0,@ANAME               ;SEND SEARCH CODE
 893  006664  017737  173164  001126          MOV     @DNAME,$BDDAT   ;READ IT BACK
 894  006672  042737  147777  001126          BIC     #147777,$BDDAT  ;SAVE ONLY SEARCH CODE
 895  006700  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 896  006706  001401                          BEQ     2$              ;BR IF OK
 897  006710  104024                          ERROR   24              ;SEARCH CODE READ/WRITE ER
 898  006712  162700  010000          2$:     SUB     #10000,R0       ;SET UP NEXT SEARCH CODE
 899  006716  162737  010000  001124          SUB     #10000,$GDDAT   ;SET UP NEXT SEARCH CODE
 900  006724  100355                          BPL     1$              ;BR IF MORE CODES TO TEST
 901
 902                                  ;;*****************************************************************
 (3)                                  ;*TEST 43       TEST THAT THE SEARCH CODE DOES NOT CHANGE WHEN ENA-0
 (3)                                  ;;*****************************************************************
 (2)  006726  000004                  TST43:  SCOPE
 (2)  006730  012737  000043  001206          MOV     #43,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 903  006736  013737  002054  001122          MOV     DNAME,$BDADR    ;SET UP REG 24 ADRS
 904  006744  012737  030000  001124          MOV     #30000,$GDDAT   ;EXPECT BOTH BITS
 905  006752  012777  070000  173070          MOV     #70000,@ANAME   ;LOAD UP SEARCH CODE
 906  006760  005077  173064                  CLR     @ANAME          ;TRY TO CLR SEARCH CODE WITH ENA 0
 907  006764  017737  173064  001126          MOV     @DNAME,$BDDAT   ;READ IT BACK
 908  006772  042737  147777  001126          BIC     #147777,$BDDAT  ;SAVE ONLY SEARCH CODE
 909  007000  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 910  007006  001401                          BEQ     TST44           ;;ADVANCE TO NEXT TEST IF OK
 911  007010  104024                          ERROR   24              ;SEARCH CODE CHANGE ENABLE ER
 912
 913                                  ;;*****************************************************************
 (3)                                  ;*TEST 44       TEST THAT RESET WILL CLEAR SEARCH CODE
 (3)                                  ;;*****************************************************************
 (2)  007012  000004                  TST44:  SCOPE
 (1)  007014  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 (2)  007022  012737  000044  001206          MOV     #44,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 914  007030  013737  002054  001122          MOV     DNAME,$BDADR    ;SET UP REG 24 ADRS
 915  007036  005037  001124                  CLR     $GDDAT          ;EXPECT RESET STATE
 916  007042  012777  070000  173000          MOV     #70000,@ANAME   ;SET BOTH SEARCH CODE BITS
 917  007050  000005                          RESET                   ;CLR THEM
 918  007052  017737  172776  001126          MOV     @DNAME,$BDDAT   ;READ REG
 919  007060  001401                          BEQ     TST45           ;;NEXT TEST IF ALL CLEARED
 920  007062  104024                          ERROR   24              ;RESET FAILED TO CLR SEARCH CODE
 921
 922                                  ;;*****************************************************************
 (3)                                  ;*TEST 45       TEST THAT THE NAME MATCH FLAG WILL SET ON 4 BIT COMPARES
 (3)                                  ;;*****************************************************************
 (2)  007064  000004                  TST45:  SCOPE
 (1)  007066  012737  000010  001166          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
 (2)  007074  012737  000045  001206          MOV     #45,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
```

```
  923  007102  012777  020000  172752          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
  924  007110  012737  000340  177776          MOV     #340,PSW        ;SET UP PRIORITY TO HIGHEST LEVEL
  925  007116  012737  007152  001110          MOV     #2$,$LPERR      ;SET UP SCOPE LOOP ADRS
  926  007124  012737  033600  001124          MOV     #33600,$GDDAT   ;START WITH MAX VALUE
  927  007132  052737  044000  001124  1$:     BIS     #44000,$GDDAT   ;SET THE SEARCH CODE ENA BIT
  928  007140  013777  001124  172702          MOV     $GDDAT,@ANAME   ;SET UP ASSOCIATIVE NAME REG
  929  007146  012700  003600                  MOV     #3600,R0        ;SET UP R0 WITH MAX VALUE
  930  007152  042737  144000  001124  2$:     BIC     #144000,$GDDAT  ;CLR MATCH FLAG & SEARCH ENA BIT
  931  007160  012737  150000  031076          MOV     #150000,BUFFER  ;SET UP NAME INSTR
  932  007166  050037  031076                  BIS     R0,BUFFER       ;SET UP NAME VALUE
  933  007172  012777  031076  172630          MOV     #BUFFER,@DPC    ;START
  934  007200  013737  002054  001122          MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
  935  007206  017737  172642  001126          MOV     @DNAME,$BDDAT   ;READ NAME REC
  936  007214  100410                          BMI     3$              ;BR IF A MATCH OCCURRED
  937  007216  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SHOULD A MATCH HAVE OCCURRED?
  938  007224  001017                          BNE     4$              ;BR IF NOT
  939  007226  052737  100000  001124          BIS     #100000,$GDDAT  ;INDICATE IT SHOULD BE SET
  940  007234  104024                          ERROR   24              ;NAME MATCH FLAG FAILED TO SET
  941  007236  042737  100000  001126  3$:     BIC     #100000,$BDDAT  ;CLR OUT MATCH FLAG BIT
  942  007244  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;MAKE SURE THERE IS A MATCH
  943  007252  001404                          BEQ     4$              ;BR IF OK
  944  007254  052737  100000  001126          BIS     #100000,$BDDAT  ;RESTORE THE MATCH FLAG BIT
  945  007262  104024                          ERROR   24              ;NAME MATCH FLAG SET IN ER
  946  007264  162700  000200          4$:     SUB     #200,R0         ;ADVANCE NAME VALUE
  947  007270  100330                          BPL     2$              ;BR IF MORE VALUES TO TEST
  948  007272  162737  000200  001124          SUB     #200,$GDDAT     ;ADVANCE VALUE FOR ASSO. NAME REG
  949  007300  022737  027600  001124          CMP     #27600,$GDDAT   ;HAVE ALL VALUES BEEN TESTED?
  950  007306  001311                          BNE     1$              ;BR IF NOT
  951
  952                                  ;;*****************************************************************
  (3)                                  ;*TEST 46       TEST THAT THE NAME MATCH FLAG WILL SET ON 8 BIT COMPARES
  (3)                                  ;;*****************************************************************
  (2)  007310  000004                  TST46:  SCOPE
  (1)  007312  012737  000004  001166          MOV     #4,$TIMES       ;;DO 4 ITERATIONS
  (2)  007320  012737  000046  001206          MOV     #46,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
  953  007326  012777  020000  172526          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
  954  007334  012737  000340  177776          MOV     #340,PSW        ;SET PRIORITY TO HIGEST LEVEL
  955  007342  012737  007376  001110          MOV     #2$,$LPERR      ;SET UP SCOPE LOOP ADRS
  956  007350  012737  023770  001124          MOV     #23770,$GDDAT   ;START WITH MAX VALUE
  957  007356  052737  044000  001124  1$:     BIS     #44000,$GDDAT   ;SET SEARCH ENA BIT
  958  007364  013777  001124  172456          MOV     $GDDAT,@ANAME   ;SET UP ASSOCIATIVE NAME REG
  959  007372  012700  003770                  MOV     #3770,R0        ;SET UP R0 WITH MAX VALUE
  960  007376  042737  144000  001124  2$:     BIC     #144000,$GDDAT  ;CLR MATCH FLAG & SEARCH ENA
  961  007404  012737  150000  031076          MOV     #150000,BUFFER  ;SET UP NAME INSTR
  962  007412  050037  031076                  BIS     R0,BUFFER       ;SET UP NAME VALUE
  963  007416  012777  031076  172404          MOV     #BUFFER,@DPC    ;START
  964  007424  013737  002054  001122          MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
  965  007432  017737  172416  001126          MOV     @DNAME,$BDDAT   ;READ NAME REG
  966  007440  100410                          BMI     3$              ;BR IF A MATCH OCCURED
  967  007442  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;SHOULD A MATCH HAVE OCCURED?
  968  007450  001017                          BNE     4$              ;BR IF NOT
  969  007452  052737  100000  001124          BIS     #100000,$GDDAT  ;INDICATE IT SHOULD BE SET
  970  007460  104024                          ERROR   24              ;NAME MATCH FLAG FAILED TO SET
  971  007462  042737  100000  001126  3$:     BIC     #100000,$BDDAT  ;CLR OUT MATCH FLAG BIT
  972  007470  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;MAKE SURE THERE IS A MATCH
  973  007476  001404                          BEQ     4$              ;BR IF OK
```

```
                                BIS     #100000,$BDDAT  ;RESTORE THE MATCH FLAG BIT
                                ERROR   24              ;NAME MATCH FLAG SET IN ER
                        4$:     SUB     #10,R0          ;ADVANCE NAME VALUE
                                BPL     2$              ;BR IF MORE VALUES TO TEST
                                SUB     #10,$GDDAT      ;ADVANCE VALUE FOR ASSO. NAME REG
                                CMP     #17770,$GDDAT   ;HAVE ALL VALUES BEEN TESTED?
                                BNE     1$              ;BR IF NOT

                        ;;*******************************************************************
                        ;*TEST 47       TEST THAT THE NAME MATCH FLAG WILL SET ON 11 BIT COMPARES
                        ;;*******************************************************************
                        TST47:  SCOPE
                                MOV     #2,$TIMES       ;;DO 2 ITERATIONS
                                MOV     #47,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
                                MOV     #2$,$LPERR      ;SET UP SCOPE LOOP ADRS
                                MOV     #10077,$GDDAT   ;START WITH 77 - WORK ONLY LOWER ORDER BITS
                        1$:     BIS     #44000,$GDDAT   ;SET SEARCH ENA BIT
                                MOV     $GDDAT,@ANAME   ;SET UP ASSOCIATIVE NAME REG
                                MOV     #77,R0          ;SET UP R0 WITH 77
                        2$:     BIC     #144000,$GDDAT  ;CLR MATCH FLAG & SEARCH ENA
                                MOV     #150000,BUFFER  ;SET UP NAME INSTR
                                BIS     R0,BUFFER       ;SET UP NAME VALUE
                                MOV     #BUFFER,@DPC    ;START
                                MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
                                NOP                     ;ALLOW TIME FOR NPR
                                MOV     @DNAME,$BDDAT   ;READ NAME REG
                                BMI     3$              ;BR IF A MATCH OCCURED
                                CMP     $GDDAT,$BDDAT   ;SHOULD A MATCH HAVE OCCURED?
                                BNE     4$              ;BR IF NOT
                                BIS     #100000,$GDDAT  ;INDICATE IT SHOULD BE SET
                                ERROR   24              ;NAME MATCH FLAG FAILED TO SET
                        3$:     BIC     #100000,$BDDAT  ;CLR OUT MATCH FLAG BIT
                                CMP     $GDDAT,$BDDAT   ;MAKE SURE THERE IS A MATCH
                                BEQ     4$              ;BR IF OK
                                BIS     #100000,$BDDAT  ;RESTORE THE MATCH FLAG BIT
                                ERROR   24              ;NAME MATCH FLAG SET IN ER
                        4$:     DEC     R0              ;ADVANCE NAME VALUE
                                BPL     2$              ;BR IF MORE VALUES TO TEST
                                DEC     $GDDAT          ;ADVANCE VALUE FOR ASSO. NAME REG
                                CMP     #7777,$GDDAT    ;HAVE ALL VALUES BEEN TESTED?
                                BNE     1$              ;BR IF NOT

                        ;;*******************************************************************
                        ;*TEST 50       TEST THAT RESET WILL CLEAR THE NAME MATCH FLAG
                        ;;*******************************************************************
                        TST50:  SCOPE
                                MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                MOV     #50,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                MOV     #BIT13,@STKPT   ;SET MAINT SWITCH 2
                                MOV     #57777,@ANAME   ;SET ANAME FOR 11 BIT COMPARE ON 3777
                                MOV     #153777,BUFFER  ;SET UP NAME INSTR
                                CLR     $GDDAT          ;EXPECT ALL ZEROS
                                MOV     #BUFFER,@DPC    ;START
                                MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
                                RESET                   ;CLR DPU NAME REG (MATCH)
```

```
1020  010032  017737  172016  001126          MOV     @DNAME,$BDDAT   ;READ IT
1021  010040  001401                          BEQ     TST51           ;;NEXT TEST IF ALL CLEARED
1022  010042  104023                          ERROR   23              ;RESET FAILED TO CLEAR NAME MATCH
1023
1024                                  ;;*****************************************************************
 (3)                                  ;*TEST 51       TEST THAT THE ASSOCIATIVE NAME REG DOES NOT CHANGE WITH ENA=0
 (3)                                  ;;*****************************************************************
 (2)  010044  000004                  TST51:  SCOPE
 (2)  010046  012737  000051  001206          MOV     #51,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1025  010054  012777  020000  172000          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1026  010062  012737  000340  177776          MOV     #340,PSW        ;DON'T ALLOW INTERRUPT
1027  010070  013737  002054  001122          MOV     DNAME,$BDADR    ;SET UP REG 24 ADRS
1028  010076  012777  057777  171744          MOV     #57777,@ANAME   ;SET UP FOR 11 BIT CMP ON VALUE #3777
1029  010104  012777  050000  171736          MOV     #50000,@ANAME   ;NOW TRY WRITING 0 WITH ENA DISABLED
1030  010112  012737  153777  031076          MOV     #153777,BUFFER  ;SET UP NAME INSTR WITH VALUE #3777
1031  010120  012777  031076  171702          MOV     #BUFFER,@DPC    ;START
1032  010126  012737  113777  001124          MOV     #113777,$GDDAT  ;EXPECT NAME MATCH ON 11 BIT CMP VALUE #3777
1033  010134  017737  171714  001126          MOV     @DNAME,$BDDAT   ;GET DPU NAME REG CONTENTS
1034  010142  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS THERE A MATCH?
1035  010150  001401                          BEQ     TST52           ;;ADVANCE TO NEXT TEST IF SO
1036  010152  104024                          ERROR   24              ;ASSOCIATIVE NAME CHANGE ENABLE ER
1037
1038                                  ;;*****************************************************************
 (3)                                  ;*TEST 52       TEST THAT AN INTERRUPT WILL OCCUR ON 11 BIT NAME MATCH COMPARES
 (3)                                  ;;*****************************************************************
 (2)  010154  000004                  TST52:  SCOPE
 (1)  010156  012737  000002  001166          MOV     #2,$TIMES       ;;DO 2 ITERATIONS
 (2)  010164  012737  000052  001206          MOV     #52,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1039  010172  012777  020000  171662          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1040  010200  012737  000340  177776          MOV     #340,PSW        ;SET PSW TO 7
1041  010206  012737  010266  001110          MOV     #2$,$LPERR      ;SET UP SCOPE LOOP ADRS
1042  010214  013701  001252                  MOV     $VECT1,R1       ;GET VECTOR ADRS
1043  010220  042701  160000                  BIC     #160000,R1      ; CLEAR PSW BITS        ** B **
1044  010224  012761  010342  000014          MOV     #3$,14(R1)      ;SET UP NAME INT SER ADRS
1045  010232  012761  000340  000016          MOV     #340,16(R1)     ;SET UP PSW TO 7 ON INT
1046  010240  012737  010077  001124          MOV     #10077,$GDDAT   ;START WITH MAX VALUE
1047  010246  052737  044000  001124  1$:     BIS     #44000,$GDDAT   ;SET SEACH CHANGE & NAME CHANGE ENABLES
1048  010254  013777  001124  171566          MOV     $GDDAT,@ANAME   ;SET UP ASSOCIATIVE NAME REG
1049  010262  012700  000077                  MOV     #77,R0          ;SET UP R0 WITH MAX VALUE
1050  010266  042737  044000  001124  2$:     BIC     #44000,$GDDAT   ;CLR ENA BITS
1051  010274  012737  150000  031076          MOV     #150000,BUFFER  ;SET UP NAME INSTR
1052  010302  050037  031076                  BIS     R0,BUFFER       ;SET UP NAME VALUE
1053  010306  012777  031076  171514          MOV     #BUFFER,@DPC    ;START
1054  010314  013737  002054  001122          MOV     DNAME,$BDADR    ;SET UP REG ADRS 24
1055  010322  005037  177776                  CLR     PSW             ;ALLOW INT TO OCCUR
1056  010326  017737  171522  001126          MOV     @DNAME,$BDDAT   ;READ THE NAME REG
1057  010334  100010                          BPL     4$              ;BR IF NO MATCH
1058  010336  104047                          ERROR   47              ;NAME MATCH FAILED TO INTERRUPT
1059  010340  000416                          BR      5$              ;GO LOOK FOR LOOP ON TEST SWITCH
1060  010342  022626                  3$:     CMP     (R6)+,(R6)+     ;FIX STK SINCE NO RTI
1061  010344  017737  171504  001126          MOV     @DNAME,$BDDAT   ;READ NAME REG
1062  010352  100401                          BMI     4$              ;BR IF MATCH FLAG SET
1063  010354  104047                          ERROR   47              ;NAME MATCH INTERRUPTED BUT NO FLAG
1064  010356  005300                  4$:     DEC     R0              ;ADVANCE NAME VALUE IN R0
1065  010360  100342                          BPL     2$              ;BR IF MORE VALUES TO TEST
1066  010362  005337  001124                  DEC     $GDDAT          ;ADVANCE VALUE FOR ASSO. NAME REG
```

```
1067  010366  022737  007777  001124          CMP     #7777,$GDDAT    ;SEE IF ALL VALUES HAVE BEEN TESTED
1068  010374  001324                          BNE     1$              ;BR IF NOT
1069  010376  004737  024454          5$:     JSR     PC,RSTVEC       ;RESTORE NAME MATCH VECTOR WITH HALT
1070
1071                                  ;;*******************************************************************
 (3)                                  ;*TEST 53       LOAD #1 INTO LINE TYPE REGISTERS
 (3)                                  ;;*******************************************************************
 (2)  010402  000004                  TST53:  SCOPE
 (2)  010404  012737  000053  001206          MOV     #53,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1072  010412  012777  020000  171442          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1073  010420  012777  100005  171352          MOV     #100005,@DBUF   ;LINE TYPE ENABLE =1 LINE TYPE -1
1074  010426  012737  000001  001124          MOV     #1,$GDDAT               ;LOAD EXPECTED
1075  010434  013777  002000  171366          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
1076  010442  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
1077  010450  017737  171356  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
1078  010456  042737  177774  001126          BIC     #177774,$BDDAT  ;MASK TO BITS 1-0
1079  010464  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1080  010472  001401                          BEQ     TST54           ;;BR IF EQUAL
1081  010474  104005                          ERROR   5               ;LINE BIT 0 FAILED TO SET
1082
1083                                  ;;*******************************************************************
 (3)                                  ;*TEST 54       LOAD #2 INTO LINE TYPE REGISTERS
 (3)                                  ;;*******************************************************************
 (2)  010476  000004                  TST54:  SCOPE
 (2)  010500  012737  000054  001206          MOV     #54,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1084  010506  012777  020000  171346          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1085  010514  012777  100006  171256          MOV     #100006,@DBUF   ;LINE TYPE ENABLE =1 LINE TYPE -2
1086  010522  012737  000002  001124          MOV     #2,$GDDAT               ;LOAD EXPECTED
1087  010530  013777  002000  171272          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
1088  010536  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
1089  010544  017737  171262  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
1090  010552  042737  177774  001126          BIC     #177774,$BDDAT  ;MASK TO BITS 1-0
1091  010560  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1092  010566  001401                          BEQ     TST55           ;;BR IF EQUAL
1093  010570  104005                          ERROR   5               ;LINE BIT 1 FAILED TO SET
1094
1095
1096                                  ;;*******************************************************************
 (3)                                  ;*TEST 55       CHECK LINE TYPE ENABLE GATE
 (3)                                  ;;*******************************************************************
 (2)  010572  000004                  TST55:  SCOPE
 (2)  010574  012737  000055  001206          MOV     #55,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1097  010602  012777  020000  171252          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1098  010610  012777  100007  171162          MOV     #100007,@DBUF   ;LINE TYPE ENABLE =1 LINE TYPE -3
1099  010616  013777  002000  171204          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
1100  010624  012777  100000  171146          MOV     #100000,@DBUF
1101  010632  012737  000003  001124          MOV     #3,$GDDAT               ;LOAD EXPECTED
1102  010640  013777  002000  171162          MOV     DBUF,@DPC
1103  010646  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
1104  010654  017737  171152  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
1105  010662  042737  177774  001126          BIC     #177774,$BDDAT  ;MASK TO BITS 1-0
1106  010670  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1107  010676  001401                          BEQ     TST56           ;;BR IF EQUAL
1108  010700  104005                          ERROR   5               ;LINE TYPE ENABLE FAILED TO INHIBIT
1109                                                                  ;CHANGING OF LINETYPE VALUE
1110
```

```
 1111                                   ;;*****************************************************************
  (3)                                   ;*TEST 56       LOAD #0 INTO BLINK REGISTERS
  (3)                                   ;;*****************************************************************
  (2) 010702 000004                     TST56:  SCOPE
  (2) 010704 012737 000056 001206               MOV     #56,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1112 010712 012777 020000 171142               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1113 010720 012777 100020 171052               MOV     #100020,@DBUF   ;BLINK ENABLE =1  BLINK =0
 1114 010726 005037 001124                      CLR     $GDDAT          ;LOAD EXPECTED
 1115 010732 013777 002000 171070               MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1116 010740 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1117 010746 017737 171060 001126               MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1118 010754 042737 177767 001126               BIC     #177767,$BDDAT  ;MASK TO BIT 3
 1119 010762 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1120 010770 001401                             BEQ     TST57           ;;BR IF EQUAL
 1121 010772 104027                             ERROR   27              ;BLINK BIT FAILED TO RESET
 1122                                   ;;*****************************************************************
  (3)                                   ;*TEST 57       LOAD #1 INTO BLINK REGISTERS
  (3)                                   ;;*****************************************************************
  (2) 010774 000004                     TST57:  SCOPE
  (2) 010776 012737 000057 001206               MOV     #57,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1123 011004 012777 020000 171050               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1124 011012 012737 100030 031076               MOV     #100030,BUFFER  ;BLINK ENABLE =1  BLINK =1
 1125 011020 012737 000010 001124               MOV     #10,$GDDAT              ;LOAD EXPECTED
 1126 011026 013777 002000 170774               MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1127 011034 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1128 011042 017737 170764 001126               MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1129 011050 042737 177767 001126               BIC     #177767,$BDDAT  ;MASK TO BIT 3
 1130 011056 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1131 011064 001401                             BEQ     TST60           ;;BR IF EQUAL
 1132 011066 104027                             ERROR   27              ;BLINK BIT FAILED TO SET
 1133
 1134
 1135                                   ;;*****************************************************************
  (3)                                   ;*TEST 60       CHECK BLINK ENABLE GATE
  (3)                                   ;;*****************************************************************
  (2) 011070 000004                     TST60:  SCOPE
  (2) 011072 012737 000060 001206               MOV     #60,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1136 011100 012777 020000 170754               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1137 011106 012777 100030 170664               MOV     #100030,@DBUF   ;LOAD BLINK ON
 1138 011114 013777 002000 170706               MOV     DBUF,@DPC
 1139 011122 012737 000010 001124               MOV     #BIT3,$GDDAT    ;LOAD EXPECTED
 1140 011130 012777 100000 170642               MOV     #100000,@DBUF   ;BLINK ENABLE =0  BLINK =0
 1141 011136 013777 002000 170664               MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1142 011144 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1143 011152 017737 170654 001126               MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1144 011160 042737 177767 001126               BIC     #177767,$BDDAT  ;MASK TO BIT 3
 1145 011166 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1146 011174 001401                             BEQ     TST61           ;;BR IF EQUAL
 1147 011176 104027                             ERROR   27              ;BLINK ENABLE FAILED TO INHIBIT
 1148                                                                   ;CHANGING OF THE BLINK BIT
 1149
 1150                                   ;;*****************************************************************
  (3)                                   ;*TEST 61       TEST FOR FALSE L.P. STATUS FLAG (L.P. - NOT ENABLED)
  (3)                                   ;;*****************************************************************
  (2) 011200 000004                     TST61:  SCOPE
  (2) 011202 012737 000061 001206               MOV     #61,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
```

```
 1151  011210  012777  020000  170644          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1152  011216  012737  100100  001124          MOV     #100100,$GDDAT  ;LP ENABLE =1 LP=0
 1153  011224  013777  001124  170546          MOV     $GDDAT,@DBUF            ;LOAD BUFFER
 1154  011232  013777  002000  170570          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1155  011240  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1156  011246  017737  170560  001126          MOV     @SREG0,$BDDAT           ;READ STATUS
 1157  011254  032737  000200  001126          BIT     #200,$BDDAT             ;
 1158  011262  001401                          BEQ     TST62           ;;BR IF EQUAL
 1159  011264  104034                          ERROR   34              ;LIGHT PEN FLAG SET IN ERROR
 1160
 1161                                  ;;**********************************************************************
  (3)                                  ;*TEST 62        TEST FOR FALSE L.P. STATUS FLAG (L.P. - ENABLED)
  (3)                                  ;;**********************************************************************
  (2)  011266  000004                  TST62:  SCOPE
  (2)  011270  012737  000062  001206          MOV     #62,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1162  011276  012777  020000  170556          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1163  011304  012737  100140  001124          MOV     #100140,$GDDAT  ;LP ENABLE =1 LP=1
 1164  011312  013777  001124  170460          MOV     $GDDAT,@DBUF            ;LOAD BUFFER
 1165  011320  013777  002000  170502          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1166  011326  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1167  011334  017737  170472  001126          MOV     @SREG0,$BDDAT           ;READ STATUS
 1168  011342  032737  000200  001126          BIT     #200,$BDDAT
 1169  011350  001401                          BEQ     TST63           ;;BR IF EQUAL
 1170  011352  104034                          ERROR   34              ;LIGHT PEN FLAG SET IN ERROR
 1171
 1172                                  ;;**********************************************************************
  (3)                                  ;*TEST 63        LOAD #4 INTO INT LEVEL REGISTER
  (3)                                  ;;**********************************************************************
  (2)  011354  000004                  TST63:  SCOPE
  (2)  011356  012737  000063  001206          MOV     #63,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1173  011364  012777  020000  170470          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1174  011372  012777  103000  170400          MOV     #103000,@DBUF   ;INTENSITY LEVEL ENABLE =1  LEVEL =4
 1175  011400  012737  002000  001124          MOV     #2000,$GDDAT            ;LOAD EXPECTED
 1176  011406  013777  002000  170414          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1177  011414  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1178  011422  017737  170404  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1179  011430  042737  174377  001126          BIC     #174377,$BDDAT  ;MASK TO BITS 8-10
 1180  011436  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1181  011444  001401                          BEQ     TST64           ;;BR IF EQUAL
 1182  011446  104006                          ERROR   6               ;INTENSITY LEVEL BIT 10 FAILED
 1183
 1184                                  ;;**********************************************************************
  (3)                                  ;*TEST 64        LOAD #2 INTO INT LEVEL REGISTER
  (3)                                  ;;**********************************************************************
  (2)  011450  000004                  TST64:  SCOPE
  (2)  011452  012737  000064  001206          MOV     #64,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1185  011460  012777  020000  170374          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1186  011466  012777  102400  170304          MOV     #102400,@DBUF   ;INTENSITY LEVEL ENABLE =1  LEVEL =2
 1187  011474  012737  001000  001124          MOV     #1000,$GDDAT            ;LOAD EXPECTED
 1188  011502  013777  002000  170320          MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1189  011510  013737  002032  001122          MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1190  011516  017737  170310  001126          MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1191  011524  042737  174377  001126          BIC     #174377,$BDDAT  ;MASK TO BITS 8-10
 1192  011532  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1193  011540  001401                          BEQ     TST65           ;;BR IF EQUAL
 1194  011542  104006                          ERROR   6               ;INTENSITY LEVEL BIT 9 FAILED
```

