AR11

DIAGNOSTIC TEST I

MD-11-DZARA-B

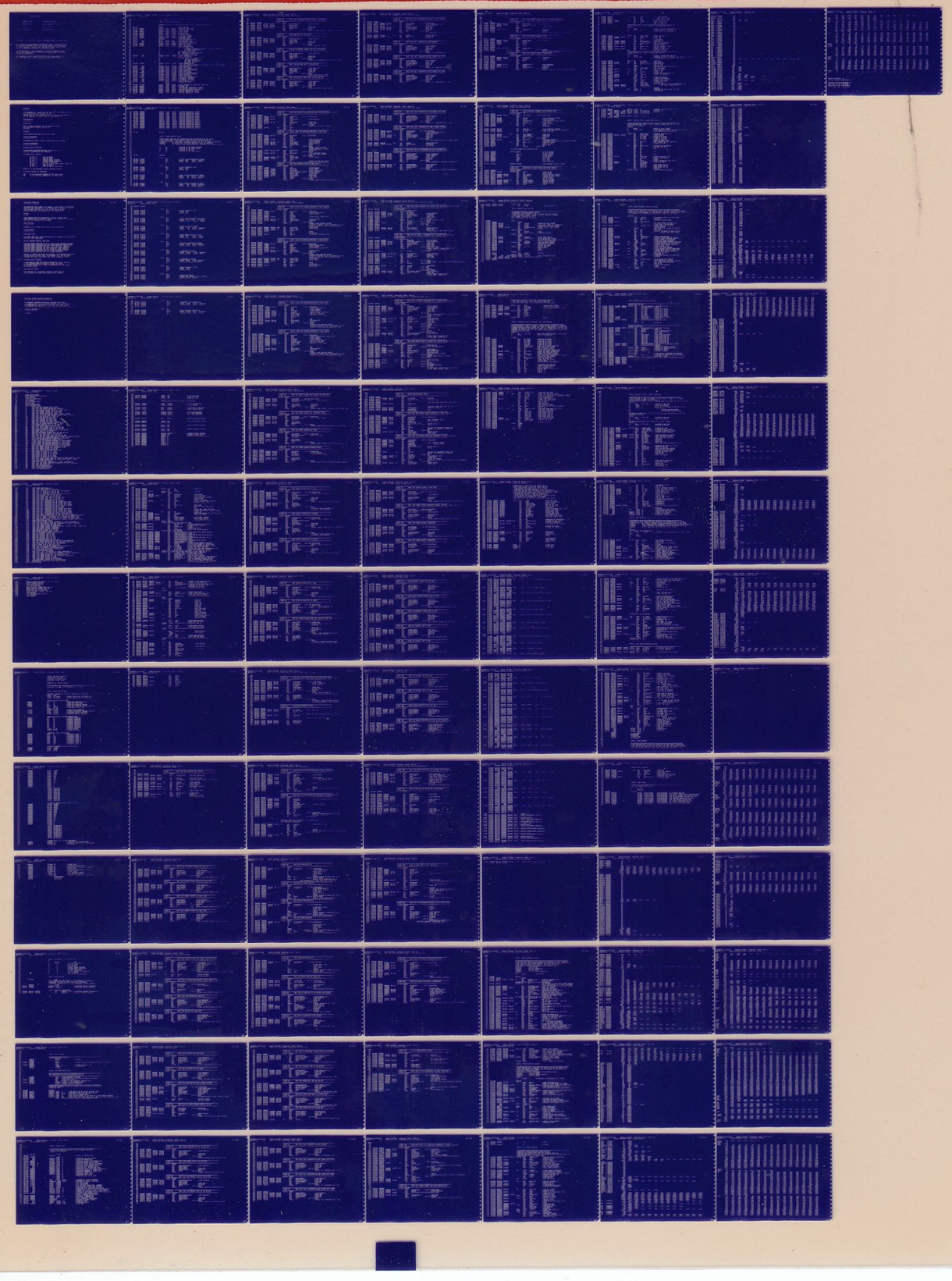
EP-DZARA-B-DL

FICHE 1 OF 1

MAY 1978

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZARA-B-D

PRODUCT NAME: AR11 DIAGNOSTIC TEST I

DATE: MAY 21, 1976

MAINTAINER: DIAGNOSTIC GROUP

1. ABSTRACT

THIS PROGRAM IS A LOGIC TEST OF THE AR-11 LAB SYSTEM OPTION. ALL FUNCTIONS OF THE OPTION WILL BE TESTED. DUE TO THE FLEXIBILITY OF THE OPTION, THE OPERATOR MAY BE REQUIRED TO SUPPLY OPTION CHARACTERISTICS.

2. REQUIREMENTS

2.1 EQUIPTMENT

PDP-11 FAMILY COMPUTER WITH 8K WORDS OF MEMORY
AR11 MEX OPTION MODULE INSTALLED
TELETYPE

2.2 STORAGE

THIS PROGRAM USES LESS THAN 8K OF MEMORY.

3. LOADING PROCEDURE

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

THIS PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT A HARDWARE SWITCH REGISTER.

STANDARD PDP-11 FORMAT

SW 15 = 1	HALT ON ERROR
SW 14 = 1	LOOP ON TEST
SW 13 = 1	INHIBIT ERROR TYPOUTS
SW 12 = 1	STORAGE SCOPE CONNECTED
SW 11 = 1	INHIBIT INTERATIONS
SW 10 = 1	ALL 1'S A TO D TEST JUMPER INSTALLED
SW 09 = 1	LOOP ON ERROR
SW 08 = 1	LOOP ON TEST IN SWR <7:0>

REFER TO 9. FOR SOFTWARE SWITCH REGISTER CONTROL

4.2 STARTING ADDRESS OR ADDRESSES

200 IS THE STARTING ADDRESS OF THE LOGIC TEST.
204 IS THE RESTART ADDRESS OF THE LOGIC TEST.

5. OPERATING PROCEDURE

THE OPERATOR MUST INSERT THE CORRECT INFORMATION IN THE SWITCH REGISTER WHEN REQUESTED BY THE PROGRAM OR AN ERROR WILL OCCUR. ONCE STARTED THE TEST WILL RUN IN ITS NORMAL MANNER WITHOUT OPERATOR INTERVENTION OR SWITCH SELECTION.

6. ERRORS

THIS PROGRAM USES THE DIAGNOSTIC 'SYSMAC' PACKAGE FOR ERROR REPORTING AND TYPEOUT. REFER TO THE "ERROR POINTER TABLE" FOR TYPE AND DESCRIPTION OF ERRORS.

7. RESTRICTIONS

EXTERNAL A TO D START INPUT MUST NOT BE CONNECTED.

8. MISCELLANEOUS

8.1 EXECUTION TIME

THE LOGIC TEST WILL TAKE APPROXIMATELY 60 SECONDS FOR COMPLETION AND WILL TYPE 'END PASS'.

8.2 DEVICE ADDRESS PROGRAM LOCATIONS

LOCATION SBASE CONTAINS THE AR11 BASE DEVICE ADDRESS <170400>
LOCATION SVECT1 CONTAINS THE AR11 BASE INTERRUPT VECTOR <340>
LOCATION ADBRL CONTAINS THE AR11 A TO D BR LEVEL <300><6>
LOCATION CKBRL CONTAINS THE AR11 CLOCK BR LEVEL <300><6>
LOCATION VCBRL CONTAINS THE AR11 SCOPE BR LEVEL <200><4>
LOCATION SFILLS CONTAINS THE TTY FILLER CHARACTER COUNT
LOCATION SNULL CONTAINS THE TTY FILLER CHARACTER

*NOTE: IF THESE LOCATIONS ARE CHANGED, THE OPERATOR MUST START THE TEST AGAIN AT LOC. 200. THE PROGRAM WILL USE THE BASE ADDRESS AND VECTOR AND UPDATE THE ACTUAL PROGRAM VALUES.

8.3 MULTIPLE AR-11 TESTING

A PROVISION IS MADE FOR TESTING SEQUENTIAL AR-11. STARTING AT BUS ADDRESS/VECTOR DEFINED BY SBASE AND SVECT1. THE HEADER TYPEOUT WILL INFORM THE OPERATOR OF THE NUMBER OF AR-11'S FOUND.

8.4 XXDP/ACT/APT NOTES

THIS PROGRAM IS A CHAINABLE PROGRAM UNDER XXDP/ACT. THE APT HOOKS HAVE BEEN INSTALLED BUT NOT TESTED.

9. SOFTWARE SWITCH REGISTER OPERATION

THE PROGRAM SUPPORTS NON-SWITCH REGISTER CPU TYPES.
A CHANGE IN SWR VALUE IS ACCOMPLISHED BY TYPING A "CTRL G".
THE RESPONSE WILL BE "SWR = " AND WAIT FOR A NEW VALUE.
THE OPERATOR NOW INPUTS THE NEW VALUE AND TERMINATES WITH A "CR".

10. TABLE OF CONTENTS

ATTACHED.

DZARAB.P11 TABLE OF CONTENTS

```
                        .TITLE  MAINDEC-11-DZARA-B
                        ;*COPYRIGHT (C) 1976
                        ;*DIGITAL EQUIPMENT CORP.
                        ;*MAYNARD, MASS. 01754
                        ;*
                        ;*PROGRAM BY RAYMOND SHOOP
                        ;*
                        ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                        ;*PACKAGE (MAINDEC-11-DZQAC-B2),NOV 21, 1975.
                        ;*

                        .SBTTL  BASIC DEFINITIONS

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

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

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

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

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
```

```
(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
     14          170000                          ABASE=170000
     15          000340                          AVECT1=340
     16          000200                          APRIOR=200
```

```
 18
 (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            STORAGE SCOPE CONNECTED
 (1)                            ;*        11            INHIBIT ITERATIONS
 (1)                            ;*        10            ALL 1'S JUMPER IS INSTALLED
 (1)                            ;*         9            LOOP ON ERROR
 (1)                            ;*         8            LOOP ON TEST IN SWR<7:0>
 19
 (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)
 (1)                            .SBTTL  STARTING ADDRESS(ES)
 (1)  000200    000137  001576          JMP     @#BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
 20   000204    000137  001602          JMP     @#BEGIN1        ;JUMP TO RESTART ADDRESS
```

```
 22
 23
(1)                             .SBTTL  ACT11 HOOKS
(1)
(2)                             ;;*****************************************************************
(1)                             ;HOOKS REQUIRED BY ACT11
(1)          000210                     $SVPC=.                 ;SAVE PC
(1)          000046                     .=46
(1)  000046  013342                     $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
(1)          000052                     .=52
(1)  000052  000000                     .WORD   0               ;;2)SET LOC.52 TO ZERO
(1)          000210                     .=$SVPC                 ;; RESTORE PC
 24          001000                     .=1000
 25
(1)                             .SBTTL  APT PARAMETER BLOCK
(1)
(2)                             ;;*****************************************************************
(1)                             ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(2)                             ;;*****************************************************************
(1)          001000                     .$=.    ;;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                     $APTHDR ;;POINT TO APT HEADER BLOCK
(1)          001000                     .=.$    ;;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             $MBITS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1)  001002  001174             $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
(1)  001004  000010             $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
(1)  001006  000010             $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(1)  001010  000200             $UNITM: .WORD   200     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(1)  001012  000052                     .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
 26
(1)                                     .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  000000                     .WORD   0
(1)  001136  177570             $SWR:   .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
(1)  001140  177570             DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
(1)  001142  177560             $TKS:   177560                  ;;TTY KBD STATUS
(1)  001144  177562             $TKB:   177562                  ;;TTY KBD BUFFER
(1)  001146  177564             $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
(1)  001150  177566             $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
(1)  001152     000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
(1)  001153     002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1)  001154     012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
(1)  001155     000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
(1)  001156  000000             $REGAD: .WORD   0               ;;CONTAINS THE ADDRESS FROM
(1)                                                             ;;WHICH  ($REG0) WAS OBTAINED
(3)  001160  000000             $REG0:  .WORD   0               ;;CONTAINS ((SREGAD)+0)
(3)  001162  000000             $REG1:  .WORD   0               ;;CONTAINS ((SREGAD)+2)
(1)  001164  000000             $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
(1)  001166  000000             $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
(1)  001170     077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
(1)  001171     015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
(1)  001172  000012             $LF:    .ASCIZ  <12>            ;;LINE FEED
```

M1

```
(2)                                 ;;**********************************************************************
(2)
(2)                                 .SBTTL  APT MAILBOX-ETABLE
(2)
(3)                                 ;;**********************************************************************
(2)                                 .EVEN
(2)  001174                         $MAIL:                          ;;APT MAILBOX
(2)  001174  000000                 $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
(2)  001176  000000                 $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
(2)  001200  000000                 $TESTN: .WORD   ATESTN  ;;TEST NUMBER
(2)  001202  000000                 $PASS:  .WORD   APASS   ;;PASS COUNT
(2)  001204  000000                 $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
(2)  001206  000000                 $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
(2)  001210  000000                 $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
(2)  001212  000000                 $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
(2)  001214                         $ETABLE:                        ;;APT ENVIRONMENT TABLE
(2)  001214     000                 $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
(2)  001215     000                 $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
(2)  001216  000000                 $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
(2)  001220  000000                 $USWR:  .WORD   AUSWR   ;;USER SWITCHES
(2)  001222  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=07,0=10
(2)                                 ;*                              BIT 10=REAL TIME CLOCK
(2)                                 ;*                              BIT  9=FLOATING POINT PROCESSOR
(2)                                 ;*                              BIT  8=MEMORY MANAGEMENT
(2)  001224     000                 $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(2)  001225     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)  001226  000000                 $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
(2)                                 ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(2)  001230     000                 $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(2)  001231     000                 $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
(2)  001232  000000                 $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(2)  001234     000                 $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(2)  001235     000                 $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
(2)  001236  000000                 $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(2)  001240     000                 $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(2)  001241     000                 $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
(2)  001242  000000                 $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
(2)  001244     340                 $VECT1: .BYTE   AVECT1  ;;INTERRUPT VECTOR#1
(2)  001245     000                 $VECT2: .BYTE   AVECT2  ;;INTERRUPT VECTOR#2
(2)  001246     200                 $PRIOR: .BYTE   APRIOR  ;;BUS PRIORITY #1,#2
(2)  001247     000                         .BYTE   0       ;;SPARE,NOT USED
(2)                                         .EVEN
(2)  001250  170400                 $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2)  001252  000000                 $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
(2)  001254  000000                 $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
(2)  001256  000000                 $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
(2)  001260  000000                 $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
(2)  001262  000000                 $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
(2)  001264  000000                 $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
```

```
(2)  001266  000000                    $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
(2)  001270  000000                    $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
(2)  001272  000000                    $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
(2)  001274  000000                    $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
(2)  001276  000000                    $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
(2)  001300  000000                    $DDW8:  .WORD   ADDW8   ;;DEVICE DESCRIPTOR WORD#8
(2)  001302  000000                    $DDW9:  .WORD   ADDW9   ;;DEVICE DESCRIPTOR WORD#9
(2)  001304  000000                    $DDW10: .WORD   ADDW10  ;;DEVICE DESCRIPTOR WORD#10
(2)  001306  000000                    $DDW11: .WORD   ADDW11  ;;DEVICE DESCRIPTOR WORD#11
(2)  001310  000000                    $DDW12: .WORD   ADDW12  ;;DEVICE DESCRIPTOR WORD#12
(2)  001312  000000                    $DDW13: .WORD   ADDW13  ;;DEVICE DESCRIPTOR WORD#13
(2)  001314  000000                    $DDW14: .WORD   ADDW14  ;;DEVICE DESCRIPTOR WORD#14
(2)  001316  000000                    $DDW15: .WORD   ADDW15  ;;DEVICE DESCRIPTOR WORD#15
(2)
(2)
(2)  001320                            $ETEND:
(2)
(1)
(1)                                     .$DTTL  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)  001320                            $ERRTB:
 27
 28                                     ;ITEM   1
 29  001320  014066                             EM1             ;A TO D STATUS REGISTER IN ERROR
 30  001322  014662                             DH1             ;$ERRPC  ADCS    ADSTAT  EXPECTED
 31  001324  015466                             DT1             ;$ERRPC ADCS    $DDDAT  $GDDAT
 32  001326  000000                             0
 33
 34                                     ;ITEM   2
 35  001330  014126                             EM2             ;A TO D INTERRUPT ERROR
 36  001332  014722                             DH2             ;$ERRPC  ADCS
 37  001334  015500                             DT2             ;$ERRPC ADCS
 38  001336  000000                             0
 39
 40                                     ;ITEM   3
 41  001340  014155                             EM3             ;CLOCK STATUS REGISTER IN ERROR
 42  001342  014740                             DH3             ;$ERRPC    CSR   CKSTAT  EXPECTED
 43  001344  015506                             DT3             ;$ERRPC   CSR   $DDDAT  $GDDAT
 44  001346  000000                             0
 45
 46                                     ;ITEM   4
 47  001350  014214                             EM4             ;CLOCK PRESET REGISTER IN ERROR
 48  001352  015000                             DH4             ;$ERRPC    CSB   CKBUFF  EXPECTED
 49  001354  015520                             DT4             ;$ERRPC   CSB   $DDDAT  $GDDAT
```

```
                                ;ITEM   5
001360  014253                          EM5             ;CLOCK INTERRUPT ERROR
001362  015040                          DM5             ;ERRPC  CSR
001364  015532                          DT5             ;SERRPC CSR
001366  000000                          0

                                ;ITEM   6
001370  014301                          EM6             ;CLOCK COUNTER REGISTER IN ERROR
001372  015055                          DM6             ;ERRPC  CSC  CKCNTR    EXPECTED
001374  015540                          DT6             ;SERRPC   CSC    SBDDAT   SGDDAT
001376  000000                          0

                                ;ITEM   7
001400  014341                          EM7             ;CLOCK COUNTED IN ERROR
001402  015055                          DM6             ;ERRPC    CSC   CKCNTR  EXPECTED
001404  015540                          DT6             ;SERRPC   CSC   SBDDAT  SGDDAT
001406  000000                          0

                                ;ITEM   10
001410  014370                          EM10            ;CLOCK REPEATABILITY FAILED
001412  015115                          DM10            ;ERRPC  CSC     TIME1   TIME2
001414  015540                          DT6             ;SERRPC  CSC    SBDDAT  SGDDAT
001416  000000                          0

                                ;ITEM   11
001420  014423                          EM11            ;VC STATUS REGISTER IN ERROR
001422  015153                          DM11            ;ERRPC  VCADR   VCSTAT EXPECTED
001424  015552                          DT11            ;SERRPC VCSTAT  SBDDAT  SGDDAT
001426  000000                          0

                                ;ITEM   12
001430  014457                          EM12            ;X REGISTER IN ERROR
001432  015213                          DM12            ;ERRPC  VCXPOS   X POS.  EXPECTED
001434  015564                          DT12            ;SERRPC VCXPOS   SBDDAT  SGDDAT
001436  000000                          0

                                ;ITEM   13
001440  014503                          EM13            ;Y REGISTER IN ERROR
001442  015254                          DM13            ;ERRPC VCYPOS   Y POS. EXPECTED
001444  015576                          DT13            ;SERRPC VCYPOS SBDDAT SGDDAT
001446  000000                          0

                                ;ITEM   14
001450  014527                          EM14            ;SCOPE INTERRUPT ERROR
001452  015315                          DM14            ;ERRPC  VCADR
001454  015610                          DT14            ;SERRPC VCSTAT
001456  000000                          0

                                ;ITEM   15
001460  014555                          EM15            ;DEVICE BUS ERROR
001462  015334                          DM15            ;ERRPC BASE ADDRESS ACTUAL ADDRESS
001464  015616                          DT15            ;SERRPC ARBADD  SBDDAT
001466  000000                          0
```

```
                                        ;ITEM   16
001470  014576                                EM16            ;INCORRECT A/D BUFFER DATA
001472  015364                                DM16            ;ERRPC  ADDBR   BUFFER  EXPECTED
001474  015626                                DT16            ;SERRPC ADDBR   SBDDAT  SGDDAT
001476  000000                                0
                                        ;ITEM   17
001500  014630                                EM17            ;DUAL REGISTER ADDRESSING
001502  015425                                DM17            ;ERRPC  BUFADR  READ    EXPECTED
001504  015640                                DT17            ;SERRPC BUFADR  SBDDAT  SGDDAT
001506  000000                                0
```

```
                                ;A TO D BR LEVEL
                                ;CLOCK BR LEVEL
                                ;SCOPE BR LEVEL

                                ;PROGRAM WILL CHANGE THE NEXT 16 LOCATIONS

                                ;A TO D STATUS REG
                                ;A TO D BUFFER

                                ;CLOCK STATUS REGISTER
                                ;CLOCK PRESET BUFFER

                                ;VC STATUS REGISTER
                                ;VC X AXIS REGISTER
                                ;VC Y AXIS REGISTER

                                ;CLOCK COUNTER REGISTER

                                ;A TO D INTERRUPT VECTOR

                                ;CLOCK INTERRUPT VECTOR

                                ;SCOPE INTERRUPT VECTOR

                                ;CURRENT DEVICE ADDRESS
                                ;CURRENT DEVICE VECTOR
                                ;NUMBER OF ADDITIONAL AR11'S
```

```
                                                
001576  005000          BEGIN:  CLR     R0              ;CLEAR R0
001600  000402                  BR      RBEG
001602  012700  177777  BEGIN1: MOV     #-1,R0          ;LOAD R0
001606  000005          RBEG:   RESET
001610  012706  001100          MOV     #STACK,SP       ;LOAD STACK
001614  012737  001642  000004  MOV     #18$,@#4        ;LOAD BUS ERROR
001622  013702  001250          MOV     $BASE,R2        ;LOAD STARTING ADDRESS
001626  005003                  CLR     R3              ;CLEAR COUNT
001630  005712          2$:     TST     (R2)            ;TEST IF EXISTENT
001632  062702  000020          ADD     #20,R2          ;EXIST, UPDATE TEST ADDRESS
001636  005203                  INC     R3              ;UPDATE # OF AR11'S
001640  000773                  BR      2$
001642  022626          1$:     CMP     (SP)+,(SP)+     ;POP STACK
001644  005703                  TST     R3              ;TEST IF FIRST DOES EXIST
001646  001001                  BNE     3$              ;BR
001650  000000                  HALT                    ;FIRST AR11 DOES NOT EXIST
                                                        ;CHECK THE PROGRAM DEVICE ADDRESS
001652  005303          3$:     DEC     R3              ;ADJUST R3
001654  010337  001556          MOV     R3,NMBEXT       ;SAVE THE NUMBER OF ADDITIONAL AR11
001660  012737  000006  000004  MOV     #6,@#4          ;RESET BUS ERROR
001666  005037  000006          CLR     @#6
001672  013737  001250  001552  MOV     $BASE,ARBADD    ;LOAD FIRST ADDRESS
001700  013737  001244  001554  MOV     $VECT1,ARBVCT   ;LOAD FIRST VECTOR
001706  013737  001556  001560  MOV     NMBEXT,NBEXT    ;LOAD NUMBER OF AR11'S
001714  000005          RBEG11: RESET
                        ;;CLEAR THE COMMON TAGS (SCMTAB) AREA
001716  012706  001100          MOV     #SCMTAB,R6      ;;FIRST LOCATION TO BE CLEARED
001722  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001724  022706  001126          CMP     #SBSDAT,R6      ;;DONE?
001730  001374                  BNE     .-6             ;;LOOP BACK IF NO
001732  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                        ;;INITIALIZE A FEW VECTORS
001736  012737  015652  000020  MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001744  012737  000340  000022  MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001752  012737  016132  000030  MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001760  012737  000340  000032  MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001766  012737  020440  000034  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001774  012737  000340  000036  MOV     #340,@#TRAPVEC+2;LEVEL 7
002002  012737  017242  000024  MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
002010  012737  000340  000026  MOV     #340,@#PWRVEC+2 ;;LEVEL 7
002016  013737  013310  013302  MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
002024  005037  001164          CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
002030  005037  001166          CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
002034  112737  000001  001115  MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
002042  012737  002042  001106  MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
002050  012737  002050  001110  MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
                        ;;SIZE FOR A HARDWARE SWITCH REGISTER, IF NOT FOUND OR IT IS
                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
002056  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
002062  012737  002120  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
002070  012737  177570  001136  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
002076  012737  177570  001140  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
002104  022777  177777  177024  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
002112  001013                  BNE     65$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
```

F2

```
   (2)  002114  005737  000001            TST     @#1             ;;FORCE A TRAP THROUGH ERRVEC
   (2)  002120  012737  000174  001136  648:  MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
   (2)  002126  012737  000174  001140        MOV     #DISPREG,DISPLAY ;;POINT TO SOFTWARE DISPLAY REG
   (2)  002134  012716  002142               MOV     #658,(SP)       ;;REPLACE OLD PC WITH NEW
   (2)  002140  000002                       RTI                     ;;RESTORE PC AND PSW
   (2)  002142  012637  000004        658:   MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
   (1)
   (1)  002146                        $ARG11:
   (2)  002146  005037  001202               CLR     $PASS           ;;CLEAR PASS COUNT
   (2)  002152  132737  000200  001215        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
   (2)  002160  001403                       BEQ     648             ;;YES,USE NON-APT SWITCH
   (2)  002162  012737  001216  001136        MOV     #$$SWREG,SWR    ;;NO,USE APT SWITCH REGISTER
   (2)  002170                        648:
   187  002170  000005               RBE821:  RESET
   188  002172  012702  000232               MOV     #232,R2                 ;LOAD R2
   189  002176  012701  000230               MOV     #230,R1                 ;LOAD R1
   190  002202  010221               58:     MOV     R2,(R1)+                ;LOAD .+2
   191  002204  005021                       CLR     (R1)+                   ;LOAD HALT
   192  002206  010102                       MOV     R1,R2                   ;LOAD R2
   193  002210  005722                       TST     (R2)+                   ;BUMP R2
   194  002212  020227  001002               CMP     R2,#1002                ;TEST FOR LAST
   195  002216  001371                       BNE     58                      ;BR UNTIL DONE
   196  002220  005700                       TST     R0                      ;TEST R0
   197  002222  001402                       BEQ     28                      ;BR IF CLEARED
   198  002224  000137  002420               JMP     48                      ;INHIBIT TYPOUT
   199  002230  005737  000042        28:    TST     @#42                    ;TEST ACT-11 OR DDP
   200  002234  001402                       BEQ     38                      ;BR IF CLEARED
   201  002236  000137  002420               JMP     48                      ;INHIBIT TYPOUT
   202  002242                        38:
   (1)  002242  104400  002250               TYPE    ,658            ;;TYPE ASCIZ STRING
   (1)  002246  000415                       BR      648             ;;GET OVER THE ASCIZ
   (1)                                ;;658:  .ASCIZ  <15><12>/AR-11 DIAGNOSTIC TEST I/
   (1)  002302                        648:
   203  002302  104400  002310               TYPE    ,678            ;;TYPE ASCIZ STRING
   (1)  002306  000414                       BR      668             ;;GET OVER THE ASCIZ
   (1)                                ;;678:  .ASCIZ  <15><12>/MAINDEC-11-DZARA-B/<15><12>
   (1)  002340                        668:
   204  002340  013746  001556               MOV     NMBEXT,-(SP)            ;PUSH ON STACK
   205  002344  104402                       TYPOS                           ;TYPE OCTAL
   206  002346  000002                       .WORD   2
   207  002350  104400  002356               TYPE    ,698            ;;TYPE ASCIZ STRING
   (1)  002354  000421                       BR      688             ;;GET OVER THE ASCIZ
   (1)                                ;;698:  .ASCIZ  /(8) ADDITIONAL AR11'S CONNECTED/<15><12>
   (1)  002420                        688:
   208
   209  002420  000240               48:     NOP
   210  002422  012700  001516               MOV     #ADCS,R0                ;LOAD POINTER
   211  002426  013720  001552        108:   MOV     ARBADD,(R0)+
   212  002432  022700  001536               CMP     #ADINT,R0               ;TEST FOR END
   213  002436  001373                       BNE     108
   214  002440  013720  001554        118:   MOV     ARBVCT,(R0)+            ;LOAD VECTOR
   215  002444  022700  001552               CMP     #ARBADD,R0
   216  002450  001373                       BNE     118
   217  002452  012700  001520               MOV     #ADDBR,R0
   218  002456  012701  000002               MOV     #2,R1
   219  002462  060120                128:   ADD     R1,(R0)+
```

```
   220  002464  005721                          TST     (R1)+
   221  002466  022701  000020                  CMP     #20,R1
   222  002472  001373                          BNE     12$
   223  002474  005720                          TST     (R0)+
   224  002476  012701  000002                  MOV     #2,R1
   225  002502  060120                  13$:    ADD     R1,(R0)+
   226  002504  005721                          TST     (R1)+
   227  002506  022701  000014                  CMP     #14,R1
   228  002512  001373                          BNE     13$
```

```
                        ;;*****************************************************************
                        ;*TEST 1         TEST EACH BUS ADDRESS FOR TIMEOUT
                        ;;*****************************************************************
002514  000004          $ST11:  SCOPE
002516  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS

002524  012737  002574  000004          MOV     #10$,@#4        ;LOAD BUS RETURN
002532  013737  001516  001126          MOV     ADCS,$BDDAT     ;LOAD WITH BUS ADDRESS TO BE TESTED

002540  005777  176362          28$:    TST     @$BDDAT         ;TEST ADDRESS
002544  023737  001126  001534          CMP     $BDDAT,CSC      ;TEST FOR LAST
002552  001404                        BEQ     38              ;BR IF DONE
002554  062737  000002  001126          ADD     #2,$BDDAT       ;MAKE NEXT ADDRESS
002562  000766                        BR      28              ;TRY MORE
002564  012737  000006  000004  38$:   MOV     #6,@#4
002572  000407                        BR      TST2            ;;BR TO NEXT TEST

002574  022626          10$:    CMP     (SP)+,(SP)+     ;CLEAN STACK
002576  012737  000006  000004          MOV     #6,@#4          ;RESET 4
002604  104015                        ERROR   15              ;DEVICE BUS ERROR
002606  000137  013166                JMP     BYPASS          ;DONT TEST ANY MORE
```

```
                                ;;*****************************************************************
                                ;*TEST 2         TEST THAT THE PRESET BUFFER CAN HOLD 00
                                ;;*****************************************************************
002612  000004                  TST2:   SCOPE
002614  012737  000000  001124          MOV     #0,@#DDAT               ;LOAD EXPECTED
002622  013777  001124  176674          MOV     @#DDAT,@#CSR            ;LOAD PRESET BUFFER
002630  017737  176670  001126          MOV     @#CSR,@#DDAT            ;READ REG.
002636  023737  001124  001126          CMP     @#DDAT,@#DDAT           ;COMPARE
002644  001401                          BEQ     TST3            ;;BR IF EQUAL
002646  104004                          ERROR   4                       ;ERROR, COUNTER PRESET FAILED TO CLEAR

                                ;;*****************************************************************
                                ;*TEST 3         TEST THE COUNTER PRESET BUFFER CAN HOLD #377
                                ;;*****************************************************************
002650  000004                  TST3:   SCOPE
002652  012737  000377  001124          MOV     #377,@#DDAT             ;LOAD EXPECTED
002660  013777  001124  176636          MOV     @#DDAT,@#CSR            ;LOAD PRESET BUFFER
002666  017737  176632  001126          MOV     @#CSR,@#DDAT            ;READ REG.
002674  023737  001124  001126          CMP     @#DDAT,@#DDAT           ;COMPARE
002702  001401                          BEQ     TST4            ;;BR IF EQUAL
002704  104004                          ERROR   4                       ;ERROR, COUNTER PRESET FAILED TO LOAD

                                ;;*****************************************************************
                                ;*TEST 4         TEST THAT PRESET BUFFER CAN HOLD #125
                                ;;*****************************************************************
002706  000004                  TST4:   SCOPE
002710  012737  000125  001124          MOV     #125,@#DDAT             ;LOAD EXPECTED
002716  013777  001124  176600          MOV     @#DDAT,@#CSR            ;LOAD PRESET BUFFER
002724  017737  176574  001126          MOV     @#CSR,@#DDAT            ;READ REG.
002732  023737  001124  001126          CMP     @#DDAT,@#DDAT           ;COMPARE
002740  001401                          BEQ     TST5            ;;BR IF EQUAL
002742  104004                          ERROR   4                       ;ERROR, COUNTER PRESET FAILED TO LOAD

                                ;;*****************************************************************
                                ;*TEST 5         TEST THAT PRESET BUFFER CAN HOLD #252
                                ;;*****************************************************************
002744  000004                  TST5:   SCOPE
002746  012737  000252  001124          MOV     #252,@#DDAT             ;LOAD EXPECTED
002754  013777  001124  176542          MOV     @#DDAT,@#CSR            ;LOAD PRESET
002762  017737  176536  001126          MOV     @#CSR,@#DDAT            ;READ REG.
002770  023737  001124  001126          CMP     @#DDAT,@#DDAT           ;COMPARE
002776  001401                          BEQ     TST6            ;;BR IF EQUAL
003000  104004                          ERROR   4                       ;ERROR, COUNTER PRESET FAILED TO LOAD
```

```
                                ;;*****************************************************************
                                ;*TEST 6         TEST THAT PRESET BUFFER CAN HOLD A COUNT PATTERN
                                ;;*****************************************************************
003002  000004                  TST6:   SCOPE
003004  012737  000100  001164          MOV     #100,STIMES     ;;DO 100 ITERATIONS
003012  012737  003026  001110          MOV     #15,$LPERR              ;LOAD ERROR SCOPE RETURN
003020  005037  001124                  CLR     $SDDAT                  ;CLEAR PATTERN
003024  013777  001124  176472  15$:    MOV     $SDDAT,@CSB             ;LOAD REG.
003032  017737  176466  001126          MOV     @CSB,$SDDAT             ;READ REG.
003040  023737  001124  001126          CMP     $SDDAT,$SDDAT           ;COMPARE
003046  001401                          BEQ     2$              ;;BR IF EQUAL
003050  104004                          ERROR   4                       ;PRESET BUFFER FAILED TO HOLD A COUNT PATTERN
                                
003052  105237  001124          2$:     INCB    $SDDAT                  ;UPDATE PATTERN
003056  001362                          BNE     1$                      ;BR UNTIL DONE
                                
                                ;;*****************************************************************
                                ;*TEST 7         TEST INIT TO CLEAR COUNT PRESET BUFFER WHEN IT IS 0-1
                                ;;*****************************************************************
003060  000004                  TST7:   SCOPE
003062  012737  000020  001164          MOV     #20,STIMES      ;;DO 20 ITERATIONS
003070  012737  000000  001124          MOV     #0,$SDDAT               ;LOAD EXPECTED
003076  012777  177777  176420          MOV     #-1,@CSB
003104  000005                          RESET
003106  017737  176412  001126          MOV     @CSB,$SDDAT             ;READ REG.
003114  001401                          BEQ     TST10           ;;BR IF EQUAL
003116  104004                          ERROR   4                       ;ERROR, INIT FAILED TO CLEAR CSB
                                
                                ;;*****************************************************************
                                ;*TEST 10        TEST THAT THE COUNTER CAN HOLD 0
                                ;;*****************************************************************
003120  000004                  TST10:  SCOPE
003122  012737  000000  001124          MOV     #0,$SDDAT               ;LOAD EXPECTED
003130  013777  001124  176366          MOV     $SDDAT,@CSB             ;LOAD PRESET BUFFER
003136  017737  176372  001126          MOV     @CSC,$SDDAT             ;READ COUNTER
003144  023737  001124  001126          CMP     $SDDAT,$SDDAT           ;COMPARE
003152  001401                          BEQ     TST11           ;;BR IF EQUAL
003154  104006                          ERROR   6                       ;ERROR, COUNTER FAILED TO CLEAR
                                
                                ;;*****************************************************************
                                ;*TEST 11        TEST THE COUNTER CAN HOLD 0377
                                ;;*****************************************************************
003156  000004                  TST11:  SCOPE
003160  012737  000377  001124          MOV     #377,$SDDAT             ;LOAD EXPECTED
003166  013777  001124  176330          MOV     $SDDAT,@CSB             ;LOAD PRESET BUFFER
003174  017737  176334  001126          MOV     @CSC,$SDDAT             ;READ COUNTER
003202  023737  001124  001126          CMP     $SDDAT,$SDDAT           ;COMPARE
003210  001401                          BEQ     TST12           ;;BR IF EQUAL
003212  104006                          ERROR   6                       ;ERROR, COUNTER FAILED TO LOAD
```

K2

```
                                ;;*****************************************************************
                                ;*TEST 12        TEST THAT COUNTER CAN HOLD 0125
                                ;;*****************************************************************
003214  000004          TST12:  SCOPE
003216  012737  000125  001124          MOV     #125,SGDDAT             ;LOAD EXPECTED
003224  013777  001124  176272          MOV     SGDDAT,@CSB             ;LOAD PRESET BUFFER
003232  017737  176276  001126          MOV     @CSC,SDDDAT             ;READ COUNTER
003240  023737  001124  001126          CMP     SGDDAT,SDDDAT           ;COMPARE
003246  001401                          BEQ     TST13           ;;BR IF EQUAL
003250  104006                          ERROR   6                       ;ERROR, COUNTER FAILED TO LOAD

                                ;;*****************************************************************
                                ;*TEST 13        TEST THAT COUNTER CAN HOLD 0252
                                ;;*****************************************************************
003252  000004          TST13:  SCOPE
003254  012737  000252  001124          MOV     #252,SGDDAT             ;LOAD EXPECTED
003262  013777  001124  176234          MOV     SGDDAT,@CSB             ;LOAD PRESET
003270  017737  176240  001126          MOV     @CSC,SDDDAT             ;READ COUNTER
003276  023737  001124  001126          CMP     SGDDAT,SDDDAT           ;COMPARE
003304  001401                          BEQ     TST14           ;;BR IF EQUAL
003306  104006                          ERROR   6                       ;ERROR, COUNTER FAILED TO LOAD

                                ;;*****************************************************************
                                ;*TEST 14        TEST THAT COUNTER CAN HOLD A COUNT PATTERN
                                ;;*****************************************************************
003310  000004          TST14:  SCOPE
003312  012737  000100  001164          MOV     #100,STIMES     ;;DO 100 ITERATIONS
003320  012737  003332  001110          MOV     #1$,$LPERR              ;LOAD ERROR SCOPE RETURN
003326  005037  001124                  CLR     SGDDAT                  ;CLEAR PATTERN
003332  013777  001124  176164  1$:     MOV     SGDDAT,@CSB             ;LOAD PRESET
003340  017737  176170  001126          MOV     @CSC,SDDDAT             ;READ COUNTER
003346  023737  001124  001126          CMP     SGDDAT,SDDDAT           ;COMPARE
003354  001401                          BEQ     2$              ;;BR IF EQUAL
003356  104006                          ERROR   6                       ;COUNTER FAILED TO HOLD A COUNT PATTERN

003360  105237  001124          2$:     INCB    SGDDAT                  ;UPDATE PATTERN
003364  001362                          BNE     1$                      ;BR UNTIL DONE

                                ;;*****************************************************************
                                ;*TEST 15        TEST INIT TO CLEAR COUNTER WHEN IT IS =-1
                                ;;*****************************************************************
003366  000004          TST15:  SCOPE
003370  012737  000020  001164          MOV     #20,STIMES      ;;DO 20 ITERATIONS
003376  012737  000000  001124          MOV     #0,SGDDAT               ;LOAD EXPECTED
003404  012777  177777  176112          MOV     #-1,@CSB
003412  000005                          RESET
003414  017737  176114  001126          MOV     @CSC,SDDDAT             ;READ COUNTER
003422  001401                          BEQ     TST16           ;;BR IF EQUAL
003424  104006                          ERROR   6                       ;ERROR, INIT FAILED TO CLEAR COUNTER
```

```
                                ;;*****************************************************************
                                ;*TEST 16        TEST ENABLE COUNTER (BIT 0) CAN BE SET
                                ;;*****************************************************************
(2)  003426  000004             TST16:  SCOPE
003430  012737  000021  001124          MOV     #BIT4!BIT0,$DDAT        ;LOAD EXPECTED
003436  013777  001124  176056          MOV     $DDAT,@CSR
003444  017737  176052  001126          MOV     @CSR,$DDAT              ;READ REG.
003452  023737  001124  001126          CMP     $DDAT,$DDAT             ;COMPARE
003460  001401                          BEQ     TST17           ;;BR IF EQUAL
003462  104003                          ERROR   3                       ;ERROR COUNTER ENABLE FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 17        TEST RATE SELECT (BIT 1) MAY BE SET
                                ;;*****************************************************************
(2)  003464  000004             TST17:  SCOPE
003466  012737  000022  001124          MOV     #BIT4!BIT1,$DDAT        ;LOAD EXPECTED
003474  013777  001124  176020          MOV     $DDAT,@CSR
003502  017737  176014  001126          MOV     @CSR,$DDAT              ;READ REG.
003510  023737  001124  001126          CMP     $DDAT,$DDAT             ;COMPARE
003516  001401                          BEQ     TST20           ;;BR IF EQUAL
003520  104003                          ERROR   3                       ;ERROR, RATE BIT 1 FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 20        TEST THAT RATE SELECT (BIT 2) MAY BE SET
                                ;;*****************************************************************
(2)  003522  000004             TST20:  SCOPE
003524  012737  000024  001124          MOV     #BIT4!BIT2,$DDAT        ;LOAD EXPECTED
003532  013777  001124  175762          MOV     $DDAT,@CSR
003540  017737  175756  001126          MOV     @CSR,$DDAT              ;READ REG.
003546  023737  001124  001126          CMP     $DDAT,$DDAT             ;COMPARE
003554  001401                          BEQ     TST21           ;;BR IF EQUAL
003556  104003                          ERROR   3                       ;ERROR, RATE BIT 2 FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 21        TEST THAT RATE SELECT (BIT 3) MAY BE SET
                                ;;*****************************************************************
(2)  003560  000004             TST21:  SCOPE
003562  012737  000030  001124          MOV     #BIT4!BIT3,$DDAT        ;LOAD EXPECTED
003570  013777  001124  175724          MOV     $DDAT,@CSR
003576  017737  175720  001126          MOV     @CSR,$DDAT              ;READ REG.
003604  023737  001124  001126          CMP     $DDAT,$DDAT             ;COMPARE
003612  001401                          BEQ     TST22           ;;BR IF EQUAL
003614  104003                          ERROR   3                       ;ERROR, RATE BIT 3 FAILED TO SET
```