```
 1195
 1196                                   ;;*****************************************************************
  (3)                                   ;*TEST 65       LOAD #1 INTO INT LEVEL REGISTER
  (3)                                   ;;*****************************************************************
  (2) 011544 000004                     TST65:  SCOPE
  (2) 011546 012737 000065 001206               MOV     #65,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1197 011554 012777 020000 170300               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1198 011562 012777 102200 170210               MOV     #102200,@DBUF   ;INTENSITY LEVEL ENABLE =1  LEVEL =1
 1199 011570 012737 000400 001124               MOV     #400,$GDDAT             ;LOAD EXPECTED
 1200 011576 013777 002000 170224               MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1201 011604 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1202 011612 017737 170214 001126               MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1203 011620 042737 174377 001126               BIC     #174377,$BDDAT  ;MASK TO BITS 8-10
 1204 011626 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1205 011634 001401                             BEQ     TST66           ;;BR IF EQUAL
 1206 011636 104006                             ERROR   6               ;INTENSITY LEVEL BIT 8 FAILED
 1207
 1208
 1209                                   ;;*****************************************************************
  (3)                                   ;*TEST 66       CHECK INTENSITY LEVEL ENABLE GATE
  (3)                                   ;;*****************************************************************
  (2) 011640 000004                     TST66:  SCOPE
  (2) 011642 012737 000066 001206               MOV     #66,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1210 011650 012777 020000 170204               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1211 011656 012777 103200 170114               MOV     #103200,@DBUF   ;LOAD LEVEL
 1212 011664 012777 100000 170110               MOV     #100000,@DBUF1  ;INTENSITY LEVEL ENABLE =0  LEVEL =0
 1213 011672 013777 002000 170130               MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1214 011700 005277 170124                      INC     @DPC            ;SINGLE STEP THE DISPLAY
 1215 011704 012737 002400 001124               MOV     #2400,$GDDAT            ;LOAD EXPECTED
 1216 011712 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1217 011720 017737 170106 001126               MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1218 011726 042737 174377 001126               BIC     #174377,$BDDAT  ;MASK TO BITS 8-10
 1219 011734 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1220 011742 001401                             BEQ     TST67           ;;BR IF EQUAL
 1221 011744 104006                             ERROR   6               ;INTENSITY LEVEL ENABLE FAILED TO INHIBIT
 1222                                                                   ;INTENSITY LEVEL CHANGE
 1223
 1224                                   ;;*****************************************************************
  (3)                                   ;*TEST 67       LOAD #0 INTO ITALIC REGISTER
  (3)                                   ;;*****************************************************************
  (2) 011746 000004                     TST67:  SCOPE
  (2) 011750 012737 000067 001206               MOV     #67,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1225 011756 012777 020000 170076               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1226 011764 012777 170040 170006               MOV     #170040,@DBUF   ;ITALICS ENABLE=1  ITALICS=0
 1227 011772 005037 001124                      CLR     $GDDAT          ;LOAD EXPECTED
 1228 011776 013777 002000 170024               MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1229 012004 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1230 012012 017737 170014 001126               MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1231 012020 042737 177757 001126               BIC     #177757,$BDDAT  ;MASK TO BIT 4
 1232 012026 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1233 012034 001401                             BEQ     TST70           ;;BR IF EQUAL
 1234 012036 104030                             ERROR   30              ;ITALICS BIT FAILED TO RESET
 1235
 1236
 1237                                   ;;*****************************************************************
  (3)                                   ;*TEST 70       LOAD #1 INTO ITALIC REGISTER
```

```
  (3)                                   ;;*****************************************************************
  (2)  012040  000004                   TST70:  SCOPE
  (2)  012042  012737  000070  001206            MOV     #70,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1238  012050  012777  020000  170004            MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1239  012056  012777  170060  167714            MOV     #170060,@DBUF   ;ITALICS ENABLE=1  ITALICS=1
 1240  012064  012737  000020  001124            MOV     #BIT4,$GDDAT    ;LOAD EXPECTED
 1241  012072  013777  002000  167730            MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1242  012100  013737  002032  001122            MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1243  012106  017737  167720  001126            MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1244  012114  042737  177757  001126            BIC     #177757,$BDDAT  ;MASK TO BIT 4
 1245  012122  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1246  012130  001401                            BEQ     TST71           ;;BR IF EQUAL
 1247  012132  104030                            ERROR   30              ;ITALICS BIT FAILED TO SET
 1248
 1249
 1250                                   ;;*****************************************************************
  (3)                                   ;*TEST 71        CHECK ITALIC ENABLE GATE
  (3)                                   ;;*****************************************************************
  (2)  012134  000004                   TST71:  SCOPE
  (2)  012136  012737  000071  001206            MOV     #71,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1251  012144  012777  020000  167710            MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1252  012152  012777  170060  167620            MOV     #170060,@DBUF   ;LOAD ENABLE
 1253  012160  012777  170000  167614            MOV     #170000,@DBUF1  ;ITALICS ENABLE=0  ITALICS=0
 1254  012166  013777  002000  167634            MOV     DBUF,@DPC
 1255  012174  012737  000020  001124            MOV     #BIT4,$GDDAT    ;LOAD EXPECTED
 1256  012202  005277  167622                    INC     @DPC            ;SINGLE STEP THE DISPLAY
 1257  012206  013737  002032  001122            MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1258  012214  017737  167612  001126            MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1259  012222  042737  177757  001126            BIC     #177757,$BDDAT  ;MASK TO BITS 4
 1260  012230  023737  001124  001126            CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1261  012236  001401                            BEQ     TST72           ;;BR IF EQUAL
 1262  012240  104030                            ERROR   30              ;ITALICS ENABLE FAILED TO INHIBIT
 1263                                                                    ;CLEARING OF ITALICS BIT
 1264
 1265                                   ;;*****************************************************************
  (3)                                   ;*TEST 72        TEST INTERNAL STOP FLAG CAN SET AND RESET
  (3)                                   ;;*****************************************************************
  (2)  012242  000004                   TST72:  SCOPE
  (2)  012244  012737  000072  001206            MOV     #72,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 1266  012252  013737  002032  001122            MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
 1267  012260  012777  020000  167574            MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1268  012266  012737  172000  031076            MOV     #172000,BUFFER  ;''STOP'' BIT =1
 1269  012274  013777  002000  167526            MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
 1270  012302  012737  100000  001124            MOV     #BIT15,$GDDAT           ;LOAD EXPECTED
 1271  012310  017737  167516  001126            MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1272  012316  100401                            BMI     1$              ;;BR IF SET
 1273  012320  104037                            ERROR   37              ;''INTERNAL-STOP'' BIT FAILED TO SET
 1274  012322  012737  160000  031076  1$:       MOV     #160000,BUFFER  ;LOAD ''NOP''
 1275  012330  012777  031076  167472            MOV     #BUFFER,@DPC    ;START DISPLAY
 1276  012336  005037  001124                    CLR     $GDDAT          ;CLEAR EXPECTED
 1277  012342  017737  167464  001126            MOV     @SREG0,$BDDAT           ;READ DISPLAY STATUS REGISTER
 1278  012350  100001                            BPL     TST73           ;;BR IF CLEARED
 1279  012352  104040                            ERROR   40              ;''INTERNAL-STOP'' BIT FAILED TO RESET
 1280
 1281                                   ;;*****************************************************************
  (3)                                   ;*TEST 73        TEST THAT RESET WILL CLEAR STOP, MODE, INTENSITY, ITALICS, BLINK & LINE
```

```
  (3)                                   ;;*****************************************************************
  (2) 012354 000004                     TST73:  SCOPE
  (1) 012356 012737 000010 001166               MOV     #10,$TIMES      ;;DO 10 ITERATIONS
  (2) 012364 012737 000073 001206               MOV     #73,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1282  012372 012777 020000 167462               MOV     #BIT13,@STKPT   ;SET UP MAINT SW 2
1283  012400 012737 102637 031076               MOV     #102637,BUFFER  ;SET UP CHAR
1284  012406 012737 170060 031100               MOV     #170060,BUFFER+2  ;SET UP ITALICS INSTR
1285  012414 012777 031076 167406               MOV     #BUFFER,@DPC    ;START
1286  012422 012737 002000 001124               MOV     #2000,$GDDAT    ;EXPECT INTENSITY ONLY
1287  012430 005277 167374                      INC     @DPC            ;PICK UP ITALICS INSTR
1288  012434 012737 172000 031102               MOV     #172000,BUFFER+4        ;SET UP STOP INSTR
1289  012442 005277 167362                      INC     @DPC            ;PICK UP STOP INSTR
1290  012446 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
1291  012454 000005                             RESET                   ;CLR REG 02
1292  012456 017737 167350 001126               MOV     @SREG0,$BDDAT   ;READ REG
1293  012464 023737 001124 001126               CMP     $GDDAT,$BDDAT   ;ALL BUT THE MSB OF INTENSITY SHOULD HAVE BEEN CLEARED
1294  012472 001401                             BEQ     TST74           ;;NEXT TEST IF OK
1295  012474 104032                             ERROR   32              ;RESET FAILED TO SET UP REG 02
1296
1297                                    ;;*****************************************************************
  (3)                                   ;*TEST 74       TEST THAT INTERNAL STOP WILL CAUSE AN INTERRUPT
  (3)                                   ;;*****************************************************************
  (2) 012476 000004                     TST74:  SCOPE
  (1) 012500 012737 000100 001166               MOV     #100,$TIMES     ;;DO 100 ITERATIONS
1298  012506 012737 000340 177776               MOV     #340,PSW        ;HIGHEST PRIORTY
1299  012514 013737 002032 001122               MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
1300  012522 012737 172000 001124               MOV     #172000,$GDDAT  ;EXPECT STOP, MODE AND INTENSITY LEVEL ONLY
1301  012530 013700 001252                      MOV     $VECT1,R0       ;GET INTR ADRS
1302  012534 042700 160000                      BIC     #160000,R0      ; CLEAR PSW BITS        ** B **
1303  012540 012710 012620                      MOV     #2$,(R0)        ;SET UP RETURN ADRS
1304  012544 012760 000340 000002              MOV     #340,2(R0)      ;HIGHEST PRIORTY ON INTR
1305  012552 012777 020040 167302              MOV     #20040,@STKPT   ;SET MAINT SW 2
1306  012560 012737 173400 031076              MOV     #173400,BUFFER  ;ENABLE STOP INTR + STOP
1307  012566 012777 031076 167234       1$:     MOV     #BUFFER,@DPC    ;START
1308  012574 005037 177776                      CLR     PSW             ;ALLOW INTR
1309  012600 021616                             CMP     (SP),(SP)       ;FUMBLE
1310  012602 005700                             TST     R0              ;EXPECT INTR?
1311  012604 001430                             BEQ     5$              ;BR IF NOT
1312  012606 017737 167220 001126               MOV     @SREG0,$BDDAT   ;READ STATUS
1313  012614 104000                             ERROR                   ;INTERNAL STOP FAILED TO INTR
1314  012616 000423                             BR      5$              ;LOOK FOR LOOP ON TEST SW
1315  012620 022626                     2$:     CMP     (SP)+,(SP)+     ;FIX STACK SINCE NO RTI
1316  012622 017737 167204 001126               MOV     @SREG0,$BDDAT   ;READ REG 02
1317  012630 005700                             TST     R0              ;EXPECT INTR?
1318  012632 001002                             BNE     3$              ;BR IF SO
1319  012634 104033                             ERROR   33              ;INTERNAL STOP INTR'ED WHEN NOT ENABLED
1320  012636 000413                             BR      5$              ;GO LOOK FOR LOOP ON TEST SW
1321  012640 023737 001124 001126       3$:     CMP     $GDDAT,$BDDAT   ;STATUS CORRECT?
1322  012646 001402                             BEQ     4$              ;BR IF SO
1323  012650 104033                             ERROR   33              ;STOP FLAG NOT SET ON INTR
1324  012652 000405                             BR      5$              ;GO LOOK FOR LOOP ON TEST SW
1325  012654 012737 173000 031076       4$:     MOV     #173000,BUFFER  ;SET UP NOW WITH INTR NOT ENABLED
1326  012662 005000                             CLR     R0              ;INDICATE NO INTR
1327  012664 000740                             BR      1$              ;REPEAT TEST
1328  012666 004737 024454              5$:     JSR     PC,RSTVEC       ;GO RESTORE VECTORS
1329
```

```
                                        ;;*****************************************************************
                                        ;*TEST 75        LOAD #0 INTO MENU REGISTER
                                        ;;*****************************************************************
(2) 012672 000004                       TST75:  SCOPE
(2) 012674 012737 000075 001206                 MOV     #75,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1331 012702 012777 020000 167152                MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1332 012710 012777 170002 167062                MOV     #170002,@DBUF   ;MENU ENABLE=1  MENU=0
1333 012716 005037 001124                       CLR     $GDDAT          ;LOAD EXPECTED
1334 012722 013777 002000 167100                MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
1335 012730 013737 002042 001122                MOV     SREG1,$BDADR    ;SET UP REG ADRS12
1336 012736 017737 167100 001126                MOV     @SREG1,$BDDAT           ;READ DISP STATUS REGISTER
1337 012744 042737 177677 001126                BIC     #177677,$BDDAT  ;MASK TO BIT 2
1338 012752 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1339 012760 001401                              BEQ     TST76           ;;BR IF EQUAL
1340 012762 104031                              ERROR   31              ;MENU BIT FAILED TO RESET
1341
1342
1343                                    ;;*****************************************************************
                                        ;*TEST 76        LOAD #1 INTO MENU REGISTER
                                        ;;*****************************************************************
(2) 012764 000004                       TST76:  SCOPE
(2) 012766 012737 000076 001206                 MOV     #76,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1344 012774 012777 020000 167060                MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1345 013002 012777 170003 166770                MOV     #170003,@DBUF   ;MENU ENABLE=1  MENU=1
1346 013010 012737 000100 001124                MOV     #BIT6,$GDDAT    ;LOAD EXPECTED
1347 013016 013777 002000 167004                MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
1348 013024 013737 002042 001122                MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
1349 013032 017737 167004 001126                MOV     @SREG1,$BDDAT           ;READ DISPLAY STATUS REGISTER
1350 013040 042737 177677 001126                BIC     #177677,$BDDAT  ;MASK TO BIT 2
1351 013046 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1352 013054 001401                              BEQ     TST77           ;;BR IF EQUAL
1353 013056 104031                              ERROR   31              ;MENU BIT FAILED TO SET
1354
1355
1356                                    ;;*****************************************************************
                                        ;*TEST 77        CHECK MENU ENABLE GATE
                                        ;;*****************************************************************
(2) 013060 000004                       TST77:  SCOPE
(2) 013062 012737 000077 001206                 MOV     #77,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
1357 013070 013737 002042 001122                MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
1358 013076 012777 020000 166756                MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1359 013104 012737 170002 031076                MOV     #170002,BUFFER  ;MENU ENABLE = 1 MENU = 0
1360 013112 012777 031076 166710                MOV     #BUFFER,@DPC    ;START DISPLAY
1361 013120 012737 170001 031100                MOV     #170001,BUFFER+2        ;MENU ENABLE=0  MENU=1
1362 013126 005277 166676                       INC     @DPC            ;RESUME
1363 013132 005037 001124                       CLR     $GDDAT          ;LOAD EXPECTED
1364 013136 017737 166700 001126                MOV     @SREG1,$BDDAT           ;READ DISPLAY STATUS REGISTER
1365 013144 042737 177677 001126                BIC     #177677,$BDDAT  ;MASK TO BIT 2
1366 013152 023737 001124 001126                CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1367 013160 001401                              BEQ     TST100          ;;BR IF EQUAL
1368 013162 104031                              ERROR   31              ;MENU ENABLE FAILED TO INHIBIT
1369                                                                    ;RESETTING OF MENU BIT
1370
1371                                    ;;*****************************************************************
                                        ;*TEST 100       TEST THAT LOAD STATUS C INSTR CAN SET THE VECTOR SCALE
                                        ;;*****************************************************************
```

```
  (2)  013164  000004                    TST100: SCOPE
  (2)  013166  012737  000100  001206             MOV     #100,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1372   013174  012777  020000  166660             MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1373   013202  012737  013216  001110             MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1374   013210  012737  000017  001124             MOV     #17,$GDDAT      ;START WITH ALL ONES
1375   013216  012737  154020  031076  1$:        MOV     #154020,BUFFER  ;SET UP INSTR AT BUFFER
1376   013224  053737  001124  031076             BIS     $GDDAT,BUFFER   ;SET UP VECTOR SCALE AT INSTR LOC
1377   013232  012777  031076  166570             MOV     #BUFFER,@DPC    ;START
1378   013240  013737  002042  001122             MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
1379   013246  017737  166570  001126             MOV     @SREG1,$BDDAT   ;READ VECTOR SCALE
1380   013254  042737  177760  001126             BIC     #177760,$BDDAT  ;ONLY INTERESTED IN VECTOR SCALE
1381   013262  023737  001124  001126             CMP     $GDDAT,$BDDAT   ;ARE THEY CORRECT?
1382   013270  001401                             BEQ     2$              ;BR IF OK
1383   013272  104025                             ERROR   25              ;VECTOR SCALE INSTR ER
1384   013274  005337  001124          2$:        DEC     $GDDAT          ;ADVANCE PATTERN
1385   013300  100346                             BPL     1$              ;BR IF NOT DONE
1386
1387                                    ;;*****************************************************************
  (3)                                  ;*TEST 101       TEST THAT THE VECTOR SCALE DOES NOT CHANGE WHEN CHANGE ENA 0
  (3)                                  ;;*****************************************************************
  (2)  013302  000004                    TST101: SCOPE
  (2)  013304  012737  000101  001206             MOV     #101,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1388   013312  012777  020000  166542             MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1389   013320  012737  154037  031076             MOV     #154037,BUFFER  ;SET UP INSTR AT BUFFER-VECTOR SCALE ENA SET
1390   013326  012737  000017  001124             MOV     #17,$GDDAT      ;SET UP VECTOR SCALE EXPECTED
1391   013334  012777  031076  166466             MOV     #BUFFER,@DPC    ;START
1392   013342  012737  154000  031100             MOV     #154000,BUFFER+2        ;SET UP INSTR AT BUFFER-NO VECTOR SCALE ENA
1393   013350  005277  166454                     INC     @DPC            ;RESUME
1394   013354  013737  002042  001122             MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
1395   013362  017737  166454  001126             MOV     @SREG1,$BDDAT   ;READ VECTOR SCALE
1396   013370  042737  177760  001126             BIC     #177760,$BDDAT  ;SAVE VECTOR SCALE ONLY
1397   013376  023737  001124  001126             CMP     $GDDAT,$BDDAT   ;DOES IT STILL EQ 17
1398   013404  001401                             BEQ     TST102          ;;ADVANCE TO NEXT TEST IF OK
1399   013406  104025                             ERROR   25              ;VECTOR SCALE CHANGE ENABLE INSTR ER
1400
1401                                    ;;*****************************************************************
  (3)                                  ;*TEST 102       TEST THAT LOAD STATUS C INSTR CAN SET THE CHARACTER SCALE
  (3)                                  ;;*****************************************************************
  (2)  013410  000004                    TST102: SCOPE
  (2)  013412  012737  000102  001206             MOV     #102,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1402   013420  012777  020000  166434             MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1403   013426  012737  013450  001110             MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1404   013434  012737  154340  031076             MOV     #154340,BUFFER  ;SET UP INSRT AT BUFFER
1405   013442  012737  001400  001124             MOV     #1400,$GDDAT    ;START WITH ALL ONES
1406   013450  012777  031076  166352  1$:        MOV     #BUFFER,@DPC    ;START
1407   013456  013737  002042  001122             MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
1408   013464  017737  166352  001126             MOV     @SREG1,$BDDAT   ;READ CHAR SCALE
1409   013472  042737  176377  001126             BIC     #176377,$BDDAT  ;SAVE CHAR SCALE ONLY
1410   013500  023737  001124  001126             CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
1411   013506  001402                             BEQ     2$              ;BR IF OK
1412   013510  001411                             BEQ     TST103          ;;ADVANCE TO NEXT TEST IF OK
1413   013512  104026                             ERROR   26              ;CHARACTER SCALE INSTR ER
1414   013514  162737  000400  001124  2$:        SUB     #400,$GDDAT     ;ADVANCE PATTERN
1415   013522  100404                             BMI     TST103          ;;ADVANCE TO NEXT TEST IF DONE
1416   013524  162737  000040  031076             SUB     #40,BUFFER      ;ADVANCE PATTERN AT INSTR LOC
1417   013532  000746                             BR      1$              ;DO NEXT CHAR SCALE
```

```
 1418
 1419                                   ;;*****************************************************************
  (3)                                   ;*TEST 103      TEST THAT THE CHAR SCALE DOES NOT CHANGE WHEN CHANGE ENA-0
  (3)                                   ;;*****************************************************************
  (2)  013534  000004                   TST103: SCOPE
  (2)  013536  012737  000103  001206           MOV     #103,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1420  013544  012777  020000  166310           MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1421  013552  012737  154340  031076           MOV     #154340,BUFFER  ;SET UP INSTR AT BUFFER-CHAR SCALE ENA SET
 1422  013560  012737  001400  001124           MOV     #1400,$GDDAT    ;SET UP CHAR SCALE EXPECTED
 1423  013566  012777  031076  166234           MOV     #BUFFER,@DPC    ;START
 1424  013574  012737  154000  031100           MOV     #154000,BUFFER+2        ;SET UP INSTR AT BUFFER-NO CHAR SCALE ENA
 1425  013602  005277  166222                   INC     @DPC            ;RESUME
 1426  013606  013737  002042  001122           MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
 1427  013614  017737  166222  001126           MOV     @SREG1,$BDDAT   ;READ CHAR SCALE
 1428  013622  042737  176377  001126           BIC     #176377,$BDDAT  ;SAVE CHAR SCALE ONLY
 1429  013630  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 1430  013636  001401                           BEQ     TST104          ;;GO TO NEXT TEST IF NO CHANGE
 1431  013640  104026                           ERROR   26              ;CHARACTER SCALE CHANGE ENABLE INSTR ER
 1432
 1433                                   ;;*****************************************************************
  (3)                                   ;*TEST 104      TEST THAT LOAD STATUS C INSTR CAN SET THE CHARACTER ROTATE
  (3)                                   ;;*****************************************************************
  (2)  013642  000004                   TST104: SCOPE
  (2)  013644  012737  000104  001206           MOV     #104,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1434  013652  012777  020000  166202           MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1435  013660  012737  013702  001110           MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
 1436  013666  012737  155400  031076           MOV     #155400,BUFFER  ;SET UP INSTR AT BUFFER
 1437  013674  012737  002000  001124           MOV     #2000,$GDDAT    ;EXPECT CHAR ROTATE BIT
 1438  013702  012777  031076  166120  1$:      MOV     #BUFFER,@DPC    ;START
 1439  013710  013737  002042  001122           MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
 1440  013716  017737  166120  001126           MOV     @SREG1,$BDDAT   ;READ CHAR ROTATE BIT
 1441  013724  042737  175777  001126           BIC     #175777,$BDDAT  ;SAVE CHAR ROTATE BIT ONLY
 1442  013732  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 1443  013740  001401                           BEQ     2$              ;BR IF OK
 1444  013742  104026                           ERROR   26              ;CHARACTER ROTATE INSTR ER
 1445  013744  162737  002000  001124  2$:      SUB     #2000,$GDDAT    ;ADVANCE PATTERN
 1446  013752  100404                           BMI     TST105          ;;ADVANCE TO NEXT TEST IF OK
 1447  013754  162737  000400  031076           SUB     #400,BUFFER     ;ADVANCE PATTERN AT INSTR LOC
 1448  013762  000747                           BR      1$              ;NOW WRITE A ZERO INTO CHAR ROTATE
 1449
 1450                                   ;;*****************************************************************
  (3)                                   ;*TEST 105      TEST THAT THE CHARACTER ROTATE DOES NOT CHANGE WHEN CHANGE ENA=0
  (3)                                   ;;*****************************************************************
  (2)  013764  000004                   TST105: SCOPE
  (2)  013766  012737  000105  001206           MOV     #105,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1451  013774  012777  020000  166060           MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1452  014002  012737  155400  031076           MOV     #155400,BUFFER  ;SET UP INSTR AT BUFFER CHAR ROTATE ENA SET
 1453  014010  012737  002000  001124           MOV     #2000,$GDDAT    ;EXPECT CHAR ROTATE BIT
 1454  014016  012777  031076  166004           MOV     #BUFFER,@DPC    ;START
 1455  014024  012737  154000  031100           MOV     #154000,BUFFER+2        ;SET UP INSTR AT BUFFER-WITHOUT CHAR ROTATE ENA
 1456  014032  005277  165772                   INC     @DPC            ;RESUME
 1457  014036  013737  002042  001122           MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
 1458  014044  017737  165772  001126           MOV     @SREG1,$BDDAT   ;READ CHAR ROTATE BIT
 1459  014052  042737  175777  001126           BIC     #175777,$BDDAT  ;SAVE CHAR ROTATE BIT
 1460  014060  023737  001124  001126           CMP     $GDDAT,$BDDAT   ;IS IT SET?
 1461  014066  001401                           BEQ     TST106          ;;ADVANCE TO NEXT IF OK
```

```
                                        ERROR   26              ;CHARACTER ROTATE CHANGE ENABLE INSTR ER


                                ;;**********************************************************************
                                ;*TEST 106       TEST THAT RESET WILL CLEAR ROTATE, CHAR SCALE, MENU & VECTOR SCALE
                                ;;**********************************************************************
                                TST106: SCOPE
                                        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                        MOV     #106,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     #BIT13,@STKPT   ;SET MAINT SW 2
                                        MOV     #155733,BUFFER  ;SET UP 'LOAD STATUS C' INSTR
                                        MOV     #170003,BUFFER+2  ;SET UP MENU INSTR
                                        MOV     #BUFFER,@DPC    ;START
                                        MOV     #404,$GDDAT     ;EXPECT MSB OF VECTOR SCALE ONLY
                                        INC     @DPC            ;PICK UP MENU INSTR
                                        MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
                                        RESET                   ;CLR REG 12
                                        MOV     @SREG1,$BDDAT   ;READ STATUS REG 1
                                        CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
                                        BEQ     TST107          ;;NEXT TEST IF OK
                                        ERROR   32              ;RESET FAILED TO SET UP REG 12

                                ;GRAPHPLOT INCREMENT REGISTER TESTS


                                ;;**********************************************************************
                                ;*TEST 107       LOAD 0-77 INTO GRAPHPLOT INCREMENT REGISTER
                                ;;**********************************************************************
                                TST107: SCOPE
                                        MOV     #107,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
                                        MOV     #BIT13,@STKPT   ;SET MAINT SW 2
                                        MOV     #174100,BUFFER  ;LOAD GRAPHPLOT INSTR
                                        CLR     $GDDAT          ;LD EXPECTED
                                1$:     MOV     #BUFFER,@DPC    ;START
                                        MOV     XPOS,$BDADR     ;SET UP REG ADRS 04
                                        MOV     @XPOS,$BDDAT    ;READ GRAPHPLOT
                                        BIC     #1777,$BDDAT    ;SAVE ONLY GRAPHPLOT
                                        CMP     $GDDAT,$BDDAT   ;CORRECT?
                                        BEQ     2$              ;BR IF SO
                                        ERROR   7               ;GRAPHPLOT FAILED TO LOAD CORRECTLY
                                2$:     INC     BUFFER          ;ADVANCE GRAPHPLOT AT INSTR
                                        ADD     #2000,$GDDAT    ;ADVANCE EXPECTED
                                        BCC     1$              ;BR IF NOT COMBS TESTED


                                ;;**********************************************************************
                                ;*TEST 110       CHECK THE GRAPHPLOT INCREMENT ENABLE REGISTER
                                ;;**********************************************************************
                                TST110: SCOPE
                                        MOV     #110,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     XPOS,$BDADR     ;SET UP REG ADRS 04
                                        MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
                                        MOV     #174100,@DBUF   ;LOAD GRAPHPLOT COUNTER WITH 0
                                        MOV     DBUF,@DPC       ;START DISPLAY
                                        MOV     #174077,BUFFER+2        ;LOAD GRAPHPLOT NO ENABLE
                                        INC     @DPC            ;RESUME
                                        CLR     $GDDAT          ;LOAD EXPECTED
```

```
                                        MOV     @XPOS,$BDDAT    ;READ INCREMENT REGISTER
                                        BIC     #1777,$BDDAT    ;MASK TO BITS 15-10
                                        CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
                                        BEQ     TST111          ;;BR IF EQUAL
                                        ERROR   7               ;GRAPHPLOT REGISTER CHANGED WITHOUT
                                                                ; THE ENABLE BEING SET


                                ;;*******************************************************************
                                ;*TEST 111      TEST THAT RESET WILL CLEAR GRAPHPLOT INCREMENT
                                ;;*******************************************************************
                                TST111: SCOPE
                                        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
                                        MOV     #111,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     XPOS,$BDADR     ;SET UP REG ADRS 04
                                        MOV     #BIT13,@STKPT   ;SET MAINT SW 2
                                        MOV     #174177,BUFFER  ;SET UP GRAPHPLOT INCREMENT INSTR
                                        MOV     #BUFFER,@DPC    ;START
                                        CLR     $GDDAT          ;EXPECT ZERO
                                        RESET                   ;CLR REG 04
                                        MOV     @XPOS,$BDDAT    ;READ REG 04
                                        BEQ     TST112          ;;NEXT TEST IF ZERO
                                        ERROR   32              ;RESET FAILED TO CLR GRAPHPLOT INC


                                ;;*******************************************************************
                                ;*TEST 112      TEST THAT LOAD STATUS B INSTR CAN SET THE COLOR LEVEL
                                ;;*******************************************************************
                                TST112: SCOPE
                                        MOV     #112,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
                                        MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
                                        MOV     #175600,BUFFER  ;SET UP INSTR AT BUFFER
                                        MOV     #14,$GDDAT      ;EXPECT ALL ONES AT START
                                1$:     MOV     #BUFFER,@DPC    ;START
                                        MOV     CONS,$BDADR     ;SET UP REG ADRS 22
                                        MOV     @CONS,$BDDAT    ;READ COLOR LEVEL
                                        BIC     #177763,$BDDAT  ;ONLY INTERESTED IN THE COLOR LEVEL
                                        CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
                                        BEQ     2$              ;BR IF OK
                                        ERROR   35              ;COLOR LEVEL INSTR ER
                                2$:     SUB     #4,$GDDAT       ;ADVANCE PATTERN
                                        BMI     TST113          ;;ADVANCE TO NEXT TEST IF ALL PATTERNS TESTED
                                        SUB     #200,BUFFER     ;SET UP NEXT PATTERN AT INSTR LOC
                                        BR      1$              ;TRY NEXT PATTERN