M2

```
                                ;;*****************************************************************
                                ;*TEST 22        TEST CLOCK INTERRUPT ENABLE (BIT 6  CAN BE SET
                                ;;*****************************************************************
(2)  003616  000004             TST22:  SCOPE
003620  012737  000120  001124          MOV     #BIT6!BIT4,SGDDAT       ;LOAD EXPECTED
003626  013777  001124  175666          MOV     SGDDAT,@CSR
003634  017737  175662  001126          MOV     @CSR,SBDDAT             ;READ REG.
003642  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
003650  001401                          BEQ     TST23           ;;BR IF EQUAL
003652  104003                          ERROR   3                       ;ERROR, CLOCK INTERRUPT ENABLE FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 23        TEST MODE (BIT 8) CAN BE SET
                                ;;*****************************************************************
(2)  003654  000004             TST23:  SCOPE
003656  012737  000420  001124          MOV     #BIT8!BIT4,SGDDAT       ;LOAD EXPECTED
003664  013777  001124  175630          MOV     SGDDAT,@CSR
003672  017737  175624  001126          MOV     @CSR,SBDDAT             ;READ REG.
003700  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
003706  001401                          BEQ     TST24           ;;BR IF EQUAL
003710  104003                          ERROR   3                       ;ERROR, CSR NOT = 420

                                ;;*****************************************************************
                                ;*TEST 24        TEST EXT INTERRUPT ENABLE (BIT 14) CAN BE SET
                                ;;*****************************************************************
(2)  003712  000004             TST24:  SCOPE
003714  012737  040020  001124          MOV     #BIT14!BIT4,SGDDAT      ;LOAD EXPECTED
003722  013777  001124  175572          MOV     SGDDAT,@CSR
003730  017737  175566  001126          MOV     @CSR,SBDDAT             ;READ REG.
003736  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
003744  001401                          BEQ     TST25           ;;BR IF EQUAL
003746  104003                          ERROR   3                       ;ERROR, EXT INTERRUPT ENABLE FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 25        TEST THAT CLK DONE (BIT 7) CAN BE SET
                                ;;*****************************************************************
(2)  003750  000004             TST25:  SCOPE
003752  012737  000220  001124          MOV     #BIT7!BIT4,SGDDAT       ;LOAD EXPECTED
003760  013777  001124  175534          MOV     SGDDAT,@CSR             ;LOAD REG.
003766  017737  175530  001126          MOV     @CSR,SBDDAT             ;READ REG
003774  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
004002  001401                          BEQ     TST26           ;;BR IF EQUAL
004004  104003                          ERROR   3                       ;CLOCK DONE FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 26        TEST THAT CLK EXT INPUT (BIT 15) CAN BE SET
                                ;;*****************************************************************
(2)  004006  000004             TST26:  SCOPE
004010  012737  100020  001124          MOV     #BIT15!BIT4,SGDDAT      ;LOAD EXPECTED
004016  013777  001124  175476          MOV     SGDDAT,@CSR             ;LOAD REG
004024  017737  175472  001126          MOV     @CSR,SBDDAT             ;READ REG.
004032  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
004040  001401                          BEQ     TST27           ;;BR IF EQUAL
004042  104003                          ERROR   3                       ;CLOCK EXT INPUT FAILED TO SET
```

```
                        ;;*****************************************************************
                        ;*TEST 27        TEST THAT THE EXT FLAG DOES NOT SET FROM THE OUTSIDE SOURCE
                        ;;*****************************************************************
004044  000004          TST27:  SCOPE
004046  012737  000010  001164          MOV     #10,STIMES      ;;DO 10 ITERATIONS
004054  005077  175442                  CLR     @CSR
004060  012737  000020  001124          MOV     #BIT4,$DDAT             ;LOAD EXPECTED
004066  005037  001566                  CLR     TEMP
004072  017737  175424  001126  18:     MOV     @CSR,$DDAT              ;READ REG
004100  005737  001126                  TST     $DDAT                   ;TEST BIT15
004104  100001                          BPL     2$              ;;BR IF CLEARED
004106  104003                          ERROR   3                       ;ERROR EXT FLG SET IN ERROR
004110  005237  001566          2$:     INC     TEMP
004114  001366                          BNE     1$
                        ;;*****************************************************************
                        ;*TEST 30        MAINT, COUNT THE COUNTER REGISTER AT RATE 1MHZ
                        ;;*****************************************************************
004116  000004          TST30:  SCOPE
004120  012737  000004  001164          MOV     #4,STIMES       ;;DO 4 ITERATIONS
004126  005077  175370                  CLR     @CSR
004132  005077  175366                  CLR     @CSB
004136  012777  000002  175356          MOV     #2,@CSR                 ;SET UP 1MHZ RATE
004144  005037  001566                  CLR     TEMP                    ;CLEAR PATTERN
004150  013737  001566  001124  18:     MOV     TEMP,$GODAT             ;LOAD EXPECTED
004156  017737  175352  001126          MOV     @CSC,$DDAT              ;READ COUNTER
004164  023737  001124  001126          CMP     $GODAT,$DDAT            ;COMP
004172  001401                          BEQ     2$              ;;BR IF EQUAL
004174  104007                          ERROR   7                       ;ERROR, CLOCK COUNTER
                                                                        ;BUFFER COUNTED IN ERROR
004176  117737  175320  001126  28:     MOVB    @CSR,$DDAT              ;READ REG
004204  100005                          BPL     3$              ;;BR IF CLEAR
004206  012737  000022  001124          MOV     #22,$GODAT              ;LOAD EXPECTED STATUS
004214  104003                          ERROR   3                       ;CLK DONE SET IN ERROR
004216  000411                          BR      TST31           ;;BR TO SCOPE
004220  052777  004000  175274  38:     BIS     #BIT11,@CSR             ;MAINT COUNT
004226  042777  004000  175266          BIC     #BIT11,@CSR
004234  105237  001566                  INCB    TEMP
004240  001343                          BNE     1$                      ;BRANCH IF NOT FULL COUNT
                        ;;*****************************************************************
                        ;*TEST 31        TEST THAT OVERFLOW SET CLK DONE (BIT 7)
                        ;;*****************************************************************
004242  000004          TST31:  SCOPE
004244  005077  175252                  CLR     @CSR                    ;CLEAR STATUS
004250  012777  177777  175246          MOV     #-1,@CSB                ;LOAD PRESET
004256  012777  000002  175236          MOV     #BIT1,@CSR              ;LOAD 1 MHZ. RATE
004264  012737  000222  001124          MOV     #BIT7!BIT4!BIT1,$GODAT  ;LOAD EXPECTED
004272  052777  004000  175222          BIS     #BIT11,@CSR             ;MAINT COUNT
004300  042777  004000  175214          BIC     #BIT11,@CSR             ;CLEAR BIT
004306  017737  175210  001126          MOV     @CSR,$DDAT              ;READ REG.
004314  105737  001126                  TSTB    $DDAT                   ;TEST BIT 7
004320  100401                          BMI     TST32           ;;BR IF MINUS
004322  104003                          ERROR   3                       ;ERROR, OVERFLOW FAILED TO SET BIT 7
```

```
                                ;;*****************************************************************
                                ;*TEST 32        MAINT, COUNT THE COUNTER REGISTER AT RATE @100KHZ
                                ;;*****************************************************************
004324  000004          TST32:  SCOPE
004326  012737  000004  001164          MOV     #4,STIMES       ;;DO 4 ITERATIONS
004334  005077  175162          CLR     @CSR                    ;CLEAR CLOCK STATUS
004340  005077  175160          CLR     @CSB                    ;CLEAR PRESET
004344  005037  001124          CLR     $BDDAT
004350  012777  000004  175144  MOV     #4,@CSR                 ;LOAD STATUS, 100KHZ RATE
004356  005037  001124          CLR     $BDDAT
004362  017737  175146  001124  1$:     MOV     @CSC,$BDDAT             ;READ COUNTER
004370  023737  001124  001126          CMP     $BDDAT,$BDDAT           ;COMPARE
004376  001401                  BEQ     2$              ;;BR IF EQUAL
004400  104007                  ERROR   7               ;ERROR, CLOCK COUNTER BUFFER
                                                        ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE
                                                        ;CLOCK UP COUNT OR RATE SELECTION LOGIC
004402  012737  000012  001562  2$:     MOV     #10.,COUNT              ;LOAD COUNT
004410  052777  004000  175104  3$:     BIS     #BIT11,@CSR             ;MAINT COUNT
004416  042777  004000  175076          BIC     #BIT11,@CSR
004424  005337  001562          DEC     COUNT                   ;DONE
004430  001367                  BNE     3$                      ;BR IF NOT
004432  105237  001124          INCB    $BDDAT                  ;INCREMENT  EXPECTED VALUE
004436  001351                  BNE     1$
                                ;;*****************************************************************
                                ;*TEST 33        MAINT, COUNT THE COUNTER REGISTER AT RATE @10KHZ
                                ;;*****************************************************************
004440  000004          TST33:  SCOPE
004442  012737  000004  001164  1$:     MOV     #4,STIMES       ;;DO 4 ITERATIONS
004450  005077  175046          CLR     @CSR                    ;CLEAR CLOCK STATUS
004454  005077  175044          CLR     @CSB                    ;CLEAR PRESET
004460  012777  000006  175034  MOV     #6,@CSR                 ;LOAD STATUS, 10KHZ RATE
004466  005037  001124          CLR     $BDDAT
004472  017737  175036  001126  1$:     MOV     @CSC,$BDDAT             ;READ COUNTER
004500  023737  001124  001126          CMP     $BDDAT,$BDDAT           ;COMPARE
004506  001401                  BEQ     2$              ;;BR IF EQUAL
004510  104007                  ERROR   7               ;ERROR, CLOCK COUNTER BUFFER
                                                        ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE
                                                        ;CLOCK UP COUNT OR RATE SELECTION LOGIC
004512  012737  000144  001562  2$:     MOV     #100.,COUNT             ;LOAD COUNT
004520  052777  004000  174774  3$:     BIS     #BIT11,@CSR             ;MAINT COUNT
004526  042777  004000  174766          BIC     #BIT11,@CSR
004534  005337  001562          DEC     COUNT                   ;DONE
004540  001367                  BNE     3$                      ;BR IF NOT
004542  105237  001124          INCB    $BDDAT                  ;UPDATE PATTERN
004546  001351                  BNE     1$                      ;BR IF NOT
```

```
                                ;;*****************************************************************
                                ;*TEST 34        MAINT, COUNT THE COUNTER REGISTER AT RATE @1KHZ
                                ;;*****************************************************************
004550  000004                TST34:  SCOPE
004552  012737  000004  001164          MOV     #4,STIMES       ;;DO 4 ITERATIONS
004560  005077  174736                  CLR     @CSR                    ;CLEAR CLOCK STATUS
004564  005077  174734                  CLR     @CSB                    ;CLEAR PRESET
004570  012737  000001  001124          MOV     #1,SBDDAT               ;LOAD EXPECTED
004576  012777  000010  174716          MOV     #10,@CSR                ;LOAD STATUS, 1 KHZ RATE
004604  012737  001750  001562          MOV     #1000.,COUNT            ;SET UP A COUNTER
004612  052777  004000  174702  18:     BIS     #BIT11,@CSR             ;GENERATE MAINT COUNT
004620  042777  004000  174674          BIC     #BIT11,@CSR
004626  005337  001562                  DEC     COUNT
004632  001367                          BNE     18                      ;BR
004634  017737  174674  001126          MOV     @CSC,SBDDAT             ;READ COUNTER
004642  023737  001124  001126          CMP     SBDDAT,SBDDAT           ;COMPARE
004650  001401                          BEQ     TST35           ;;BR IF EQUAL
004652  104007                          ERROR   7                       ;ERROR, CLOCK COUNTER BUFFER
                                                                        ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE
                                                                        ;CLOCK UP COUNT OR RATE SELECTION LOGIC
                                ;;*****************************************************************
                                ;*TEST 35        MAINT, COUNT THE COUNTER REGISTER AT RATE @100HZ
                                ;;*****************************************************************
004654  000004                TST35:  SCOPE
004656  012737  000004  001164          MOV     #4,STIMES       ;;DO 4 ITERATIONS
004664  005077  174632                  CLR     @CSR                    ;CLEAR CLOCK STATUS, 100HZ RATE
004670  005077  174630                  CLR     @CSB                    ;CLEAR PRESET
004674  012737  000001  001124          MOV     #1,SBDDAT               ;LOAD EXPECTED
004702  012777  000012  174612          MOV     #12,@CSR                ;LOAD STATUS, 100 HZ
004710  012737  023420  001562          MOV     #10000.,COUNT           ;SET UP A COUNTER
004716  052777  004000  174576  18:     BIS     #BIT11,@CSR             ;GENERATE MAINT COUNT
004724  042777  004000  174570          BIC     #BIT11,@CSR
004732  005337  001562                  DEC     COUNT
004736  001367                          BNE     18                      ;BR
004740  017737  174570  001126          MOV     @CSC,SBDDAT             ;READ COUNTER
004746  023737  001124  001126          CMP     SBDDAT,SBDDAT           ;COMPARE
004754  001401                          BEQ     TST36           ;;BR IF EQUAL
004756  104007                          ERROR   7                       ;ERROR, CLOCK COUNTER BUFFER
                                                                        ;COUNTED IN ERROR, FAULT IS PROBABLY IN THE
                                                                        ;CLOCK UP COUNT OR RATE SELECTION LOGIC
```

```
                                ;;*****************************************************************
                                ;*TEST 36        TEST THAT RESET CLEARS RATE SELECT AND MODE BITS
                                ;;*****************************************************************
(2)  004760  000004             TST36:  SCOPE
(1)  004762  012737  000010  001164        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
549  004770  012777  000416  174524        MOV     #BIT8!BIT3!BIT2!BIT1,@#CSR      ;SET MODE BITS
550  004776  012737  000020  001124        MOV     #20,$SDDAT
551  005004  000005                        RESET
552  005006  017737  174510  001126        MOV     @#CSR,$SDDAT            ;READ REG.
553  005014  023737  001124  001126        CMP     $SDDAT,$SDDAT           ;COMPARE
554  005022  001401                        BEQ     TST37           ;;BR IF EQUAL
555  005024  104003                        ERROR   3                       ;ERROR, RESET FAILED TO CLEAR RATE OR MODE BITS
556
557                                 ;;*****************************************************************
(3)                                 ;*TEST 37        TEST THAT RESET CLEARS CLK INTERRUPT ENABLE
(3)                                 ;;*****************************************************************
(2)  005026  000004             TST37:  SCOPE
(1)  005030  012737  000010  001164        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
558  005036  012777  040100  174456        MOV     #BIT6!BIT14,@#CSR       ;SET CLK INT ENABLE
559  005044  012737  000020  001124        MOV     #20,$SDDAT
560  005052  000005                        RESET
561  005054  017737  174442  001126        MOV     @#CSR,$SDDAT
562  005062  023737  001124  001126        CMP     $SDDAT,$SDDAT
563  005070  001401                        BEQ     TST40           ;;BR IF EQUAL
564  005072  104003                        ERROR   3                       ;ERROR, RESET FAILED TO CLEAR CLK INT ENABLE
565
566                                 ;;*****************************************************************
(3)                                 ;*TEST 40        TEST THAT RESET CLEARS CLK FLAGS
(3)                                 ;;*****************************************************************
(2)  005074  000004             TST40:  SCOPE
(1)  005076  012737  000010  001164        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
567  005104  012777  100200  174410        MOV     #BIT7!BIT15,@#CSR       ;SET CLK FLAGS
568  005112  012737  000020  001124        MOV     #20,$SDDAT
569  005120  000005                        RESET
570  005122  017737  174374  001126        MOV     @#CSR,$SDDAT
571  005130  023737  001124  001126        CMP     $SDDAT,$SDDAT
572  005136  001401                        BEQ     TST41           ;;BR IF EQUAL
573  005140  104003                        ERROR   3                       ;ERROR, RESET FAILED TO CLEAR CLK FLAGS
574
575                                 ;;*****************************************************************
(3)                                 ;*TEST 41        TEST THAT RESET CLEARS COUNTER ENABLE
(3)                                 ;;*****************************************************************
(2)  005142  000004             TST41:  SCOPE
(1)  005144  012737  000010  001164        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
576  005152  012777  000000  174344        MOV     #0,@#CSR                ;CLEAR COUNTER
577  005160  012777  000001  174334        MOV     #BIT0,@#CSR             ;LOAD COUNTER ENABLE
578  005166  012737  000020  001126        MOV     #20,$SDDAT
579  005174  000005                        RESET
580  005176  017737  174320  001126        MOV     @#CSR,$SDDAT
581  005204  023737  001124  001126        CMP     $SDDAT,$SDDAT
582  005212  001401                        BEQ     TST42           ;;BR IF EQUAL
583  005214  104003                        ERROR   3                       ;ERROR, RESET FAILED TO CLEAR COUNTER ENABLE
584
585
```

```
587                                     ;;*****************************************************************
(3)                                     ;*TEST 42        TEST CLOCK TO COUNT UP AT 1 MHZ
(3)                                     ;;*****************************************************************
(2)  005216  000004                     TST42:  SCOPE
(1)  005220  012737  000100  001164             MOV     #100,STIMES     ;;DO 100 ITERATIONS
588  005226  012737  000003  014056             MOV     #3,RATE                 ;SELECT MODE 0, 1MHZ., GO
589  005234  012737  000222  001124             MOV     #BIT7!BIT4!BIT1,SGDDAT
590  005242  004737  013376                     JSR     PC,UPCNT
591  005246  017737  174250  001126             MOV     @CSR,SBDDAT             ;READ REG
592  005254  023737  001124  001126             CMP     SGDDAT,SBDDAT           ;COMPARE
593  005262  001401                             BEQ     1$              ;;BR IF EQUAL
594  005264  104003                             ERROR   3                       ;ERROR, 1MHZ RATE FAILED TO SET DONE
595  005266  012737  000000  001124  1$:       MOV     #0,SGDDAT
596  005274  017737  174234  001126             MOV     @CSC,SBDDAT
597  005302  023737  001124  001126             CMP     SGDDAT,SBDDAT
598  005310  001401                             BEQ     TST43           ;;BR IF EQUAL
599  005312  104006                             ERROR   6                       ;IN MODE 0 CLOCK COUNTER WAS
600                                                                             ;LOADED ON CLK OVERFLOW
601
602                                     ;;*****************************************************************
(3)                                     ;*TEST 43        TEST CLOCK TO COUNT UP AT 100KHZ
(3)                                     ;;*****************************************************************
(2)  005314  000004                     TST43:  SCOPE
(1)  005316  012737  000100  001164             MOV     #100,STIMES     ;;DO 100 ITERATIONS
603  005324  012737  000005  014056             MOV     #5,RATE                 ;SELECT MODE 0, 100KHZ., GO
604  005332  012737  000224  001124             MOV     #BIT7!BIT4!BIT2,SGDDAT
605  005340  004737  013376                     JSR     PC,UPCNT
606  005344  017737  174152  001126             MOV     @CSR,SBDDAT             ;READ REG
607  005352  023737  001124  001126             CMP     SGDDAT,SBDDAT           ;COMPARE
608  005360  001401                             BEQ     1$              ;;BR IF EQUAL
609  005362  104003                             ERROR   3                       ;ERROR, 100KHZ. RATE FAILED TO SET DONE
610  005364  012737  000000  001124  1$:       MOV     #0,SGDDAT
611  005372  017737  174136  001126             MOV     @CSC,SBDDAT
612  005400  023737  001124  001126             CMP     SGDDAT,SBDDAT
613  005406  001401                             BEQ     TST44           ;;BR IF EQUAL
614  005410  104006                             ERROR   6                       ;IN MODE 0 CLOCK COUNTER WAS
615                                                                             ;LOADED ON CLK OVERFLOW
616
617                                     ;;*****************************************************************
(3)                                     ;*TEST 44        TEST CLOCK TO COUNT UP AT 10 KHZ
(3)                                     ;;*****************************************************************
(2)  005412  000004                     TST44:  SCOPE
(1)  005414  012737  000100  001164             MOV     #100,STIMES     ;;DO 100 ITERATIONS
618  005422  012737  000007  014056             MOV     #7,RATE                 ;SELECT MODE 0, 10KHZ., GO
619  005430  012737  000226  001124             MOV     #BIT7!BIT4!BIT2!BIT1,SGDDAT
620  005436  004737  013376                     JSR     PC,UPCNT
621  005442  017737  174054  001126             MOV     @CSR,SBDDAT             ;READ REG
622  005450  023737  001124  001126             CMP     SGDDAT,SBDDAT           ;COMPARE
623  005456  001401                             BEQ     TST45           ;;BR IF EQUAL
624  005460  104007                             ERROR   7                       ;ERROR, 10 KHZ. FAILED TO SET DONE
625
```

```
                                ;;*****************************************************************
                                ;*TEST 45        TEST CLOCK TO COUNT UP AT 1KHZ
                                ;;*****************************************************************
005462  000004          TST45:  SCOPE
005464  012737  000100  001164          MOV     #100,STIMES     ;;DO 100 ITERATIONS
005472  012737  000011  014056          MOV     #11,RATE                ;SELECT MODE 0, 1 KHZ, GO
005500  012737  000230  001124          MOV     #BIT7!BIT4!BIT3,SGDDAT
005506  004737  013376                  JSR     PC,UPCNT
005512  017737  174004  001126          MOV     @CSR,SBDDAT             ;READ REG
005520  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
005526  001401                          BEQ     TST46           ;;BR IF EQUAL
005530  104007                          ERROR   7                       ;ERROR, 1KHZ FAILED TO SET DONE

                                ;;*****************************************************************
                                ;*TEST 46        TEST CLOCK TO COUNT UP AT 100HZ
                                ;;*****************************************************************
005532  000004          TST46:  SCOPE
005534  012737  000100  001164          MOV     #100,STIMES     ;;DO 100 ITERATIONS
005542  012737  000013  014056          MOV     #13,RATE                ;SELECT MODE 0, 100 HZ, GO