                                ;;*******************************************************************
                                ;*TEST 113      TEST THAT THE COLOR LEVEL DOES NOT CHANGE WHEN CHANGE ENA-0
                                ;;*******************************************************************
                                TST113: SCOPE
                                        MOV     #113,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
                                        MOV     #175600,BUFFER  ;SET UP INSTR AT BUFFER-COLOR LEVEL ENA SET
                                        MOV     #14,$GDDAT      ;SET UP COLOR LEVEL EXPECTED
                                        MOV     #BUFFER,@DPC    ;START
                                        MOV     #174000,BUFFER+2        ;SET UP INSTR AT BUFFER-NO COLOR LEVEL ENA
                                        INC     @DPC            ;RESUME
                                        MOV     CONS,$BDADR     ;SET UP REG ADRS 22
```

```
 1548  014712  017737  165134  001126          MOV     @CONS,$BDDAT    ;READ COLOR LEVEL
 1549  014720  042737  177763  001126          BIC     #177763,$BDDAT  ;ONLY INTERESTED IN THE COLOR LEVEL
 1550  014726  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 1551  014734  001401                          BEQ     TST114          ;;ADVANCE TO NEXT TEST IF OK
 1552  014736  104035                          ERROR   35              ;COLOR LEVEL CHANGE ENABLE INSTR ER
 1553
 1554                                  ;;*******************************************************************
  (3)                                  ;*TEST 114       TEST THAT NOP INSTR CAN SET & RESET INTENSITY ENABLED-CONSOLE 0
  (3)                                  ;;*******************************************************************
  (2)  014740  000004                  TST114: SCOPE
  (2)  014742  012737  000114  001206          MOV     #114,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1555  014750  012777  020000  165104          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1556  014756  012737  015000  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
 1557  014764  012737  100000  001124          MOV     #100000,$GDDAT  ;EXPECT IT TO BE SET INITIALLY
 1558  014772  012737  164300  031076          MOV     #164300,BUFFER  ;SET UP INSTR LOC
 1559  015000  012777  031076  165022  1$:     MOV     #BUFFER,@DPC    ;START
 1560  015006  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
 1561  015014  017737  165032  001126          MOV     @CONS,$BDDAT    ;READ CONSOLE 0 INT ENA REG
 1562  015022  042737  077777  001126          BIC     #77777,$BDDAT   ;SAVE ONLY INTENSITY ENABLE BIT
 1563  015030  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 1564  015036  001401                          BEQ     2$              ;BR IF SO
 1565  015040  104036                          ERROR   36              ;CONSOLE 0 INTENSITY ENABLE INSTR ER
 1566  015042  162737  100000  001124  2$:     SUB     #100000,$GDDAT  ;GO TO RESET STATE
 1567  015050  100404                          BMI     TST115          ;;GO TO NEXT TEST IF BOTH STATES TESTED
 1568  015052  042737  000100  031076          BIC     #100,BUFFER     ;CLR INT ENABLED BIT AT INSTR LOC
 1569  015060  000747                          BR      1$              ;GO TRY RESET STATE
 1570
 1571                                  ;;*******************************************************************
  (3)                                  ;*TEST 115       TEST THAT CONSOLE 0 INTENSITY ENABLED DOES NOT RESET WHEN ENA=0
  (3)                                  ;;*******************************************************************
  (2)  015062  000004                  TST115: SCOPE
  (2)  015064  012737  000115  001206          MOV     #115,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1572  015072  012777  020000  164762          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1573  015100  012737  100000  001124          MOV     #100000,$GDDAT  ;EXPECT INTENSITY TO BE SET
 1574  015106  012737  164300  031076          MOV     #164300,BUFFER  ;SET UP INSTR
 1575  015114  012777  031076  164706          MOV     #BUFFER,@DPC    ;START
 1576  015122  012737  164000  031100          MOV     #164000,BUFFER+2        ;SET UP INSTR WITH ENA=0
 1577  015130  005277  164674                  INC     @DPC            ;ADVANCE TO NEXT INSTR
 1578  015134  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
 1579  015142  005777  164704                  TST     @CONS           ;IS INTENSITY STILL SET?
 1580  015146  100403                          BMI     TST116          ;;ADVANCE TO NEXT TEST IF STILL SET
 1581  015150  005037  001126                  CLR     $BDDAT          ;INDICATE IT WAS CLEARED
 1582  015154  104036                          ERROR   36              ;CONSOLE 0 INTENSITY CHANGE ENABLE INSTR ER
 1583
 1584                                  ;;*******************************************************************
  (3)                                  ;*TEST 116       TEST THAT NOP INSTR CAN SET & RESET INTENSITY ENABLED-CONSOLE 1
  (3)                                  ;;*******************************************************************
  (2)  015156  000004                  TST116: SCOPE
  (2)  015160  012737  000116  001206          MOV     #116,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1585  015166  012777  020000  164666          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1586  015174  013737  015216  001110          MOV     1$,$LPERR       ;SET UP SCOPE LOOP ADRS
 1587  015202  012737  001000  001124          MOV     #1000,$GDDAT    ;EXPECT IT TO BE SET INITIALLY
 1588  015210  012737  164700  031076          MOV     #164700,BUFFER  ;SET UP INSTR LOCATION
 1589  015216  012777  031076  164604  1$:     MOV     #BUFFER,@DPC    ;START
 1590  015224  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
 1591  015232  017737  164614  001126          MOV     @CONS,$BDDAT    ;READ CONSOLE 1 INT ENA REG
```

```
 1592  015240  042737  176777  001126          BIC     #176777,$BDDAT  ;SAVE ONLY INT ENA
 1593  015246  023737  001124  001126          CMP     $GDDAT $BDDAT   ;IS IT CORRECT?
 1594  015254  001401                          BEQ     2$              ;BR IF OK
 1595  015256  104036                          ERROR   36              ;CONSOLE 1 INTENSITY ENABLE INSTR ER
 1596  015260  162737  001000  001124  2$:     SUB     #1000,$GDDAT    ;GO TO RESET STATE
 1597  015266  100404                          BMI     TST117          ;;GO TO NEXT TEST IF BOTH STATES TESTED
 1598  015270  042737  000100  031076          BIC     #100,BUFFER     ;CLR INT ENABLED BIT AT INSTR LOC
 1599  015276  000747                          BR      1$              ;GO TRY RESET STATE
 1600
 1601                                  ;;*******************************************************************
  (3)                                  ;*TEST 117      TEST THAT CONSOLE 1 INTENSITY ENABLED DOES NOT RESET WHEN ENA=0
  (3)                                  ;;*******************************************************************
  (2)  015300  000004                  TST117: SCOPE
  (2)  015302  012737  000117  001206          MOV     #117,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1602  015310  012777  020000  164544          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1603  015316  012737  001000  001124          MOV     #1000,$GDDAT    ;EXPECT INTENSITY TO BE SET
 1604  015324  012737  164700  031076          MOV     #164700,BUFFER  ;SET UP INSTR
 1605  015332  012777  031076  164470          MOV     #BUFFER,@DPC    ;START
 1606  015340  012737  164400  031100          MOV     #164400,BUFFER+2        ;SET UP INSTR WITH ENA=0
 1607  015346  005277  164456                  INC     @DPC            ;ADVANCE TO NEXT INSTR
 1608  015352  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
 1609  015360  032777  001000  164464          BIT     #1000,@CONS     ;IS INTENSITY STILL SET?
 1610  015366  001003                          BNE     TST120          ;;ADVANCE TO NEXT TEST IF STILL SET
 1611  015370  005037  001126                  CLR     $BDDAT          ;INDICATE IT WAS CLEARED
 1612  015374  104036                          ERROR   36              ;CONSOLE 1 INTENSITY CHANGE ENABLE INSTR ER
 1613
 1614                                  ;;*******************************************************************
  (3)                                  ;*TEST 120      TEST THAT NOP INSTR CAN SET & RESET L.P. INTR ENABLED-CONSOLE 0
  (3)                                  ;;*******************************************************************
  (2)  015376  000004                  TST120: SCOPE
  (2)  015400  012737  000120  001206          MOV     #120,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1615  015406  012777  020000  164446          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1616  015414  012737  015436  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
 1617  015422  012737  004000  001124          MOV     #4000,$GDDAT    ;EXPECT IT TO BE SET INITIALLY
 1618  015430  012737  164060  031076          MOV     #164060,BUFFER  ;SET UP INSTR LOC
 1619  015436  012777  031076  164364  1$:     MOV     #BUFFER,@DPC    ;START
 1620  015444  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
 1621  015452  017737  164374  001126          MOV     @CONS,$BDDAT    ;READ CONSOLE 0 LP INTR ENABLED REG
 1622  015460  042737  173777  001126          BIC     #173777,$BDDAT  ;SAVE ONLY LP INTR ENABLED BIT
 1623  015466  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 1624  015474  001401                          BEQ     2$              ;BR IF OK
 1625  015476  104042                          ERROR   42              ;CONSOLE 0 L.P. INTR ENABLE INSTR ER
 1626  015500  162737  004000  001124  2$:     SUB     #4000,$GDDAT    ;GO TO RESET STATE
 1627  015506  100404                          BMI     TST121          ;;ADVANCE TO NEXT TEST IF BOTH STATES TESTED
 1628  015510  042737  000020  031076          BIC     #20,BUFFER      ;CLR THE ENABLED BIT AT INSTR LOC
 1629  015516  000747                          BR      1$              ;GO TRY RESET STATE
 1630
 1631                                  ;;*******************************************************************
  (3)                                  ;*TEST 121      TEST THAT CONSOLE 0 L.P. INTR ENABLED DOES NOT RESET WHEN ENA=0
  (3)                                  ;;*******************************************************************
  (2)  015520  000004                  TST121: SCOPE
  (2)  015522  012737  000121  001206          MOV     #121,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1632  015530  012777  020000  164324          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1633  015536  012737  004000  001124          MOV     #4000,$GDDAT    ;EXPECT L.P. INTR ENABLED TO BE SET
 1634  015544  012737  164060  031076          MOV     #164060,BUFFER  ;SET UP INSTR LOC
 1635  015552  012777  031076  164250          MOV     #BUFFER,@DPC    ;START
```

```
 1636  015560  012737  164000  031100          MOV     #164000,BUFFER+2        ;SET UP INSTR WITH ENA=0
 1637  015566  005277  164236                  INC     @DPC                    ;ADVANCE TO NEXT INSTR
 1638  015572  013737  002052  001122          MOV     CONS,$BDADR             ;SET UP REG ADRS 22
 1639  015600  032777  004000  164244          BIT     #4000,@CONS             ;IS L.P. INTR ENABLED STILL SET?
 1640  015606  001003                          BNE     TST122                  ;;ADVANCE TO NEXT TEST IF STILL SET
 1641  015610  005037  001126                  CLR     $BDDAT                  ;INDICATE IT WAS CLEARED
 1642  015614  104042                          ERROR   42                      ;CONSOLE 0 L.P. CHANGE INTR ENABLE INSTR ER
 1643
 1644                                  ;;*****************************************************************
  (3)                                  ;*TEST 122       TEST THAT NOP INSTR CAN SET & RESET L.P. INTR ENABLED-CONSOLE 1
  (3)                                  ;;*****************************************************************
  (2)  015616  000004                  TST122: SCOPE
  (2)  015620  012737  000122  001206          MOV     #122,$TESTN             ;;SET TEST NUMBER IN APT MAIL BOX
 1645  015626  012777  020000  164226          MOV     #BIT13,@STKPT                   ;SET MAINT SWITCH #2
 1646  015634  012737  015656  001110          MOV     #1$,$LPERR              ;SET UP SCOPE LOOP ADRS
 1647  015642  012737  000040  001124          MOV     #40,$GDDAT              ;EXPECT IT TO BE SET INITIALLY
 1648  015650  012737  164460  031076          MOV     #164460,BUFFER          ;SET UP INSTR LOC
 1649  015656  012777  031076  164144  1$:     MOV     #BUFFER,@DPC            ;START
 1650  015664  013737  002052  001122          MOV     CONS,$BDADR             ;SET UP REG ADRS 22
 1651  015672  017737  164154  001126          MOV     @CONS,$BDDAT            ;READ CONSOLE 1 LP INTR ENABLED REG
 1652  015700  042737  177737  001126          BIC     #177737,$BDDAT          ;SAVE ONLY LP INTR ENABLED BIT
 1653  015706  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;IS IT CORRECT?
 1654  015714  001401                          BEQ     2$                      ;BR IF OK
 1655  015716  104042                          ERROR   42                      ;CONSOLE 1 L.P. INTR ENABLE INSTR ER
 1656  015720  162737  000040  001124  2$:     SUB     #40,$GDDAT              ;GO TO RESET STATE
 1657  015726  100404                          BMI     TST123                  ;;ADVANCE TO NEXT TEST IF BOTH STATES TESTED
 1658  015730  042737  000020  031076          BIC     #20,BUFFER              ;CLR THE ENABLED BIT AT INSTR LOC
 1659  015736  000747                          BR      1$                      ;GO TRY RESET STATE
 1660
 1661                                  ;;*****************************************************************
  (3)                                  ;*TEST 123       TEST THAT CONSOLE 1 L.P. INTR ENABLED DOES NOT RESET WHEN ENA=0
  (3)                                  ;;*****************************************************************
  (2)  015740  000004                  TST123: SCOPE
  (2)  015742  012737  000123  001206          MOV     #123,$TESTN             ;;SET TEST NUMBER IN APT MAIL BOX
 1662  015750  012777  020000  164104          MOV     #BIT13,@STKPT                   ;SET MAINT SWITCH #2
 1663  015756  012737  000040  001124          MOV     #40,$GDDAT              ;EXPECT L.P. INTR ENABLED TO BE SET
 1664  015764  012737  164460  031076          MOV     #164460,BUFFER          ;SET UP INSTR LOC
 1665  015772  012777  031076  164030          MOV     #BUFFER,@DPC            ;START
 1666  016000  012737  164400  031100          MOV     #164400,BUFFER+2        ;SET UP INSTR WITH ENA=0
 1667  016006  005277  164016                  INC     @DPC                    ;ADVANCE TO NEXT INSTR
 1668  016012  013737  002052  001122          MOV     CONS,$BDADR             ;SET UP REG ADRS 22
 1669  016020  032777  000040  164024          BIT     #40,@CONS               ;IS L.P. INTR ENABLED STILL SET?
 1670  016026  001003                          BNE     TST124                  ;;ADVANCE TO NEXT TEST IF STILL SET
 1671  016030  005037  001126                  CLR     $BDDAT                  ;INDICATE IT WAS CLEARED
 1672  016034  104042                          ERROR   42                      ;CONSOLE 1 L.P. CHANGE INTR ENABLE INSTR ER
 1673
 1674                                  ;;*****************************************************************
  (3)                                  ;*TEST 124       TEST THAT NOP INSTR CAN SET & RESET L.P. SWITCH INTR ENABLED-CONSOLE 0
  (3)                                  ;;*****************************************************************
  (2)  016036  000004                  TST124: SCOPE
  (2)  016040  012737  000124  001206          MOV     #124,$TESTN             ;;SET TEST NUMBER IN APT MAIL BOX
 1675  016046  012777  020000  164006          MOV     #BIT13,@STKPT                   ;SET MAINT SWITCH #2
 1676  016054  012737  016076  001110          MOV     #1$,$LPERR              ;SET UP SCOPE LOOP ADRS
 1677  016062  012737  002000  001124          MOV     #2000,$GDDAT            ;EXPECT LP SW INTR ENABLED BIT
 1678  016070  012737  164014  031076          MOV     #164014,BUFFER          ;SET UP INSTR
 1679  016076  012777  031076  163724  1$:     MOV     #BUFFER,@DPC            ;START
```

```
1680  016104  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
1681  016112  017737  163734  001126          MOV     @CONS,$BDDAT    ;READ L.P. SWITCH INTR ENABLED REG
1682  016120  042737  175777  001126          BIC     #175777,$BDDAT  ;SAVE ONLY L.P. SWITCH INTR ENABLED BIT
1683  016126  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
1684  016134  001401                          BEQ     2$              ;BR IF OK
1685  016136  104043                          ERROR   43              ;CONSOLE 0 L P. SW INTR ENABLE INSTR ER
1686  016140  162737  002000  001124  2$:     SUB     #2000,$GDDAT    ;GO TO RESET STATE
1687  016146  100404                          BMI     TST125          ;;ADVANCE TO NEXT TEST IF BOTH STATES TESTED
1688  016150  042737  000004  031076          BIC     #4,BUFFER       ;CLR THE ENABLED BIT AT INSTR LOC
1689  016156  000747                          BR      1$              ;GO TRY RESET STATE
1690
1691                                  ;;*****************************************************************
                                      ;*TEST 125       TEST THAT L.P. SWITCH INTR ENABLED DOES NOT RESET WHEN ENA=0
                                      ;;*****************************************************************
 (3)                                  TST125: SCOPE
 (3)
 (2)  016160  000004
 (2)  016162  012737  000125  001206          MOV     #125,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1692  016170  012777  020000  163664          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1693  016176  012737  002000  001124          MOV     #2000,$GDDAT    ;EXPECT L.P. SWITCH INTR ENABLED TO BE SET
1694  016204  012737  164014  031076          MOV     #164014,BUFFER  ;SET UP INSTR LOC
1695  016212  012777  031076  163610          MOV     #BUFFER,@DPC    ;START
1696  016220  012737  164000  031100          MOV     #164000,BUFFER+2        ;SET UP INSTR WITH ENA=0
1697  016226  005277  163576                  INC     @DPC            ;ADVANCE TO NEXT INSTR
1698  016232  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
1699  016240  032777  002000  163604          BIT     #2000,@CONS     ;IS L.P. SWITCH INTR ENABLED STILL SET?
1700  016246  001003                          BNE     TST126          ;;ADVANCE TO NEXT TEST IF STILL SET
1701  016250  005037  001126                  CLR     $BDDAT          ;INDICATE IT WAS CLEARED
1702  016254  104043                          ERROR   43              ;CONSOLE 0 L.P. SW CHANGE INTR ENABLE INSTR ER
1703
1704                                  ;;*****************************************************************
                                      ;*TEST 126       TEST THAT NOP INSTR CAN SET & RESET L.P. SWITCH INTR ENABLED-CONSOLE 1
 (3)                                  ;;*****************************************************************
 (3)                                  TST126: SCOPE
 (2)  016256  000004
 (2)  016260  012737  000126  001206          MOV     #126,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1705  016266  012777  020000  163566          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1706  016274  012737  016316  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1707  016302  012737  000020  001124          MOV     #20,$GDDAT      ;EXPECT L.P. SWITCH INTR ENABLED BIT
1708  016310  012737  164414  031076          MOV     #164414,BUFFER  ;SET UP INSTR
1709  016316  012777  031076  163504  1$:     MOV     #BUFFER,@DPC    ;START
1710  016324  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
1711  016332  017737  163514  001126          MOV     @CONS,$BDDAT    ;READ L.P. SWITCH INTR ENABLED REG
1712  016340  042737  177757  001126          BIC     #177757,$BDDAT  ;SAVE ONLY L.P. SWITCH INTR ENABLED BIT
1713  016346  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
1714  016354  001401                          BEQ     2$              ;BR IF OK
1715  016356  104043                          ERROR   43              ;CONSOLE 1 L.P. SW INTR ENABLE INSTR ER
1716  016360  162737  000020  001124  2$:     SUB     #20,$GDDAT      ;GO TO RESET STATE
1717  016366  100404                          BMI     TST127          ;;ADVANCE TO NEXT TEST IF BOTH STATES TESTED
1718  016370  042737  000004  031076          BIC     #4,BUFFER       ;CLR THE ENABLED BIT AT INSTR LOC
1719  016376  000747                          BR      1$              ;GO TRY RESET STATE
1720
1721                                  ;;*****************************************************************
                                      ;*TEST 127       TEST THAT L.P. SWITCH INTR ENABLED DOES NOT RESET WHEN ENA=0
 (3)                                  ;;*****************************************************************
 (3)                                  TST127: SCOPE
 (2)  016400  000004
 (2)  016402  012737  000127  001206          MOV     #127,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1722  016410  012777  020000  163444          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1723  016416  012737  000020  001124          MOV     #20,$GDDAT      ;EXPECT L.P. SW INTR ENABLED TO BE SET
```

```
1724  016424  012737  164414  031076          MOV     #164414,BUFFER  ;SET UP INSTR LOC
1725  016432  012777  031076  163370          MOV     #BUFFER,@DPC    ;START
1726  016440  012737  164400  031100          MOV     #164400,BUFFER+2        ;SET UP INSTR LOC WITH ENA=0
1727  016446  005277  163356                  INC     @DPC            ;ADVANCE TO NEXT INSTR
1728  016452  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
1729  016460  032777  000020  163364          BIT     #20,@CONS       ;IS L.P. SW INTR ENABLED STILL SET?
1730  016466  001003                          BNE     TST130          ;;ADVANCE TO NEXT TEST IF STILL SET
1731  016470  005037  001126                  CLR     $BDDAT          ;INDICATE IT WAS CLEARED
1732  016474  104043                          ERROR   43              ;CONSOLE 1 L.P. SW CHANGE INTR ENABLE INSTR ER
1733
1734
1735                                  ;;*****************************************************************
 (3)                                  ;*TEST 130       TEST THAT RESET WILL CLEAR L.P. INTR ENABLES & COLOR
 (3)                                  ;;*****************************************************************
 (2)  016476  000004                  TST130: SCOPE
 (1)  016500  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 (2)  016506  012737  000130  001206          MOV     #130,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1736  016514  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
1737  016522  012777  020000  163332          MOV     #BIT13,@STKPT   ;SET MAINT SW 2
1738  016530  012737  164274  031076          MOV     #164274,BUFFER  ;SET UP INSTR FOR CONSOLE 0 L.P. INTR ENAB
1739  016536  012777  031076  163264          MOV     #BUFFER,@DPC    ;START
1740  016544  012737  164774  031100          MOV     #164774,BUFFER+2        ;CONS 1 L.P. INTR ENA
1741  016552  005277  163252                  INC     @DPC            ;RESUME
1742  016556  012737  175600  031102          MOV     #175600,BUFFER+4        ;COLOR LEVEL
1743  016564  005277  163240                  INC     @DPC            ;RESUME
1744  016570  012737  100000  001124          MOV     #BIT15,$GDDAT   ;EXPECT INTENSITY ENABLE CONSOLE 0 ONLY
1745  016576  000005                          RESET                   ;CLR FLAGS
1746  016600  017737  163246  001126          MOV     @CONS,$BDDAT    ;READ FLAGS
1747  016606  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS REG 22 CORRECT?
1748  016614  001401                          BEQ     TST131          ;;NEXT TEST IF SO
1749  016616  104032                          ERROR   32              ;RESET FAILED TO SET UP REG 22
1750
1751                                  ;;*****************************************************************
 (3)                                  ;*TEST 131       JUMP TEST TO LOC. 25252
 (3)                                  ;;*****************************************************************
 (2)  016620  000004                  TST131: SCOPE
 (2)  016622  012737  000131  001206          MOV     #131,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1752  016630  012777  020000  163224          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1753  016636  012777  160000  163134          MOV     #160000,@DBUF   ;MOVE DJUMP INTO THE BUFFER
1754  016644  012737  025252  001124          MOV     #25252,$GDDAT   ;MOVE 25252 INTO THE NEXT LOC.
1755  016652  013777  002000  163150          MOV     DBUF,@DPC       ;START THE DISPLAY
1756  016660  013737  001124  031100          MOV     $GDDAT,BUFFER+2 ;LD ADRS
1757  016666  005277  163136                  INC     @DPC            ;SINGLE STEP THE DISPLAY
1758  016672  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
1759  016700  017737  163124  001126          MOV     @DPC,$BDDAT             ;READ THE DISPLAY P.C.
1760  016706  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1761  016714  001401                          BEQ     TST132          ;;BR IF EQUAL
1762  016716  104010                          ERROR   10              ;DJUMP FAILED TO LOAD THE NEW
1763                                                                  ;DISPLAY P.C. PROPERLY
1764
1765                                  ;;*****************************************************************
 (3)                                  ;*TEST 132       JUMP TEST TO LOC. 12524
 (3)                                  ;;*****************************************************************
 (2)  016720  000004                  TST132: SCOPE
 (2)  016722  012737  000132  001206          MOV     #132,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1766  016730  012777  020000  163124          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
```

```
 1767  016736  012777  160000  163034          MOV     #160000,@DBUF   ;MOVE DJUMP INTO THE BUFFER
 1768  016744  012737  012524  001124          MOV     #12524,$GDDAT   ;MOVE 12524 INTO THE NEXT LOC.
 1769  016752  013777  002000  163050          MOV     DBUF,@DPC       ;START THE DISPLAY
 1770  016760  013737  001124  031100          MOV     $GDDAT,BUFFFR+2 ;LD ADRS
 1771  016766  005277  163036                  INC     @DPC            ;SINGLE STEP THE DISPLAY
 1772  016772  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
 1773  017000  017737  163024  001126          MOV     @DPC,$BDDAT             ;READ THE DISPLAY P.C.
 1774  017006  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 1775  017014  001401                          BEQ     TST133          ;;BR IF EQUAL
 1776  017016  104010                          ERROR   ^O              ;DJUMP FAILED TO LOAD THE NEW
 1777                                                                  ;DISPLAY P.C. PROPERLY
 1778
 1779                                  ;;*****************************************************************
  (3)                                  ;*TEST 133       JUMP TO EACH EXISTING LOCATION IN LOWER BANK
  (3)                                  ;;*****************************************************************
  (2)  017020  000004                  TST133: SCOPE
  (1)  017022  012737  000020  001166          MOV     #20,$TIMES      ;;DO 20 ITERATIONS
  (2)  017030  012737  000133  001206          MOV     #133,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1780  017036  013737  026576  001124          MOV     $LSTAD,$GDDAT   ;LOAD JUMP DESTINATION
 1781  017044  012777  020000  163010          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1782  017052  012737  160000  031076          MOV     #160000,BUFFER  ;LOAD 'DISPLAY JUMP'' INST
 1783  017060  012777  031076  162742  1$:     MOV     #BUFFER,@DPC    ;START DISPLAY
 1784  017066  013737  001124  031100          MOV     $GDDAT,BUFFER+2 ;SET UP ADRS
 1785  017074  005277  162730                  INC     @DPC            ;SINGLE STEP THE DISPLAY
 1786  017100  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ARS 00
 1787  017106  017737  162716  001126          MOV     @DPC,$BDDAT     ;READ DESTINATION D.P.C.
 1788  017114  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;TEST IF EXPECTED
 1789  017122  001401                          BEQ     2$              ;;BR IF SAME
 1790  017124  104010                          ERROR   10              ;JUMP TO LOCATION 'N'' FAILED
 1791                                                                  ;N=ADDRESS IN $GDDAT
 1792  017126  162737  000002  001124  2$:     SUB     #2,$GDDAT       ;ADJUST EXPECTED ADDRESS
 1793  017134  103351                          BCC     1$              ;BR IF MORE LOCATIONS TO TEST
 1794
 1795                                  ;;*****************************************************************
  (3)                                  ;*TEST 134       TEST 'JUMP RELATIVE' INSTR (BIT 8=0) AT LOC 'BUFFER' & VERIFY NEW DPC AD
  (3)                                  ;;*****************************************************************
  (2)  017136  000004                  TST134: SCOPE
  (2)  017140  012737  000134  001206          MOV     #134,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1796  017146  012737  017170  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
 1797  017154  012777  020000  162700          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1798  017162  012700  000400                  MOV     #400,R0         ;SET UP MSB FIRST
 1799  017166  000423                          BR      2$              ;GO CALCULATE ADRS
 1800  017170  012737  161000  031076  1$:     MOV     #161000,BUFFER  ;SET UP JUMP RELATIVE INSTR
 1801  017176  050037  031076                  BIS     R0,BUFFER       ;SET UP RELATIVE ADRS PATTERN
 1802  017202  012777  031076  162620          MOV     #BUFFER,@DPC    ;START
 1803  017210  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
 1804  017216  017737  162606  001126          MOV     @DPC,$BDDAT     ;READ NEW DPC ADRS
 1805  017224  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DOES IT EQUAL THE CALCULATED ADRS?
 1806  017232  001401                          BEQ     2$              ;BR IF OK
 1807  017234  104011                          ERROR   11              ;JUMP RELATIVE POSITIVE FAILED
 1808  017236  006200                  2$:     ASR     R0              ;SHIFT RIGHT TO NEXT BIT
 1809  017240  103410                          BCS     TST135          ;;ADVANCE TO NEXT TEST IF DONE
 1810  017242  010037  001124                  MOV     R0,$GDDAT       ;SET UP OFFSET VALUE
 1811  017246  006337  001124                  ASL     $GDDAT          ;MAKE INTO WORD OFFSET
 1812  017252  062737  031100  001124          ADD     #BUFFER+2,$GDDAT        ;ADD IN ADRS OF INSTR PLUS 2
 1813  017260  000743                          BR      1$              ;TRY THIS RELATIVE ADRS
```