005550  012737  000232  001124          MOV     #BIT7!BIT4!BIT3!BIT1,SGDDAT     ;LOAD EXPECTED
005556  004737  013376                  JSR     PC,UPCNT
005562  017737  173734  001126          MOV     @CSR,SBDDAT             ;READ REG.
005570  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
005576  001401                          BEQ     TST47           ;;BR IF EQUAL
005600  104007                          ERROR   7                       ;ERROR, 100HZ FAILED TO SET DONE

                                ;;*****************************************************************
                                ;*TEST 47        TEST THAT CLOCK ENABLE DOES NOT CLEAR ON DONE (MODE 1) 1 KHZ
                                ;;*****************************************************************
005602  000004          TST47:  SCOPE
005604  012737  000100  001164          MOV     #100,STIMES     ;;DO 100 ITERATIONS
005612  012737  000411  014056          MOV     #411,RATE               ;SELECT MODE 1, 1KHZ., GO
005620  012737  000031  001124          MOV     #31,SBDDAT              ;LOAD EXP
005626  004737  013376                  JSR     PC,UPCNT
005632  017737  173664  001126          MOV     @CSR,SBDDAT             ;READ REG
005640  042777  000001  173654          BIC     #BIT0,@CSR              ;CLEAR ENABLE
005646  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
005654  001401                          BEQ     1$              ;;BR IF EQUAL
005656  104007                          ERROR   7                       ;ERROR, 1 KHZ, FAILED TO OVERFLOW
                                                                        ; AND CONTINUE COUNTING
005660  012737  000376  001124  1$:      MOV     #376,SBDDAT
005666  017737  173642  001126          MOV     @CSC,SBDDAT
005674  023737  001124  001126          CMP     SBDDAT,SBDDAT
005702  001401                          BEQ     TST50           ;;BR IF EQUAL
005704  104006                          ERROR   6                       ;IN MODE 1, CLOCK COUNTER FAILED
                                                                        ;TO BE RE-LOADED ON CLOCK OVERFLOW
```

```
                                ;;*****************************************************************
                                ;*TEST 50        TEST THAT CLOCK ENABLE DOES NOT CLEAR DONE (MODE 1 100HZ)
                                ;;*****************************************************************
                        TST50:  SCOPE
 005706  000004                 
 005710  012737  000100  001164         MOV     #100,STIMES     ;;DO 100 ITERATIONS
 005716  012737  000413  014056         MOV     #413,RATE               ;MODE 1 100HZ GO
 005724  012737  000633  001124         MOV     #633,SGDDAT             ;LOAD EXPECTED
 005732  004737  013376                 JSR     PC,UPCNT
 005736  017737  173560  001126         MOV     @CSR,SBDDAT             ;READ REG
 005744  042777  000001  173550         BIC     #BIT0,@CSR              ;CLEAR ENABLE
 005752  023737  001124  001126         CMP     SGDDAT,SBDDAT           ;COMPARE
 005760  001401                         BEQ     1$              ;;BR IF EQUAL
 005762  104007                         ERROR   7                       ;ERROR, 100HZ FAILED TO
                                                                 ; OVERFLOW AND CONTINUE COUNTING
 005764  012737  000376  001124  1$:    MOV     #376,SGDDAT
 005772  017737  173536  001126         MOV     @CSC,SBDDAT
 006000  023737  001124  001126         CMP     SGDDAT,SBDDAT
 006006  001401                         BEQ     TST51           ;;BR IF EQUAL
 006010  104006                         ERROR   6                       ;IN MODE 1, CLOCK COUNTER FAILED
                                                                 ;TO BE RE-LOADED ON CLOCK OVERFLOW
                                ;;*****************************************************************
                                ;*TEST 51        CLOCK PRE-INTERRUPT SETUP
                                ;;*****************************************************************
                        TST51:  SCOPE
 006012  000004                 
 006014  012737  000001  001164         MOV     #1,STIMES       ;;DO 1 ITERATION
 006022  042737  177437  001512         BIC     #177437,CKBRL
 006030  001001                         BNE     .+4
 006032  000000                         HALT                    ;ERROR, BR LEVEL INDICATED FOR CLOCK WAS 0
 006034  022737  000340  001512         CMP     #340,CKBRL
 006042  001001                         BNE     .+4
 006044  000000                         HALT                    ;ERROR BR LEVEL FOR CLOCK WAS 7

 006046  013737  001512  001572         MOV     CKBRL,BRLEV1
 006054  162737  000040  001572         SUB     #40,BRLEV1
 006062  013737  001512  001574         MOV     CKBRL,BRLEV2
 006070  000005                         RESET
```

```
                        ;;*****************************************************************
                        ;*TEST 52        TEST THAT THE CLOCK INTERRUPTS AT LEVEL INDICATED -1
                        ;;*****************************************************************
(2)  006072  000004     TST52:  SCOPE
696  006074  012777  006160  173440          MOV     #28,@KWIV               ;LOAD VECTOR
697  006102  005077  173414                  CLR     @CSR
698  006106  012777  177777  173410          MOV     #-1,@CSB                ;LOAD PRESET
699  006114  012777  000102  173400          MOV     #BIT6!BIT1,@CSR         ;LOAD RATE AND INT ENABLE
700  006122  012700  002000                  MOV     #2000,R0                        ;SET UP DELAY
701  006126  013737  001572  177776          MOV     @RLEV1,PSW
702  006134  052777  004000  173360          BIS     #BIT11,@CSR             ;CLOCK
703  006142  042777  004000  173352          BIC     #BIT11,@CSR
704  006150  005300          1$:     DEC     R0
705  006152  001376                  BNE     1$                      ;DELAY
706  006154  104005                  ERROR   5                       ;ERROR, CLOCK FAILED TO INTERRUPT
707  006156  000401                  BR      TST53           ;;NEXT TEST
708  006160  022626          2$:     CMP     (SP)+,(SP)+
709
710                     ;;*****************************************************************
                        ;*TEST 53        TEST THAT THE CLOCK DOES NOT INTERRUPT AT LEVEL INDICATED
                        ;;*****************************************************************
(2)  006162  000004     TST53:  SCOPE
711  006164  012777  006246  173350          MOV     #3$,@KWIV               ;SET UP INTERRUPT RETURN STATUS
712  006172  012777  000000  173344          MOV     #0,@KWIVS
713  006200  005077  173316                  CLR     @CSR
714  006204  012777  177774  173312          MOV     #-4,@CSB
715  006212  012777  000102  173302          MOV     #BIT6!BIT1,@CSR         ;ENABLE INTERRUPT
716  006220  013737  001574  177776          MOV     @RLEV2,PSW
717  006226  052777  000001  173266          BIS     #BIT0,@CSR
718  006234  012700  001000                  MOV     #1000,R0
719  006240  005300                  DEC     R0
720  006242  001376                  BNE     .-2
721  006244  000403                  BR      2$              ;;BR TO NEXT TEST
722  006246  104005          1$:     ERROR   5                       ;ERROR, INTERRUPT OCCURED IN ERROR
723  006250  022626                  CMP     (SP)+,(SP)+             ;AT LEVEL INDICATED
724  006252  000421                  BR      TST54           ;;BR TO SCOPE
725
726
727                     ;SUBTEST, TEST THAT IF PRIORITY IS LOWERED AGAIN
728                     ;NO INTERRUPT SHOULD OCCUR
729
730  006254  012777  006302  173260  2$:     MOV     #3$,@KWIV               ;RESET THE VECTOR
731  006262  012700  000010                  MOV     #10,R0
732  006266  005037  177776                  CLR     PSW
733  006272  005300                  DEC     R0
734  006274  001376                  BNE     .-2                     ;DELAY
735  006276  104005                  ERROR   5                       ;ERROR, INIT DONE FAILED TO CLEAR INT REQ.
736  006300  000406                  BR      TST54           ;;BR TO NEXT TEST
737  006302  022626          3$:     CMP     (SP)+,(SP)+
738  006304  013777  001544  173238          MOV     KWIVS,@KWIV
739  006312  005077  173226                  CLR     @KWIVS
```

```
                                        ;;*********************************************************
                                        ;*TEST 54        TEST 1MHZ REPEATABILITY
                                        ;;*********************************************************
000316  000004                  TST54:  SCOPE
000320  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
000326  004537  013676                  JSR     R5,REPEAT               ;TEST REPEATIBILITY
000332  000002                          2                               ;CLOCK RATE, 1MHZ
000334  000002                          2                               ;CLOCK DEV.
000336  000001                          1                               ;MIN. COUNT
000340  000036                          30.                             ;DELAY
000342  104007                          ERROR   7                       ;ERROR, FAILED TO REACH MIN. COUNT
000344  003401                          BLE     TST55           ;;BR IF LESS OR EQUAL
000346  104010                          ERROR   10                      ;ERROR, CLOCK REPEATIBILITY >2
                                                        ;TRY CHANGING I.C. "E98" FROM 8648 TO DEC 388
                                        ;;*********************************************************
                                        ;*TEST 55        TEST 100 KHZ REPEATABILITY
                                        ;;*********************************************************
000350  000004                  TST55:  SCOPE
000352  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
000360  004537  013676                  JSR     R5,REPEAT               ;TEST REPEATIBILITY
000364  000004                          4                               ;CLOCK RATE, 100KHZ
000366  000002                          2                               ;CLOCK DEV.
000370  000001                          1                               ;MIN. COUNT .
000372  000456                          300.                            ;DELAY
000374  104007                          ERROR   7                       ;ERROR, FAILED TO REACH MIN. COUNT
000376  003401                          BLE     TST56           ;;BR IF LESS OR EQUAL
000400  104010                          ERROR   10                      ;ERROR, CLOCK REPEATIBILITY >2
                                        ;;*********************************************************
                                        ;*TEST 56        TEST 10 KHZ REPEATABILITY
                                        ;;*********************************************************
000402  000004                  TST56:  SCOPE
000404  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
000412  004537  013676                  JSR     R5,REPEAT
000416  000006                          6                               ;CLOCK RATE, 10KHZ
000420  000002                          2                               ;CLOCK DEV.
000422  000001                          1                               ;MIN. COUNT
000424  005670                          3000.                           ;DELAY
000426  104007                          ERROR   7                       ;ERROR, FAILED TO REACH MIN. COUNT
000430  003401                          BLE     TST57           ;;BR IF LESS OR EQUAL
000432  104010                          ERROR   10                      ;ERROR CLOCK REPEATABILITY >2
                                        ;;*********************************************************
                                        ;*TEST 57        TEST 1KHZ REPEATABILITY
                                        ;;*********************************************************
000434  000004                  TST57:  SCOPE
000436  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
000444  004537  013676                  JSR     R5,REPEAT               ;
000450  000010                          10                              ;CLOCK RATE, 1KHZ
000452  000002                          2                               ;CLOCK DEV.
000454  000001                          1                               ;MIN. COUNT
000456  072460                          30000.                          ;DELAY
000460  104007                          ERROR   7                       ;ERROR, FAILED TO REACH MIN. COUNT
000462  003401                          BLE     TST60           ;;BR IF LESS OR EQUAL
000464  104010                          ERROR   10                      ;ERROR CLOCK REPEATABILITY >2
```

```
                                ;;*****************************************************************
                                ;*TEST 60        TEST 100HZ REPEATIBILITY
                                ;;*****************************************************************
                        TST60:  SCOPE
006466  000004                 
006470  012737  000100  001164          MOV     #100,STIMES     ;;DO 100 ITERATIONS
006476  004537  013676                  JSR     R5,REPEAT               ;
006502  000012                          12                              ;CLOCK RATE, 100HZ
006504  000002                          2                               ;CLOCK DEV.
006506  000001                          1                               ;MIN. COUNT
006510  177777                          -1                              ;DELAY
006512  104007                          ERROR   7                       ;ERROR, FAILED TO REACH MIN. COUNT
006514  003401                          BLE     TST61           ;;BR IF LESS OR EQUAL
006516  104010                          ERROR   10                      ;ERROR CLOCK REPEATABILITY >2

                                ;;*****************************************************************
                                ;*TEST 61        TEST THAT RESET SETS VC READY BIT
                                ;;*****************************************************************
                        TST61:  SCOPE
006520  000004                 
006522  012737  000010  001164          MOV     #10,STIMES      ;;DO 10 ITERATIONS
006530  012737  000200  001124          MOV     #BIT7,SGDDAT            ;LOAD EXPECTED
006536  000005                          RESET
006540  017737  172762  001126          MOV     @VCSTAT,SBDDAT          ;READ REGISTER
006546  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
006554  001401                          BEQ     TST62           ;;BR IF SET
006556  104011                          ERROR   11                      ;RESET FAILED TO SET READY

                                ;;*****************************************************************
                                ;*TEST 62        TEST THAT VC MODE BIT 2 CAN BE SET AND CLEARED
                                ;;*****************************************************************
                        TST62:  SCOPE
006560  000004                 
006562  012777  000004  172736          MOV     #BIT2,@VCSTAT           ;LOAD DISPLAY STATUS
006570  012737  000204  001124          MOV     #BIT7!BIT2,SGDDAT       ;LOAD EXPECTED
006576  017737  172724  001126          MOV     @VCSTAT,SBDDAT          ;READ REG
006604  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
006612  001401                          BEQ     1$                      ;;BR IF EQUAL
006614  104011                          ERROR   11                      ;ERROR, VC STATUS NOT = 204
006616  005077  172704          1$:     CLR     @VCSTAT                 ;CLEAR STATUS
006622  012737  000200  001124          MOV     #BIT7,SGDDAT            ;LOAD EXPECTED
006630  017737  172672  001126          MOV     @VCSTAT,SBDDAT          ;READ REG
006636  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
006644  001401                          BEQ     TST63           ;;BR IF CLEARED
006646  104011                          ERROR   11                      ;MODE FAILED TO CLEAR

                                ;;*****************************************************************
                                ;*TEST 63        TEST THAT VC MODE BIT 3 CAN BE SET
                                ;;*****************************************************************
                        TST63:  SCOPE
006650  000004                 
006652  012777  000010  172646          MOV     #BIT3,@VCSTAT           ;LOAD
006660  012737  000210  001124          MOV     #BIT7!BIT3,SGDDAT       ;LOAD EXPECTED
006666  017737  172634  001126          MOV     @VCSTAT,SBDDAT          ;READ REG
006674  023737  001124  001126          CMP     SGDDAT,SBDDAT           ;COMPARE
006702  001401                          BEQ     TST64           ;;BR IF EQUAL
006704  104011                          ERROR   11                      ;ERROR, VC STATUS NOT = 210
```

```
                                ;;*****************************************************************
                                ;*TEST 64        TEST THAT VC INTERRUPT ENABLE (BIT 6) CAN BE SET
                                ;;*****************************************************************
(2)  006706  000004             TST64:  SCOPE
006710  012777  000100  172610          MOV     #BIT6,@VCSTAT           ;LOAD DISPLAY STATUS
006716  012737  000300  001124          MOV     #BIT7!BIT6,$GDDAT       ;LOAD EXPECTED
006724  017737  172576  001126          MOV     @VCSTAT,$BDDAT          ;READ REG
006732  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
006740  001401                          BEQ     TST65           ;;BR IF EQUAL
006742  104011                          ERROR   11                      ;ERROR, VC STATUS NOT = 300

                                ;;*****************************************************************
                                ;*TEST 65        TEST THAT CHANNEL (BIT 9) CAN BE SET
                                ;;*****************************************************************
(2)  006744  000004             TST65:  SCOPE
006746  012777  001000  172552          MOV     #BIT9,@VCSTAT           ;LOAD DISPLAY STATUS
006754  012737  001200  001124          MOV     #BIT9!BIT7,$GDDAT       ;LOAD EXPECTED
006762  017737  172540  001126          MOV     @VCSTAT,$BDDAT          ;READ REG
006770  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
006776  001401                          BEQ     TST66           ;;BR IF EQUAL
007000  104011                          ERROR   11                      ;ERROR, VC STATUS NOT = 1200

                                ;;*****************************************************************
                                ;*TEST 66        TEST THAT STORE (BIT 10) CAN BE SET
                                ;;*****************************************************************
(2)  007002  000004             TST66:  SCOPE
007004  012777  002000  172514          MOV     #BIT10,@VCSTAT          ;LOAD DISPLAY STATUS
007012  012737  002200  001124          MOV     #BIT10!BIT7,$GDDAT      ;LOAD EXPECTED
007020  017737  172502  001126          MOV     @VCSTAT,$BDDAT          ;READ REG
007026  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007034  001401                          BEQ     TST67           ;;BR IF EQUAL
007036  104011                          ERROR   11                      ;ERROR, VC STATUS NOT = 2200

                                ;;*****************************************************************
                                ;*TEST 67        TEST THAT WRITE THRU (BIT 11) CAN BE SET
                                ;;*****************************************************************
(2)  007040  000004             TST67:  SCOPE
007042  012777  004000  172456          MOV     #BIT11,@VCSTAT          ;LOAD DISPLAY STATUS
007050  012737  004200  001124          MOV     #BIT11!BIT7,$GDDAT      ;LOAD EXPECTED
007056  017737  172444  001126          MOV     @VCSTAT,$BDDAT          ;READ REG
007064  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007072  001401                          BEQ     TST70           ;;BR IF EQUAL
007074  104011                          ERROR   11                      ;ERROR, VC STATUS NOT = 4200

                                ;;*****************************************************************
                                ;*TEST 70        TEST THAT ERASE (BIT 12) CAN BE SET
                                ;;*****************************************************************
(2)  007076  000004             TST70:  SCOPE
007100  012777  010000  172420          MOV     #BIT12,@VCSTAT          ;SET ERASE BIT
007106  012737  010000  001124          MOV     #BIT12,$GDDAT           ;LOAD EXPECTED
007114  017737  172406  001126          MOV     @VCSTAT,$BDDAT          ;READ REG
007122  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007130  001401                          BEQ     18              ;;BR IF SET
007132  104011                          ERROR   11                      ;ERROR, ERASE BIT FAILED TO SET
007134  005077  172366          18:     CLR     @VCSTAT
```

```
                        ;;*****************************************************************
                        ;*TEST 71        TEST THAT THE X REGISTER CAN BE CLEARED
                        ;;*****************************************************************
007140  000004          TST711: SCOPE
007142  012737  000000  001124          MOV     #0,$SDDAT               ;LOAD EXPECTED
007150  013777  001124  172352          MOV     $SDDAT,@VCXREG          ;LOAD REG
007156  017737  172346  001126          MOV     @VCXREG,$SDDAT          ;READ REG
007164  023737  001124  001126          CMP     $SDDAT,$SDDAT           ;COMPARE
007172  001401                          BEQ     TST72           ;;BR IF EQUAL
007174  104012                          ERROR   12                      ;ERROR, VC XREGISTER NOT = 0

                        ;;*****************************************************************
                        ;*TEST 72        TEST THAT THE X REGISTER CAN BE LOADED WITH 01777
                        ;;*****************************************************************
007176  000004          TST72:  SCOPE
007200  012737  001777  001124          MOV     #1777,$SDDAT            ;LOAD EXPECTED
007206  013777  001124  172314          MOV     $SDDAT,@VCXREG          ;LOAD REG
007214  017737  172310  001126          MOV     @VCXREG,$SDDAT          ;READ REG
007222  023737  001124  001126          CMP     $SDDAT,$SDDAT           ;COMPARE
007230  001401                          BEQ     TST73           ;;BR IF EQUAL
007232  104012                          ERROR   12                      ;ERROR, VC X REGISTER NOT = 1777

                        ;;*****************************************************************
                        ;*TEST 73        TEST THAT THE X REGISTER CAN BE LOADED WITH 0525
                        ;;*****************************************************************
007234  000004          TST731: SCOPE
007236  012737  000525  001124          MOV     #525,$SDDAT             ;LOAD EXPECTED
007244  013777  001124  172256          MOV     $SDDAT,@VCXREG          ;LOAD REG.
007252  017737  172252  001126          MOV     @VCXREG,$SDDAT          ;READ REG
007260  023737  001124  001126          CMP     $SDDAT,$SDDAT           ;COMPARE
007266  001401                          BEQ     TST74           ;;BR IF EQUAL
007270  104012                          ERROR   12                      ;ERROR, VC X REGISTER NOT = 525

                        ;;*****************************************************************
                        ;*TEST 74        TEST THAT THE X REGISTER CAN BE LOADED WITH 01252
                        ;;*****************************************************************
007272  000004          TST741: SCOPE
007274  012737  001252  001124          MOV     #1252,$SDDAT            ;LOAD EXPECTED
007302  013777  001124  172220          MOV     $SDDAT,@VCXREG          ;LOAD REG.
007310  017737  172214  001126          MOV     @VCXREG,$SDDAT          ;READ REG
007316  023737  001124  001126          CMP     $SDDAT,$SDDAT           ;COMPARE
007324  001401                          BEQ     TST75           ;;BR IF EQUAL
007326  104012                          ERROR   12                      ;ERROR, VC X REGISTER NOT = 1252
```