```
1814
1815                                    ;;*****************************************************************
 (3)                                    ;*TEST 135       TEST 'JUMP RELATIVE' INSTR (BIT 8=1) AT LOC 'BUFFER' & VERIFY NEW DPC
 (3)                                    ;;*****************************************************************
 (2) 017262 000004                      TST135: SCOPE
 (2) 017264 012737 000135 001206                MOV     #135,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1816 017272 012737 017314 001110                MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1817 017300 012777 020000 162554                MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1818 017306 012700 000400                       MOV     #400,R0         ;SET UP MSB FIRST
1819 017312 000423                              BR      2$              ;GO CALCULATE ADRS
1820 017314 012737 161400 031076        1$:     MOV     #161400,BUFFER  ;SET UP JUMP RELATIVE INSTR
1821 017322 050037 031076                       BIS     R0,BUFFER       ;SET UP RELATIVE ADRS PATTERN
1822 017326 012777 031076 162474                MOV     #BUFFER,@DPC    ;START
1823 017334 013737 002030 001122                MOV     DPC,$BDADR      ;SET UP REG ADRS 00
1824 017342 017737 162462 001126                MOV     @DPC,$BDDAT     ;READ NEW DPC ADRS
1825 017350 023737 001124 001126                CMP     $GDDAT,$BDDAT   ;DOES IT EQUAL THE CALCULATED VALUE?
1826 017356 001401                              BEQ     2$              ;BR IF OK
1827 017360 104011                              ERROR   11              ;JUMP RELATIVE NEGATIVE FAILED
1828 017362 006200                      2$:     ASR     R0              ;SHIFT RIGHT TO NEXT BIT
1829 017364 103413                              BCS     TST136          ;;ADVANCE TO NEXT TEST IF DONE
1830 017366 012737 031100 001124                MOV     #BUFFER+2,$GDDAT        ;SET UP CURRENT ADRS
1831 017374 010001                              MOV     R0,R1           ;GET RELATIVE ADRS VALUE
1832 017376 005401                              NEG     R1              ;MAKE NEG
1833 017400 006301                              ASL     R1              ;CORRECT FOR WORD OFFSET
1834 017402 042701 177000                       BIC     #177000,R1      ;GET RID OF MSBS
1835 017406 160137 001124                       SUB     R1,$GDDAT       ;SUB OFFSET FROM DPC VALUE
1836 017412 000740                              BR      1$              ;TRY NEXT RELATIVE ADRS
1837
1838                                    ;;*****************************************************************
 (3)                                    ;*TEST 136       TEST THAT JUMP ABS INSTR WILL SETUP DPC WITH A FLOATING ONE
 (3)                                    ;;*****************************************************************
 (2) 017414 000004                      TST136: SCOPE
 (2) 017416 012737 000136 001206                MOV     #136,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1839 017424 012777 020000 162430                MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1840 017432 012737 160000 031076                MOV     #160000,BUFFER  ;SET UP JUMP ABS INSTRUCTION AT 'BUFFER'
1841 017440 005037 001124                       CLR     $GDDAT          ;EXPECT ZEROS FIRST TIME
1842 017444 012737 000001 031100                MOV     #1,BUFFER+2     ;SET UP ABS ADRS TO #1
1843 017452 012777 031076 162350        1$:     MOV     #BUFFER,@DPC    ;START
1844 017460 012737 017452 001110                MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1845 017466 005277 162336                       INC     @DPC            ;PICK UP ABS ADRS       ** B **
1846 017472 013737 002030 001122                MOV     DPC,$BDADR      ;SET UP REG ADRS 00     ** B **
1847 017500 000240                              NOP                     ;ALLOW TIME FOR 2ND NPR
1848 017502 017737 162322 001126                MOV     @DPC,$BDDAT     ;GET ADRS JUMPED TO
1849 017510 023737 001124 001126                CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
1850 017516 001401                              BEQ     2$              ;BR IF OK
1851 017520 104010                              ERROR   10              ;JUMP ABSOLUTE FAILED (FLOAT A 1 ADDRESS)
1852 017522 006337 031100               2$:     ASL     BUFFER+2                ;ADVANCE ONE BIT LEFT
1853 017526 103404                              BCS     TST137          ;;ADVANCE TO NEXT TEST IF ALL BITS TESTED
1854 017530 013737 031100 001124                MOV     BUFFER+2,$GDDAT ;SET UP EXPECTED ADRS
1855 017536 000745                              BR      1$              ;TRY NEXT BIT
1856
1857                                    ;;*****************************************************************
 (3)                                    ;*TEST 137       TEST THAT JUMP ABS INSTR WILL SETUP DPC WITH A FLOATING ZERO
 (3)                                    ;;*****************************************************************
 (2) 017540 000004                      TST137: SCOPE
 (2) 017542 012737 000137 001206                MOV     #137,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
```

```
 1858  017550  012737  017600  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
 1859  017556  012777  020000  162276          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1860  017564  012737  160000  031076          MOV     #160000,BUFFER  ;SET UP JUMP ABS INSTR AT 'BUFFER'
 1861  017572  012737  177774  001124          MOV     #177774,$GDDAT  ;SET LSB TO ZERO
 1862  017600  012777  031076  162222  1$:     MOV     #BUFFER,@DPC    ;START
 1863  017606  013737  001124  031100          MOV     $GDDAT,BUFFER+2 ;SET UP ADRS
 1864  017614  005277  162210                  INC     @DPC            ;PICK UP ABS ADRS
 1865  017620  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
 1866  017626  017737  162176  001126          MOV     @DPC,$BDDAT     ;GET ADRS JUMPED TO
 1867  017634  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 1868  017642  001401                          BEQ     2$              ;BR IF OK
 1869  017644  104010                          ERROR   10              ;JUMP ABSOLUTE FAILED (TO A FLOATING ZERO)
 1870  017646  006337  001124          2$:     ASL     $GDDAT          ;SHIFT ZERO BIT LEFT
 1871  017652  103004                          BCC     TST140          ;;ADVANCE TO NEXT TEST IF ALL BITS TESTED
 1872  017654  062737  000002  001124          ADD     #2,$GDDAT       ;REPLACE LSB
 1873  017662  000746                          BR      1$              ;TRY NEXT BIT
 1874
 1875                                  ;;*****************************************************************
  (3)                                  ;*TEST 140      TEST THAT THE 'JSR ABSOLUTE' INSTR WILL SET 'JSR' BIT IN REG 32
  (3)                                  ;;*****************************************************************
  (2)  017664  000004                  TST140: SCOPE
  (2)  017666  012737  000140  001206          MOV     #140,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1876  017674  012777  020000  162160          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1877  017702  012737  162000  031076          MOV     #162000,BUFFER  ;SET UP JSR ABS INSTR
 1878  017710  005037  031100                  CLR     BUFFER+2                ;CLR ABS ADRS LOC
 1879  017714  012737  001000  001124          MOV     #BIT9,$GDDAT    ;EXPECT JSR BIT
 1880  017722  012777  031076  162100          MOV     #BUFFER,@DPC    ;START
 1881  017730  013737  002062  001122          MOV     STKPT,$BDADR    ;SET UP REG ARS 32
 1882  017736  032777  001000  162116          BIT     #BIT9,@STKPT    ;IS JSR BIT SET?
 1883  017744  001003                          BNE     TST141          ;;ADVANCE TO NEXT TEST IF OK
 1884  017746  005037  001126                  CLR     $BDDAT          ;SHOW THAT IT WAS ZERO
 1885  017752  104012                          ERROR   12              ;CALL ERROR ROUTINE USING MSG 23
 1886
 1887                                  ;;*****************************************************************
  (3)                                  ;*TEST 141      TEST 'JSR RELATIVE' INSTR (BIT 8=0) AT LOC 'BUFFER' & VERIFY NEW DPC ADR
  (3)                                  ;;*****************************************************************
  (2)  017754  000004                  TST141: SCOPE
  (2)  017756  012737  000141  001206          MOV     #141,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 1888  017764  012777  020000  162070          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 1889  017772  012737  020006  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
 1890  020000  012700  000400                  MOV     #400,R0         ;SET UP MSB FIRST
 1891  020004  000426                          BR      2$              ;GO CALCULATE ADRS
 1892  020006  052777  000040  162046  1$:     BIS     #40,@STKPT      ;RESET STK PTR
 1893  020014  012737  163000  031076          MOV     #163000,BUFFER  ;SET UP JSR RELATIVE INSTR
 1894  020022  050037  031076                  BIS     R0,BUFFER       ;SET UP RELATIVE ADRS PATTERN
 1895  020026  012777  031076  161774          MOV     #BUFFER @DPC    ;START
 1896  020034  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
 1897  020042  017737  161762  001126          MOV     @DPC,$BDDAT     ;READ NEW DPC ADRS
 1898  020050  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DOES IT EQUAL THE CALCULATED ADRS?
 1899  020056  001401                          BEQ     2$              ;BR IF OK
 1900  020060  104013                          ERROR   13              ;JSR RELATIVE 'POSITIVE' FAILED
 1901  020062  006200                  2$:     ASR     R0              ;SHIFT RIGHT TO NEXT BIT
 1902  020064  103410                          BCS     TST142          ;;ADVANCE TO NEXT TEST IF DONE
 1903  020066  010037  001124                  MOV     R0,$GDDAT       ;SET UP OFFSET VALUE
 1904  020072  006337  001124                  ASL     $GDDAT          ;MAKE INTO WORD OFFSET
 1905  020076  062737  031100  001124          ADD     #BUFFER+2,$GDDAT        ;ADD IN ADRS OF INSTR PLUS 2
```

```
1906  020104  000740                          BR      1$              ;TRY THIS RELATIVE ADRS
1907
1908                                  ;;*****************************************************************
 (3)                                  ;*TEST 142      TEST 'JSR RELATIVE' INSTR (BIT 8=1) AT LOC 'BUFFER' & VERIFY NEW DPC ADR
 (3)                                  ;;*****************************************************************
 (2)  020106  000004                  TST142: SCOPE
 (2)  020110  012737  000142  001206          MOV     #1+2,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1909  020116  012777  020000  161736          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1910  020124  012737  020140  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1911  020132  012700  000400                  MOV     #400,R0         ;SET UP MSB FIRST
1912  020136  000426                          BR      2$              ;GO CALCULATE ADRS
1913  020140  052777  000040  161714  1$:     BIS     #40,@STKPT      ;RESET STK PTR
1914  020146  012737  163400  031076          MOV     #163400,BUFFER  ;SET UP JSR RELATIVE INSTR
1915  020154  050037  031076                  BIS     R0,BUFFER       ;SET UP RELATIVE ADRS PATTERN
1916  020160  012777  031076  161642          MOV     #BUFFER,@DPC    ;START
1917  020166  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
1918  020174  017737  161630  001126          MOV     @DPC,$BDDAT     ;READ NEW DPC ADRS
1919  020202  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DOES IT EQUAL THE CALCULATED VALUE?
1920  020210  001401                          BEQ     2$              ;BR IF OK
1921  020212  104013                          ERROR   13              ;JSR RELATIVE NEGATIVE TO ADDRESS FAILED
1922  020214  006200                  2$:     ASR     R0              ;SHIFT RIGHT TO NEXT BIT
1923  020216  103413                          BCS     TST143          ;;ADVANCE TO NEXT TEST IF DONE
1924  020220  012737  031100  001124          MOV     #BUFFER+2,$GDDAT        ;SET UP CURRENT ADRS
1925  020226  010001                          MOV     R0,R1           ;GET RELATIVE ADRS VALUE
1926  020230  005401                          NEG     R1              ;MAKE NEG
1927  020232  006301                          ASL     R1              ;CORRECT FOR WORD OFFSET
1928  020234  042701  177000                  BIC     #177000,R1      ;GET RID OF MSBS
1929  020240  160137  001124                  SUB     R1,$GDDAT       ;SUB OFFSET FROM DPC VALUE
1930  020244  000735                          BR      1$              ;TRY NEXT RELATIVE ADRS
1931
1932                                  ;;*****************************************************************
 (3)                                  ;*TEST 143      TEST THAT 'JSR ABS' INSTR WILL SET UP DPC WITH A FLOATING ONE
 (3)                                  ;;*****************************************************************
 (2)  020246  000004                  TST143: SCOPE
 (2)  020250  012737  000143  001206          MOV     #143,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1933  020256  012777  020000  161576          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1934  020264  012737  162000  031076          MOV     #162000,BUFFER  ;SET UP JSR ABSOLUTE INSTR AT 'BUFFER'
1935  020272  005037  001124                  CLR     $GDDAT          ;EXPECT ZEROS FIRST TIME
1936  020276  012737  000001  031100          MOV     #1,BUFFER+2     ;SET UP ABS ADRS TO 1
1937  020304  052777  000040  161550  1$:     BIS     #40,@STKPT      ;RESET STK PTR
1938  020312  012777  031076  161510          MOV     #BUFFER,@DPC    ;START
1939  020320  012737  020304  001110          MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1940  020326  005277  161476                  INC     @DPC            ;ADVANCE TO ABS ADRS
1941  020332  013737  002030  001122          MOV     DPC,$BDADR      ;SET UP REG ADRS 00
1942  020340  017737  161464  001126          MOV     @DPC,$BDDAT     ;GET ADRS JUMPED TO
1943  020346  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
1944  020354  001401                          BEQ     2$              ;BR IF OK
1945  020356  104012                          ERROR   12              ;JSR ABSOLUTE FAILED (TO A FLOATING ONE)
1946  020360  006337  031100          2$:     ASL     BUFFER+2                ;ADVANCE ONE BIT LEFT
1947  020364  103404                          BCS     TST144          ;;ADVANCE TO NEXT TEST IF ALL BITS TESTED
1948  020366  013737  031100  001124          MOV     BUFFER+2,$GDDAT ;SET UP EXPECTED ADRS
1949  020374  000743                          BR      1$              ;TRY NEXT BIT
1950
1951                                  ;;*****************************************************************
 (3)                                  ;*TEST 144      TEST THAT JSR ABS INSTR WILL SET UP DPC WITH A FLOATING ZERO
 (3)                                  ;;*****************************************************************
```

```
  (2) 020376 000004                    TST144: SCOPE
  (2) 020400 012737 000144 001206              MOV     #144,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1952 020406 012777 020000 161446               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1953 020414 012737 020436 001110               MOV     #1$,$LPERR      ;SET UP SCOPE LOOP ADRS
1954 020422 012737 162000 031076               MOV     #162000,BUFFER  ;SET UP JSR ABSOLUTE AT 'BUFFER'
1955 020430 012737 177774 001124               MOV     #177774,$GDDAT  ;SET LSB TO ZERO
1956 020436 052777 000040 161416 1$:           BIS     #40,@STKPT      ;RESET STK PTR
1957 020444 012777 031076 161356               MOV     #BUFFER,@DPC    ;START
1958 020452 013737 001124 031100               MOV     $GDDAT,BUFFER+2 ;SET UP ADRS
1959 020460 005277 161344                      INC     @DPC            ;ADVANCE TO ABS ADRS
1960 020464 013737 002030 001122               MOV     DPC,$BDADR      ;SET UP REG ADRS 00
1961 020472 017737 161332 001126               MOV     @DPC,$BDDAT     ;GET ADRS JUMPED TO
1962 020500 023737 001124 001126               CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
1963 020506 001401                             BEQ     2$              ;BR IF OK
1964 020510 104012                             ERROR   12              ;JSR ABSOLUTE FAILED (TO A FLOATING ZERO)
1965 020512 006337 001124              2$:     ASL     $GDDAT          ;SHIFT ZERO BIT LEFT
1966 020516 103004                             BCC     TST145          ;;ADVANCE TO NEXT TEST IF ALL BITS TESTED
1967 020520 062737 000002 001124               ADD     #2,$GDDAT       ;REPLACE LSB
1968 020526 000743                             BR      1$              ;TRY NEXT BIT
1969
1970                                 ;;*****************************************************************
  (3)                                ;*TEST 145      TEST THAT GRAPHPOLT X CAN LOAD X POSITION REGISTER WITH #1252
  (3)                                ;;*****************************************************************
  (2) 020530 000004                    TST145: SCOPE
  (2) 020532 012737 000145 001206              MOV     #145,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1971 020540 013737 002034 001122               MOV     XPOS,$BDADR     ;SET UP REG ADRS 04
1972 020546 012777 020000 161306               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1973 020554 012700 031076                      MOV     #BUFFER,R0      ;LOAD BUFFER POINTER
1974 020560 012720 122000                      MOV     #122000,(R0)+   ;LOW INTERSITY - SET GRAPH PLOT X MODE
1975 020564 012720 001252                      MOV     #1252,(R0)+     ;SET X POSITION
1976 020570 013777 002000 161232               MOV     DBUF,@DPC       ;LOAD DISPLAY P.C.
1977 020576 012737 001252 001124               MOV     #1252,$GDDAT            ;LOAD EXPECTED
1978 020604 005277 161220                      INC     @DPC            ;SINGLE STEP THE DISPLAY
1979 020610 004737 024514                      JSR     PC,DLAY         ;STALL
1980 020614 017737 161214 001126               MOV     @XPOS,$BDDAT    ;READ X POSITION
1981 020622 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
1982 020630 001401                             BEQ     TST146          ;;BR IF EQUAL
1983 020632 104014                             ERROR   14              ;X POSITION REGISTER FAILED TO LOAD
1984                                                                   ;PROPERLY USING GRAPH PLOT X MODE
1985
1986                                 ;;*****************************************************************
  (3)                                ;*TEST 146      TEST THAT GRAPHPLOT X CAN LOAD X POSITION REGISTER WITH #525
  (3)                                ;;*****************************************************************
  (2) 020634 000004                    TST146: SCOPE
  (2) 020636 012737 000146 001206              MOV     #146,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
1987 020644 013737 002034 001122               MOV     XPOS,$BDADR     ;SET UP REG ADRS 04
1988 020652 012777 020000 161202               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
1989 020660 012700 031076                      MOV     #BUFFER,R0      ;LOAD BUFFER POINTER
1990 020664 012720 122000                      MOV     #122000,(R0)+   ;LOW INTENSITY - SET GRAPH PLOT X MODE
1991 020670 012720 000525                      MOV     #525,(R0)+      ;SET X POSITION
1992 020674 012777 031076 161126               MOV     #BUFFER,@DPC            ;STAR DISPLAY
1993 020702 012737 000525 001124               MOV     #525,$GDDAT             ;LOAD EXPECTED
1994 020710 005277 161114                      INC     @DPC            ;SINGLE STEP THE DISPLAY
1995 020714 004737 024514                      JSR     PC,DLAY         ;STALL
1996 020720 017737 161110 001126               MOV     @XPOS,$BDDAT    ;READ X POSITION
1997 020726 023737 001124 001126               CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
```

```
 1998  020734  001401                          BEQ     TST147          ;;BR IF EQUAL
 1999  020736  104014                          ERROR   14              ;X POSITION REGISTER FAILED TO LOAD
 2000                                                                  ;PROPERLY USING GRAPH PLOT X MODE
 2001
 2002                                  ;;*****************************************************************
  (3)                                  ;*TEST 147      TEST THAT GRAPHPLOT Y CAN LOAD Y POSITION REGISTER WITH #1252
  (3)                                  ;;*****************************************************************
  (2)  020740  000004                  TST147: SCOPE
  (2)  020742  012737  000147  001206          MOV     #147,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2003  020750  013737  002036  001122          MOV     YPOS,$BDADR     ;SET UP REG ADRS 06
 2004  020756  012777  020000  161076          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 2005  020764  012700  031076                  MOV     #BUFFER,R0      ;LOAD BUFFER POINTER
 2006  020770  012720  126000                  MOV     #126000,(R0)+   ;LOW INTENSITY - SET GRAPHPLOT Y
 2007  020774  012720  001252                  MOV     #1252,(R0)+     ;SET Y POSITION
 2008  021000  013777  002000  161022          MOV     DBUF,@DPC       ;LOAD THE DISPLAY P.C.
 2009  021006  012737  001252  001124          MOV     #1252,$GDDAT    ;LOAD EXPECTED
 2010  021014  005277  161010                  INC     @DPC            ;SINGLE STEP THE DISPLAY
 2011  021020  004737  024514                  JSR     PC,DLAY         ;STALL
 2012  021024  017737  161006  001126          MOV     @YPOS,$BDDAT    ;READ Y POSITION
 2013  021032  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 2014  021040  001401                          BEQ     TST150          ;;BR IF EQUAL
 2015  021042  104015                          ERROR   15              ;Y POSITION REGISTER FAILED TO LOAD
 2016                                                                  ;PROPERLY USING GRAPHPLOT Y MODE
 2017
 2018                                  ;;*****************************************************************
  (3)                                  ;*TEST 150      TEST THAT GRAPHPLOT Y CAN LOAD Y POSITION REGISTER WITH #525
  (3)                                  ;;*****************************************************************
  (2)  021044  000004                  TST150: SCOPE
  (2)  021046  012737  000150  001206          MOV     #150,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2019  021054  013737  002036  001122          MOV     YPOS,$BDADR     ;SET UP REG ADRS 06
 2020  021062  012777  020000  160772          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 2021  021070  012700  031076                  MOV     #BUFFER,R0      ;LOAD BUFFER POINTER
 2022  021074  012720  126000                  MOV     #126000,(R0)+   ;LOW INTENSITY - SET GRAPHPLOT Y MODE
 2023  021100  012720  000525                  MOV     #525,(R0)+      ;SET Y POSITION
 2024  021104  013777  002000  160716          MOV     DBUF,@DPC       ;LOAD THE DISPLAY P.C.
 2025  021112  012737  000525  001124          MOV     #525,$GDDAT     ;LD EXPECTED
 2026  021120  005277  160704                  INC     @DPC            ;SINGLE STEP THE DISPLAY
 2027  021124  004737  024514                  JSR     PC,DLAY         ;STALL
 2028  021130  017737  160702  001126          MOV     @YPOS,$BDDAT    ;READ Y POSITION
 2029  021136  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
 2030  021144  001401                          BEQ     TST151          ;;BR IF EQUAL
 2031  021146  104015                          ERROR   15              ;Y POSITION REGISTER FAILED TO LOAD
 2032                                                                  ;PROPERLY USING GRAPHPLOT Y MODE
 2033
 2034
 2035                                  ;;*****************************************************************
  (3)                                  ;*TEST 151      TEST FOR PROPER SELECTION OF X AND Y REGISTERS
  (3)                                  ;;*****************************************************************
  (2)  021150  000004                  TST151: SCOPE
  (2)  021152  012737  000151  001206          MOV     #151,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2036  021160  013737  002034  001122          MOV     XPOS,$BDADR     ;SET UP REG ADRS 04
 2037  021166  012777  020000  160666          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 2038  021174  012700  031076                  MOV     #BUFFER,R0      ;LOAD BUFFER POINTER
 2039  021200  012720  122000                  MOV     #122000,(R0)+   ;LOW INTENSITY - SET GRAPHPLOT X MODE
 2040  021204  012720  001234                  MOV     #1234,(R0)+     ;SET X POSITION
 2041  021210  012720  126000                  MOV     #126000,(R0)+   ;SET GRAPHPLOT Y MODE
```

```
2042  021214  012720  001432                  MOV     #1432,(R0)+     ;SET Y POSITION
2043  021220  013777  002000  160602          MOV     DBUF,@DPC       ;LOAD THE DISPLAY P.C.
2044  021226  012737  001234  001124          MOV     #1234,$GDDAT    ;LD EXPECTED
2045  021234  005277  160570                  INC     @DPC            ;SINGLE STEP THE DISPLAY
2046  021240  004737  024514                  JSR     PC,DLAY         ;STALL
2047  021244  017737  160564  001126          MOV     @XPOS,$BDDAT    ;READ X POSITION
2048  021252  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
2049  021260  001402                          BEQ     1$
2050  021262  104014                          ERROR   14              ;GRAPHPLOT X MODE FAILED TO SELECT
2051  021264  000424                          BR      TST152          ;;
2052
2053  021266                          1$:
 (1)  021266  005277  160536                  INC     @DPC            ;SINGLE STEP THE DISPLAY
2054  021272  013737  002036  001122          MOV     YPOS,$BDADR     ;SET UP REG ADRS 06
2055  021300  005277  160524                  INC     @DPC            ;SINGLE STEP THE DISPLAY
2056  021304  012737  001432  001124          MOV     #1432,$GDDAT    ;LOAD EXPECTED
2057  021312  004737  024514                  JSR     PC,DLAY         ;STALL
2058  021316  017737  160514  001126          MOV     @YPOS,$BDDAT    ;READ Y POSITION
2059  021324  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE EXPCT TO RCVD
2060  021332  001401                          BEQ     TST152          ;;BR IF EQUAL
2061  021334  104015                          ERROR   15              ;Y POSITION REGISTER FAILED TO LOAD
2062                                                                  ;PROPERLY USING GRAPHPLOT Y MODE
2063
2064                                  ;;**********************************************************************
 (3)                                  ;*TEST 152       TEST THAT RESET CLEARS X AND Y POSITION REG.
 (3)                                  ;;**********************************************************************
 (2)  021336  000004                  TST152: SCOPE
 (1)  021340  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 (2)  021346  012737  000152  001206          MOV     #152,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2065  021354  013737  002034  001122          MOV     XPOS,$BDADR     ;SET UP REG ADRS 04
2066  021362  012737  114000  031076          MOV     #114000,BUFFER  ;LOAD 'POINT' INST.
2067  021370  012777  020000  160464          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
2068  021376  012777  031076  160424          MOV     #BUFFER,@DPC    ;START DISPLAY
2069  021404  012737  001777  031100          MOV     #1777,BUFFER+2  ;X=1777
2070  021412  005277  160412                  INC     @DPC            ;SINGLE STEP THE DISPLAY
2071  021416  012737  001777  031102          MOV     #1777,BUFFER+4  ;Y=1777
2072  021424  005277  160400                  INC     @DPC            ;SINGLE STEP THE DISPLAY
2073  021430  004737  024514                  JSR     PC,DLAY         ;STALL
2074
2075  021434  000005                          RESET                   ;INITILIZE
2076  021436  005037  001124                  CLR     $GDDAT          ;CLEAR EXPECTED
2077  021442  017737  160366  001126          MOV     @XPOS,$BDDAT    ;READ X POSITION
2078  021450  001401                          BEQ     1$              ;;BR IF CLEARED
2079  021452  104032                          ERROR   32              ;RESET FAILED TO CLEAR X POS. REGISTER
2080
2081  021454  013737  002036  001122  1$:     MOV     YPOS,$BDADR     ;SET UP REG ADRS 06
2082  021462  017737  160350  001126          MOV     @YPOS,$BDDAT    ;READ Y POSITION
2083  021470  001401                          BEQ     TST153          ;;BR IF CLARED
2084  021472  104032                          ERROR   32              ;RESET FAILED TO CLEAR Y POS. REGISTER
2085                                  ;;**********************************************************************
 (3)                                  ;*TEST 153       TEST THAT DISPLAY BUSY WILL SET & THEN CLR ON INTERNAL STOP
 (3)                                  ;;**********************************************************************
 (2)  021474  000004                  TST153: SCOPE
 (2)  021476  012737  000153  001206          MOV     #153,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2086  021504  012777  020000  160350          MOV     #BIT13,@STKPT   ;SET MAINT SW 2
2087  021512  012737  100000  001124          MOV     #100000,$GDDAT  ;EXPECT DISPLAY BUSY INITIALLY
```

```
2088  021520  012737  164000  031076          MOV     #164000,BUFFER  ;SET UP A NOP INSTR
2089  021526  012737  172000  031100          MOV     #172000,BUFFER+2        ;SET UP LD STATUS A INSTR WITH STOP
2090  021534  012777  031076  160266          MOV     #BUFFER,@DPC    ;START
2091  021542  013737  002042  001122          MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
2092  021550  005777  160266                  TST     @SREG1          ;SEE THAT DISPLAY BUSY IS SET
2093  021554  100403                          BMI     1$              ;BR IF SO
2094  021556  005037  001126                  CLR     $BDDAT          ;INDICATE DISPLAY BUSY NOT SET
2095  021562  104044                          ERROR   44              ;DISPLAY BUSY NOT SET
2096  021564  005277  160240          1$:     INC     @DPC            ;ADVANCE TO STOP INSTR
2097  021570  005037  001124                  CLR     $GDDAT          ;EXPECTE ZERO
2098  021574  005777  160242                  TST     @SREG1          ;SEE THAT DISPLAY BUSY HAS CLEARED
2099  021600  100004                          BPL     TST154          ;;GO TO NEXT TEST IF OK
2100  021602  012737  100000  001126          MOV     #100000,$BDDAT  ;INDICATE DISPLAY BUSY WAS STILL SET
2101  021610  104044                          ERROR   44              ;DISPLAY BUSY FAILED TO CLR ON STOP INSTR
2102
2103
2104                                  ;;**********************************************************************
 (3)                                  ;*TEST 154       TEST THAT RESET WILL CLEAR DISPLAY BUSY
 (3)                                  ;;**********************************************************************
 (2)  021612  000004                  TST154: SCOPE
 (1)  021614  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 (2)  021622  012737  000154  001206          MOV     #154,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2105  021630  013737  002042  001122          MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
2106  021636  012777  020000  160216          MOV     #BIT13,@STKPT   ;SET MAINT SW 2
2107  021644  012737  164000  031076          MOV     #164000,BUFFER  ;SET UP NOP INSTR
2108  021652  012777  031076  160150          MOV     #BUFFER,@DPC    ;START
2109  021660  012737  000404  001124          MOV     #404,$GDDAT     ;EXPECT ONLY CHAR & VECTOR SCALE BITS
2110  021666  000005                          RESET                   ;CLR BUSY
2111  021670  017737  160146  001126          MOV     @SREG1,$BDDAT   ;READ REG 12
2112  021676  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
2113  021704  001401                          BEQ     TST155          ;;NEXT TEST IF OK
2114  021706  104044                          ERROR   44              ;RESET FAILED TO CLR DISPLAY BUSY
2115
2116                                  ;;**********************************************************************
 (3)                                  ;*TEST 155       TEST THAT EXTERNAL STOP WILL SET & RESET WITH DISPLAY OFF
 (3)                                  ;;**********************************************************************
 (2)  021710  000004                  TST155: SCOPE
 (1)  021712  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 (2)  021720  012737  000155  001206          MOV     #155,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2117  021726  012777  020000  160126          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
2118  021734  012737  000340  177776          MOV     #340,PSW        ;SET PRIORITY TO HIGEST LEVEL
2119  021742  000005                          RESET                   ;MAKE SURE DISPLAY IS OFF
2120  021744  013737  002042  001122          MOV     SREG1,$BDADR    ;SET UP REG 12 ADRS
2121  021752  012737  000200  001124          MOV     #200,$GDDAT     ;EXPECT EXTERNAL STOP TO BE SET
2122  021760  013777  001124  160054          MOV     $GDDAT,@SREG1   ;SET EXTERNAL STOP
2123  021766  105777  160050                  TSTB    @SREG1          ;SEE IF BIT SET
2124  021772  100403                          BMI     1$              ;BR IF IT DID SET
2125  021774  005037  001126                  CLR     $BDDAT          ;INDICATE IT WAS NOT SET
2126  022000  104045                          ERROR   45              ;EXTERNAL STOP FAILED TO SET
2127  022002  012777  031076  160020  1$:     MOV     #BUFFER,@DPC    ;CLR EXTERNAL STOP BIT
2128  022010  105777  160026                  TSTB    @SREG1          ;CHECK THAT IT WAS CLEARED
2129  022014  100006                          BPL     TST156          ;;BR TO NEXT TEST IF OK
2130  022016  005037  001124                  CLR     $GDDAT          ;EXPECTED ZERO
2131  022022  012737  000200  001126          MOV     #200,$BDDAT     ;INDICATE THAT IT DID NOT CLR
2132  022030  104045                          ERROR   45              ;EXTERNAL STOP FAILED TO CLR ON START
2133
```