M3

```
                                        ;;*********************************************************************
(3)                                     ;*TEST 75        TEST THAT THE X REGISTER CAN HOLD A COUNT PATTERN
(3)                                     ;;*********************************************************************
(2)  007330  000004                     TST75:  SCOPE
(1)  007332  012737  000100  001144             MOV     #100,STIMES     ;;DO 100 ITERATIONS
893  007340  012737  007352  001110             MOV     #1$,SLPERR              ;LOAD SCOPE ERROR RETURN
894  007346  005037  001124                     CLR     SGDDAT                  ;CLEAR EXPECTED
895  007352  013777  001124  172150     1$:     MOV     SGDDAT,@VCXREG          ;LOAD REG
896  007360  017737  172144  001126             MOV     @VCXREG,SBDDAT          ;READ REG
897  007366  023737  001124  001126             CMP     SGDDAT,SBDDAT           ;COMPARE
898  007374  001401                             BEQ     2$              ;;BR IF EQUAL
899  007376  104012                             ERROR   12                      ;VC X REG FAILED TO HOLD A COUNT PATTERN
900
901  007400  005237  001124             2$:     INC     SGDDAT                  ;UPDATE PATTERN
902  007404  022737  002000  001124             CMP     #2000,SGDDAT            ;FINISHED?
903  007412  001357                             BNE     1$                      ;BR IF NOT
904
905                                     ;;*********************************************************************
(3)                                     ;*TEST 76        TEST THAT THE Y REGISTER CAN BE CLEARED
(3)                                     ;;*********************************************************************
(2)  007414  000004                     TST76:  SCOPE
906  007416  012737  000000  001124             MOV     #0,SGDDAT               ;LOAD EXPECTED
907  007424  013777  001124  172100             MOV     SGDDAT,@VCYREG          ;LOAD Y REG
908  007432  017737  172074  001126             MOV     @VCYREG,SBDDAT          ;READ REG
909  007440  023737  001124  001126             CMP     SGDDAT,SBDDAT           ;COMPARE
910  007446  001401                             BEQ     TST77           ;;BR IF EQUAL
911  007450  104013                             ERROR   13                      ;ERROR, VC Y REGISTER NOT = 0
912
913                                     ;;*********************************************************************
(3)                                     ;*TEST 77        TEST THAT THE Y REGISTER CAN BE LOADED WITH #1777
(3)                                     ;;*********************************************************************
(2)  007452  000004                     TST77:  SCOPE
914  007454  012737  001777  001124             MOV     #1777,SGDDAT            ;LOAD EXPECTED
915  007462  013777  001124  172042             MOV     SGDDAT,@VCYREG          ;LOAD Y REG
916  007470  017737  172036  001126             MOV     @VCYREG,SBDDAT          ;READ REG
917  007476  023737  001124  001126             CMP     SGDDAT,SBDDAT           ;COMPARE
918  007504  001401                             BEQ     TST100          ;;BR IF EQUAL
919  007506  104013                             ERROR   13                      ;ERROR, VC Y REGISTER NOT = 1777
920
921                                     ;;*********************************************************************
(3)                                     ;*TEST 100       TEST THAT THE Y REGISTER CAN BE LOADED WITH #525
(3)                                     ;;*********************************************************************
(2)  007510  000004                     TST100: SCOPE
922  007512  012737  000525  001124             MOV     #525,SGDDAT             ;LOAD EXPECTED
923  007520  013777  001124  172004             MOV     SGDDAT,@VCYREG          ;LOAD Y REG
924  007526  017737  172000  001126             MOV     @VCYREG,SBDDAT          ;READ REG
925  007534  023737  001124  001126             CMP     SGDDAT,SBDDAT           ;COMPARE
926  007542  001401                             BEQ     TST101          ;;BR IF EQUAL
927  007544  104013                             ERROR   13                      ;ERROR, VC Y REGISTER NOT = 525
928
```

```
                        ;;*****************************************************************
                        ;*TEST 101      TEST THAT THE V REGISTER CAN BE LOADED WITH #1252
                        ;;*****************************************************************
007546  000004          TST101: SCOPE
007550  012737  001252  001124          MOV     #1252,$GDDAT            ;LOAD EXPECTED
007556  013777  001124  171746          MOV     $GDDAT,@VCVREG          ;LOAD V REG
007564  017737  171742  001126          MOV     @VCVREG,$BDDAT          ;READ REG
007572  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007600  001401                        BEQ     TST102          ;;BR IF EQUAL
007602  104013                        ERROR   13                      ;ERROR, VC V REGISTER NOT = 1252

                        ;;*****************************************************************
                        ;*TEST 102      TEST THAT THE V REGISTER CAN HOLD A COUNT PATTERN
                        ;;*****************************************************************
007604  000004          TST102: SCOPE
007606  012737  000100  001164          MOV     #100,$TIMES     ;;DO 100 ITERATIONS
007614  012737  007626  001110          MOV     #18,$LPERR              ;LOAD SCOPE ERROR RETURN
007622  005037  001124                  CLR     $GDDAT                  ;CLEAR EXPECTED
007626  013777  001124  171676  181:    MOV     $GDDAT,@VCVREG          ;LOAD REG
007634  017737  171672  001126          MOV     @VCVREG,$BDDAT          ;READ REG
007642  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007650  001401                        BEQ     28              ;;BR IF EQUAL
007652  104013                        ERROR   13                      ;VC V REG FAILED TO HOLD A COUNT PATTERN

007654  005237  001124          281:    INC     $GDDAT                  ;UPDATE PATTERN
007660  022737  002000  001124          CMP     #2000,$GDDAT            ;FINISHED?
007666  001357                        BNE     18                      ;BR IF NOT

                        ;;*****************************************************************
                        ;*TEST 103      TEST THAT THE X-V REGISTERS CAN HOLD DIFFERENT DATA
                        ;;*****************************************************************
007670  000004          TST103: SCOPE
007672  012777  001252  171638          MOV     #1252,@VCXREG           ;LOAD X REGISTER
007700  012737  000525  001124          MOV     #525,$BDDAT             ;LOAD EXPECTED
007706  013777  001124  171616          MOV     $GDDAT,@VCVREG          ;LOAD V REG
007714  017737  171612  001126          MOV     @VCVREG,$BDDAT          ;READ REG
007722  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007730  001401                        BEQ     18              ;;BR IF EQUAL
007732  104013                        ERROR   13                      ;ERROR, SELECTED V REGISTER INCORRECTLY

007734  012737  001252  001124  181:    MOV     #1252,$GDDAT            ;LOAD EXPECTED
007742  017737  171562  001126          MOV     @VCXREG,$BDDAT          ;READ REG
007750  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
007756  001401                        BEQ     TST104          ;;BR IF EQUAL
007760  104012                        ERROR   12                      ;ERROR, SELECTED X REGISTER INCORRECTLY
```

```
                        ;;*****************************************************************
                        ;*TEST 104      TEST THAT WHEN INTENSIFY BIT IS SET THAT THE VC READY BIT CLEARS
                        ;;*****************************************************************
                        TST104: SCOPE
                        ; AND THEN SETS AFTER A DELAY
                                MOV     #1000,R0
                                MOV     #0,SGDDAT               ;LOAD EXPECTED
                                MOV     #BIT0,@VCSTAT           ;INTENSIFY
                                MOV     @VCSTAT,SBDDAT          ;READ REG
                                TSTB    SBDDAT                  ;TEST READY
                                BPL     1$              ;;BR IF NOT SET
                                ERROR   11                      ;READY FAILED TO CLEAR
                                BR      TST105          ;;BR TO SCOPE
                        1$:     TSTB    @VCSTAT                 ;TEST READY
                                BMI     TST105          ;;NEXT TEST
                                DEC     R0                      ;DELAY
                                BNE     1$                      ;
                                MOV     #BIT7,SGDDAT            ;LOAD EXPECTED
                                MOV     @VCSTAT,SBDDAT          ;READ REG
                                ERROR   11                      ;READY FAILED TO SET AFTER A DELAY

                        ;;*****************************************************************
                        ;*TEST 105      TEST THAT VC MODE 1 (INTENSIFY ON X) CLEARS THE READY FLAG
                        ;;*****************************************************************
                        TST105: SCOPE
                        ; AND THEN SETS IT
                                MOV     #1000,R0                ;SET UP DELAY
                                MOV     #BIT7!BIT2,SGDDAT       ;LOAD EXPECTED
                                MOV     #BIT2,@VCSTAT           ;LOAD MODE 1
                                MOV     @VCSTAT,SBDDAT          ;READ REG
                                TSTB    SBDDAT                  ;TEST READY
                                BMI     2$              ;;BR IF READY STILL SET
                                ERROR   11                      ;ERROR, IN MODE 1 READY SHOULD NOT
                                                                ; CLEAR UNTIL X IS LOADED
                                BR      TST106          ;;BR TO SCOPE

                                MOV     #BIT2,SGDDAT            ;LOAD EXPECTED
                        2$:     CLR     @VCXREG                 ;ADDRESS X AXIS
                                MOV     @VCSTAT,SBDDAT          ;READ REG
                                TSTB    SBDDAT                  ;TEST READY
                                NOP
                                BPL     1$              ;;BR IF CLEAR
                                ERROR   11                      ;ERROR, MODE 1 LOAD X FAILED TO CLEAR READY FLAG
                                BR      TST106          ;;BR TO SCOPE

                        1$:     TSTB    @VCSTAT                 ;TEST READY
                                BMI     TST106          ;;NEXT TEST
                                DEC     R0                      ;DELAY
                                BNE     1$              ;;TEST READY AGAIN
                                MOV     #BIT7!BIT2,SGDDAT       ;LOAD EXPECTED
                                MOV     @VCSTAT,SBDDAT          ;READ REG
                                ERROR   11                      ;ERROR, READY FAILED TO SET
                                                                ; AFTER MODE 1 OPERATION
```

```
                                ;;*****************************************************************
                                ;*TEST 186       TEST THAT VC MODE 2 (INTENSIFY ON Y) CLEARS THE READY FLAG
                                ;;*****************************************************************
010202  000004                  TST186: SCOPE
                                ; AND THEN SETS IT
010204  012700  001000                  MOV     #1000,R0                ;SET UP DELAY
010210  012737  000210  001124          MOV     #BIT7!BIT3,SCODAT       ;LOAD EXPECTED
010216  012777  000010  171302          MOV     #BIT3,@VCSTAT           ;LOAD MODE 2
010224  017737  171276  001126          MOV     @VCSTAT,$BDDAT          ;READ REG
010232  105737  001126                  TSTB    $BDDAT                  ;TEST READY
010236  100402                          BMI     2$              ;;BR IF SET
010240  104011                          ERROR   11                      ;ERROR, IN MODE 2 READY SHOULD NOT CLEAR
                                                                        ;UNTIL Y IS LOADED
010242  000431                          BR      TST187          ;;BR TO SCOPE
010244  012737  000010  001124  2$:      MOV     #BIT3,SCODAT            ;LOAD EXPECTED
010252  005077  171254                  CLR     @VCYREG                 ;ADDRESS Y AXIS
010256  017737  171244  001126          MOV     @VCSTAT,$BDDAT          ;READ REG
010264  105737  001126                  TSTB    $BDDAT                  ;TEST READY
010270  100002                          BPL     1$              ;;BR IF CLEARED
010272  104011                          ERROR   11                      ;ERROR, MODE 2 LOAD Y FAILED TO CLEAR READY FLAG
010274  000414                          BR      TST187          ;;BR TO SCOPE
010276  105777  171224          1$:     TSTB    @VCSTAT                 ;TEST READY
010302  100411                          BMI     TST187          ;;NEXT TEST
010304  005300                          DEC     R0                      ;DELAY
010306  001373                          BNE     1$              ;;TEST READY AGAIN
010310  012737  000210  001124          MOV     #BIT7!BIT3,SCODAT       ;LOAD EXPECTED
010316  017737  171204  001126          MOV     @VCSTAT,$BDDAT          ;
010324  104011                          ERROR   11                      ;ERROR, READY FAILED TO SET
                                                                        ; AFTER MODE 2 OPERATION
                                ;;*****************************************************************
                                ;*TEST 187       TEST WHEN ERASE IS SET, VC READY BIT CLEARS AND SET AFTER DELAY
                                ;;*****************************************************************
010326  000004                  TST187: SCOPE
010330  032777  010000  170600          BIT     #BIT12,@SWR             ;TEST BIT 12
010336  001430                          BEQ     TST110          ;;BYPASS IF NO STORAGE SCOPE
010340  012700  000010                  MOV     #10,R0
010344  005037  001566                  CLR     TEMP                    ;CLEAR DELAY
010350  012777  002000  171150          MOV     #BIT10,@VCSTAT          ;SET STORE MODE
010356  052777  010000  171142          BIS     #BIT12,@VCSTAT          ;SET ERASE BIT
010364  105777  171136                  TSTB    @VCSTAT                 ;TEST THAT READY CLEARS
010370  100002                          BPL     1$              ;;BR IF CLEARED
010372  104011                          ERROR   11                      ;ERROR, READY FAILED TO RESET
010374  000411                          BR      TST110          ;;BR TO SCOPE
010376  105777  171120          1$:     TSTB    @VCSTAT                 ;TEST FOR READY
010402  100406                          BMI     TST110          ;;BR IF SET
010404  005337  001566                  DEC     TEMP                    ;DELAY
010410  001372                          BNE     1$              ;;BR IF NOT READY
010412  005300                          DEC     R0                      ;DECREMENT COUNTER
010414  001370                          BNE     1$              ;;BR IF NOT DONE
010416  104011                          ERROR   11                      ;ERROR, ERASE CLEARED READY AND FAILED
                                                                        ;TO SET READY AFTER A DELAY
```

```
                                ;;*****************************************************************
                                ;*TEST 110       SCOPE PRE-INTERRUPT SETUP
                                ;;*****************************************************************
                                TST110: SCOPE
010420  000004                         MOV     #1,$TIMES       ;;DO 1 ITERATION
010422  012737  177437  001164         BIC     #177437,VCBRL           ;MASK TO  PSW
010430  042737  177437  001514         BNE     .+4
010436  001001                         HALT                            ;LOCATION VCBRL CONTAINED A BR LEVEL 0
010440  000000                         CMP     #340,VCBRL              ;
010442  022737  000340  001514         BNE     .+4
010450  001001                         HALT                            ;LOCATION VCBRL CONTAINS BR LEVEL 7
010452  000000                         MOV     VCBRL,BRLEV1            ;SET UP BR LEVELS
010454  013737  001514  001572         SUB     #40,BRLEV1              ; -1
010462  162737  000040  001572         MOV     VCBRL,BRLEV2            ; 0
010470  013737  001514  001574
                                ;;*****************************************************************
                                ;*TEST 111       TEST THAT THE DISPLAY DOES INTERRUPT AT LEVEL INDICATED -1
                                ;;*****************************************************************
                                TST111: SCOPE
010476  000004                         MOV     #40,$TIMES      ;;DO 40 ITERATIONS
010500  012737  000040  001164         MOV     #340,PSW
010506  012737  000340  177776         MOV     #10,@VCIV               ;SET INTERRUPT VECTOR
010514  012777  010556  171024         MOV     #400,R0                 ;SET UP DELAY
010522  012700  000400                 MOV     BRLEV1,PSW
010526  013737  001572  177776         MOV     #BIT6!BIT0,@VCSTAT      ;START DISPLAY
010534  012777  000101  170764         DEC     R0                      ;DELAY
010542  005300                         BNE     .-2
010544  001376                         CLR     @VCSTAT                 ;DO NOT LET INTERRUPT ENABLE SET
010546  005077  170754                 ERROR   14                      ;ERROR, VC FAILED TO INTERRUPT
010552  104014                         BR      TST112          ;;NEXT TEST
010554  000404                 18:     MOV     VCIV8,@VCIV             ;RESET VECTOR
010556  013777  001550  170762         CMP     (SP)+,(SP)+             ;POP STACK
010564  022626
                                ;;*****************************************************************
                                ;*TEST 112       TEST THAT THE DISPLAY DOES NOT INTERRUPT AT LEVEL INDICATED
                                ;;*****************************************************************
                                TST112: SCOPE
010566  000004                         MOV     #40,$TIMES      ;;DO 40 ITERATIONS
010570  012737  000040  001164         MOV     #340,PSW
010576  012737  000340  177776         MOV     #10,@VCIV               ;SET INTERRUPT VECTOR
010604  012777  010640  170734         MOV     #4000,R0                        ;SET UP DELAY
010612  012700  004000                 MOV     BRLEV2,PSW
010616  013737  001574  177776         MOV     #BIT6!BIT0,@VCSTAT      ;START DISPLAY
010624  012777  000101  170674         DEC     R0                      ;DELAY
010632  005300                         BNE     .-2
010634  001376                         BR      28              ;;
010636  000404                 18:     CLR     @VCSTAT                 ;DO NOT LET INTERRUPT ENABLE SET
010640  005077  170662                 ERROR   14                      ;ERROR VC INTERRUPTED IN ERROR
010644  104014                         BR      TST113          ;;NEXT TEST
010646  000416                 28:     MOV     #38,@VCIV               ;LOAD RETURN VECTOR
010650  012777  010674  170670         BIC     #340,PSW                ;LOWER PSW
010656  042737  000340  177776         CLR     @VCSTAT                 ;CLEAR INT ENABLE
010664  005077  170636                 ERROR   14                      ;LOWERING THE PRIORITY FAILED TO ALLOW INTERRUPT
010670  104014                         BR      TST113          ;;NEXT TEST
010672  000404                 38:     MOV     VCIV8,@VCIV             ;RESET VECTOR
010674  013777  001550  170644         CMP     (SP)+,(SP)+             ;POP STACK
010702  022626
```

```
1107                                    ;;*****************************************************************
 (3)                                    ;*TEST 113      TEST THAT RESET CLEARS VC MODE BITS
 (3)                                    ;;*****************************************************************