```
2134                                    ;;*****************************************************************
 (3)                                    ;*TEST 156      TEST THAT EXTERNAL STOP WILL SET & RESET WITH DISPLAY BUSY
 (3)                                    ;;*****************************************************************
 (2) 022032 000004                      TST156: SCOPE
 (2) 022034 012737 000156 001206               MOV     #156,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2135 022042 012777 020000 160012               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
2136 022050 012737 000340 177776               MOV     #340,PSW        ;SET PSW TO 7
2137 022056 012737 000200 001124               MOV     #200,$GDDAT     ;EXPECT EXTERNAL STOP TO BE SET INITIALLY
2138 022064 012737 164000 031076               MOV     #164000,BUFFER  ;SET UP A NOP INSTR
2139 022072 012777 031076 157730               MOV     #BUFFER,@DPC    ;START
2140 022100 012737 164000 031100               MOV     #164000,BUFFER+2        ;SET UP ANOTHER NOP
2141 022106 013777 001124 157726               MOV     $GDDAT,@SREG1   ;ISSUE A SOFTWARE STOP
2142 022114 005277 157710                      INC     @DPC            ;THIS RESUME SHOULD ALLOW EXTERNAL STOP TO SET
2143 022120 013737 002042 001122               MOV     SREG1,$BDADR    ;SET UP REG ADRS 12
2144 022126 032777 000200 157706               BIT     #200,@SREG1     ;SEE IF IT SET
2145 022134 001004                             BNE     1$              ;BR IF SO
2146 022136 005037 001126                      CLR     $BDDAT          ;INDICATE IT DID NOT SET
2147 022142 104045                             ERROR   45              ;EXTERNAL STOP FAILED TO SET WHEN DISPLAY BUSY
2148 022144 000415                             BR      TST157          ;;GO LOOK FOR LOOP ON TEST SW
2149 022146 012777 031076 157654 1$:           MOV     #BUFFER,@DPC    ;START - THIS SHOULD CLR EXT STOP
2150 022154 032777 000200 157660               BIT     #200,@SREG1     ;CHECK THAT IT DID CLEAR
2151 022162 001406                             BEQ     TST157          ;;GO TO NEXT TEST IF EXTERNAL STOP CLEARED
2152 022164 005037 001124                      CLR     $GDDAT          ;EXPECTED ZERO
2153 022170 012737 000200 001126               MOV     #200,$BDDAT     ;INDICATE IT DID NOT CLR
2154 022176 104045                             ERROR   45              ;EXTERNAL STOP FAILED TO CLR ON START
2155
2156                                    ;;*****************************************************************
 (3)                                    ;*TEST 157      TEST THAT RESET WILL CLEAR EXTERNAL STOP
 (3)                                    ;;*****************************************************************
 (2) 022200 000004                      TST157: SCOPE
 (1) 022202 012737 000010 001166               MOV     #10,$TIMES      ;;DO 10 ITERATIONS
 (2) 022210 012737 000157 001206               MOV     #157,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2157 022216 000005                             RESET                   ;MAKE SURE DISPLAY IS OFF
2158 022220 013737 002042 001122               MOV     SREG1,$BDADR    ;SET UP REG 12 ADRS
2159 022226 005037 001124                      CLR     $GDDAT          ;EXPECT RESET STATE
2160 022232 012777 000200 157602               MOV     #200,@SREG1     ;SET EXTERNAL STOP
2161 022240 000005                             RESET                   ;CLR IT
2162 022242 032777 000200 157572               BIT     #200,@SREG1     ;DID IT CLR?
2163 022250 001404                             BEQ     TS1160          ;;NEXT TEST IF SO
2164 022252 012737 000200 001126               MOV     #200,$BDDAT     ;SHOW THAT IT WAS STILL SET
2165 022260 104045                             ERROR   45              ;RESET FAILED TO CLR EXTERNAL STOP
2166
2167                                    ;;*****************************************************************
 (3)                                    ;*TEST 160      TEST THAT EXTERNAL STOP WILL CAUSE AN INTERRUPT
 (3)                                    ;;*****************************************************************
 (2) 022262 000004                      TST160: SCOPE
 (1) 022264 012737 000040 001166               MOV     #40,$TIMES      ;;DO 40 ITERATIONS
 (2) 022272 012737 000160 001206               MOV     #160,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2168 022300 012777 020000 157554               MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
2169 022306 000005                             RESET                   ;MAKE SURE DISPLAY IS OFF
2170 022310 012737 000340 177776               MOV     #340,PSW        ;SET PSW TO 7
2171 022316 013737 002042 001122               MOV     SREG1,$BDADR    ;SET UP REG 12 ADRS
2172 022324 012737 000200 001124               MOV     #200,$GDDAT     ;EXPECT EXTERNAL STOP FLAG
2173 022332 013701 001252                      MOV     $VECT1,R1       ;GET BASIC VECTOR ADRS
2174 022336 042701 160000                      BIC     #160000,R1      ; CLEAR PSW BITS        ** B **
2175 022342 012711 022406                      MOV     #1$,(R1)        ;SET UP STOP INT ADRS
```

```
 2176  022346  012761  000340  000002          MOV     #340,2(R1)      ;SET UP PSW TO 7 ON INT
 2177  022354  012777  000200  157460          MOV     #200,@SREG1     ;CAUSE AN EXTERNAL STOP INT
 2178  022362  005037  177776                  CLR     PSW             ;ALLOW INT TO OCCUR
 2179  022366  017737  157450  001126          MOV     @SREG1,$BDDAT   ;READ REG 12
 2180  022374  042737  177577  001126          BIC     #177577,$BDDAT  ;SAVE ONLY STOP FLAG
 2181  022402  104033                          ERROR   33              ;EXTERNAL STOP FAILED TO INTERRUPT
 2182  022404  000410                          BR      2$              ;GO LOOK FOR LOOP ON TEST SWITCH
 2183  022406  022626                  1$:     CMP     (R6)+,(R6)+     ;FIX STK SINCE NO RETURN
 2184  022410  032777  000200  157424          BIT     #200,@SREG1     ;CK THAT STOP FLAG IS SET
 2185  022416  001003                          BNE     2$              ;BR IF SO
 2186  022420  005037  001126                  CLR     $BDDAT          ;INDICATE FLAG WAS NOT SET
 2187  022424  104033                          ERROR   33              ;EXTERNAL STOP INTERRUPTED BUT FLAG NOT SET
 2188  022426  004737  024454          2$:     JSR     PC,RSTVEC       ;RESTORE EXTERNAL STOP VECTOR WITH HALT
 2189
 2190                                  ;;*******************************************************************
 . (3)                                ;*TEST 161      TEST THAT TIME OUT WILL CAUSE AN INTERRUPT
   (3)                                ;;*******************************************************************
   (2)  022432  000004                  TST161: SCOPE
   (1)  022434  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
   (2)  022442  012737  000161  001206          MOV     #161,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2191  022450  012777  020000  157404          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 2192  022456  012737  000340  177776          MOV     #340,PSW        ;SET PRIORITY TO HIGHEST LEVEL
 2193  022464  013701  001252                  MOV     $VECT1,R1       ;GET BASIC VECTOR ADRS
 2194  022470  042701  160000                  BIC     #160000,R1      ; CLEAR PSW BITS        ** B **
 2195  022474  012761  022566  000010          MOV     #1$,10(R1)      ;SET UP TIMEOUT INT SERVICE ADRS
 2196  022502  012761  000340  000012          MOV     #340,12(R1)     ;SET UP PRIORITY TO 7 ON INT
 2197  022510  013737  002042  001122          MOV     SREG1,$BDADR    ;SET UP REG 12 ADRS
 2198  022516  012737  004060  001124          MOV     #4060,$GDDAT    ;EXPECT TIMEOUT & DPC 16,17
 2199  022524  012777  007600  157306          MOV     #7600,@RLO      ;RELOCATE TO ADRS 760000
 2200  022532  005037  177776                  CLR     PSW             ;ALLOW INTERRUPT TO OCCUR
 2201  022536  012777  000000  157264          MOV     #0,@DPC         ;DISPLAY WILL TIMEOUT AT ADRS 760000
 2202  022544  021616                          CMP     (SP),(SP)       ;ALLOW TIME
 2203  022546  017737  157270  001126          MOV     @SREG1,$BDDAT   ;READ REG 12
 2204  022554  042737  173717  001126          BIC     #173717,$BDDAT  ;SAVE ONLY TIMEOUT BIT & DPC 16,17
 2205  022562  104046                          ERROR   46              ;VS60 FAILED TO INTERRUPT ON TIME OUT
 2206  022564  000414                          BR      2$              ;GO LOOK FOR LOOP ON TEST SWITCH
 2207  022566  022626                  1$:     CMP     (R6)+,(R6)+     ;FIX STACK SINCE NO RETURN
 2208  022570  017737  157246  001126          MOV     @SREG1,$BDDAT   ;READ STATUS
 2209  022576  042737  173717  001126          BIC     #173717,$BDDAT  ;SAVE TIMEOUT & DPC 16,17
 2210  022604  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CORRECT?
 2211  022612  001401                          BEQ     2$              ;BR IF SO
 2212  022614  104046                          ERROR   46              ;TIMEOUT BIT FAILED TO SET ON INTERRUPT
 2213  022616  004737  024454          2$:     JSR     PC,RSTVEC       ;RESET TIMEOUT VECTOR WITH HALT
 2214  022622  000005                          RESET                   ;CLR RELOCATE REG BEFORE ADVANCING
 2215
 2216                                  ;;*******************************************************************
   (3)                                ;*TEST 162      TEST THAT RESET WILL CLEAR TIMEOUT FLAG
   (3)                                ;;*******************************************************************
   (2)  022624  000004                  TST162: SCOPE
   (1)  022626  012737  000040  001166          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
   (2)  022634  012737  000162  001206          MOV     #162,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2217  022642  013737  002042  001122          MOV     SREG1,$BDADR    ;SET L.P. REG ADRS 12
 2218  022650  012777  020000  157204          MOV     #BIT13,@STKPT   ;SET MAINT SW 2
 2219  022656  012737  000340  177776          MOV     #340,PSW        ;DON'T LET INTR OCCUR.
 2220  022664  012737  000404  001124          MOV     #404,$GDDAT     ;EXPECT ONLY CHAR & VECTOR SCALE BITS
 2221  022672  012777  007600  157140          MOV     #7600,@RLO      ;RELOCATE TO ADRS 760000
```

```
 2222  022700  012777  000000  157122          MOV     #0,@DPC          ;DISPLAY SHOULD TIMEOUT AT ADRS 760000
 2223  022706  021616                          CMP     (SP),(SP)        ;ALLOW TIME
 2224  022710  000005                          RESET                    ;CLR REG REG 12
 2225  022712  017737  157124  001126          MOV     @SREG1,$BDDAT    ;READ REG 12
 2226  022720  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;IS IT CORRECT?
 2227  022726  001401                          BEQ     TST163           ;;NEXT TEST IF OK
 2228  022730  104032                          ERROR   32               ;RESET FAILED TO CLR TIMEOUT
 2229
 2230                                  ;;*****************************************************************
  (3)                                  ;*TEST 163      TEST THAT LIGHT PEN FLAG WILL SET THEN CLEAR ON RESET - CONS 0
  (3)                                  ;;*****************************************************************
  (2)  022732  000004                  TST163: SCOPE
  (1)  022734  012737  000040  001166          MOV     #40,$TIMES       ;;DO 40 ITERATIONS
  (2)  022742  012737  000163  001206          MOV     #163,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 2231  022750  012737  000340  177776          MOV     #340,PSW         ;HIGHEST PRIORITY
 2232  022756  012777  020000  157076          MOV     #20000,@STKPT    ;SET MAINT SW 2
 2233  022764  012737  144000  001124          MOV     #144000,$GDDAT   ;EXPECT L.P. FLAG INITIALLY
 2234  022772  012737  164360  031076          MOV     #164360,BUFFER   ;ENABLE L.P. FLG INTR
 2235  023000  012777  031076  157022          MOV     #BUFFER,@DPC     ;START
 2236  023006  052777  040000  157046          BIS     #40000,@STKPT    ;SET MAINT SW 3
 2237  023014  012737  122000  031100          MOV     #122000,BUFFER+2         ;GRAPH X MODE
 2238  023022  005277  157002                  INC     @DPC             ;RESUME
 2239  023026  012737  001777  031102          MOV     #1777,BUFFER+4   ;X=1777
 2240  023034  005277  156770                  INC     @DPC             ;RESUME
 2241  023040  013737  002052  001122          MOV     CONS,$BDADR      ;SET UP REG ADRS 22
 2242  023046  052777  040000  156776          BIS     #40000,@CONS     ;SET L.P. FLAG
 2243  023054  017737  156772  001126          MOV     @CONS,$BDDAT     ;READ FLAGS REG 22
 2244  023062  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;IS IT CORRECT?
 2245  023070  001401                          BEQ     1$               ;BR IF SO
 2246  023072  104034                          ERROR   34               ;LIGHT PEN FLAG FAILED TO SET - CONSOLE 0
 2247  023074  000005                  1$:     RESET                    ;CLR FLAGS
 2248  023076  012737  100000  001124          MOV     #100000,$GDDAT   ;EXPECT ONLY INT ENA CONSOLE 0
 2249  023104  017737  156742  001126          MOV     @CONS,$BDDAT     ;READ REG 22
 2250  023112  023737  001124  001126          CMP     $GDDAT,$BDDAT    ;DID REG 22 RESET?
 2251  023120  001401                          BEQ     TST164           ;;NEXT TEST IF SO
 2252  023122  104032                          ERROR   32               ;RESET FAILED TO CLEAR FLAGS - CONSOLE 0
 2253
 2254                                  ;;*****************************************************************
  (3)                                  ;*TEST 164      TEST THAT LIGHT PEN FLAG WILL SET THEN CLEAR ON RESET - CONS 1
  (3)                                  ;;*****************************************************************
  (2)  023124  000004                  TST164: SCOPE
  (1)  023126  012737  000040  001166          MOV     #40,$TIMES       ;;DO 40 ITERATIONS
  (2)  023134  012737  000164  001206          MOV     #164,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 2255  023142  012737  000340  177776          MOV     #340,PSW         ;HIGHEST PRIORTY
 2256  023150  012777  020000  156704          MOV     #20000,@STKPT    ;SET MAINT SW 2
 2257  023156  012737  101440  001124          MOV     #101440,$GDDAT   ;EXPECT L.P. FLAG INITIALLY
 2258  023164  012737  164760  031076          MOV     #164760,BUFFER   ;ENABLE L.P. FLG
 2259  023172  012777  031076  156630          MOV     #BUFFER,@DPC     ;START
 2260  023200  052777  040000  156654          BIS     #40000,@STKPT    ;SET MAINT SW 3
 2261  023206  012737  122000  031100          MOV     #122000,BUFFER+2         ;GRAPH X MODE
 2262  023214  005277  156610                  INC     @DPC             ;RESUME
 2263  023220  012737  001777  031102          MOV     #1777,BUFFER+4   ;X=1777
 2264  023226  005277  156576                  INC     @DPC             ;RESUME
 2265  023232  013737  002052  001122          MOV     CONS,$BDADR      ;SET UP REG ADRS 22
 2266  023240  052777  000400  156604          BIS     #400,@CONS       ;SET L.P. FLAG
 2267  023246  017737  156600  001126          MOV     @CONS,$BDDAT     ;READ IT BACK
```

```
                                        CMP     $GDDAT,$BDDAT   ;CORRECT?
                                        BEQ     1$              ;BR IF SO
                                        ERROR   34              ;LIGHT PEN FLAG FAILED TO SET - CONSOLE 1
                                1$:     RESET                   ;CLR FLAGS
                                        MOV     #100000,$GDDAT  ;EXPECT ONLY INT EN FOR CONSOLE 0
                                        MOV     @CONS,$BDDAT    ;READ REG 22
                                        CMP     $GDDAT,$BDDAT   ;CORRECT?
                                        BEQ     TST165          ;;NEXT TEST IF SO
                                        ERROR   32              ;RESET FAILED TO CLR FLAGS - CONSOLE 1

                                ;;*******************************************************************
                                ;*TEST 165      TEST THAT LIGHT PEN FLAG - CONSOLE 0 WILL CAUSE AN INTERRUPT
                                ;;*******************************************************************
                                TST165: SCOPE
                                        MOV     #100,$TIMES     ;;DO 100 ITERATIONS
                                        MOV     #165,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
                                        MOV     #340,PSW        ;SET PRIORITY TO 7
                                        MOV     $VECT1,R1       ;GET BASIC VS60 VECTOR ADRS
                                        BIC     #160000,R1      ; CLEAR PSW BITS        ** B **
                                        MOV     #3$,4(R1)       ;SET UP INTR SERVICE ADRS
                                        MOV     #340,6(R1)      ;SET UP PRIORITY TO 7 ON INT
                                        MOV     #144000,$GDDAT  ;EXPECT INT EN, L.P. FLG, L.P. INT EN
                                        MOV     #1,R0           ;R0 WHEN NON-ZERO MEANS INT EXPECTED
                                        MOV     #20040,@STKPT   ;SET MAINT SW 2
                                        MOV     #164360,BUFFER  ;SET UP TO ENAB L.P. INT & INTENSITY EN
                                1$:     MOV     #BUFFER,@DPC    ;START
                                        BIS     #40000,@STKPT   ;SET MAINT SW 3
                                        MOV     #122000,BUFFER+2        ;GRAPH X MODE
                                        INC     @DPC            ;PICK UP GRAPH X INSTR - INT WILL OCCUR IN DATA MODE
                                        MOV     #1777,BUFFER+4  ;X=1777
                                        INC     @DPC            ;RESUME FOR X
                                        MOV     CONS,$BDADR     ;SET UP REG ADRS 22
                                        BIS     #40000,@CONS    ;SET L.P. FLAG
                                        CLR     PSW             ;ALLOW INTR
                                        MOV     #40,R1          ;SET UP DELAY
                                2$:     DEC     R1              ;COUNT AWAY
                                        BNE     2$              ;STAY UNTIL DONE
                                        TST     R0              ;DO WE EXPECT INT THIS PASS - NO IF ZERO
                                        BEQ     7$              ;BR IF NOT
                                        MOV     @CONS,$BDDAT    ;READ L.P. STATUS
                                        ERROR   50              ;LIGHT PEN CONSOLE 0 INTERRUPT FAILURE
                                        BR      8$              ;GO LOOK FOR LOOP ON TEST SW
                                3$:     CMP     (R6)+,(R6)+     ;FIX STACK SINCE NO RTI
                                        MOV     @CONS,$BDDAT    ;GET CONTENTS OF REG 22
                                        TST     R0              ;DID WE EXPECT AN L.P. INTERRUPT THIS PASS?
                                        BNE     4$              ;BR IF SO
                                        ERROR   50              ;L.P. FLAG INTERRUPTED WHEN NOT ENABLED
                                        BR      8$              ;GO LOOK FOR LOOP ON TEST SW
                                4$:     CMP     $GDDAT,$BDDAT   ;SEE IF STATUS IS CORRECT AFTER INT
                                        BEQ     5$              ;BR IF SO
                                        ERROR   50              ;INCORRECT INTERRUPT STATUS
                                        BR      8$              ;GO LOOK FOR LOOP ON TEST SW
                                5$:     BIT     #200,@SREG0     ;L.P. FLG SET IN REG 02 ALSO?
                                        BNE     6$              ;BR IF SO
                                        MOV     SREG0,$BDADR    ;SET UP REG ADRS 02
                                        MOV     #200,$GDDAT     ;EXPECTED L.P. FLAG
```

```
2319  023602  005037  001126                  CLR     $BDDAT          ;SHOW IT WAS NOT SET
2320  023606  104050                          ERROR   50              ;L.P. FLAG FAILED TO SET AT REG 02
2321  023610  000420                          BR      8$              ;GO LOOK FOR LOOP ON TEST SW
2322  023612  012737  100000  001124  6$:     MOV     #100000,$GDDAT  ;SET UP EXPECTED WITH L.P. FLAG INT DISABLED
2323  023620  012737  164340  031076          MOV     #164340,BUFFER  ;SET UP INSTR WITH L.P. FLAG INT DISABLED
2324  023626  005000                          CLR     R0              ;SET TEST SWITCH TO NO INTERRUPT EXPECTED
2325  023630  000671                          BR      1$              ;GO REPEAT TEST WITH INT EN DISABLED
2326  023632  017737  156214  001126  7$:     MOV     @CONS,$BDDAT    ;READ REG 22
2327  023640  023737  001126  001124          CMP     $BDDAT,$GDDAT   ;FLAG CLR ON START?
2328  023646  001401                          BEQ     8$              ;BR IF SO
2329  023650  104034                          ERROR   34              ;START FAILED TO CLR L.P. FLAG
2330  023652  004737  024454          8$:     JSR     PC,RSTVEC       ;RESET L.P. INTERRUPT VECTOR WITH HALT
2331
2332                                          ;;*****************************************************************
 (3)                                          ;*TEST 166      TEST THAT LIGHT PEN FLAG - CONSOLE 1 WILL CAUSE AN INTERRUPT
 (3)                                          ;;*****************************************************************
 (2)  023656  000004                  TST166: SCOPE
 (1)  023660  012737  000100  001166          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
 (2)  023666  012737  000166  001206          MOV     #166,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
2333  023674  012737  000340  177776          MOV     #340,PSW        ;SET PRIORITY TO 7
2334  023702  013701  001252                  MOV     $VECT1,R1       ;GET BASIC VS60 VECTOR ADRS
2335  023706  042701  160000                  BIC     #160000,R1      ; CLEAR PSW BITS        ** B **
2336  023712  012761  024062  000004          MOV     #3$,4(R1)       ;SET UP INTR SERVICE ADRS
2337  023720  012761  000340  000006          MOV     #340,6(R1)      ;SET UP PRIORITY TO 7 ON INT
2338  023726  012737  101440  001124          MOV     #101440,$GDDAT  ;EXPECT INT EN, L.P. FLG, L.P. INT EN
2339  023734  012700  000001                  MOV     #1,R0           ;R0 WHEN NON-ZERO MEANS INT EXPECTED
2340  023740  012777  020040  156114          MOV     #20040,@STKPT   ;SET MAINT SW 2
2341  023746  012737  164760  031076          MOV     #164760,BUFFER  ;SET UP TO ENAB L.P. INT & INTENSITY EN
2342  023754  012777  031076  156046  1$:     MOV     #BUFFER,@DPC    ;START
2343  023762  052777  040000  156072          BIS     #40000,@STKPT   ;SET MAINT SW 3
2344  023770  012737  122000  031100          MOV     #122000,BUFFER+2        ;GRAPH X MODE
2345  023776  005277  156026                  INC     @DPC            ;PICK UP GRAPH X INSTR - INT WILL OCCUR IN DATA MODE
2346  024002  012737  001777  031102          MOV     #1777,BUFFER+4  ;X=1777
2347  024010  005277  156014                  INC     @DPC            ;RESUME FOR X
2348  024014  013737  002052  001122          MOV     CONS,$BDADR     ;SET UP REG ADRS 22
2349  024022  052777  000400  156022          BIS     #400,@CONS      ;SET L.P. FLAG
2350  024030  005037  177776                  CLR     PSW             ;ALLOW INTR
2351  024034  012701  000040                  MOV     #40,R1          ;SET UP DELAY
2352  024040  005301                  2$:     DEC     R1              ;COUNT AWAY
2353  024042  001376                          BNE     2$              ;STAY UNTIL DONE
2354  024044  005700                          TST     R0              ;DO WE EXPECT INT THIS PASS - NO IF ZERO
2355  024046  001433                          BEQ     6$              ;BR IF NOT
2356  024050  017737  155776  001126          MOV     @CONS,$BDDAT    ;READ L.P. STATUS
2357  024056  104050                          ERROR   50              ;LIGHT PEN CONSOLE 1 INTERRUPT FAILURE
2358  024060  000436                          BR      7$              ;GO LOOK FOR LOOP ON TEST SW
2359  024062  022626                  3$:     CMP     (R6)+,(R6)+     ;FIX STACK SINCE NO RTI
2360  024064  017737  155762  001126          MOV     @CONS,$BDDAT    ;GET L.P. STATUS
2361  024072  005700                          TST     R0              ;DID WE EXPECT AN L.P. INTERRUPT THIS PASS?
2362  024074  001002                          BNE     4$              ;BR IF SO
2363  024076  104050                          ERROR   50              ;L.P. FLAG INTERRUPTED WHEN NOT ENABLED
2364  024100  000426                          BR      7$              ;GO LOOK FOR LOOP ON TEST SW
2365  024102  023737  001124  001126  4$:     CMP     $GDDAT,$BDDAT   ;SEE IF STATUS IS CORRECT AFTER INT
2366  024110  001402                          BEQ     5$              ;BR IF SO
2367  024112  104050                          ERROR   50              ;INCORRECT L.P. STATUS
2368  024114  000420                          BR      7$              ;GO LOOK FOR LOOP ON TEST SW
2369  024116  012737  101000  001124  5$:     MOV     #101000,$GDDAT  ;SET UP TEST WITH L.P. FLAG INT DISABLED
```

```
 2370  024124  012737  164740  031076          MOV     #164740,BUFFER  ;SET UP INSTR WITH L.P. FLAG DISABLED
 2371  024132  005000                          CLR     R0              ;SET TEST SWITCH TO NO INTERRUPT EXPECTED
 2372  024134  000707                          BR      1$              ;GO REPEAT TEST WITH INT EN DISABLED
 2373  024136  017737  155710  001126  6$:     MOV     @CONS,$BDDAT    ;READ REG 22
 2374  024144  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;L.P. FLG CLR ON START?
 2375  024152  001401                          BEQ     7$              ;BR IF SO
 2376  024154  104034                          ERROR   34              ;START FAILED TO CLR L.P. FLAG
 2377  024156  004737  024454          7$:     JSR     PC,RSTVEC       ;RESET L.P. INTERRUPT VECTOR WITH HALT
 2378
 2379                                  ;;*******************************************************************
  (3)                                  ;*TEST 167       TEST THAT WORD 0,1 & 2 INDICATORS WILL SET
  (3)                                  ;;*******************************************************************
  (2)  024162  000004                  TST167: SCOPE
  (2)  024164  012737  000167  001206          MOV     #167,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2380  024172  012777  020000  155662          MOV     #BIT13,@STKPT           ;SET MAINT SWITCH #2
 2381  024200  012737  000100  001124          MOV     #100,$GDDAT     ;EXPECT WORD 0 INDICATION INITIALLY
 2382  024206  012737  110000  031076          MOV     #110000,BUFFER  ;SET UP LONG VECTOR INSTR
 2383  024214  005037  031100                  CLR     BUFFER+2                ;CLR X,Y
 2384  024220  005037  031102                  CLR     BUFFER+4
 2385  024224  012777  031076  155576          MOV     #BUFFER,@DPC    ;START
 2386  024232  000402                          BR      2$              ;GO ALLOW TIME FOR INSTR NPR
 2387  024234  005277  155570          1$:     INC     @DPC            ;PICK UP DATA WORD
 2388  024240  013737  002062  001122  2$:     MOV     STKPT,$BDADR    ;SET UP REG ADRS 32
 2389  024246  017737  155610  001126          MOV     @STKPT,$BDDAT   ;READ REG 32
 2390  024254  042737  177077  001126          BIC     #177077,$BDDAT  ;SAVE ONLY WORD INDICATORS
 2391  024262  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CORRECT?
 2392  024270  001401                          BEQ     3$              ;BR IF SO
 2393  024272  104041                          ERROR   41              ;WORD INDICATOR FAILURE
 2394  024274  006337  001124          3$:     ASL     $GDDAT          ;ADVANCE TO NEXT WORD
 2395  024300  022737  001000  001124          CMP     #1000,$GDDAT    ;HAVE ALL BEEN TESTED?
 2396  024306  001352                          BNE     1$              ;BR IF NOT
 2397
 2398                                  ;;*******************************************************************
  (3)                                  ;*TEST 170       TEST FOR HALT AT END OF DIAGNOSTIC (SW12 SET)
  (3)                                  ;;*******************************************************************
  (2)  024310  000004                  TST170: SCOPE
  (2)  024312  012737  000170  001206          MOV     #170,$TESTN     ;;SET TEST NUMBER IN APT MAIL BOX
 2399  024320  032777  010000  154612          BIT     #BIT12,@SWR     ;IS SW12 SET?
 2400  024326  001401                          BEQ     $EOP            ;BR IF NOT
 2401  024330  000000                          HALT                    ;COMPLETED PASS - SW12 SAYS HALT
```