 (2)  010704  000004                    TST1131: SCOPE
 (1)  010706  012737  000040  001164            MOV     #40,STIMES      ;;DO 40 ITERATIONS
1108  010714  012777  000014  170604            MOV     #BIT3!BIT2,@VCSTAT
1109  010722  012737  000200  001124            MOV     #BIT7,$DDDAT            ;LOAD EXPECTED
1110  010730  000005                            RESET
1111  010732  017737  170570  001126            MOV     @VCSTAT,$DDDAT          ;READ STATUS
1112  010740  023737  001124  001126            CMP     $DDDAT,$DDDAT           ;COMPARE
1113  010746  001401                            BEQ     TST114          ;;BR IF EQUAL
1114  010750  104011                            ERROR   11                      ;ERROR, RESET FAILED TO CLEAR VC STATUS REG
1115
1116                                    ;;*****************************************************************
 (3)                                    ;*TEST 114      TEST THAT RESET CLEARS INTERRUPT ENABLE, CHANNEL, STORE, WRITE THRU
 (3)                                    ;;*****************************************************************
 (2)  010752  000004                    TST1141: SCOPE
 (1)  010754  012737  000040  001164            MOV     #40,STIMES      ;;DO 40 ITERATIONS
1117  010762  012777  007100  170536            MOV     #BIT11!BIT10!BIT9!BIT6,@VCSTAT
1118  010770  012737  000200  001124            MOV     #BIT7,$DDDAT            ;LOAD EXPECTED
1119  010776  000005                            RESET
1120  011000  017737  170522  001126            MOV     @VCSTAT,$DDDAT          ;READ STATUS
1121  011006  023737  001124  001126            CMP     $DDDAT,$DDDAT           ;COMPARE
1122  011014  001401                            BEQ     TST115          ;;BR IF EQUAL
1123  011016  104011                            ERROR   11                      ;ERROR, RESET FAILED TO CLEAR VC STATUS
1124
1125                                    ;;*****************************************************************
 (3)                                    ;*TEST 115      TEST THAT RESET CLEARS X REGISTER
 (3)                                    ;;*****************************************************************
 (2)  011020  000004                    TST1151: SCOPE
 (1)  011022  012737  000040  001164            MOV     #40,STIMES      ;;DO 40 ITERATIONS
1126  011030  012777  177777  170472            MOV     #-1,@VCXREG
1127  011036  012737  000000  001124            MOV     #0,$DDDAT               ;LOAD EXPECTED
1128  011044  000005                            RESET
1129  011046  017737  170456  001126            MOV     @VCXREG,$DDDAT          ;READ REG
1130  011054  023737  001124  001126            CMP     $DDDAT,$DDDAT           ;COMPARE
1131  011062  001401                            BEQ     TST116          ;;BR IF EQUAL
1132  011064  104011                            ERROR   11                      ;ERROR, RESET FAILED TO CLEAR VC X REGISTER
1133
1134                                    ;;*****************************************************************
 (3)                                    ;*TEST 116      TEST THAT RESET CLEARS Y REGISTER
 (3)                                    ;;*****************************************************************
 (2)  011066  000004                    TST1161: SCOPE
 (1)  011070  012737  000040  001164            MOV     #40,STIMES      ;;DO 40 ITERATIONS
1135  011076  012777  177777  170426            MOV     #-1,@VCYREG
1136  011104  012737  000000  001124            MOV     #0,$DDDAT               ;LOAD EXPECTED
1137  011112  000005                            RESET
1138  011114  017737  170412  001126            MOV     @VCYREG,$DDDAT          ;READ REG
1139  011122  023737  001124  001126            CMP     $DDDAT,$DDDAT           ;COMPARE
1140  011130  001401                            BEQ     TST117          ;;BR IF EQUAL
1141  011132  104011                            ERROR   11                      ;ERROR, RESET FAILED TO CLEAR VC Y REGISTER
1142
```

```
                                ;;*****************************************************************
                                ;*TEST 117       DOES EXTERNAL ENABLE (BIT 4) SET
                                ;;*****************************************************************
011134  000004                  TST117: SCOPE
011136  012777  000020  170352          MOV     #BIT4,@ADCS             ;SET BIT 4
011144  012737  000020  001124          MOV     #BIT4,$GDDAT            ;LOAD EXPECTED
011152  017737  170340  001126          MOV     @ADCS,$BDDAT            ;READ REG
011160  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011166  001401                          BEQ     TST120          ;;BR IF SET
011170  104001                          ERROR   1                       ;EXT ENABLE BIT 4 FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 120       DOES CLOCK OVERFLOW ENABLE (BIT 5) SET
                                ;;*****************************************************************
011172  000004                  TST120: SCOPE
011174  012777  000040  170314          MOV     #BIT5,@ADCS             ;SET BIT 5
011202  012737  000040  001124          MOV     #BIT5,$GDDAT            ;LOAD EXPECTED
011210  017737  170302  001126          MOV     @ADCS,$BDDAT            ;READ REG
011216  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011224  001401                          BEQ     TST121          ;;BR IF SET
011226  104001                          ERROR   1                       ;CLOCK OVERFLOW ENABLE FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 121       DOES AD INTERRUPT ENABLE (BIT 6) SET
                                ;;*****************************************************************
011230  000004                  TST121: SCOPE
011232  012777  000100  170256          MOV     #BIT6,@ADCS             ;SET BIT 6
011240  012737  000100  001124          MOV     #BIT6,$GDDAT            ;LOAD EXPECTED
011246  017737  170244  001126          MOV     @ADCS,$BDDAT            ;READ REG
011254  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011262  001401                          BEQ     TST122          ;;BR IF SET
011264  104001                          ERROR   1                       ;AD INTERRUPT ENABLE FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 122       DOES MUX CHANNEL (BIT 8) SET
                                ;;*****************************************************************
011266  000004                  TST122: SCOPE
011270  012777  000400  170220          MOV     #BIT8,@ADCS             ;SET BIT 8
011276  012737  000400  001124          MOV     #BIT8,$GDDAT            ;LOAD EXPECTED
011304  017737  170206  001126          MOV     @ADCS,$BDDAT            ;READ REG
011312  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011320  001401                          BEQ     TST123          ;;BR IF SET
011322  104001                          ERROR   1                       ;MUX BIT 8 FAILED TO SET

                                ;;*****************************************************************
                                ;*TEST 123       DOES MUX CHANNEL (BIT 9) SET
                                ;;*****************************************************************
011324  000004                  TST123: SCOPE
011326  012777  001000  170162          MOV     #BIT9,@ADCS             ;SET BIT 9
011334  012737  001000  001124          MOV     #BIT9,$GDDAT            ;LOAD EXPECTED
011342  017737  170150  001126          MOV     @ADCS,$BDDAT            ;READ REGISTER
011350  023737  001124  001126          CMP     $GDDAT,$BDDAT           ;COMPARE
011356  001401                          BEQ     TST124          ;;BR IF EQUAL
011360  104001                          ERROR   1                       ;MUX BIT 9 FAILED TO SET
```

```
                        ;;*****************************************************************
                        ;*TEST 124       DOES MUX CHANNEL (BIT 10) SET
                        ;;*****************************************************************
(2)  011362  000004     TST124: SCOPE
1185 011364  012777  002000  170124        MOV     #BIT10,@ADCS            ;SET BIT 10
1186 011372  012737  002000  001124        MOV     #BIT10,$GDDAT           ;LOAD EXPECTED
1187 011400  017737  170112  001126        MOV     @ADCS,$BDDAT            ;READ REG.
1188 011406  023737  001124  001126        CMP     $GDDAT,$BDDAT           ;COMPARE
1189 011414  001401                        BEQ     TST125          ;;BR IF EQUAL
1190 011416  104001                        ERROR   1                       ;MUX BIT 10 FAILED TO SET
1191
1192                    ;;*****************************************************************
                        ;*TEST 125       DOES MUX CHANNEL (BIT 11) SET
                        ;;*****************************************************************
(2)  011420  000004     TST125: SCOPE
1193 011422  012777  004000  170066        MOV     #BIT11,@ADCS            ;SET BIT 11
1194 011430  012737  004000  001124        MOV     #BIT11,$GDDAT           ;LOAD EXPECTED
1195 011436  017737  170054  001126        MOV     @ADCS,$BDDAT            ;READ REGISTER
1196 011444  023737  001124  001126        CMP     $GDDAT,$BDDAT           ;COMPARE
1197 011452  001401                        BEQ     TST126          ;;BR IF EQUAL
1198 011454  104001                        ERROR   1                       ;MUX BIT 11 FAILED TO SET
1199
1200                    ;;*****************************************************************
                        ;*TEST 126       DOES UNIPOLAR/BIPOLAR (BIT 13) SET
                        ;;*****************************************************************
(2)  011456  000004     TST126: SCOPE
1201 011460  012777  020000  170030        MOV     #BIT13,@ADCS            ;SET BIT 13
1202 011466  012737  020000  001124        MOV     #BIT13,$GDDAT           ;LOAD EXPECTED
1203 011474  017737  170016  001126        MOV     @ADCS,$BDDAT            ;READ REGISTER
1204 011502  023737  001124  001126        CMP     $GDDAT,$BDDAT           ;COMPARE
1205 011510  001401                        BEQ     TST127          ;;BR IF EQUAL
1206 011512  104001                        ERROR   1                       ;UNIPOLAR /BIPOLAR BIT FAILED TO SET
1207
1208                    ;;*****************************************************************
                        ;*TEST 127       TEST FOR PROPER SELECTION OF THE LOW BYTE OPERATION
                        ;;*****************************************************************
(2)  011514  000004     TST127: SCOPE
1209 011516  012737  001000  001124        MOV     #BIT9,$GDDAT            ;LOAD EXPECTED VALUE
1210 011524  012777  001040  167764        MOV     #BIT9!BIT5,@ADCS        ;LOAD A TO D REG.
1211 011532  105077  167760                CLRB    @ADCS                   ;CLEAR THE LOW BYTE
1212 011536  017737  167754  001126        MOV     @ADCS,$BDDAT            ;READ RESULT
1213 011544  023737  001124  001126        CMP     $GDDAT,$BDDAT           ;COMPARE RESULTS
1214 011552  001401                        BEQ     TST130          ;;BR IF THE SAME
1215 011554  104001                        ERROR   1                       ;FAILED TO CLEAR ONLY THE LOW BYTE OF A TO D
1216
```

```
1218                                    ;;*****************************************************************
 (3)                                    ;*TEST 138      TEST FOR PROPER SELECTION OF THE HIGH BYTE OPERATION
 (3)                                    ;;*****************************************************************
 (2)  011556  000004                    TST138: SCOPE
1219  011560  012737  000040  001124            MOV     #BIT5,SGNDAT            ;LOAD EXPECTED
1220  011566  013737  001516  001566            MOV     ADCS,TEMP               ;GET BASE ADDRESS OF A TO D
1221  011574  005237  001566                    INC     TEMP                    ;MAKE HIGH BYTE ADDRESS
1222  011600  012777  001040  167710            MOV     #BIT9!BIT5,@ADCS        ;LOAD A TO D REG
1223  011606  105077  167754                    CLRB    @TEMP                   ;SHOULD CLEAR HIGH BYTE ONLY
1224  011612  017737  167700  001126            MOV     @ADCS,SBDDAT            ;READ A TO D REG.
1225  011620  023737  001124  001126            CMP     SGNDAT,SBDDAT           ;COMPARE RESULTS
1226  011626  001401                            BEQ     TST131          ;;BR IF THE SAME
1227  011630  104001                            ERROR   1                       ;FAILED TO CLEAR ONLY THE HIGH BYTE OF A TO D
1228
1229                                    ;;*****************************************************************
 (3)                                    ;*TEST 131      TEST AD GO (BIT 0) CAN BE SET AND CLEARED
 (3)                                    ;;*****************************************************************
 (2)  011632  000004                    TST131: SCOPE
1230
1231  011634  017700  167660                    MOV     @ADDBR,R0               ;FAKE READ
1232  011640  005077  167652                    CLR     @ADCS
1233  011644  012737  000001  001124            MOV     #BIT0,SGDDAT            ;LOAD EXPECTED
1234  011652  013777  001124  167636            MOV     SGNDAT,@ADCS
1235  011660  017737  167632  001126            MOV     @ADCS,SBDDAT            ;READ REGISTER
1236  011666  023737  001124  001126            CMP     SGDDAT,SBDDAT           ;COMPARE
1237  011674  001401                            BEQ     1$              ;;BR IF SET
1238  011676  104001                            ERROR   1                       ;A TO D GO FAILED TO SET
1239  011700  012700  001000            1$:     MOV     #1000,R0
1240  011704  005300                            DEC     R0
1241  011706  001376                            BNE     .-2
1242  011710  017737  167602  001126            MOV     @ADCS,SBDDAT            ;READ REG.
1243  011716  012737  000200  001124            MOV     #BIT7,SGDDAT            ;LOAD EXPECTED
1244  011724  032737  000001  001126            BIT     #BIT0,SBDDAT            ;TEST BIT
1245  011732  001401                            BEQ     TST132          ;;BR IF CLEARED
1246  011734  104001                            ERROR   1                       ;A TO D GO FAILED TO CLEAR
```

```
                                ;;*****************************************************************
                                ;*TEST 132      DOES AD DONE (BIT 7) SET AND CLEAR
                                ;;*****************************************************************
 012736  000004                 TST132: SCOPE
 012740  017700  167554                 MOV     @ADDBR,R0
 012744  012700  000000                 MOV     #0,R0
 012750  012777  000001  167540         MOV     #BIT0,@ADCS
 012756  012737  000200  001124         MOV     #BIT7,$GDDAT            ;LOAD EXPECTED
 012764  105777  167526         1$:     TSTB    @ADCS
 012770  100406                         BMI     2$
 012772  005200                         INC     R0
 012774  001373                         BNE     1$
 012776  017737  167514  001126         MOV     @ADCS,$BDDAT            ;READ REGISTER
 013004  104001                         ERROR   1                       ;ERROR, A TO D DONE FAILED TO SET
 013006  017700  167506         2$:     MOV     @ADDBR,R0
 013012  012737  000000  001124         MOV     #0,$GDDAT                       ;CLEAR EXPECTED
 013020  017737  167472  001126         MOV     @ADCS,$BDDAT            ;READ REG
 013026  105737  001126                 TSTB    $BDDAT
 013032  100001                         BPL     3$                      ;BR IF CLEARED
 013034  104001                         ERROR   1                       ;ERROR, DONE FAILED TO CLEAR UPON
                                                                         ;READING CONVERTED VALUE
 013036  017737  167454  001126 3$:     MOV     @ADCS,$BDDAT            ;READ REG
 013044  005037  001124                 CLR     $GDDAT                  ;CLEAR EXPECTED
 013050  032737  000001  001126         BIT     #BIT0,$BDDAT
 013056  001401                         BEQ     TST133                  ;BR IF CLEARED
 013060  104001                         ERROR   1                       ;ERROR, AD GO FAILED TO CLEAR


                                ;;*****************************************************************
                                ;*TEST 133      TEST THAT THE CONVERTED NUMBER = 1777 (SW BIT 10=1)
                                ;;*****************************************************************
 013062  000004                 TST133: SCOPE
 013064  005077  167426                 CLR     @ADCS                   ;INIT
 013070  005277  167422                 INC     @ADCS                   ;CONVERT
 013074  105777  167416         1$:     TSTB    @ADCS                   ;LOOP
 013100  100375                         BPL     1$
 013102  012737  001777  001124         MOV     #1777,$GDDAT            ;LOAD EXPECTED
 013110  017737  167404  001126         MOV     @ADDBR,$BDDAT           ;READ REG
 013116  032777  002000  167012         BIT     #BIT10,@SWR             ;TEST BIT 10
 013124  001405                         BEQ     TST134          ;;BR IF CLEARED
 013126  023737  001124  001126         CMP     $GDDAT,$BDDAT           ;COMPARE
 013134  001401                         BEQ     TST134          ;;BR IF EQUAL
 013136  104016                         ERROR   16                      ;CONVERTED VALUE NOT = 1777
                                                ;OPERATOR INFORMED THE PROGRAM THAT THE
                                                ; ALL 1'S JUMPER HAD BEEN INSTALLED
```

```
 1289
 1290                                   ;;*****************************************************************
  (3)                                   ;*TEST 134      TEST THAT NO EXTERNAL CONVERSIONS INPUT
  (3)                                   ;;*****************************************************************
  (2)  012140  000004                   TST134: SCOPE
  (1)  012142  012737  000004  001164           MOV     #4,STIMES       ;;DO 4 ITERATIONS
 1291  012150  005077  167342                   CLR     @ADCS
 1292  012154  012737  000020  001124           MOV     #BIT4,$DDAT             ;LOAD EXPECTED
 1293  012162  017700  167332                   MOV     @ADDBR,R0
 1294  012166  052777  000020  167322           BIS     #BIT4,@ADCS
 1295  012174  012700  000000                   MOV     #0,R0
 1296  012200  005200                   18:     INC     R0
 1297  012202  001376                           BNE     18
 1298  012204  017737  167306  001126           MOV     @ADCS,$DDAT             ;READ REG
 1299  012212  105737  001126                   TSTB    $DDAT
 1300  012216  100001                           BPL     TST135          ;;BR IF PLUS
 1301  012220  104001                           ERROR   1                       ;ERROR EXTERNAL CONVERSION
 1302
 1303                                   ;;*****************************************************************
  (3)                                   ;*TEST 135      TEST THAT CLOCK CAN START A CONVERSION
  (3)                                   ;;*****************************************************************
  (2)  012222  000004                   TST135: SCOPE
  (1)  012224  012737  000004  001164           MOV     #4,STIMES       ;;DO 4 ITERATIONS
 1304  012232  005077  167260                   CLR     @ADCS           ;
 1305  012236  005077  167260                   CLR     @CSR            ;CLEAR CLOCK
 1306  012242  012737  000240  001124           MOV     #BIT7!BIT5,$DDAT        ;LOAD EXPECTED DATA
 1307  012250  012737  000003  014056           MOV     #3,RATE                 ;LOAD CLOCK RATE
 1308  012256  052777  000040  167232           BIS     #BIT5,@ADCS             ;ENABLE CLOCK OVERFLOW
 1309  012264  004737  013376                   JSR     PC,UPCNT                ;ENABLE THE CLOCK
 1310  012270  012700  000040                   MOV     #BIT5,R0                ;SETUP DELAY
 1311  012274  005300                   181:    DEC     R0                      ;DELAY
 1312  012276  001376                           BNE     18
 1313  012300  017737  167212  001126           MOV     @ADCS,$DDAT             ;READ REGISTER
 1314  012306  023737  001124  001126           CMP     $DDAT,$DDAT             ;COMPARE
 1315  012314  001401                           BEQ     TST136          ;;BR IF EQUAL
 1316  012316  104001                           ERROR   1                       ;CLOCK OVERFLOW FAILED TO START A CONVERSION
```