```
2403                                    .SBTTL  END OF PASS ROUTINE
 (1)
 (2)                                    ;;**********************************************************************
 (1)                                    ;*INCREMENT THE PASS NUMBER ($PASS)
 (1)                                    ;*INDICATE END-OF-PROGRAM AFTER 1 PASSES THRU THE PROGRAM
 (1)                                    ;*TYPE 'END PASS #XXXXX'' (WHERE XXXXX IS A DECIMAL NUMBER)
 (1)                                    ;*IF THERES A MONITOR GO TO IT
 (1)                                    ;*IF THERE ISN'T JUMP TO RESTRT
 (1)
 (1)  024332                            $EOP:
 (1)  024332  000004                            SCOPE
 (1)  024334  005037  001102                    CLR     $TSTNM          ;;ZERO THE TEST NUMBER
 (1)  024340  005037  001166                    CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
 (1)  024344  005237  001210                    INC     $PASS           ;;INCREMENT THE PASS NUMBER
 (1)  024350  042737  100000  001210           BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
 (1)  024356  005327                            DEC     (PC)+           ;;LOOP?
 (1)  024360  000001                    $EOPCT: .WORD   1
 (1)  024362  003022                            BGT     $DOAGN          ;;YES
 (1)  024364  012737                            MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
 (1)  024366  000001                    $ENDCT: .WORD   1
 (1)  024370  024360                            $EOPCT
 (1)  024372  104401  024437                    TYPE    ,$ENDMG         ;;TYPE 'END PASS #''
 (2)  024376  013746  001210                    MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
 (2)  024402  104405                            TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 (1)  024404  104401  024434                    TYPE    ,$ENULL         ;;TYPE A NULL CHARACTER
 (1)  024410  013700  000042            $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
 (1)  024414  001405                            BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
 (1)  024416  000005                            RESET                   ;;CLEAR THE WORLD
 (1)  024420  004710                    $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
 (1)  024422  000240                            NOP                     ;;SAVE ROOM
 (1)  024424  000240                            NOP                     ;;FOR
 (1)  024426  000240                            NOP                     ;;ACT11
 (1)  024430                            $DOAGN:
 (1)  024430  000137                            JMP     @(PC)+          ;;RETURN
 (1)  024432  002472                    $RTNAD: .WORD   RESTRT
 (1)  024434     377     377     000    $ENULL: .BYTE   -1,-1,0         ;;NULL CHARACTER STRING
 (1)  024437     015  042412  042116    $ENDMG: .ASCIZ  <15><12>/END PASS #/
2404
```

```
2406                                    ;;*****************************************************************
2407                                            ;THIS CODE RESETS ALL VS60 VECTORS WITH HALTS
2408                                    ;;*****************************************************************
2409  024454  012737  000340  177776  RSTVEC: MOV     #340,PSW        ;RESET PRIORITY TO HIGHEST LEVEL
2410  024462  012701  000004                  MOV     #4,R1           ;SET UP VECTOR LOCATION COUNT
2411  024466  013700  001252                  MOV     $VECT1,R0       ;GET 1ST VS60 VECTOR ADRS
2412  024472  042700  160000                  BIC     #160000,R0      ; CLEAR PSW BITS        ** B **
2413  024476  010010                  1$:     MOV     R0,(R0)         ;SET UP ADRS TO HALT
2414  024500  062720  000002                  ADD     #2,(R0)+        ;POINT IT TO HALT
2415  024504  005020                          CLR     (R0)+           ;SET UP HALT
2416  024506  005301                          DEC     R1              ;COUNT THIS VECTOR RESTORE
2417  024510  001372                          BNE     1$              ;BR IF MORE TO RESTORE
2418  024512  000207                          RTS     PC              ;RETURN IF ALL DONE
2419
2420                                    ;;*****************************************************************
2421                                            ;THIS IS A GENERAL PURPOSE DELAY ROUTINE (REG CNT OF 20)
2422                                    ;;*****************************************************************
2423  024514  013700  002026          DLAY:   MOV     DELAY,R0        ;GET COUNT
2424  024520  005300                  1$:     DEC     R0              ;COUNT AWAY
2425  024522  001376                          BNE     1$              ;BR TILL DONE
2426  024524  000207                          RTS     PC              ;EXIT
2427
```

```
 2429                                   .SBTTL  SCOPE HANDLER ROUTINE
  (1)
  (2)                                   ;;*****************************************************************
  (1)                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
  (1)                                   ;*AND LOAD THE TEST NUMBER($TS NM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
  (1)                                   ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
  (1)                                   ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
  (1)                                   ;*SW14=1          LOOP ON TEST
  (1)                                   ;*SW11=1          INHIBIT ITERATIONS
  (1)                                   ;*SW09=1          LOOP ON ERROR
  (1)                                   ;*SW08-1          LOOP ON TEST IN SWR<7:0>
  (1)                                   ;*CALL
  (1)                                   ;*        SCOPE             ;;SCOPE-IOT
  (1)
  (1)  024526                           $SCOPE:
  (1)  024526  032777  040000  154404   1$:       BIT       #BIT14,@SWR       ;;LOOP ON PRESENT TEST?
  (1)  024534  001114                            BNE       $OVER             ;;YES IF SW14-1
  (1)                                   ;#####START OF CODE FOR THE XOR TESTER#####
  (1)  024536  000416                   $XTSTR: BR        6$                ;;IF RUNNING ON THE "XOR" TESTER CHANGE
  (1)                                                                       ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
  (1)  024540  013746  000004                   MOV       @#ERRVEC,-(SP)    ;;SAVE THE CONTENTS OF THE ERROR VECTOR
  (1)  024544  012737  024564  000004          MOV       #5$,@#ERRVEC      ;;SET FOR TIMEOUT
  (1)  024552  005737  177060                   TST       @#177060          ;;TIME OUT ON XOR?
  (1)  024556  012637  000004                   MOV       (SP)+,@#ERRVEC    ;;RESTORE THE ERROR VECTOR
  (1)  024562  000463                           BR        $SVLAD            ;;GO TO THE NEXT TEST
  (1)  024564  022626                   5$:     CMP       (SP)+,(SP)+       ;;CLEAR THE STACK AFTER A TIME OUT
  (1)  024566  012637  000004                   MOV       (SP)+,@#ERRVEC    ;;RESTORE THE ERROR VECTOR
  (1)  024572  000423                           BR        7$                ;;LOOP ON THE PRESENT TEST
  (1)  024574                           6$:;#####END OF CODE FOR THE XOR TESTER#####
  (1)  024574  032777  000400  154336           BIT       #BIT08,@SWR       ;;LOOP ON SPEC. TEST?
  (1)  024602  001404                           BEQ       2$                ;;BR IF NO
  (1)  024604  127737  154330  001102          CMPB      @SWR,$TSTNM       ;;ON THE RIGHT TEST?    SWR<7:0>
  (1)  024612  001465                           BEQ       $OVER             ;;BR IF YES
  (1)  024614  105737  001103          2$:     TSTB      $ERFLG            ;;HAS AN ERROR OCCURRED?
  (1)  024620  001421                           BEQ       3$                ;;BR IF NO
  (1)  024622  123737  001115  001103          CMPB      $ERMAX,$ERFLG     ;;MAX. ERRORS FOR THIS TEST OCCURRED?
  (1)  024630  101015                           BHI       3$                ;;BR IF NO
  (1)  024632  032777  001000  154300          BIT       #BIT09,@SWR       ;;LOOP ON ERROR?
  (1)  024640  001404                           BEQ       4$                ;;BR IF NO
  (1)  024642  013737  001110  001106  7$:     MOV       $LPERR,$LPADR     ;;SET LOOP ADDRESS TO LAST SCOPE
  (1)  024650  000446                           BR        $OVER
  (1)  024652  105037  001103          4$:     CLRB      $ERFLG            ;;ZERO THE ERROR FLAG
  (1)  024656  005037  001166                   CLR       $TIMES            ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
  (1)  024662  000415                           BR        1$                ;;ESCAPE TO THE NEXT TEST
  (1)  024664  032777  004000  154246  3$:     BIT       #BIT11,@SWR       ;;INHIBIT ITERATIONS?
  (1)  024672  001011                           BNE       1$                ;;BR IF YES
  (1)  024674  005737  001210                   TST       $PASS             ;;IF FIRST PASS OF PROGRAM
  (1)  024700  001406                           BEQ       1$                ;;        INHIBIT ITERATIONS
  (1)  024702  005237  001104                   INC       $ICNT             ;;INCREMENT ITERATION COUNT
  (1)  024706  023737  001166  001104          CMP       $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
  (1)  024714  002024                           BGE       $OVER             ;;BR IF MORE ITERATION REQUIRED
  (1)  024716  012737  000001  001104  1$:     MOV       #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
  (1)  024724  013737  025002  001166          MOV       $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
  (1)  024732  105237  001102          $SVLAD: INCB      $TSTNM            ;;COUNT TEST NUMBERS
  (1)  024736  113737  001102  001206          MOVB      $TSTNM,$TESTN     ;;SET TEST NUMBER IN APT MAILBOX
  (1)  024744  011637  001106                   MOV       (SP),$LPADR       ;;SAVE SCOPE LOOP ADDRESS
```

```
  (1)  024750  011637  001110                MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
  (1)  024754  005037  001170                CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
  (1)  024760  112737  000001  001115        MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
  (1)  024766  013777  001102  154146 $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
  (1)  024774  013716  001106                MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
  (1)  025000  000002                        RTI                     ;;FIXES PS
  (1)  025002  003720                $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
 2430                                .SBTTL  APT COMMUNICATIONS ROUTINE
  (1)
  (2)                                ;;**************************************************************************
  (1)  025004  112737  000001  025250 $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
  (1)  025012  112737  000001  025246 $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
  (1)  025020  000403                        BR      $ATYC
  (1)  025022  112737  000001  025250 $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
  (1)  025030                        $ATYC:
  (3)  025030  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
  (3)  025032  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
  (1)  025034  105737  025246                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
  (1)  025040  001450                        BEQ     5$              ;;IF NOT:  BR
  (1)  025042  122737  000001  001222        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
  (1)  025050  001031                        BNE     3$              ;;IF NOT:  BR
  (1)  025052  132737  000100  001223        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
  (1)  025060  001425                        BEQ     3$              ;;IF NOT:  BR
  (1)  025062  017600  000004                MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
  (1)  025066  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
  (1)  025074  005737  001202        1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
  (1)  025100  001375                        BNE     1$              ;;IF NOT:  WAIT
  (1)  025102  010037  001216                MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
  (1)  025106  105720                2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
  (1)  025110  001376                        BNE     2$
  (1)  025112  163700  001216                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
  (1)  025116  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
  (1)  025120  010037  001220                MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
  (1)  025124  012737  000004  001202        MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
  (1)  025132  000413                        BR      5$
  (1)  025134  017637  000004  025160 3$:    MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
  (1)  025142  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
  (3)  025150  013746  177776                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
  (1)  025154  004737  026600                JSR     PC,$TYPE        ;;CALL TYPE MACRO
  (1)  025160  000000                4$:     .WORD   0
  (1)  025162                        5$:
  (1)  025162  105737  025250        10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
  (1)  025166  001416                        BEQ     12$             ;;IF NOT:  BR
  (1)  025170  005737  001222                TST     $ENV            ;;RUNNING UNDER APT?
  (1)  025174  001413                        BEQ     12$             ;;IF NOT:  BR
  (1)  025176  005737  001202        11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
  (1)  025202  001375                        BNE     11$             ;;IF NOT:  WAIT
  (1)  025204  017637  000004  001204        MOV     @4(SP),$FATAL   ;;GET ERROR #
  (1)  025212  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
  (1)  025220  005237  001202                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
  (1)  025224  105037  025250        12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
  (1)  025230  105037  025247                CLRB    $LFLG           ;;CLEAR LOG FLAG
  (1)  025234  105037  025246                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
  (3)  025240  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
  (3)  025242  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
  (1)  025244  000207                        RTS     PC              ;;RETURN
```

```
(1)  025246  000                  $MFLG:  .BYTE   0               ;:MESSG. FLAG
(1)  025247  000                  $LFLG:  .BYTE   0               ;:LOG FLAG
(1)  025250  000                  $FFLG:  .BYTE   0               ;:FATAL FLAG
(1)          025252                       .EVEN
(1)          000200               APTSIZE=200
(1)          000001               APTENV=001
(1)          000100               APTSPOOL=100
(1)          000040               APTCSUP=040
2431                              .SBTTL  ERROR HANDLER ROUTINE
(1)
(2)                               ;;**********************************************************************
(1)                               ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
(1)                               ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
(1)                               ;*AND GO TO $ERRTYP ON ERROR
(1)                               ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)                               ;*SW15=1        HALT ON ERROR
(1)                               ;*SW13=1        INHIBIT ERROR TYPEOUTS
(1)                               ;*SW10=1        BELL ON ERROR
(1)                               ;*SW09=1        LOOP ON ERROR
(1)                               ;*CALL
(1)                               ;*      ERROR   N       ;;ERROR-EMT AND N ERROR ITEM NUMBER
(1)
(1)  025252                       $ERROR:
(2)  025252  113737  001102  002024          MOVB    $TSTNM,TSTNUM
(1)  025260  105237  001103          7$:     INCB    $ERFLG          ;:SET THE ERROR FLAG
(1)  025264  001775                          BEQ     7$              ;:DON'T LET THE FLAG GO TO ZERO
(1)  025266  013777  001102  153646          MOV     $TSTNM,@DISPLAY ;:DISPLAY TEST NUMBER AND ERROR FLAG
(1)  025274  032777  002000  153636          BIT     #BIT10,@SWR     ;:BELL ON ERROR?
(1)  025302  001402                          BEQ     1$              ;:NO - SKIP
(1)  025304  104401  001172                  TYPE    ,$BELL          ;:RING BELL
(1)  025310  005237  001112          1$:     INC     $ERTTL          ;:COUNT THE NUMBER OF ERRORS
(1)  025314  011637  001116                  MOV     (SP),$ERRPC     ;:GET ADDRESS OF ERROR INSTRUCTION
(1)  025320  162737  000002  001116          SUB     #2,$ERRPC
(1)  025326  117737  153564  001114          MOVB    @$ERRPC,$ITEMB  ;:STRIP AND SAVE THE ERROR ITEM CODE
(1)  025334  032777  020000  153576          BIT     #BIT13,@SWR     ;:SKIP TYPEOUT IF SET
(1)  025342  001004                          BNE     20$             ;:SKIP TYPEOUTS
(1)  025344  004737  025454                  JSR     PC,$ERRTYP      ;:GO TO USER ERROR ROUTINE
(1)  025350  104401  001177                  TYPE    ,$CRLF
(1)  025354                       20$:
(1)  025354  122737  000001  001222          CMPB    #APTENV,$ENV    ;:RUNNING IN APT MODE
(1)  025362  001007                          BNE     2$              ;:NO,SKIP APT ERROR REPORT
(1)  025364  113737  001114  025376          MOVB    $ITEMB,21$      ;:SET ITEM NUMBER AS ERROR NUMBER
(1)  025372  004737  025022                  JSR     PC,$ATY4        ;:REPORT FATAL ERROR TO APT
(1)  025376     000               21$:     .BYTE   0
(1)  025377     000                        .BYTE   0
(1)  025400  000777               22$:     BR      22$             ;:APT ERROR LOOP
(1)  025402  005777  153532          2$:     TST     @SWR            ;:HALT ON ERROR
(1)  025406  100001                          BPL     3$              ;:SKIP IF CONTINUE
(1)  025410  000000                          HALT                    ;:HALT ON ERROR!
(1)  025412  032777  001000  153520  3$:     BIT     #BIT09,@SWR     ;:LOOP ON ERROR SWITCH SET?
(1)  025420  001402                          BEQ     4$              ;:BR IF NO
(1)  025422  013716  001110                  MOV     $LPERR,(SP)     ;:FUDGE RETURN FOR LOOPING
(1)  025426  005737  001170          4$:     TST     $ESCAPE         ;:CHECK FOR AN ESCAPE ADDRESS
(1)  025432  001402                          BEQ     5$              ;:BR IF NONE
(1)  025434  013716  001170                  MOV     $ESCAPE,(SP)    ;:FUDGE RETURN ADDRESS FOR ESCAPE
(1)  025440                       5$:
```

```
(1)  025440  022737  024420  000042          CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
(1)  025446  001001                          BNE     6$              ;;BRANCH IF NO
(1)  025450  000000                          HALT                    ;;YES
(1)  025452                          6$:
(1)  025452  000002                          RTI                     ;;RETURN
```

```
2433
2434                                    .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 (1)
 (2)                                    ;;**********************  ************************************************
 (1)                                    ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 (1)                                    ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
 (1)                                    ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 (1)
 (1)  025454                            $ERRTYP:
 (1)  025454  104401  001177                    TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 (1)  025460  010046                            MOV     R0,-(SP)        ;;SAVE R0
 (1)  025462  005000                            CLR     R0              ;;PICKUP THE ITEM INDEX
 (1)  025464  153700  001114                    BISB    @#$ITEMB,R0
 (1)  025470  001004                            BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
 (1)                                                                    ;;TYPE THE PC OF THE ERROR
 (2)  025472  013746  001116                    MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
 (2)                                                                    ;;ERROR ADDRESS
 (2)  025476  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 (1)  025500  000426                            BR      6$              ;;GET OUT
 (1)  025502  005300                    1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
 (1)  025504  006300                            ASL     R0              ;;      WORK FOR THE ERROR TABLE
 (1)  025506  006300                            ASL     R0
 (1)  025510  006300                            ASL     R0
 (1)  025512  062700  001270                    ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
 (1)  025516  012037  025526                    MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
 (1)  025522  001404                            BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
 (1)  025524  104401                            TYPE                    ;;TYPE THE "ERROR MESSAGE"
 (1)  025526  000000                    2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
 (1)  025530  104401  001177                    TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 (1)  025534  012037  025544            3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
 (1)  025540  001404                            BEQ     5$              ;;SKIP TYPEOUT IF 0
 (1)  025542  104401                            TYPE                    ;;TYPE THE "DATA HEADER"
 (1)  025544  000000                    4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
 (1)  025546  104401  001177                    TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 (1)  025552  011000                    5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
 (1)  025554  001004                            BNE     7$              ;;GO TYPE THE DATA
 (1)  025556  012600                    6$:     MOV     (SP)+,R0        ;;RESTORE R0
 (1)  025560  104401  001177                    TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
 (1)  025564  000207                            RTS     PC              ;;RETURN
 (1)  025566                            7$:
 (2)  025566  013046                            MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
 (2)  025570  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 (1)  025572  005710                            TST     (R0)            ;;IS THERE ANOTHER NUMBER?
 (1)  025574  001770                            BEQ     6$              ;;BR IF NO
 (1)  025576  104401  025604                    TYPE    ,8$             ;;TYPE TWO(2) SPACES
 (1)  025602  000771                            BR      7$              ;;LOOP
 (1)  025604  020040      000          8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
 (1)          025610                            .EVEN
```

```
2436
2437                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 (1)
 (2)                                    ;;**********************************************************************
 (1)                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 (1)                                    ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 (1)                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
 (1)                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 (1)                                    ;*      .BYTE   M               ;;M=1 OR 0
 (1)                                    ;*                                      ;;1=TYPE LEADING ZEROS
 (1)                                    ;*                                      ;;0-SUPPRESS LEADING ZEROS
 (1)                                    ;*
 (1)                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 (1)                                    ;*$TYPOS OR $TYPOC
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
 (1)                                    ;*
 (1)                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 (1)                                    ;*CALL:
 (1)                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
 (1)
 (1)  025610  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 (1)  025614  116637  000001  026033            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 (1)  025622  112637  026035                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 (1)  025626  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 (1)  025632  000406                            BR      $TYPON
 (1)  025634  112737  000001  026033    $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 (1)  025642  112737  000006  026035            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 (1)  025650  112737  000005  026032    $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 (1)  025656  010346                            MOV     R3,-(SP)        ;;SAVE R3
 (1)  025660  010446                            MOV     R4,-(SP)        ;;SAVE R4
 (1)  025662  010546                            MOV     R5,-(SP)        ;;SAVE R5
 (1)  025664  113704  026035                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 (1)  025670  005404                            NEG     R4
 (1)  025672  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 (1)  025676  110437  026034                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 (1)  025702  113704  026033                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 (1)  025706  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 (1)  025712  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
 (1)  025714  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 (1)  025716  000404                            BR      3$              ;;GO DO MSB
 (1)  025720  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
 (1)  025722  006105                            ROL     R5
 (1)  025724  006105                            ROL     R5
 (1)  025726  010503                            MOV     R5,R3
 (1)  025730  006103                    3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 (1)  025732  105337  026034                    DECB    $OMODE          ;;TYPE THIS DIGIT?
 (1)  025736  100016                            BPL     7$              ;;BR IF NO
 (1)  025740  042703  177770                    BIC     #177770,R3      ;;GET RID OF JUNK
 (1)  025744  001002                            BNE     4$              ;;TEST FOR 0
 (1)  025746  005704                            TST     R4              ;;SUPPRESS THIS 0?
```

```
(1)  025750  001403                        BEQ     5$              ;;BR IF YES
(1)  025752  005204                4$:      INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
(1)  025754  052703  000060                BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
(1)  025760  052703  000040                BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
(1)  025764  110337  026030                MOVB    R3,8$           ;;SAVE FOR TYPING
(1)  025770  104401  026030                TYPE    ,8$             ;;GO TYPE THIS DIGIT
(1)  025774  105337  026032                DECB    $OCNT           ;;COUNT BY 1
(1)  026000  003347                        BGT     2$              ;;BR IF MORE TO DO
(1)  026002  002402                        BLT     6$              ;;BR IF DONE
(1)  026004  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
(1)  026006  000744                        BR      2$              ;;GO DO THE LAST DIGIT
(1)  026010  012605                6$:     MOV     (SP)+,R5        ;;RESTOPE R5
(1)  026012  012604                        MOV     (SP)+,R4        ;;RESTORE R4
(1)  026014  012603                        MOV     (SP)+,R3        ;;RESTORE R3
(1)  026016  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
(1)  026024  012616                        MOV     (SP)+,(SP)
(1)  026026  000002                        RTI                     ;;RETURN
(1)  026030     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
(1)  026031     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
(1)  026032     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
(1)  026033     000                $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
(1)  026034  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
2438                               .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
(1)
(2)                                ;;**************************************************************************
(1)                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
(1)                                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
(1)                                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
(1)                                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
(1)                                ;*REPLACED WITH SPACES.
(1)                                ;*CALL:
(1)                                ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
(1)                                ;*      TYPDS                   ;;GO TO THE ROUTINE
(1)
(1)  026036                        $TYPDS:
(3)  026036  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  026040  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(3)  026042  010246                        MOV     R2,-(SP)        ;;PUSH R2 ON STACK
(3)  026044  010346                        MOV     R3,-(SP)        ;;PUSH R3 ON STACK
(3)  026046  010546                        MOV     R5,-(SP)        ;;PUSH R5 ON STACK
(1)  026050  012746  020200                MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
(1)  026054  016605  000020                MOV     20(SP),R5       ;;GET THE INPUT NUMBER
(1)  026060  100004                        BPL     1$              ;;BR IF INPUT IS POS.
(1)  026062  005405                        NEG     R5              ;;MAKE THE BINARY NUMBER POS.
(1)  026064  112766  000055  000001        MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
(1)  026072  005000                1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
(1)  026074  012703  026252                MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
(1)  026100  112723  000040                MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
(1)  026104  005002                2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
(1)  026106  016001  026242                MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
(1)  026112  160105                3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
(1)  026114  002402                        BLT     4$              ;;BR IF DONE
(1)  026116  005202                        INC     R2              ;;INCREASE THE BCD DIGIT BY 1
(1)  026120  000774                        BR      3$
(1)  026122  060105                4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
(1)  026124  005702                        TST     R2              ;;CHECK IF BCD DIGIT=0
```

```
(1)  026126  001002                        BNE     5$              ;;FALL THROUGH IF 0
(1)  026130  105716                        TSTB    (SP)            ;;STILL DOING LEADING 0'S?
(1)  026132  100407                        BMI     7$              ;;BR IF YES
(1)  026134  106316                5$:     ASLB    (SP)            ;;MSD?
(1)  026136  103003                        BCC     6$              ;;BR IF NO
(1)  026140  116663  000001 177777        MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
(1)  026146  052702  000060        6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
(1)  026152  052702  000040        7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1)  026156  110223                        MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1)  026160  005720                        TST     (R0)+           ;;JUST INCREMENTING
(1)  026162  020027  000010                CMP     R0,#10          ;;CHECK THE TABLE INDEX
(1)  026166  002746                        BLT     2$              ;;GO DO THE NEXT DIGIT
(1)  026170  003002                        BGT     8$              ;;GO TO EXIT
(1)  026172  010502                        MOV     R5,R2           ;;GET THE LSD
(1)  026174  000764                        BR      6$              ;;GO CHANGE TO ASCII
(1)  026176  105726                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
(1)  026200  100003                        BPL     9$              ;;BR IF NO
(1)  026202  116663  177777 177776        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
(1)  026210  105013                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
(3)  026212  012605                        MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  026214  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  026216  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  026220  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  026222  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  026224  104401  026252                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  026230  016666  000002 000004         MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  026236  012616                        MOV     (SP)+,(SP)
(1)  026240  000002                        RTI                     ;;RETURN TO USER
(1)  026242  023420                $DTBL:  10000.
(1)  026244  001750                        1000.
(1)  026246  000144                        100.
(1)  026250  000012                        10.
(1)  026252  000004                $DBLK:  .BLKW   4
```

```
2440
2441                                    .SBTTL  POWER DOWN AND UP ROUTINES
 (1)
 (2)                                    ;;**********************************************************************
 (1)                                    ;POWER DOWN ROUTINE
 (1)  026262  012737  026426  000024    $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 (1)  026270  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
 (3)  026276  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 (3)  026300  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 (3)  026302  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 (3)  026304  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 (3)  026306  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 (3)  026310  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 (3)  026312  017746  152622                    MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
 (1)  026316  010637  026432                    MOV     SP,$SAVR6       ;;SAVE SP
 (1)  026322  012737  026334  000024            MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 (1)  026330  000000                            HALT
 (1)  026332  000776                            BR      .-2             ;;HANG UP
 (1)
 (2)                                    ;;**********************************************************************
 (1)                                    ;POWER UP ROUTINE
 (1)  026334  012737  026426  000024    $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 (1)  026342  013706  026432                    MOV     $SAVR6,SP       ;;GET SP
 (1)  026346  005037  026432                    CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 (1)  026352  005237  026432            1$:     INC     $SAVR6          ;;WAIT FOR THE INC
 (1)  026356  001375                            BNE     1$              ;;OF  WORD
 (3)  026360  012677  152554                    MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
 (3)  026364  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
 (3)  026366  012604                            MOV     (SP)+,R4        ;;POP STACK INTO R4
 (3)  026370  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
 (3)  026372  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
 (3)  026374  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 (3)  026376  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
 (1)  026400  012737  026202  000024            MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 (1)  026406  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
 (1)  026414  104401                            TYPE                    ;;REPORT THE POWER FAILURE
 (1)  026416  026434                    $PWRMG: .WORD   PWRMSG          ;;POWER FAIL MESSAGE POINTER
 (1)  026420  012716                            MOV     (PC)+,(SP)      ;;RESTART AT BEGIN
 (1)  026422  002110                    $PWRAD: .WORD   BEGIN           ;;RESTART ADDRESS
 (1)  026424  000002                            RTI
 (1)  026426  000000                    $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
 (1)  026430  000776                            BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 (1)  026432  000000                    $SAVR6: 0                       ;;PUT THE SP HERE
2442  026434  005015  042522  052123    PWRMSG: .ASCIZ  <15><12>/RESTARTING AFTER A POWER FAILURE/<15><12>
2443        026502                              .EVEN
```