K4

```
                                ;;*****************************************************************
                                ;*TEST 136       A TO D PRE-INTERRUPT SETUP
                                ;;*****************************************************************
                        TST136: SCOPE
012320  000004                  
012322  012737  000001  001164          MOV     #1,$TIMES       ;;DO 1 ITERATION
012330  042737  177437  001510          BIC     #177437,ADBRL           ;MASK TO BITS
012336  001001                          BNE     .+4
012340  000000                          HALT                            ;BR LEVEL INDICATED IS 0
012342  022737  000340  001510          CMP     #340,ADBRL              ;IS IT BR LEVEL 7
012350  001001                          BNE     .+4
012352  000000                          HALT                            ;BR LEVEL INDICATED IS 7
012354  013737  001510  001572          MOV     ADBRL,BRLEV1
012362  162737  000040  001572          SUB     #40,BRLEV1
012370  013737  001510  001574          MOV     ADBRL,BRLEV2

                                ;;*****************************************************************
                                ;*TEST 137       TEST THAT A TO D INTERRUPTS AT LEVEL INDICATED -1
                                ;;*****************************************************************
                        TST137: SCOPE
012376  000004                  
012400  012737  000040  001164          MOV     #40,$TIMES      ;;DO 40 ITERATIONS
012406  005077  167104                  CLR     @ADCS                   ;CLEAR A TO D STATUS
012412  017700  167102                  MOV     @ADDBR,R0               ;READ ADBUFF
012416  012777  012454  167112          MOV     #10$,@ADINT             ;LOAD RETURN ADDRESS
012424  013737  001572  177776          MOV     BRLEV1,PSW              ;BR LEVEL -1
012432  052777  000101  167056          BIS     #BIT6!BIT0,@ADCS        ;SET THE AD GO AND INTERRUPT ENABLE BIT
012440  012700  000100                  MOV     #100,R0
012444  005300                          DEC     R0                      ;DELAY
012446  001376                          BNE     .-2
012450  104002                          ERROR   2                       ;ERROR, A TO D FAILED TO INTERRUPT
012452  000403                          BR      TST140          ;;NEXT TEST
012454  005077  167036          10$:    CLR     @ADCS                   ;CLEAR ERROR BIT
012460  022626                          CMP     (SP)+,(SP)+
```

```
                                ;;*****************************************************************
 (3)                            ;*TEST 140      TEST THAT THE A TO D DOES NOT INTERRUPT AT LEVEL INDICATED
 (3)                            ;;*****************************************************************
 (2) 012462 000004              TST140: SCOPE
 (1) 012464 012737 000040 001164        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1346 012472 005077 167020               CLR     @#ADCS                  ;CLEAR A TO D STATUS
1347 012476 017700 167016               MOV     @#ADDBR,R0              ;READ ADBUFF
1348 012502 013737 001574 177776        MOV     $BRLEV2,PSW             ;BR LEVEL
1349 012510 012777 012536 167020        MOV     #18,@#ADINT             ;LOAD RETURN ADDRESS
1350 012516 052777 000101 166772        BIS     #BIT6!BIT0,@#ADCS       ;SET AD GO AND INTERRUPT ENABLE BIT
1351 012524 012700 001000               MOV     #1000,R0
1352 012530 005300                      DEC     R0                      ;DELAY
1353 012532 001376                      BNE     .-2
1354 012534 000405                      BR      3$              ;;<NEXT SUB-TEST>
1355 012536 005077 166754       18$:    CLR     @#ADCS                  ;CLEAR ERROR FLAG
1356 012542 022626                      CMP     (SP)+,(SP)+
1357 012544 104002                      ERROR   2                       ;A TO D INTERRUPTED ON LEVEL INDICATED
1358 012546 000423                      BR      TST141          ;;<NEXT TEST>
1359 012550 012777 012576 166760 38$:   MOV     #28,@#ADINT             ;RELOAD VECTOR
1360 012556 012700 001000               MOV     #1000,R0                ;LOAD COUNT
1361 012562 005037 177776               CLR     PSW                     ;LOWER PSW
1362 012566 005300                      DEC     R0
1363 012570 001376                      BNE     .-2                     ;DELAY
1364 012572 104002                      ERROR   2                       ;LOWERING PSW FAILED TO ALLOW AD INTERRUPT
1365 012574 000410                      BR      TST141          ;;NEXT TEST
1366 012576 022626              28$:    CMP     (SP)+,(SP)+
1367 012600 005077 166712               CLR     @#ADCS
1368 012604 013777 001540 166724        MOV     ADINT1,@#ADINT
1369 012612 005077 166722               CLR     @#ADINT1
1370
1371                            ;;*****************************************************************
 (3)                            ;*TEST 141      TEST THAT RESET CLEARS MUX AND UNIPOLAR BITS
 (3)                            ;;*****************************************************************
 (2) 012616 000004              TST141: SCOPE
 (1) 012620 012737 000040 001164        MOV     #40,$TIMES      ;;DO 40 ITERATIONS
1372 012626 005037 001124               CLR     $GDDAT                  ;CLEAR EXPECTED
1373 012632 012777 007400 166656        MOV     #BIT13!BIT11!BIT10!BIT9!BIT8,@#ADCS
1374 012640 000005                      RESET
1375 012642 017737 166650 001126        MOV     @#ADCS,$GDDAT           ;READ REG
1376 012650 001401                      BEQ     TST142          ;;BR IF CLEARED
1377 012652 104001                      ERROR   1                       ;ERROR, RESET FAILED TO CLEAR MUX BITS
1378
```

```
                                ;;*****************************************************************
                                ;*TEST 142       TEST THAT RESET CLEARS EXT AND INTERRUPT ENABLE BITS
                                ;;*****************************************************************
012654  000004                TST142: SCOPE
012656  012737  000040  001164          MOV     #40,STIMES      ;;DO 40 ITERATIONS
012664  012777  000120  166624          MOV     #BIT6!BIT4,@ADCS
012672  005037  001124                  CLR     SGDDAT                  ;CLEAR EXPECTED
012676  000005                          RESET
012700  017737  166612  001126          MOV     @ADCS,SRDDAT            ;READ REG
012706  001401                          BEQ     TST143          ;;BR IF CLEARED
012710  104001                          ERROR   1                       ;ERROR, RESET FAILED TO CLEAR ENABLES

                                ;;*****************************************************************
                                ;*TEST 143       TEST THAT RESET CLEARS AD DONE
                                ;;*****************************************************************
012712  000004                TST143: SCOPE
012714  012737  000040  001164          MOV     #40,STIMES      ;;DO 40 ITERATIONS
012722  012777  000001  166566          MOV     #BIT0,@ADCS
012730  005037  001124                  CLR     SGDDAT
012734  105777  166556                  TSTB    @ADCS
012740  100375                          BPL     .-4
012742  000005                          RESET
012744  017737  166546  001126          MOV     @ADCS,SRDDAT
012752  001401                          BEQ     TST144          ;;BR IF CLEARED
012754  104001                          ERROR   1                       ;ERROR, RESET FAILED TO CLEAR DONE FLAG

                                ;;*****************************************************************
                                ;*TEST 144       TEST THAT RESET CLEARS AD BUFFER REG
                                ;;*****************************************************************
012756  000004                TST144: SCOPE
012760  012737  000040  001164          MOV     #40,STIMES      ;;DO 40 OPERATIONS
012766  012777  000001  166522          MOV     #1,@ADCS                ;CONVERT
012774  105777  166516          1$:     TSTB    @ADCS                   ;WAIT
013000  100375                          BPL     1$
013002  000005                          RESET
013004  017737  166510  001126          MOV     @ADDBR,SRDDAT           ;READ REGISTER
013012  005037  001124                  CLR     SGDDAT                  ;LOAD EXPECTED
013016  023737  001124  001126          CMP     SGDDAT,SRDDAT           ;COMPARE
013024  001401                          BEQ     TST145          ;;BR IF CLEARED
013026  104016                          ERROR   16                      ;RESET FAILED TO CLEAR AD BUFFER REGISTER
```

```
1409                                    ;;*****************************************************************
 (3)                                    ;*TEST 145      LOAD DIFFERENT NUMBERS INTO DIFFERENT REG.
 (3)                                    ;;*****************************************************************
 (2)  013030  000004                    TST145: SCOPE
 (1)  013032  012737  000020  001164            MOV     #20,STIMES      ;;DO 20 ITERATIONS
1410  013040  000005                            RESET
1411  013042  013700  001534                    MOV     CSC,R0                  ;LOAD STARTING ADDRESS
1412  013046  062700  000002                    ADD     #2,R0                   ;ADJUST ADDRESS
1413  013052  012701  013162                    MOV     #BUFNUM+20,R1           ;LOAD STARTING TABLE ADDRESS
1414  013056  012702  000010                    MOV     #8.,R2                  ;LOAD COUNT
1415  013062  014140                    18:     MOV     -(R1),-(R0)             ;LOAD TMP REG
1416  013064  005302                            DEC     R2                      ;DONE ALL
1417  013066  001375                            BNE     18                      ;BR IF NOT
1418
1419  013070  013700  001516                    MOV     ADCB,R0                 ;LOAD STARTING POINTER
1420  013074  012701  013142                    MOV     #BUFNUM,R1              ;LOAD STARTING POINTER <EXPECTED>
1421  013100  012702  000010                    MOV     #8.,R2                  ;LOAD # OF REG
1422
1423  013104  011137  001124            38:     MOV     (R1),SGDDAT             ;READ REG
1424  013110  011037  001126                    MOV     (R0),SBDDAT             ;READ REG
1425  013114  023737  001124  001126            CMP     SGDDAT,SBDDAT           ;COMPARE
1426  013122  001403                            BEQ     48              ;;BR IF EQUAL
1427  013124  010037  013162                    MOV     R0,BUFADR               ;SAVE BUS ADDRESS
1428  013130  104017                            ERROR   17                      ;INCORRECT DATA, REG. WAS CHANGED IN ERROR
1429  013132  022021                    48:     CMP     (R0)+,(R1)+             ;UPDATE POINTERS
1430  013134  005302                            DEC     R2                      ;DONE ALL REG
1431  013136  001362                            BNE     38                      ;BR IF NOT
1432  013140  000411                            BR      TST146          ;;NEXT TEST
1433
1434                                    ;# TO BE LOADED INTO DIFFERENT REG
1435
1436  013142  027560                    BUFNUM: 27560                   ;A TO D STATUS
1437  013144  000000                            0                       ;A TO D BUFFER
1438  013146  140736                            140736                  ;CLOCK STATUS
1439  013150  000377                            377                     ;CLOCK PRESET
1440  013152  007214                            7214                    ;VC STATUS
1441  013154  001252                            1252                    ;VC X POS
1442  013156  000525                            525                     ;VC Y POS
1443  013160  000377                            377                     ;CLOCK COUNTER
1444
1445  013162  170400                    BUFADR: 170400                  ;BUS ADDRESS OF REG IN ERROR
1446
1447                                    ;;*****************************************************************
 (3)                                    ;*TEST 146      DETERMINE IF MORE AR11'S ARE TO BE TESTED
 (3)                                    ;;*****************************************************************
 (2)  013164  000004                    TST146: SCOPE
1448  013166  005737  001560            BYPAS1: TST     NBEXT                   ;TEST IF ANY
1449  013172  001411                            BEQ     18                      ;BR IF NONE
1450  013174  062737  000020  001552            ADD     #20,ARBADD              ;UPDATE DEVICE ADDRESS
1451  013202  062737  000020  001554            ADD     #20,ARBVCT              ;UPDATE DEVICE VECTOR
1452  013210  005337  001560                    DEC     NBEXT                   ;ANOTHER ONE ?
1453  013214  000413                            BR      BYPAS1                  ;BR IF ANOTHER
1454  013216  013737  001250  001552    18:     MOV     $BASE,ARBADD            ;RELOAD ADDRESS
1455  013224  013737  001244  001554            MOV     $VECT1,ARBVCT           ;RELOAD VECTOR
1456  013232  013737  001556  001560            MOV     NMBEXT,NBEXT            ;RELOAD NUMBER
1457  013240  000137  013254                    JMP     @EOP                    ;DONE
```

```
 1458  013244  012700  177777          BYPAS1: MOV     #-1,R0
 1459  013250  000137  002170                  JMP     AREG2                   ;TEST ANOTHER UNIT
 1460
  (1)                                  .SBTTL  END OF PASS ROUTINE
  (1)
  (2)                                  ;;*****************************************************************
  (1)                                  ;*INCREMENT THE PASS NUMBER (SPASS)
  (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 BYPAS1
  (1)
  (1)  013254                          SEOP:
  (1)  013254  000004                          SCOPE
  (1)  013256  005037  001102                  CLR     STSTNM          ;;ZERO THE TEST NUMBER
  (1)  013262  005037  001164                  CLR     STIMES          ;;ZERO THE NUMBER OF ITERATIONS
  (1)  013266  005237  001202                  INC     SPASS           ;;INCREMENT THE PASS NUMBER
  (1)  013272  042737  100000  001202          BIC     #100000,SPASS   ;;DON'T ALLOW A NEG. NUMBER
  (1)  013300  005327                          DEC     (PC)+           ;;LOOP?
  (1)  013302  000001                  SEOPCT: .WORD   1
  (1)  013304  003022                          BGT     SDOAGN          ;;YES
  (1)  013306  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
  (1)  013310  000001                  SENDCT: .WORD   1
  (1)  013312  013302                          SEOPCT
  (1)  013314  104400  013361                  TYPE    ,SENDMS         ;;TYPE "END PASS #"
  (2)  013320  013746  001202                  MOV     SPASS,-(SP)     ;;SAVE SPASS FOR TYPEOUT
  (2)  013324  104404                          TYPOS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
  (1)  013326  104400  013356                  TYPE    ,SENULL         ;;TYPE A NULL CHARACTER
  (1)  013332  013700  000042          SGET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
  (1)  013336  001405                          BEQ     SDOAGN          ;;BRANCH IF NO MONITOR
  (1)  013340  000005                          RESET                   ;;CLEAR THE WORLD
  (1)  013342  004710                  SENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
  (1)  013344  000240                          NOP                     ;;SAVE ROOM
  (1)  013346  000240                          NOP                     ;;FOR
  (1)  013350  000240                          NOP                     ;;ACT11
  (1)  013352                          SDOAGN:
  (1)  013352  000137                          JMP     @(PC)+          ;;RETURN
  (1)  013354  013244                  SRTNAD: .WORD   BYPAS1
  (1)  013356     377     377     000  SENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
  (1)  013361     015  042412  042116  SENDMS: .ASCIZ  <15><12>/END PASS #/
  (1)  013366  050040  051501  020123
  (1)  013374  000043
 1461
```

```
  1463
  1464                                  ;SUBROUTINE TO LOAD -2 INTO THE CLOCK PRESET REGISTER
  1465                                  ; AND LOAD CLOCK RATE INTO CLOCK STATUS REGISTER
  1466                                  ; AND START THE CLOCK, WAIT A PERIOD OF TIME THEN EXIT
  1467
  1468  013376  005077  166128  UPCNT: CLR     @CSR                    ;CLEAR CLOCK STATUS
  1469  013402  012777  177776  166114         MOV     #-2,@CSB        ;MOVE -2 INT PRESET
  1470  013410  013777  014056  166104         MOV     RATE,@CSR       ;LOAD RATE AND ENABLE CLOCK
  1471  013416  012737  000000  001564         MOV     #0,DELAY
  1472  013424  105777  166072  UPCNTA: TSTB    @CSR                    ;
  1473  013430  100407                 BMI     UPCNTB                  ;
  1474  013432  022727  000000  000000         CMP     #0,#0
  1475  013440  062737  000001  001564         ADD     #1,DELAY        ;
  1476  013446  001366                 BNE     UPCNTA                  ;
  1477  013450  000207  UPCNTB: RTS     7                               ;EXIT
  1478
  1479
   (1)                                  .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)
   (2)  013452                          $TYPDS:
   (3)  013452  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
   (3)  013454  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   (3)  013456  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
   (3)  013460  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
   (3)  013462  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
   (1)  013464  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
   (1)  013470  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
   (1)  013474  100004                          BPL     1$              ;;BR IF INPUT IS POS.
   (1)  013476  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
   (1)  013500  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
   (1)  013506  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
   (1)  013510  012703  013666                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
   (1)  013514  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
   (1)  013520  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
   (1)  013522  016001  013656                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
   (1)  013526  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
   (1)  013530  002402                          BLT     4$              ;;BR IF DONE
   (1)  013532  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
   (1)  013534  000774                          BR      3$
   (1)  013536  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
   (1)  013540  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
   (1)  013542  001002                          BNE     5$              ;;FALL THROUGH IF 0
   (1)  013544  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
   (1)  013546  100407                          BMI     7$              ;;BR IF YES
   (1)  013550  106316                  5$:     ASLB    (SP)            ;;MSD?
   (1)  013552  103003                          BCC     6$              ;;BR IF NO
```

```
(1)  013554  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
(1)  013562  052702  000060          68$:    BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
(1)  013566  052702  000040          78$:    BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
(1)  013572  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
(1)  013574  005720                          TST     (R0)+           ;;JUST INCREMENTING
(1)  013576  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
(1)  013602  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
(1)  013604  003002                          BGT     8$              ;;GO TO EXIT
(1)  013606  010502                          MOV     R5,R2           ;;SET THE LSD
(1)  013610  000764                          BR      6$              ;;GO CHANGE TO ASCII
(1)  013612  105726                  88:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
(1)  013614  100003                          BPL     9$              ;;BR IF NO
(1)  013616  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
(1)  013624  105013                  98:     CLRB    (R3)            ;;SET THE TERMINATOR
(3)  013626  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
(3)  013630  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
(3)  013632  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
(3)  013634  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  013636  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  013640  104400  013666                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
(1)  013644  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
(1)  013652  012616                          MOV     (SP)+,(SP)
(1)  013654  000002                          RTI                     ;;RETURN TO USER
(1)  013656  023420                  $DTBL:  10000.
(1)  013660  001750                          1000.
(1)  013662  000144                          100.
(1)  013664  000012                          10.
(1)  013666  000004                  $DBLK:  .DLKW   4
```

```
                                        ;SUBROUTINE TO TEST THE CLOCK REPEATIBILITY
                                        ;FIRST CLEAR CLOCK STATUS AND PRESET BUFFER
                                        ; THEN ENABLE THE CLOCK TO COUNT AT A RATE.
                                        ; DECREMENT R0 FOR SOME PERIOD OF TIME, WHEN R0 = 0
                                        ; SAVE THE COUNTER VALUE AND REPEAT THIS OPERATION AGAIN
                                        ;THEN COMPARE THE FIRST TIMED VALUE TO THE SECOND TIMED VALUE
                                        ; <MACHINE AND MEMORY TIMING NOT IMPORTANT>
                                        ; TO BE WITHIN THE VALUE SPECIFIED BY LOCATION CNTDEV
                                        ; IF GREATER THAN EXPECTED IT IS AN ERROR.
                                        ;ALSO TEST THAT THE COUNTER HAS REACHED A MIN. COUNT

013676  005077  165620          REPEAT: CLR     @#CSR                   ;STOP THE CLOCK
013702  005077  165616                  CLR     @#CSB                   ;CLEAR THE BUFFER
013706  012537  014056                  MOV     (R5)+,RATE              ;SET UP RATE
013712  012537  014060                  MOV     (R5)+,CNTDEV            ;SET UP CNT. DEV
013716  012537  014062                  MOV     (R5)+,MINCNT            ;SET UP MIN COUNT
013722  012537  014064                  MOV     (5)+,CKDLY              ;SAVE DELAY
013726  004737  014016                  JSR     PC,10$                  ;DUMMY TO CHARGE THE "CACHE"
013732  004737  014016                  JSR     PC,10$                  ;ENABLE THE CLOCK
013736  010037  001124                  MOV     R0,$GDDAT               ;SAVE FIRST TIME
013742  004737  014016                  JSR     PC,10$                  ;ENABLE THE CLOCK
013746  010037  001126                  MOV     R0,$DDDAT               ;SAVE SECOND TIME
013752  013700  001124                  MOV     $GDDAT,R0               ;GET FIRST TIME AGAIN
013756  163700  001126                  SUB     $GDDAT,R0               ;SUBTRACT SECOND TIME
013762  100001                          BPL     3$
013764  005400                          NEG     R0                      ;MAGNITUDE OF DIFFERENCE IN R0
013766  023737  001126  014062  3$:     CMP     $GDDAT,MINCNT           ;COMPARE TO MIN. COUNT
013774  002004                          BGE     4$                      ;BRANCH IF GREATER
013776  013737  014062  001124          MOV     MINCNT,$GDDAT           ;LOAD $GDDAT FOR TYPE-OUT
014004  000205                          RTS     R5
014006  005725                  4$:     TST     (R5)+                   ;UPDATE THE STACK
014010  020037  014060                  CMP     R0,CNTDEV               ;COMPARE TO DEVIATION
014014  000205                          RTS     R5

014016  013700  014064          10$:    MOV     CKDLY,R0                ;GET DELAY
014022  013777  014056  165472          MOV     RATE,@#CSR              ;LOAD RATE
014030  005277  165466                  INC     @#CSR                   ;ENABLE CLOCK
014034  005300                  1$:     DEC     R0                      ;DELAY
014036  001376                          BNE     1$
014040  017700  165470                  MOV     @#CSC,R0                ;READ COUNTER
014044  005077  165452                  CLR     @#CSR                   ;STOP CLOCK
014050  005077  165450                  CLR     @#CSB                   ;CLEAR BUFFER
014054  000207                          RTS     PC                      ;EXIT

014056  000000          RATE:   0                                       ;CLOCK RATE
014060  000000          CNTDEV: 0                                       ;CLOCK DEV.
014062  000000          MINCNT: 0                                       ;MIN. COUNT
014064  000000          CKDLY:  0
```

```
  1530  014066  020101  047524  042040  EM1:    .ASCIZ  /A TO D STATUS REGISTER IN ERROR/
        014074  051440  040524  052524
        014102  020123  042522  044507
        014110  052123  051105  044440
        014116  020116  051105  047522
        014124  000122
  1531  014126  020101  047524  042040  EM2:    .ASCIZ  /A TO D INTERRUPT ERROR/
        014134  044440  052116  051105
        014142  052522  052120  042440
        014150  051122  051117     000
  1532  014155     103  047514  045503  EM3:    .ASCIZ  /CLOCK STATUS REGISTER IN ERROR/
        014162  051440  040524  052524
        014170  020123  042522  044507
        014176  052123  051105  044440
        014204  020116  051105  047522
        014212  000122
  1533  014214  046103  041517  020113  EM4:    .ASCIZ  /CLOCK PRESET REGISTER IN ERROR/
        014222  051120  051505  052105
        014230  051040  043505  051511
        014236  042524  020122  047111
        014244  042440  051122  051117
        014252     000
  1534  014253     103  047514  045503  EM5:    .ASCIZ  /CLOCK INTERRUPT ERROR/
        014260  044440  052116  051105
        014266  052522  052120  042440
        014274  051122  051117     000
  1535  014301     103  047514  045503  EM6:    .ASCIZ  /CLOCK COUNTER REGISTER IN ERROR/
        014306  041440  052517  052116
        014314  051105  051040  043505
        014322  051511  042524  020122
        014330  047111  042440  051122
        014336  051117     000
  1536  014341     103  047514  045503  EM7:    .ASCIZ  /CLOCK COUNTED IN ERROR/
        014346  041440  052517  052116
        014354  042105  044440  020116
        014362  051105  047522  000122
  1537  014370  046103  041517  020113  EM10:   .ASCIZ  /CLOCK REPEATABILITY FAILED/
        014376  042522  042520  052101
        014404  041101  046111  052111
        014412  020131  040506  046111
        014420  042105     000
  1538
  1539  014423     126  020103  052123  EM11:   .ASCIZ  /VC STATUS REGISTER IN ERROR/
        014430  052101  051525  051040
        014436  043505  051511  042524
        014444  020122  047111  042440
        014452  051122  051117     000
  1540  014457     130  051040  043505  EM12:   .ASCIZ  /X REGISTER IN ERROR/
        014464  051511  042524  020122
        014472  047111  042440  051122
        014500  051117     000
  1541  014503     131  051040  043505  EM13:   .ASCIZ  /Y REGISTER IN ERROR/
        014510  051511  042524  020122
        014516  047111  042440  051122
        014524  051117     000
  1542  014527     123  047503  042520  EM14:   .ASCIZ  /SCOPE INTERRUPT ERROR/
```

```
        014534  044440  052116  051105
        014542  052522  052120  042440
        014550  051122  051117     000
  1543  014555     104  053105  041511  EM15:   .ASCIZ  /DEVICE BUS ERROR/
        014562  020105  052502  020123
        014570  051105  047522  000122
  1544
  1545  014576  047111  047503  051122  EM16:   .ASCIZ  "INCORRECT A/D BUFFER DATA"
        014604  041505  020124  027501
        014612  020104  052502  043106
        014620  051105  042040  052101
        014626  000101
  1546  014630  044103  047101  042507  EM17:   .ASCIZ  /CHANGED REGISTER IN ERROR/
        014636  020104  042522  044507
        014644  052123  051105  044440
        014652  020116  051105  047522
        014660  000122
  1547  014662  051105  050122  020103  DM1:    .ASCIZ  /ERRPC    ADCS   ADSTAT EXPECTED/
        014670  020040  040440  041504
        014676  020123  020040  042101
        014704  052123  052101  042440
        014712  050130  041505  042524
        014720  000104
  1548  014722  051105  050122  020103  DM2:    .ASCIZ  /ERRPC    ADCS/
        014730  020040  040440  041504
        014736  000123
  1549  014740  051105  050122  020103  DM3:    .ASCIZ  /ERRPC    CSR    CKSTAT EXPECTED/
        014746  020040  041440  051123
        014754  020040  020040  045503
        014762  052123  052101  042440
        014770  050130  041505  042524
        014776  000104
  1550  015000  051105  050122  020103  DM4:    .ASCIZ  /ERRPC    CSB    CKBUFF EXPECTED/
        015006  020040  041440  041123
        015014  020040  020040  045503
        015022  052502  043106  042440
        015030  050130  041505  042524
        015036  000104
  1551  015040  051105  050122  020103  DM5:    .ASCIZ  /ERRPC    CSR/
        015046  020040  041440  051123
        015054     000
  1552  015055     105  051122  041520  DM6:    .ASCIZ  /ERRPC    CSC    CKCNTR EXPECTED/
        015062  020040  020040  051503
        015070  020103  020040  041440
        015076  041513  052116  020122
        015104  054105  042520  052103
        015112  042105     000
  1553  015115     105  051122  041520  DM10:   .ASCIZ  /ERRPC    CSC    TIME1    TIME2/
        015122  020040  020040  051503
        015130  020103  020040  052040
        015136  046511  030505  020040
        015144  052040  046511  051105
        015152     000
  1554  015153     105  051122  041520  DM11:   .ASCIZ  /ERRPC    VCADR  VCSTAT EXPECTED/
        015160  020040  053040  040503
        015166  051104  020040  053040
```

```
                015174  051503  040524  020124
                015202  054105  042520  052103
                015210  042105      000
1555            015213      105  051122  041520  DM12:   .ASCIZ  /ERRPC   VCXREG  X POS.  EXPECTED/
                015220  020040  053040  054103
                015226  042522  020107  054040
                015234  050040  051517  020056
                015242  042440  050130  041505
                015250  042524  000104
1556            015254  051105  050122  020103  DM13:   .ASCIZ  /ERRPC   VCYREG  Y POS.  EXPECTED/
                015262  020040  041526  051131
                015270  043505  020040  020131
                015276  047520  027123  020040
                015304  054105  042520  052103
                015312  042105      000
1557            015315      105  051122  041520  DM14:   .ASCIZ  /ERRPC   VCSTAT/
                015322  020040  053040  051503
                015330  040524  000124
1558            015334  051105  050122  020103  DM15:   .ASCIZ  /ERRPC    BASE    ACTUAL/
                015342  020040  041040  051501
                015350  020105  020040  040440
                015356  052103  040525  000114
1559            015364  051105  050122  020103  DM16:   .ASCIZ  /ERRPC   ADDBR    READ    EXPECTED/
                015372  020040  042101  041104
                015400  020122  020040  051040
                015406  040505  020104  020040
                015414  054105  042520  052103
                015422  042105      000
1560            015425      105  051122  041520  DM17:   .ASCIZ  /ERRPC   BUFADR   READ    EXPECTED/
                015432  020040  041040  043125
                015440  042101  020122  020040
                015446  042522  042101  020040
                015454  042440  050130  041505
                015462  042524  000104
1561                                                    .EVEN
1562            015466  001116  001516  001126  DT1:    SERRPC,ADCS,SBDDAT,SGDDAT,0
                015474  001124  000000
1563            015500  001116  001516  000000  DT2:    SERRPC,ADCS,0
1564            015506  001116  001522  001126  DT3:    SERRPC,CSR,SBDDAT,SGDDAT,0
                015514  001124  000000
1565            015520  001116  001524  001126  DT4:    SERRPC,CSB,SBDDAT,SGDDAT,0
                015526  001124  000000
1566            015532  001116  001522  000000  DT5:    SERRPC,CSR,0
1567            015540  001116  001534  001126  DT6:    SERRPC,CSC,SBDDAT,SGDDAT,0
                015546  001124  000000
1568            015552  001116  001526  001126  DT11:   SERRPC,VCSTAT,SBDDAT,SGDDAT,0
                015560  001124  000000
1569            015564  001116  001530  001126  DT12:   SERRPC,VCXREG,SBDDAT,SGDDAT,0
                015572  001124  000000
1570            015576  001116  001532  001126  DT13:   SERRPC,VCYREG,SBDDAT,SGDDAT,0
                015604  001124  000000
1571            015610  001116  001526  000000  DT14:   SERRPC,VCSTAT,0
1572            015616  001116  001552  001126  DT15:   SERRPC,ARBADD,SBDDAT,0
                015624  000000
1573            015626  001116  001520  001126  DT16:   SERRPC,ADDBR,SBDDAT,SGDDAT,0
                015634  001124  000000
```

```
 1574  015640  001116  013162  001126  DT17:   SERRPC,BUFADR,SRDDAT,SGDDAT,0
       015644  001124  000000
```

```
 1576
 1577
  (1)                                   .SBTTL  SCOPE HANDLER ROUTINE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
  (1)                                   ;*AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
  (1)                                   ;*AND LOAD THE ERROR FLAG (SERFLG) 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)  015652                           $SCOPE:
  (2)  015652  104405                           CKSWR
  (1)  015654  032777  040000  163254   18$:    BIT     #BIT14,@SWR       ;;LOOP ON PRESENT TEST?
  (1)  015662  001114                           BNE     $OVER             ;;YES IF SW14=1
  (1)                                   ;;;;;;START OF CODE FOR THE XOR TESTER;;;;;;
  (1)  015664  000416                   $XTSTR: BR      6$                ;;IF RUNNING ON THE "XOR" TESTER CHANGE
  (1)                                                                     ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
  (1)  015666  013746  000004                   MOV     @#ERRVEC,-(SP)    ;;SAVE THE CONTENTS OF THE ERROR VECTOR
  (1)  015672  012737  015712  000004           MOV     #5$,@#ERRVEC      ;;SET FOR TIMEOUT
  (1)  015700  005737  177060                   TST     @#177060          ;;TIME OUT ON XOR?
  (1)  015704  012637  000004                   MOV     (SP)+,@#ERRVEC    ;;RESTORE THE ERROR VECTOR
  (1)  015710  000463                           BR      $SVLAD            ;;GO TO THE NEXT TEST
  (1)  015712  022626                   5$:     CMP     (SP)+,(SP)+       ;;CLEAR THE STACK AFTER A TIME OUT
  (1)  015714  012637  000004                   MOV     (SP)+,@#ERRVEC    ;;RESTORE THE ERROR VECTOR
  (1)  015720  000423                           BR      7$                ;;LOOP ON THE PRESENT TEST
  (1)  015722                           6$:;;;;;END OF CODE FOR THE XOR TESTER;;;;;;
  (1)  015722  032777  000400  163206           BIT     #BIT08,@SWR       ;;LOOP ON SPEC. TEST?
  (1)  015730  001404                           BEQ     2$                ;;BR IF NO
  (1)  015732  127737  163200  001102           CMPB    @SWR,STSTNM       ;;ON THE RIGHT TEST?    SWR<7:0>
  (1)  015740  001465                           BEQ     $OVER             ;;BR IF YES
  (1)  015742  105737  001103           2$:     TSTB    SERFLG            ;;HAS AN ERROR OCCURRED?
  (1)  015746  001421                           BEQ     3$                ;;BR IF NO
  (1)  015750  123737  001115  001103           CMPB    SERMAX,SERFLG     ;;MAX. ERRORS FOR THIS TEST OCCURRED?
  (1)  015756  101015                           BHI     3$                ;;BR IF NO
  (1)  015760  032777  001000  163150           BIT     #BIT09,@SWR       ;;LOOP ON ERROR?
  (1)  015766  001404                           BEQ     4$                ;;BR IF NO
  (1)  015770  013737  001110  001106   7$:     MOV     $LPERR,$LPADR     ;;SET LOOP ADDRESS TO LAST SCOPE
  (1)  015776  000446                           BR      $OVER
  (1)  016000  105037  001103           4$:     CLRB    SERFLG            ;;ZERO THE ERROR FLAG
  (1)  016004  005037  001164                   CLR     $TIMES            ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
  (1)  016010  000415                           BR      1$                ;;ESCAPE TO THE NEXT TEST
  (1)  016012  032777  004000  163116   5$:     BIT     #BIT11,@SWR       ;;INHIBIT ITERATIONS?
  (1)  016020  001011                           BNE     1$                ;;BR IF YES
  (1)  016022  005737  001202                   TST     $PASS             ;;IF FIRST PASS OF PROGRAM
  (1)  016026  001406                           BEQ     1$                ;;      INHIBIT ITERATIONS
  (1)  016030  005237  001104                   INC     $ICNT             ;;INCREMENT ITERATION COUNT
  (1)  016034  023737  001164  001104           CMP     $TIMES,$ICNT      ;;CHECK THE NUMBER OF ITERATIONS MADE
  (1)  016042  002024                           BGE     $OVER             ;;BR IF MORE ITERATION REQUIRED
  (1)  016044  012737  000001  001104   1$:     MOV     #1,$ICNT          ;;REINITIALIZE THE ITERATION COUNTER
  (1)  016052  013737  016130  001164           MOV     $MXCNT,$TIMES     ;;SET NUMBER OF ITERATIONS TO DO
```