```
 2445                                   .SBTTL  ROUTINE TO SIZE MEMORY
  (1)
  (2)                                   ;;*********************************************************************
  (1)                                   ;*CALL:
  (1)                                   ;*      JSR     PC,$SIZE
  (1)                                   ;*      RETURN
  (1)                                   ;*$LSTAD WILL CONTAIN THE LAST AVAILABLE MEMORY LOCATION
  (1)
  (1)  026502  010046                   $SIZE:  MOV     R0,-(SP)        ;;SAVE R0 ON THE STACK
  (1)  026504  010146                           MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
  (1)  026506  013746  000004                   MOV     @#ERRVEC,-(SP)  ;;SAVE PRESENT ERROR VECTOR PS & PC
  (1)  026512  013746  000006                   MOV     @#ERRVEC+2,-(SP)
  (1)  026516  010600                           MOV     SP,R0           ;;SAVE THE STACK POINTER
  (1)                                   ;;SET THE ERRVEC PS TO THE PRESENT PS
  (2)  026520  104400                           TRAP                    ;;PUSH OLD PSW AND PC ON STACK
  (2)  026522  012637  000006                   MOV     (SP)+,@#ERRVEC+2        ;;SAVE THE PSW IN @#ERRVEC+2
  (1)  026526  012737  026546  000004           MOV     #2$,@#ERRVEC    ;;SET FOR TIMEOUT
  (1)  026534  012701  020000                   MOV     #20000,R1       ;;FIRST ADDRESS
  (1)  026540  005711                   1$:     TST     (R1)            ;;TEST THIS ADDRESS
  (1)  026542  005721                           TST     (R1)+           ;;STEP TO NEXT ADDRESS
  (1)  026544  000775                           BR      1$              ;;TRY ANOTHER
  (1)  026546  162701  000002           2$:     SUB     #2,R1           ;;DROP BACK
  (1)  026552  010006                           MOV     R0,SP           ;;RESTORE THE STACK
  (1)  026554  012637  000006                   MOV     (SP)+,@#ERRVEC+2 ;;RESTORE ERROR VECTOR
  (1)  026560  012637  000004                   MOV     (SP)+,@#ERRVEC
  (1)  026564  010137  026576                   MOV     R1,$LSTAD       ;;LAST ADDRESS
  (1)  026570  012601                           MOV     (SP)+,R1        ;;RESTORE R1
  (1)  026572  012600                           MOV     (SP)+,R0        ;;RESTORE R0
  (1)  026576  000207                           RTS     PC
  (1)  026576  000000                   $LSTAD: .WORD   0               ;;CONTAINS THE LAST ADDRESS
```

```
 2447
 2448                                   .SBTTL  TYPE ROUTINE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
  (1)                                   ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
  (1)                                   ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
  (1)                                   ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
  (1)                                   ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
  (1)                                   ;*
  (1)                                   ;*CALL:
  (1)                                   ;*1) USING A TRAP INSTRUCTION
  (1)                                   ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
  (1)                                   ;*OR
  (1)                                   ;*      TYPE
  (1)                                   ;*      MESADR
  (1)                                   ;*
  (1)
  (1)  026600  105737  001157           $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
  (1)  026604  100002                           BPL     1$              ;;BR IF YES
  (1)  026606  000000                           HALT                    ;;HALT HERE IF NO TERMINAL
  (1)  026610  000430                           BR      3$              ;;LEAVE
  (1)  026612  010046                   1$:     MOV     R0,-(SP)        ;;SAVE R0
  (1)  026614  017600  000002                   MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
  (1)  026620  122737  000001  001222           CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
  (1)  026626  001011                           BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
  (1)  026630  132737  000100  001223           BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
  (1)  026636  001405                           BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
  (1)  026640  010037  026650                   MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
  (1)  026644  004737  025012                   JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
  (1)  026650  000000                   61$:    .WORD   0               ;;MESSAGE ADDRESS
  (1)  026652  132737  000040  001223   62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
  (1)  026660  001003                           BNE     60$             ;;YES,SKIP TYPE OUT
  (1)  026662  112046                   2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
  (1)  026664  001005                           BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
  (1)  026666  005726                           TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
  (1)  026670  012600                   60$:    MOV     (SP)+,R0        ;;RESTORE R0
  (1)  026672  062716  000002           3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
  (1)  026676  000002                           RTI                     ;;RETURN
  (1)  026700  122716  000011           4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
  (1)  026704  001430                           BEQ     8$
  (1)  026706  122716  000200                   CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
  (1)  026712  001006                           BNE     5$
  (1)  026714  005726                           TST     (SP)+           ;;POP  <CR><LF> EQUIV
  (1)  026716  104401                           TYPE                    ;;TYPE A CR AND LF
  (1)  026720  001177                           $CRLF
  (1)  026722  105037  027056                   CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
  (1)  026726  000755                           BR      2$              ;;GET NEXT CHARACTER
  (1)  026730  004737  027012           5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
  (1)  026734  123726  001156           6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
  (1)  026740  001350                           BNE     2$              ;;IF NO GO GET NEXT CHAR.
  (1)  026742  013746  001154                   MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
  (1)                                           ;;AND THE NULL CHAR.
  (1)  026746  105366  000001           7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
  (1)  026752  002770                           BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
  (1)  026754  004737  027012                   JSR     PC,$TYPEC       ;;GO TYPE A NULL
```

```
   (1)  026760  105337  027056              DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
   (1)  026764  000770                      BR      7$              ;;LOOP
   (1)
   (1)                                  ;HORIZONTAL TAB PROCESSOR
   (1)
   (1)  026766  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
   (1)  026772  004737  027012          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
   (1)  026776  132737  000007  027056          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
   (1)  027004  001372                          BNE     9$              ;;TAB STOP
   (1)  027006  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
   (1)  027010  000724                          BR      2$              ;;GET NEXT CHARACTER
   (1)  027012  105777  152132          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
   (1)  027016  100375                          BPL     $TYPEC
   (1)  027020  116677  000002  152124          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
   (1)  027026  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
   (1)  027034  001003                          BNE     1$              ;;BRANCH IF NO
   (1)  027036  105037  027056                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
   (1)  027042  000406                          BR      $TYPEX          ;;EXIT
   (1)  027044  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
   (1)  027052  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
   (1)  027054  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
   (1)  027056  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
   (1)  027060  000207          $TYPEX: RTS     PC
   (1)
  2449
```

```
2451                                    .SBTTL  TRAP DECODER
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
 (1)                                    ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 (1)                                    ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 (1)                                    ;*GO TO THAT ROUTINE.
 (1)
 (1)  027062  010046                    $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
 (1)  027064  016600  000002                    MOV     2(SP),R0        ;;GET TRAP ADDRESS
 (1)  027070  005740                            TST     -(R0)           ;;BACKUP BY 2
 (1)  027072  111000                            MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 (1)  027074  006300                            ASL     R0              ;;POSITION FOR INDEXING
 (1)  027076  016000  027116                    MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 (1)  027102  000200                            RTS     R0              ;;GO TO ROUTINE
 (1)
 (1)
 (1)                                    ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO
 (1)
 (1)  027104  011646                    $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 (1)  027106  016666  000004  000002            MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 (1)  027114  000002                            RTI                     ;;RESTORE THE PSW
 (1)
 (3)                                    .SBTTL  TRAP TABLE
 (3)
 (3)                                    ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
 (3)                                    ;*BY THE ''TRAP'' INSTRUCTION.
 (3)
 (3)                                    ;       ROUTINE
 (3)                                    ;       -------
 (3)  027116  027104                    $TRPAD: .WORD   $TRAP2
 (3)  027120  026600                            $TYPE   ;;CALL-TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 (3)  027122  025634                            $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 (3)  027124  025610                            $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 (3)  027126  025650                            $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 (3)  027130  026036                            $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 (1)
 (1)
2452
```

```
2454                                                    .SBTTL  ASCII MESSAGES
2455  027132  005015  051412  040524  HEADER: .ASCIZ  <15><12><12>/START OF CZVSA-B  VS60 INSTRUCTION TEST PART 1/<15><12>
2456
2457  027216  040515  047111  027124  EM1:    .ASCIZ  /MAINT. SWITCH REG/
2458  027240  047514  042101  047111  EM2:    .ASCIZ  /LOADING D.P.C. REG <STATIC>/
2459  027274  047514  042101  047111  EM3:    .ASCIZ  /LOADING MODE REG/
2460  027315     123  050505  042525  EM4:    .ASCIZ  /SEQUENCING DPC REG/
2461  027340  047514  042101  047111  EM5:    .ASCIZ  /LOADING LINE TYPE REG/
2462  027366  047514  042101  047111  EM6:    .ASCIZ  /LOADING INTENSITY LEVEL REG/
2463  027422  047514  042101  047111  EM7:    .ASCIZ  /LOADING GRAPHPLOT INC REG/
2464  027454  044504  050123  040514  EM10:   .ASCIZ  /DISPLAY JUMP ABS TO AN ADRS/
2465  027510  044504  050123  040514  EM11:   .ASCIZ  /DISPLAY JUMP REL TO AN ADRS/
2466  027544  044504  050123  040514  EM12:   .ASCIZ  /DISPLAY JSR ABS TO AN ADRS/
2467  027577     104  051511  046120  EM13:   .ASCIZ  /DISPLAY JSR REL TO AN ADRS/
2468  027632  047514  042101  047111  EM14:   .ASCIZ  /LOADING X POS REG/
2469  027654  047514  042101  047111  EM15:   .ASCIZ  /LOADING Y POS REG/
2470  027676  052502  020123  044524  EM16:   .ASCIZ  /BUS TIME-OUT ER/
2471  027716  044103  051101  052040  EM17:   .ASCIZ  /CHAR TERMINATE ER/
2472  027740  020131  054504  040516  EM20:   .ASCIZ  /Y DYNAMIC OFFSET ER/
2473  027764  020130  054504  040516  EM21:   .ASCIZ  /X DYNAMIC OFFSET ER/
2474  030010  042522  047514  040503  EM22:   .ASCIZ  /RELOCATE REG ER/
2475  030030  027104  027120  027125  EM23:   .ASCIZ  /D.P.U. NAME REG ER/
2476  030053     123  040505  041522  EM24:   .ASCIZ  /SEARCH CODE OF ASSOC. NAME REG ER/
2477  030115     126  041505  047524  EM25:   .ASCIZ  /VECTOR SCALE ER/
2478  030135     103  040510  020122  EM26:   .ASCIZ  /CHAR SCALE ER/
2479  030153     102  044514  045516  EM27:   .ASCIZ  /BLINK LOGIC ER/
2480  030172  052111  046101  041511  EM30:   .ASCIZ  /ITALIC LOGIC ER/
2481  030212  042515  052516  046040  EM31:   .ASCIZ  /MENU LOGIC ER/
2482  030230  042522  042523  020124  EM32:   .ASCIZ  /RESET FAILED TO CLEAR A REG/
2483  030264  052123  050117  044440  EM33:   .ASCIZ  /STOP INTR ER/
2484  030301     114  050056  020056  EM34:   .ASCIZ  /L.P. FLAG ER/
2485  030316  047503  047514  020122  EM35:   .ASCIZ  /COLOR LEVEL ER/
2486  030335     111  042524  051516  EM36:   .ASCIZ  /ITENSITY ENABLE - CONS 0 + 1 ER/
2487  030375     111  052116  051105  EM37:   .ASCIZ  /INTERNAL STOP FLAG FAILED TO SET/
2488  030436  047111  042524  047122  EM40:   .ASCIZ  /INTERNAL STOP FLAG FAILED TO CLEAR/
2489  030501     127  051117  020104  EM41:   .ASCIZ  /WORD 0-1-2 INDICATOR ER/
2490  030531     114  050056  020056  EM42:   .ASCIZ  /L.P. INTR ENABLE - CONS 0 + 1 ER/
2491  030572  027114  027120  051440  EM43:   .ASCIZ  /L.P. SWITCH INTR ENABLE - CONS 0 + 1 ER/
2492  030642  044504  050123  040514  EM44:   .ASCIZ  /DISPLAY BUSY ER/
2493  030662  054105  042524  047122  EM45:   .ASCIZ  /EXTERNAL STOP LOGIC ER/
2494  030711     124  046511  026505  EM46:   .ASCIZ  /TIME-OUT INTR ER/
2495  030732  040516  042515  046440  EM47:   .ASCIZ  /NAME MATCH INTR ER/
2496  030755     114  050056  020056  EM50:   .ASCIZ  /L.P. FLAG INTR ER/
2497  030777     123  040524  045503  EM51:   .ASCIZ  /STACK PTR ER/
2498  031014  051105  050122  020103  DH1:    .ASCIZ  /ERRPC   TSTNUM  BUSADR  EXPCT   RCVD/
2499          031062                          .EVEN
2500  031062  001116  002024  001122  DT1:    $ERRPC,TSTNUM,$BDADR,$GDDAT,$BDDAT,0
2501
2502                                  ;;*****************************************************************
2503                                          ;THIS IS THE WORKING AREA FOR ALL VS60 NPR'S (INSTRS & DATA)
2504                                          ;FROM HERE TO THE END OF 28K
2505                                  ;;*****************************************************************
2506  031076  000000                  BUFFER: 0
2507
2508          000001                          .END
```

```
ABASE = 172000              19#     32
ACDW1 = 000000              32
ACDW2 = 000000              32
ACPUOP= 000000              32
ADDW0 = 000000              32
ADDW1 = 000000              32
ADDW10= 000000              32
ADDW11= 000000              32
ADDW12= 000000              32
ADDW13= 000000              32
ADDW14= 000000              32
ADDW15- 000000              32
ADDW2 = 000000              32
ADDW3 = 000000              32
ADDW4 = 000000              32
ADDW5 = 000000              32
ADDW6 = 000000              32
ADDW7   000000              32
ADDW8 = 000000              32
ADDW9 - 000000              32
ADEVCT= 000000              32
ADEVM = 000000              32
AENV  = 000000              32
AENVM - 000000              32
AFATAL- 000000              32
AMADR1- 000000              32
AMADR2= 000000              32
AMADR3= 000000              32
AMADR4= 000000              32
AMAMS1= 000000              32
AMAMS2= 000000              32
AMAMS3= 000000              32
AMAMS4= 000000              32
AMSGAD= 000000              32
AMSGLG= 000000              32
AMSGTY= 000000              32
AMTYP1= 000000              32
AMTYP2= 000000              32
AMTYP3= 000000              32
AMTYP4- 000000              32
ANAME   002050            364#    892*    905*    906*    916*    928*    958*    987*   1014*   1028*   1029*   1048*
APASS = 000000              32
APRIOR= 000000              32
APTCSU= 000040            2430#   2448
APTENV= 000001            2430#   2431    2448
APTSIZ= 000200             388    2430#
APTSPO= 000100            2430#   2448
ASWREG= 000000              32
ATESTN= 000000              32
AUNIT = 000000              32
AUSWR = 000000              32
AVECT1= 100320              20#     32
AVECT2= 000000              32
BEGIN   002110              24     388#    400     2441
BIT0  = 000001              17#
BIT00 - 000001              17#
```

```
BIT01 = 000002            17#
BIT02 = 000004            17#
BIT03 = 000010            17#
BIT04 = 000020            17#
BIT05 = 000040            17#
BIT06 = 000100            17#
BIT07 = 000200            17#
BIT08 = 000400            17#  2429
BIT09 = 001000            17#  2429  2431
BIT1  = 000002            17#   693
BIT10 = 002000            17#  2431
BIT11 = 004000            17#  2429
BIT12 = 010000            17#   694   706   723  2399
BIT13 = 020000            17#   737   750   762   774   785   821   843   859   877   923   953   983
                         1013  1025  1039  1072  1084  1097  1112  1123  1136  1151  1162  1173  1185
                         1197  1210  1225  1238  1251  1267  1282  1331  1344  1358  1372  1388  1402
                         1420  1434  1451  1466  1483  1499  1514  1524  1541  1555  1572  1585  1602
                         1615  1632  1645  1662  1675  1692  1705  1722  1737  1752  1766  1781  1797
                         1817  1839  1859  1876  1888  1909  1933  1952  1972  1988  2004  2020  2037
                         2067  2086  2106  2117  2135  2168  2191  2218  2380  2431
BIT14 = 040000            17#  2429
BIT15 = 100000            17#  1270  1744
BIT2  = 000004            17#
BIT3  = 000010            17#  1139
BIT4  = 000020            17#  1240  1255
BIT5  = 000040            17#
BIT6  = 000100            17#  1346
BIT7  = 000200            17#
BIT8  = 000400            17#
BIT9  = 001000            17#  1879  1882
BPTVEC= 000014            17#
BUFFER  031076             326   327   328   329   330   331   733*  751*  763*  775*  786   792   827
                           846*  847*  848   864*  865*  866   879*  880   931*  932*  933   961*  962*
                           963   990*  991*  992  1015* 1017  1030* 1031  1051* 1052* 1053  1124* 1268*
                          1274* 1275  1283* 1284* 1285  1288* 1306* 1307  1325* 1359* 1360  1361* 1375*
                          1376* 1377  1389* 1391  1392* 1404* 1406  1416* 1421* 1423  1424* 1436* 1438
                          1447* 1452* 1454  1455* 1467* 1468* 1469  1484* 1486  1493* 1502* 1515* 1516
                          1526* 1528  1537* 1542* 1544  1545* 1558* 1559  1568* 1574* 1575  1576* 1588*
                          1589  1598* 1604* 1605  1606* 1618* 1619  1628* 1634* 1635  1636* 1648* 1649
                          1658* 1664* 1665  1666* 1678* 1679  1688* 1694* 1695  1696* 1708* 1709  1718*
                          1724* 1725  1726* 1738* 1739  1740* 1742* 1756* 1770* 1782* 1783  1784* 1800*
                          1801* 1802  1812  1820* 1821* 1822  1830  1840* 1842* 1843  1852* 1854  1860*
                          1862  1863* 1877* 1878* 1880  1893* 1894* 1895  1905  1914* 1915* 1916  1924
                          1934* 1936* 1938  1946* 1948  1954* 1957  1958* 1973  1989  1992  2005  2021
                          2038  2066* 2068  2069* 2071* 2088* 2089* 2090  2107* 2108  2127  2138* 2139
                          2140* 2149  2234* 2235  2237* 2239* 2258* 2259  2261* 2263* 2287* 2288  2290*
                          2292* 2323* 2341* 2342  2344* 2346* 2370* 2382* 2383* 2384* 2385  2506#
CNTR    002022             335#
CONS    002052             365#  1529  1530  1547  1548  1560  1561  1578  1579  1590  1591  1608  1609
                          1620  1621  1638  1639  1650  1651  1668  1669  1680  1681  1698  1699  1710
                          1711  1728  1729  1736  1746  2241  2242* 2243  2249  2265  2266* 2267  2273
                          2294  2295* 2302  2306  2326  2348  2349* 2356  2360  2373
CR    = 000015            17#  2448
CRLF  = 000200            17#  2448
DBUF    002000             326#   739   752   764   776   791   822   826  1073* 1075  1085* 1087  1098*
                          1099  1100* 1102  1113* 1115  1126  1137* 1138  1140* 1141  1153* 1154  1164*
```

```
                    1165    1174*   1176    1186*   1188    1198*   1200    1211*   1213    1226*   1228    1239*   1241
                    1252*   1254    1269    1332*   1334    1345*   1347    1500*   1501    1753*   1755    1767*   1769
                    1975    2008    2024    2043
DBUF1    002002      327#    1212*   1253*
DBUF2    002004      328#
DBUF3    002006      329#
DBUF4    002010      330#
DBUF5    002012      331#
DDISP =  177570       17#      32     388
DDONE    002070      373#     393     395
DDONE1   002072      374#
DELAY    002026      337#    2423
DH1      031014       37      44      5    58      65      72      79      86      93     100     107     114     121
                     128     135     142     149     156     163     171     178     185     192     199     206     213
                     221     228     235     242     249     256     263     271     278     285     291     298     305
                     312     319    2498#
DISPLA   001142       32#     388*    2429*   2431*
DISPRE   000174       24#     388
DLAY     024514    1979    1995    2011    2027    2046    2057    2073    2423#
DNAME    002054      366#     849     850     867     868     881     883     889     893     903     907     914     918
                     934     935     964     965     993     995    1018    1020    1027    1033    1054    1056    1061
DPC      002030      356#     389     692     695*    696     707     708*    712     722     725*    727     739*    752*
                     764*    776*    784     791*    793     798*    800     805*    807     812*    814     820     826*
                     831     839*    848*    866*    880*    933*    963*    992*   1017*   1031*   1053*   1075*   1087*
                    1099*   1102*   1115*   1126*   1138*   1141*   1154*   1165*   1176*   1188*   1200*   1213*   1214*
                    1228*   1241*   1254*   1256*   1269*   1275*   1285*   1287*   1289*   1307*   1334*   1347*   1360*
                    1362*   1377*   1391*   1393*   1406*   1423*   1425*   1438*   1454*   1456*   1469*   1471*   1486*
                    1501*   1503*   1516*   1528*   1544*   1546*   1559*   1575*   1577*   1589*   1605*   1607*   1619*
                    1635*   1637*   1649*   1665*   1667*   1679*   1695*   1697*   1709*   1725*   1727*   1739*   1741*
                    1743*   1755*   1757*   1758    1759    1769*   1771*   1772    1773    1783*   1785*   1786    1787
                    1802*   1803    1804    1822*   1823    1824    1843*   1845*   1846    1848    1862*   1864*   1865
                    1866    1880*   1895*   1896    1897    1916*   1917    1918    1938*   1940*   1941    1942    1957*
                    1959*   1960    1961    1976*   1978*   1992*   1994*   2008*   2010*   2024*   2026*   2043*   2045*
                    2053*   2055*   2068*   2070*   2072*   2090*   2096*   2108*   2127*   2139*   2142*   2149*   2201*
                    2222*   2235*   2238*   2240*   2259*   2262*   2264*   2288*   2291*   2293*   2342*   2345*   2347*
                    2385*   2387*
DSAVE    002014      332#
DSAVE1   002016      333#
DSWR     177570       17#      32     388
DT1      031062       38      45      52      59      66      73      80      87      94     101     108     115     122
                     129     136     143     150     157     164     172     179     186     193     200     207     214
                     222     229     236     243     250     257     264     272     279     286     292     299     306
                     313     320    2500#
EMTVEC=  000030       17#     388*
EM1      027216       36    2457#
EM10     027454       85    2464#
EM11     027510       92    2465#
EM12     027544       99    2466#
EM13     027577      106    2467#
EM14     027632      113    2468#
EM15     027654      120    2469#
EM16     027676      127    2470#
EM17     027716      134    2471#
EM2      027240       43    2458#
EM20     027740      141    2472#
EM21     027764      148    2473#
```

```
EM22    030010              155    2474#
EM23    030030              162    2475#
EM24    030053              170    2476#
EM25    030115              177    2477#
EM26    030135              184    2478#
EM27    030153              191    2479#
EM3     027274               50    2459#
EM30    030172              198    2480#
EM31    030212              205    2481#
EM32    030230              212    2482#
EM33    030264              220    2483#
EM34    030301              227    2484#
EM35    030316              234    2485#
EM36    030335              241    2486#
EM37    030375              248    2487#
EM4     027315               57    2460#
EM40    030436              255    2488#
EM41    030501              262    2489#
EM42    030531              270    2490#
EM43    030572              277    2491#
EM44    030642              284    2492#
EM45    030662              290    2493#
EM46    030711              297    2494#
EM47    030732              304    2495#
EM5     027340               64    2461#
EM50    030755              311    2496#
EM51    030777              318    2497#
EM6     027366               71    2462#
EM7     027422               78    2463#
ERRVEC= 000004               17#    388*    417*    418*    428*    429*   2429*   2445*
GNS   = ****** U             24    2451
HEADER  027132              409    2455#
HT    = 000011               17#   2448
IOTVEC  000020               17#    388*
KIPARO- ****** U            2445
LF    = 000012               17#   2448
LPVCT   002074              376#
LPVCT1  002076              377#
NAMEVT  002104              382#
NAMEV1  002106              383#
PIRQ  - 177772               17#
PIRQVE= 000240               17#
PR0   - 000000               17#
PR1     000040               17#
PR2   - 000100               17#
PR3   - 000140               17#
PR4     000200               17#
PR5   = 000240               17#
PR6   = 000300               17#
PR7   = 000340               17#
PS    = 177776               17#
PSW     177776               17#    410*    924*    954*   1026*   1040*   1055*   1298*   1308*   2118*   2136*   2170*   2178*
                            2192*  2200*   2219*   2231*   2255*   2279*   2296*   2333*   2350*   2409*
PWRMSG  026434              2441   2442#
PWRVEC  000024               17#    388*   2441*
RESTRT  002472               25     407     410#   2403
```

```
RESVEC= 000010   17#
RLO     002040   360# 560  562* 563  572  576* 577  583* 599  601* 603  711* 2199*
                 2221*
RSTVEC  024454   402  1069 1328 2188 2213 2330 2377 2409#
SETUP1  002360   389#
SETUP2  002370   391# 394
SETUP3  002420   398# 401
SIZE    002020   334# 404* 405* 824  828
SREG0   002032   357# 736  741  749  754  761  766  773  778  1076 1077 1088 1089
                 1103 1104 1116 1117 1127 1128 1142 1143 1155 1156 1166 1167 1177
                 1178 1189 1190 1201 1202 1216 1217 1229 1230 1242 1243 1257 1258
                 1266 1271 1277 1290 1292 1299 1312 1316 2315 2317
SREG1   002042   361# 1335 1336 1348 1349 1357 1364 1378 1379 1394 1395 1407 1408
                 1426 1427 1439 1440 1457 1458 1472 1474 2091 2092 2098 2105 2111
                 2120 2122* 2123 2128 2141* 2143 2144 2150 2158 2160* 2162 2171 2177*
                 2179 2184 2197 2203 2208 2217 2225
STACK - 001100   17#  388  411
STKLMT= 177774   17#
STKPT   002062   369# 586  589* 590  647  649* 651  658  660* 661  670  672* 673*
                 674  680* 682  694* 706* 723* 737* 750* 762* 774* 785* 821* 843*
                 859* 877* 923* 953* 983* 1013* 1025* 1039* 1072* 1084* 1097* 1112* 1123*
                 1136* 1151* 1162* 1173* 1185* 1197* 1210* 1225* 1238* 1251* 1267* 1282* 1305*
                 1331* 1344* 1358* 1372* 1388* 1402* 1420* 1434* 1451* 1466* 1483* 1499* 1514*
                 1524* 1541* 1555* 1572* 1585* 1602* 1615* 1632* 1645* 1662* 1675* 1692* 1705*
                 1722* 1737* 1752* 1766* 1781* 1797* 1817* 1839* 1859* 1876* 1881 1882 1888*
                 1892* 1909* 1913* 1933* 1937* 1952* 1956* 1972* 1988* 2004* 2020* 2037* 2067*
                 2086* 2106* 2117* 2135* 2168* 2191* 2218* 2232* 2236* 2256* 2260* 2286* 2289*
                 2340* 2343* 2380* 2388 2389
STKVAL  002056   367#
SWR     000176   32#  388* 2399 2429 2431 2441*
SWREG   000176   24#  388
SW0     = 000001 17#
SW00    = 000001 17#
SW01    = 000002 17#
SW02    = 000004 17#
SW03    = 000010 17#
SW04    = 000020 17#
SW05    = 000040 17#
SW06    - 000100 17#
SW07    = 000200 17#
SW08    = 000400 17#
SW09    = 001000 17#
SW1     = 000002 17#
SW10    = 002000 17#
SW11    = 004000 17#
SW12    = 010000 17#
SW13    = 020000 17#
SW14    = 040000 17#
SW15    = 100000 17#
SW2     = 000004 17#
SW3     - 000010 17#
SW4     - 000020 17#
SW5     = 000040 17#
SW6     = 000100 17#
SW7     = 000200 17#
SW8     = 000400 17#
```

```
SW9    = 001000              17#
TBITVE= 000014               17#
TERMCH  002060               368#    433     437*    438     447     450*    451     461     463*    464*    465     638     640*
                             642
TIMEVT  002100               379#
TKVEC  = 000060              17#
TMEVT1  002102               380#
TPVEC  = 000064              17#
TRAPVE= 000034               17#     388*
TRTVEC= 000014               17#
TSTNUM  002024               336#    2431*   2500
TST1    002512               415#
TST10   003362               512#
TST100  013164              1367     1371#
TST101  013302              1387#
TST102  013410              1398     1401#
TST103  013534              1412     1415    1419#
TST104  013642              1430     1433#
TST105  013764              1446     1450#
TST106  014072              1461     1465#
TST107  014202              1476     1481#
TST11   003466               528#
TST110  014316              1497#
TST111  014422              1508     1512#
TST112  014510              1520     1523#
TST113  014632              1536     1540#
TST114  014740              1551     1554#
TST115  015062              1567     1571#
TST116  015156              1580     1584#
TST117  015300              1597     1601#
TST12   003570               543#
TST120  015376              1610     1614#
TST121  015520              1627     1631#
TST122  015616              1640     1644#
TST123  015740              1657     1661#
TST124  016036              1670     1674#
TST125  016160              1687     1691#
TST126  016256              1700     1704#
TST127  016400              1717     1721#
TST13   003674               558#
TST130  016476              1730     1735#
TST131  016620              1748     1751#
TST132  016720              1761     1765#
TST133  017020              1775     1779#
TST134  017136              1795#
TST135  017262              1809     1815#
TST136  017414              1829     1838#
TST137  017540              1853     1857#
TST14   003762               570#
TST140  017664              1871     1875#
TST141  017754              1883     1887#
TST142  020106              1902     1908#
TST143  020246              1923     1932#
TST144  020376              1947     1951#
TST145  020530              1966     1970#
TST146  020634              1982     1986#
```

```
TST147  020740            1998    2002#
TST15   004060             585#
TST150  021044            2014    2018#
TST151  021150            2030    2035#
TST152  021336            2051    2060    2064#
TST153  021474            2083    2085#
TST154  021612            2099    2104#
TST155  021710            2113    2116#
TST156  022032            2129    2134#
TST157  022200            2148    2151    2156#
TST16   004156             598#
TST160  022262            2163    2167#
TST161  022432            2190#
TST162  022624            2216#
TST163  022732            2227    2230#
TST164  023124            2251    2254#
TST165  023316            2275    2278#
TST166  023656            2332#
TST167  024162            2379#
TST17   004230             604     607#
TST170  024310            2398#
TST2    002614             431#
TST20   004302             613     616#
TST21   004354             622     625#
TST22   004450             634     637#
TST23   004522             643     646#
TST24   004602             653     656#
TST25   004676             669#
TST26   005030             684     691#
TST27   005126             700     704#
TST3    002710             445#
TST30   005244             717     721#
TST31   005324             728     735#
TST32   005420             744     748#
TST33   005514             757     760#
TST34   005610             769     772#
TST35   005704             781     783#
TST36   006134             797     804     811     816     819#
TST37   006272             835     838     842#
TST4    003010             460#
TST40   006410             858#
TST41   006534             876#
TST42   006622             884     887#
TST43   006726             902#
TST44   007012             910     913#
TST45   007064             919     922#
TST46   007310             952#
TST47   007534             982#
TST5    003066             467     470#
TST50   007750            1012#
TST51   010044            1021    1024#
TST52   010154            1035    1038#
TST53   010402            1071#
TST54   010476            1080    1083#
TST55   010572            1092    1096#
TST56   010702            1107    1111#
```

```
TST57   010774              1120    1122#
TST6    003162               483#
TST60   011070              1131    1135#
TST61   011200              1146    1150#
TST62   011266              1158    1161#
TST63   011354              1169    1172#
TST64   011450              1181    1184#
TST65   011544              1193    1196#
TST66   011640              1205    1209#
TST67   011746              1220    1224#
TST7    003266               499#
TST70   012040              1233    1237#
TST71   012134              1246    1250#
TST72   012242              1261    1265#
TST73   012354              1278    1281#
TST74   012476              1294    1297#
TST75   012672              1330#
TST76   012764              1339    1343#
TST77   013060              1352    1356#
TYPDS = 104405              2403    2451#
TYPE  = 104401               408    2403    2431    2434    2437    2438    2441    2448    2451#
TYPOC = 104402              2434    2451#
TYPON = 104404              2451#
TYPOS = 104403              2451#
XDOFF   002044               362#    501     503*    504     514     518*    519     526*    548*    608     610*    612     628*
XPOS    002034               358#   1487    1488    1498    1505    1513    1519    1971    1980    1987    1996    2036    2047
                            2065    2077
YDOFF   002046               363#    472     474*    475     485     489*    490     497*    533*    617     619*    621     629*
YPOS    002036               359#   2003    2012    2019    2028    2054    2058    2081    2082
ZDOFF   002066               371#    530     534     545     549     626     631
ZPOS    002064               370#
$APTHD  001000                30#
$ASTAT= ****** U             2430
$ATYC   025030               2430#
$ATY1   025004               2430#
$ATY3   025012               2430#   2448
$ATY4   025022               2430#   2431
$AUTOB  001134                32#
$BASE   001256                32#    390     419
$BDADR  001122                32#    419*    421     422*    433*    447*    461*    472*    485*    501*    514*    530*    545*
                             560*    572*    586*    599*    608*    617*    626*    638*    647*    658*    670*    692*    707*
                             722*    736*    749*    761*    773*    784*    820*    849*    867*    881*    889*    903*    914*
                             934*    964*    993*   1018*   1027*   1054*   1076*   1088*   1103*   1116*   1127*   1142*   1155*
                            1166*   1177*   1189*   1201*   1216*   1229*   1242*   1257*   1266*   1290*   1299*   1335*   1348*
                            1357*   1378*   1394*   1407*   1426*   1439*   1457*   1472*   1487*   1498*   1513*   1529*   1547*
                            1560*   1578*   1590*   1608*   1620*   1638*   1650*   1668*   1680*   1698*   1710*   1728*   1736*
                            1758*   1772*   1786*   1803*   1823*   1846*   1865*   1881*   1896*   1917*   1941*   1960*   1971*
                            1987*   2003*   2019*   2036*   2054*   2065*   2081*   2091*   2105*   2120*   2143*   2158*   2171*
                            2197*   2217*   2241*   2265*   2294*   2317*   2348*   2388*   2500
$BDDAT  001126                32#    438*    439     451*    452     465*    466     475*    476*    477     490*    491*    492
                             504*    505*    506     519*    520*    521     534*    535*    536     549*    550*    551     563*
                             564     577*    578     590*    591*    592     603*    612*    621*    631*    632*    633     642*
                             651*    652*    661*    662*    663     674*    675*    676     682*    683     696*    697     712*
                             713     727*    741*    742*    743     754*    755*    756     766*    767*    768     778*    779*
                             780     793*    794     800*    801     807*    808     814*    815     831*    832     850*    851*
                             852     868*    869*    870     883*    893*    894*    895     907*    908*    909     918*    935*
```

```
                        937     941*    942     944*    965*    967     971*    972     974*    995*    997     1001*   1002
                       1004*   1020*   1033*   1034    1056*   1061*   1077*   1078*   1079    1089*   1090*   1091    1104*
                       1105*   1106    1117*   1118*   1119    1128*   1129*   1130    1143*   1144*   1145    1156*   1157
                       1167*   1168    1173*   1179*   1180    1190*   1191*   1192    1202*   1203*   1204    1217*   1218*
                       1219    1230*   1231*   1232    1243*   1244*   1245    1258*   1259*   1260    1271*   1277*   1292*
                       1293    1312*   1316*   1321    1336*   1337*   1338    1349*   1350*   1351    1364*   1365*   1366
                       1379*   1380*   1381    1395*   1396*   1397    1408*   1409*   1410    1427*   1428*   1429    1440*
                       1441*   1442    1458*   1459*   1460    1474*   1475    1488*   1489*   1490    1505*   1506*   1507
                       1519*   1530*   1531*   1532    1548*   1549*   1550    1561*   1562*   1563    1581*   1591*   1592*
                       1593    1611*   1621*   1622*   1623    1641*   1651*   1652*   1653    1671*   1681*   1682*   1683
                       1701*   1711*   1712*   1713    1731*   1746*   1747    1759*   1760    1773*   1774    1787*   1788
                       1804*   1805    1824*   1825    1848*   1849    1866*   1867    1884*   1897*   1898    1918*   1919
                       1942*   1943    1961*   1962    1980*   1981    1996*   1997    2012*   2013    2028*   2029    2047*
                       2048    2058*   2059    2077*   2082*   2094*   2100*   2111*   2112    2125*   2131*   2146*   2153*
                       2164*   2179*   2180*   2186*   2203*   2204*   2208*   2209*   2210    2225*   2226    2243*   2244
                       2249*   2250    2267*   2268    2273*   2274    2302*   2306*   2311    2319*   2326*   2327    2356*
                       2360*   2365    2373*   2374    2389*   2390*   2391    2500
$BELL   001172          32#     2431
$CDW1   001262          32#
$CDW2   001264          32#
$CHARC  027056        2448#*
$CKSWR= ****** U      2451
$CMTAG  001100          32#     388
$CM1  = 000002          32#
$CM2  = 000004          32#
$CM3  = 000002          32#
$CPUOP  001230          32#
$CRLF   001177          32#     2431    2434    2448
$DBLK   026252        2438#
$DDWO   001266          32#
$DEVCT  001212          32#
$DEVM   001260          32#
$DOAGN  024430        2403#
$DTBL   026242        2436#
$ENDAD  024420          27      2403#   2431
$ENDCT  024366         388      2403#
$ENDMG  024437        2403#
$ENULL  024434        2403#
$ENV    001222          32#     2430    2431    2448
$ENVM   001223          32#      388    2430    2448
$EOP    024332        2400      2403#
$EOPCT  024360         388*     2403#
$ERFLG  001103          32#     2429*   2431*
$ERMAX  001115          32#      388*   2429*
$ERROR  025252         388      2431#
$ERRPC  001116          32#     2431*   2434    2500
$ERRTB  001270          32#     2434
$ERRTY  025454        2431      2434#
$ERTTL  001112          32#     2431*
$ESCAP  001170          32#      388*   2429*   2431
$ETABL  001222          32#
$ETEND  001270          30       32#
$FATAL  001204          32#     2430*
$FFLG   025250        2430#*
$FILLC  001156          32#     2448
$FILLS  001155          32#     2448
```

```
$GDADR  001120          32#
$GDDAT  001124          32#   434*  435   439   442*  448*  452   456*  457*  462*  466   473*  474
                        477   480*  487*  488*  492   502*  503   506   509*  516*  517*  521   532*
                        536   539*  547*  551   554*  555*  561*  562   564   567*  574*  575*  578
                        588*  589   592   595*  600*  609*  618*  627*  633   639*  648*  659*  660
                        663   666*  671*  676   679*  683   693*  695   697   701*  709*  713   716*
                        724*  740*  743   753*  756   765*  768   777*  780   792*  794   799*  801
                        806*  808   813*  815   827*  830*  832   837   845*  847   852   855*  862*
                        863*  865   870   878*  890*  895   899*  904*  909   915*  926*  927*  928
                        930*  937   939*  942   948*  949   956*  957*  958   960*  967   969*  972
                        978*  979   985*  986*  987   989*  997   999*  1002  1008*  1009  1016*  1032*
                        1034  1046*  1047*  1048  1050*  1066*  1067  1074*  1079  1086*  1091  1101*  1106
                        1114*  1119  1125*  1130  1139*  1145  1152*  1153  1163*  1164  1175*  1180  1187*
                        1192  1199*  1204  1215*  1219  1227*  1232  1240*  1245  1255*  1260  1270*  1276*
                        1286*  1293  1300*  1321  1333*  1338  1346*  1351  1363*  1366  1374*  1376  1381
                        1384*  1390*  1397  1405*  1410  1414*  1422*  1429  1437*  1442  1445*  1453*  1460
                        1470*  1475  1485*  1490  1494*  1504*  1507  1517*  1527*  1532  1535*  1543*  1550
                        1557*  1563  1566*  1573*  1587*  1593  1596*  1603*  1617*  1623  1626*  1633*  1647*
                        1653  1656*  1663*  1677*  1683  1686*  1693*  1707*  1713  1716*  1723*  1744*  1747
                        1754*  1756  1760  1768*  1770  1774  1780*  1784  1788  1792*  1805  1810*  1811*
                        1812*  1825  1830*  1835*  1841*  1849  1854*  1861*  1863  1867  1870*  1872*  1879*
                        1898  1903*  1904*  1905*  1919  1924*  1929*  1935*  1943  1948*  1955*  1958  1962
                        1965*  1967*  1977*  1981  1993*  1997  2009*  2013  2025*  2029  2044*  2048  2056*
                        2059  2076*  2087*  2097*  2109*  2112  2121*  2122  2130*  2137*  2141  2152*  2159*
                        2172*  2198*  2210  2220*  2226  2233*  2244  2248*  2250  2257*  2268  2272*  2274
                        2284*  2311  2318*  2322*  2327  2338*  2365  2369*  2374  2381*  2391  2394*  2395
                        2500
$GET42  024410          2403#
$GTSWR- ****** U        2451
$HD   - 000000          13
$HIBTS  001000          30#
$ICNT   001104          32#   2429*
$ILLUP  026426          2441#
$INTAG  001135          32#
$ITEMB  001114          32#   2431*  2434
$LF     001200          32#   2431   2448
$LFLG   025247          2430#*
$LPADR  001106          32#   388*  2429*
$LPERR  001110          32#   388*  416*  432*  446*  471*  484*  500*  513*  529*  544*  559*  571*
                        587*  657*  705*  844*  860*  888*  925*  955*  984*  1041*  1373*  1403*  1435*
                        1482*  1525*  1556*  1586*  1616*  1646*  1676*  1706*  1796*  1816*  1844*  1858*  1889*
                        1910*  1939*  1953*  2429*  2431
$LSTAD  026576          404   1780   2445#*
$MADR1  001234          32#
$MADR2  001240          32#
$MADR3  001244          32#
$MADR4  001250          32#
$MAIL   001202          30    32#   388   415   431   445   460   470   483   499   512   528   543
                        558   570   585   598   607   616   625   637   646   656   669   691   704
                        721   735   748   760   772   783   819   842   858   876   887   902   913
                        922   952   982   1012  1024  1038  1071  1083  1096  1111  1122  1135  1150
                        1161  1172  1184  1196  1209  1224  1237  1250  1265  1281  1330  1343  1356
                        1371  1387  1401  1419  1433  1450  1465  1481  1497  1512  1523  1540  1554
                        1571  1584  1601  1614  1631  1644  1661  1674  1691  1704  1721  1735  1751
                        1765  1779  1795  1815  1838  1857  1875  1887  1908  1932  1951  1970  1986
                        2002  2018  2035  2064  2085  2104  2116  2134  2156  2167  2190  2216  2230
```

```
                        2254    2278    2332    2379    2398    2429    2431    2448
$MAMS1  001232            32#
$MAMS2  001236            32#
$MAMS3  001242            32#
$MAMS4  001246            32#
$MBADR  001002            30#
$MFLG   025246          2430#*
$MSGAD  001216            32#   2430*
$MSGLG  001220            32#   2430*
$MSGTY  001202            32#   2430*
$MTYP1  001233            32#
$MTYP2  001237            32#
$MTYP3  001243            32#
$MTYP4  001247            32#
$MXCNT  025002          2429#
$NULL   001154            32#   2448
$NWTST  000001           415#   431#   445#   460#   470#   483#   499#   512#   528#   543#   558#   570#   585#
                         598#   607#   616#   625#   637#   646#   656#   669#   691#   704#   721#   735#   748#
                         760#   772#   783#   819#   842#   858#   876#   887#   902#   913#   922#   952#   982#
                        1012#  1024#  1038#  1071#  1083#  1096#  1111#  1122#  1135#  1150#  1161#  1172#  1184#
                        1196#  1209#  1224#  1237#  1250#  1265#  1281#  1297#  1330#  1343#  1356#  1371#  1387#
                        1401#  1419#  1433#  1450#  1465#  1481#  1497#  1512#  1523#  1540#  1554#  1571#  1584#
                        1601#  1614#  1631#  1644#  1661#  1674#  1691#  1704#  1721#  1735#  1751#  1765#  1779#
                        1795#  1815#  1838#  1857#  1875#  1887#  1908#  1932#  1951#  1970#  1986#  2002#  2018#
                        2035#  2064#  2085#  2104#  2116#  2134#  2156#  2167#  2190#  2216#  2230#  2254#  2278#
                        2332#  2379#  2398#
$OCNT   026032          2437#*
$OMODE  026034          2437#*
$OVER   024766          2429#
$PASS   001210            32#    388*   2403*   2429
$PASTM  001006            30#
$PWRAD  026422          2441#
$PWRDN  026262           388    2441#
$PWRMG  026416          2441#
$PWRUP  026334          2441#
$QUES   001176            32#   2431    2448
$RDCHR= ****** U        2451
$RDDEC= ****** U        2451
$RDLIN= ****** U        2451
$RDOCT= ****** U        2451
$REGAD  001160            32#
$REG0   001162            32#
$REG1   001164            32#
$RTNAD  024432          2403#
$R2A  - ****** U        2451
$SAVRE= ****** U        2451
$SAVR6  026432          2441#*
$SCOPE  024526           388   2429#
$SETUP= 000037           387#   388    2403    2429    2431
$SIZE   026502           403   2445#
$STUP = 177777           387#
$SVLAD  024732          2429#
$SVPC = 000210            27#
$SWR  - 167400             5#     13      22      32     388     415     431     445     460     470     485     499     512
                         528     543     558     570     585     598     607     616     625     637     646     656     669
                         691     704     721     735     748     760     772     783     819     842     858     876     887
```

```
                        902     913     922     952     982    1012    1024    1038    1071    1083    1096    1111    1122
                       1135    1150    1161    1172    1184    1196    1209    1224    1237    1250    1265    1281    1297
                       1330    1343    1356    1371    1387    1401    1419    1433    1450    1465    1481    1497    1512
                       1523    1540    1554    1571    1584    1601    1614    1631    1644    1661    1674    1691    1704
                       1721    1735    1751    1765    1779    1795    1815    1838    1857    1875    1887    1908    1932
                       1951    1970    1986    2002    2018    2035    2064    2085    2104    2116    2134    2156    2167
                       2190    2216    2230    2254    2278    2332    2379    2398    2403    2429    2431    2441
$SWREG  001224          32#     388
$SWRMK- 000000          22     2429
$TESTN  001206          32#     415*    431*    445*    460*    470*    483*    499*    512*    528*    543*    558*    570*
                       585*    598*    607*    616*    625*    637*    646*    656*    669*    691*    704*    721*    735*
                       748*    760*    772*    783*    819*    842*    858*    876*    887*    902*    913*    922*    952*
                       982*   1012*   1024*   1038*   1071*   1083*   1096*   1111*   1122*   1135*   1150*   1161*   1172*
                      1184*   1196*   1209*   1224*   1237*   1250*   1265*   1281*   1330*   1343*   1356*   1371*   1387*
                      1401*   1419*   1433*   1450*   1465*   1481*   1497*   1512*   1523*   1540*   1554*   1571*   1584*
                      1601*   1614*   1631*   1644*   1661*   1674*   1691*   1704*   1721*   1735*   1751*   1765*   1779*
                      1795*   1815*   1838*   1857*   1875*   1887*   1908*   1932*   1951*   1970*   1986*   2002*   2018*
                      2035*   2064*   2085*   2104*   2116*   2134*   2156*   2167*   2190*   2216*   2230*   2254*   2278*
                      2332*   2379*   2398*   2429*
$TIMES  001166          32#     388*    598*    607*    616*    625*    637*    646*    669*    691*    704*    721*    819*
                       876*    913*    922*    952*    982*   1012*   1038*   1281*   1297*   1465*   1512*   1735*   1779*
                      2064*   2104*   2116*   2156*   2167*   2190*   2216*   2230*   2254*   2278*   2332*   2403*   2429*
$TKB    001146          32#
$TKS    001144          32#
$TN   = 000171           6#      13     415#    431#    445#    460#    467     470#    483#    499#    512#    528#    543#
                       558#    570#    585#    598#    604     607#    613     616#    622     625#    634     637#    643
                       646#    653     656#    669#    684     691#    700     704#    717     721#    728     735#    744
                       748#    757     760#    769     772#    781     783#    797     804     811     816     819#    835
                       838     842#    858#    876#    884     887#    902#    910     913#    919     922#    952#    982#
                      1012#   1021    1024#   1035    1038#   1071#   1080    1083#   1092    1096#   1107    1111#   1120
                      1122#   1131    1135#   1146    1150#   1158    1161#   1169    1172#   1181    1184#   1193    1196#
                      1205    1209#   1220    1224#   1233    1237#   1246    1250#   1261    1265#   1278    1281#   1294
                      1297#   1330#   1339    1343#   1352    1356#   1367    1371#   1387#   1398    1401#   1412    1415
                      1419#   1430    1433#   1446    1450#   1461    1465#   1476    1481#   1497#   1508    1512#   1520
                      1523#   1536    1540#   1551    1554#   1567    1571#   1580    1584#   1597    1601#   1610    1614#
                      1627    1631#   1640    1644#   1657    1661#   1670    1674#   1687    1691#   1700    1704#   1717
                      1721#   1730    1735#   1748    1751#   1761    1765#   1775    1779#   1795#   1809    1815#   1829
                      1838#   1853    1857#   1871    1875#   1883    1887#   1902    1908#   1923    1932#   1947    1951#
                      1966    1970#   1982    1986#   1998    2002#   2014    2018#   2030    2035#   2051    2060    2064#
                      2083    2085#   2099    2104#   2113    2116#   2129    2134#   2148    2151    2156#   2163    2167#
                      2190#   2216#   2227    2230#   2251    2254#   2275    2278#   2332#   2379#   2398#
$TPB    001152          32#    2448*
$TPFLG  001157          32#    2448
$TPS    001150          32#    2448
$TRAP   027062         388     2451#
$TRAP2  027104        2451#
$TRP  = 000006        2451#
$TRPAD  027116        2451#
$TSTM   001004          30#
$TSTNM  001102          32#     413*   2403*   2429*   2431
$TYPBN= ****** U      2451
$TYPDS  026036        2438#   2451
$TYPE   026600        2430    2448#   2451
$TYPEC  027012        2448#
$TYPEX  027060        2448#
$TYPOC  025634        2437#   2451
```

```
$TYPON  025650            2437#   2451
$TYPOS  025610            2437#   2451
$UNIT   001214              32#
$UNITM  001010              30#
$USWR   001226              32#
$VECT1  001252              32#    396   1042   1301   2173   2193   2280   2334   2411
$VECT2  001254              32#
$XTSTR  024536            2429#
$$GET4= 000000            2403#
$OFILL  026033            2437#*
$40CAT- ****** U          2429    2431
.       - 031100            24#     27#    29#    30#    32#    388   2403   2429   2430#  2431   2434#  2438#  2441
                          2443#   2448   2499#
.$ASTA- ****** U          2430
.$X     001000              30#
```

```
COMMEN    17#
DELAY     347#
DELAY1    351#
ENDCOM    17#
ERROR     17#   427   441   454   468   479   494   508   523   538   553   566   580   594   605
          614   623   635   644   654   665   678   685   699   715   729   745   758   770   782
          796   803   810   817   834   854   872   885   897   911   920   940   945   970   975
         1000  1005  1022  1036  1058  1063  1081  1093  1108  1121  1132  1147  1159  1170  1182
         1194  1206  1221  1234  1247  1262  1273  1279  1295  1313  1319  1323  1340  1353  1368
         1383  1399  1413  1431  1444  1462  1477  1492  1509  1521  1534  1552  1565  1582  1595
         1612  1625  1642  1655  1672  1685  1702  1715  1732  1749  1762  1776  1790  1807  1827
         1851  1869  1885  1900  1921  1945  1964  1983  1999  2015  2031  2050  2061  2079  2084
         2095  2101  2114  2126  2132  2147  2154  2165  2181  2187  2205  2212  2228  2246  2252
         2270  2276  2303  2309  2313  2320  2329  2357  2363  2367  2376  2393
ESCAPE    17#
GETPRI    17#   2445
GETSWR    17#
MAINT1    338#  694   706   723
MAINT2    341#  737   750   762   774   785   821   843   859   923   953   983  1025  1039  1072
         1084  1097  1112  1123  1136  1151  1162  1173  1185  1197  1210  1225  1238  1251  1267
         1331  1344  1358  1372  1388  1402  1420  1434  1451  1499  1524  1541  1555  1572  1585
         1602  1615  1632  1645  1662  1675  1692  1705  1722  1752  1766  1781  1797  1817  1839
         1859  1876  1888  1909  1933  1952  1972  1988  2004  2020  2037  2067  2117  2135  2168
         2191  2380
MULT      17#
NEWTST    17#   415   431   445   460   470   483   499   512   528   543   558   570   585   598
          607   616   625   637   646   656   669   691   704   721   735   748   760   772   783
          819   842   858   876   887   902   913   922   952   982  1012  1024  1038  1071  1083
         1096  1111  1122  1135  1150  1161  1172  1184  1196  1209  1224  1237  1250  1265  1281
         1297  1330  1343  1356  1371  1387  1401  1419  1433  1450  1465  1481  1497  1512  1523
         1540  1554  1571  1584  1601  1614  1631  1644  1661  1674  1691  1704  1721  1735  1751
         1765  1779  1795  1815  1838  1857  1875  1887  1908  1932  1951  1970  1986  2002  2018
         2035  2064  2085  2104  2116  2134  2156  2167  2190  2216  2230  2254  2278  2332  2379
         2398
POP       17#   2430  2438  2441
PUSH      17#   2430  2438  2441
REPORT    17#
RESUME    344#  805   812   839  1214  1256  1757  1771  1785  1978  1994  2010  2026  2045  2053
         2055  2070  2072
SCOPE     17#   415   431   445   460   470   483   499   512   528   543   558   570   585   598
          607   616   625   637   646   656   669   691   704   721   735   748   760   772   783
          819   842   858   876   887   902   913   922   952   982  1012  1024  1038  1071  1083
         1096  1111  1122  1135  1150  1161  1172  1184  1196  1209  1224  1237  1250  1265  1281
         1297  1330  1343  1356  1371  1387  1401  1419  1433  1450  1465  1481  1497  1512  1523
         1540  1554  1571  1584  1601  1614  1631  1644  1661  1674  1691  1704  1721  1735  1751
         1765  1779  1795  1815  1838  1857  1875  1887  1908  1932  1951  1970  1986  2002  2018
         2035  2064  2085  2104  2116  2134  2156  2167  2190  2216  2230  2254  2278  2332  2379
         2398  2403
SETPRI    17#
SETTRA    2451#
SETUP     17#   388
SKIP      17#   467   604   613   622   634   643   653   684   698   700   717   728   744   757
          769   781   797   804   811   816   835   838   840   884   910   919  1021  1035  1080
         1092  1107  1120  1131  1146  1158  1169  1181  1193  1205  1220  1233  1246  1261  1272
         1278  1294  1339  1352  1367  1398  1412  1415  1430  1446  1461  1476  1508  1520  1536
         1551  1567  1580  1597  1610  1627  1640  1657  1670  1687  1700  1717  1730  1748  1761
```

```
        1775    1789    1809    1829    1853    1871    1883    1902    1923    1947    1966    1982    1998    2014    2030
        2051    2060    2078    2083    2099    2113    2129    2148    2151    2163    2227    2251    2275
SLASH     17#
SPACE     17#
STARS     17#     27      30      32     415     431     445     460     470     483     499     512     528     543     558
         570     585     598     607     616     625     637     646     656     669     691     704     721     735     748
         760     772     783     819     842     858     876     887     902     913     922     952     982    1012    1024
        1038    1071    1083    1096    1111    1122    1135    1150    1161    1172    1184    1196    1209    1224    1237
        1250    1265    1281    1297    1330    1343    1356    1371    1387    1401    1419    1433    1450    1465    1481
        1497    1512    1523    1540    1554    1571    1584    1601    1614    1631    1644    1661    1674    1691    1704
        1721    1735    1751    1765    1779    1795    1815    1838    1857    1875    1887    1908    1932    1951    1970
        1986    2002    2018    2035    2064    2085    2104    2116    2134    2156    2167    2190    2216    2230    2254
        2278    2332    2379    2398    2403    2406    2408    2420    2422    2429    2430    2431    2434    2437    2438
        2441    2445    2448    2451    2502    2505
SWRSU     17#   388#
TRMTRP  2451#
TYPBIN    17#
TYPDEC    17#   2403
TYPNAM    17#
TYPNUM    17#
TYPOCS    17#
TYPOCT    17#   2434
TYPTXT    17#
$$CMRE    32#
$$CMTM    32#
$$ESCA    17#
$$NEWT    17#     415     431     445     460     470     483     499     512     528     543     558     570     585     598
         607     616     625     637     646     656     669     691     704     721     735     748     760     772     783
         819     842     858     876     887     902     913     922     952     982    1012    1024    1038    1071    1083
        1096    1111    1122    1135    1150    1161    1172    1184    1196    1209    1224    1237    1250    1265    1281
        1297    1330    1343    1356    1371    1387    1401    1419    1433    1450    1465    1481    1497    1512    1523
        1540    1554    1571    1584    1601    1614    1631    1644    1661    1674    1691    1704    1721    1735    1751
        1765    1779    1795    1815    1838    1857    1875    1887    1908    1932    1951    1970    1986    2002    2018
        2035    2064    2085    2104    2116    2134    2156    2167    2190    2216    2230    2254    2278    2332    2379
        2398
$$SFT   2451#
$$SETM   388#
$$SKIP    7#     467     604     613     622     634     643     653     684     700     717     728     744     757     769
         781     797     804     811     816     835     838     884     910     919    1021    1035    1080    1092    1107
        1120    1131    1146    1158    1169    1181    1193    1205    1220    1233    1246    1261    1278    1294    1339
        1352    1367    1398    1412    1415    1430    1446    1461    1476    1508    1520    1536    1551    1567    1580
        1597    1610    1627    1640    1657    1670    1687    1700    1717    1730    1748    1761    1775    1809    1829
        1853    1871    1883    1902    1923    1947    1966    1982    1998    2014    2030    2051    2060    2083    2099
        2113    2129    2148    2151    2163    2227    2251    2275
.EQUAT     7#     17
.HEADE     7#     13
.KT11      9#
.SETUP     8#    387
.SWRHI     9#     22
.SWRLO    22#
.$ACT1    10#     27
.$APTB    10#    32#
.$APTH    10#     30
.$APTY    10#   2430
.$CATC     7#     24
.$CMTA     7#     32
```

```
.$EOP        7#   2403
.$ERRO       7#   2431
.$ERRT       9#   2434
.$PARM       8#
.$POWE       8#   2441
.$READ       8#
.$SAVE       8#
.$SCOP       8#   2429
.$SIZE       9#   2445
.$SPAC       8#
.$SWDO       8#
.$TRAP       8#   2451
.$TYPD       9#   2438
.$TYPE       7#      8#   2448
.$TYPO       7#   2437


. ABS.  031100     000     CON   RW    ABS   GBL   I


 ERRORS DETECTED:  0

 CZVSAB,CZVSAB/CRF=CZVSAB
 RUN-TIME: 36 23 2 SECONDS
 RUN-TIME RATIO: 134/62=2.1
 CORE USED:  26K  (51 PAGES)
```