K5

```
(1)  016060  105237  001102          $SVLAD: INCB    STSTNM          ;;COUNT TEST NUMBERS
(1)  016064  113737  001102  001200          MOVB    STSTNM,STESTN   ;;SET TEST NUMBER IN APT MAILBOX
(1)  016072  011637  001106                  MOV     (SP),SLPADR     ;;SAVE SCOPE LOOP ADDRESS
(1)  016076  011637  001110                  MOV     (SP),SLPERR     ;;SAVE ERROR LOOP ADDRESS
(1)  016102  005037  001164                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
(1)  016106  112737  000001  001115          MOVB    #1,SERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
(1)  016114  013777  001102  163016  $OVER:  MOV     STSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
(1)  016122  013716  001106                  MOV     SLPADR,(SP)     ;;FUDGE RETURN ADDRESS
(1)  016126  000002                          RTI                     ;;FIXES PS
(1)  016130  003720                  $MXCNT: 2000.                   ;;MAX. NUMBER OF ITERATIONS
1570
(1)                                  .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)                                  ;*SW09=1        LOOP ON ERROR
(1)                                  ;*CALL
(1)                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
(1)
(1)  016132                          $ERROR:
(1)  016132  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
(1)  016136  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
(1)  016140  013777  001102  162772          MOV     STSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
(1)  016146  005237  001112                  INC     $ERTTL          ;;INC THE ERROR COUNT
(1)  016152  011637  001116                  MOV     (SP),$ERRPC     ;;SET ADDRESS OF ERROR INSTRUCTION
(1)  016156  162737  000002  001116          SUB     #2,$ERRPC
(1)  016164  117737  162726  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
(1)  016172  032777  020000  162736          BIT     #BIT13,@$SWR    ;;SKIP TYPEOUT IF SET
(1)  016200  001004                          BNE     20$             ;;SKIP TYPEOUTS
(1)  016202  004737  017106                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
(1)  016206  104400  001171                  TYPE    ,$CRLF
(1)  016212                          20$:
(1)  016212  122737  000001  001214          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
(1)  016220  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
(1)  016222  113737  001114  016234          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
(1)  016230  004737  020210                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
(1)  016234     000                  21$:    .BYTE   0
(1)  016235     000                          .BYTE   0
(1)  016236  000777                  22$:    BR      22$             ;;APT ERROR LOOP
(1)  016240  005777  162672          2$:     TST     @$SWR           ;;HALT ON ERROR
(1)  016244  100001                          BPL     3$              ;;SKIP IF CONTINUE
(1)  016246  000000                          HALT                    ;;HALT ON ERROR)
(1)  016250  032777  001000  162660  3$:     BIT     #BIT09,@$SWR    ;;LOOP ON ERROR SWITCH SET?
(1)  016256  001402                          BEQ     4$              ;;BR IF NO
(1)  016260  013716  001110                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
(1)  016264  005737  001166          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
(1)  016270  001402                          BEQ     5$              ;;BR IF NONE
(1)  016272  013716  001166                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
(1)  016276                          5$:
(1)  016276  022737  013342  000042          CMP     @#ENDAD,#42     ;;ACT-11 AUTO-ACCEPT?
(1)  016304  001001                          BNE     6$              ;;BRANCH IF NO
```

```
(1)  016306  000000                    HALT                    ;;YES
(1)  016310                      68$:
(1)  016310  000002                    RTI                     ;;RETURN
1579
(1)                              .SBTTL  TTY INPUT ROUTINE
(1)
(2)                              ;;**********************************************************************
(1)                              ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
(1)                              ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
(1)                              ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
(1)                              ;*WHEN OPERATING IN TTY FLAG MODE.
(1)  016312  022737  000176  001136  SCKSWR: CMP     @#SWREG,SWR     ;;IS THE SOFT-SWR SELECTED?
(1)  016320  001073                    BNE     14$             ;;BRANCH IF NO
(1)  016322  105777  162614            TSTB    @#STKS          ;;CHAR THERE?
(1)  016326  100070                    BPL     14$             ;;IF NO, DON'T WAIT AROUND
(1)  016330  117746  162610      28$:  MOVB    @#STKB,-(SP)    ;;SAVE THE CHAR
(1)  016334  042716  177600            BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
(1)  016340  022726  000007            CMP     #7,(SP)+        ;;IS IT A CONTROL G?
(1)  016344  001061                    BNE     14$             ;;NO, RETURN TO USER
(1)  016346  104400  016755            TYPE    ,SCNTLG         ;;YES, ECHO CONTROL G
(1)
(1)  016352  104400  016762      68$:  TYPE    ,SWSWR          ;;TYPE CURRENT CONTENTS
(2)  016356  013746  000176            MOV     @#SWREG,-(SP)   ;;SAVE SWREG FOR TYPEOUT
(2)  016362  104401                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
(1)  016364  104400  016773            TYPE    ,SWNEW          ;;PROMPT FOR NEW SWR
(1)  016370  005046                    CLR     -(SP)           ;;CLEAR COUNTER
(1)  016372  005046                    CLR     -(SP)           ;;THE NEW SWR
(1)  016374  104406              78$:  RDCHR                   ;;GET NEXT CHAR
(1)
(1)  016376  022716  000025      88$:  CMP     #25,(SP)        ;;IS IT A CONTROL U?
(1)  016402  001005                    BNE     9$              ;;BRANCH IF NO
(1)  016404  104400  016750            TYPE    ,SCNTLU         ;;YES, ECHO IT
(1)  016410  062706  000006            ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
(1)  016414  000756                    BR      6$              ;;LET'S TRY IT AGAIN
(1)
(1)  016416  022716  000015      98$:  CMP     #15,(SP)        ;;IS IT A <CR>?
(1)  016422  001011                    BNE     11$             ;;BRANCH IF NO
(1)  016424  005766  000004            TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
(1)  016430  001403                    BEQ     10$             ;;BRANCH IF YES
(1)  016432  016677  000002  162476    MOV     2(SP),@#SWR     ;;SAVE NEW SWR
(1)  016440  062706  000006      108:  ADD     #6,SP           ;;CLEAR UP STACK
(1)  016444  000417                    BR      13$             ;;RETURN TO USER
(1)  016446  022716  000012      118:  CMP     #12,(SP)        ;;IS IT A <LF>
(1)  016452  001017                    BNE     15$             ;;BRANCH IF NO
(1)  016454  005766  000004            TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
(1)  016460  001403                    BEQ     12$             ;;YES
(1)  016462  016677  000002  162446    MOV     2(SP),@#SWR     ;;SAVE NEW SWR
(1)  016470  062706  000006      128:  ADD     #6,SP           ;;CLEAR UP STACK
(1)  016474  013716  000046            MOV     @#46,(SP)       ;;GET RESTART
(1)  016500  062716  000010            ADD     #10,(SP)        ;;ADDRESS
(1)  016504  104400  001171      138:  TYPE    ,SCRLF          ;;ECHO <CR> AND <LF>
(1)  016510  000002              148:  RTI                     ;;RETURN
(1)  016512  004737  020122      158:  JSR     PC,@#TYPEC      ;;ECHO CHAR
(1)  016516  042726  177770            BIC     #177770,(SP)+   ;;RESTRICT TO 0-7
(1)  016522  005766  000002            TST     2(SP)           ;;IS THIS THE FIRST CHAR
(1)  016526  001403                    BEQ     16$             ;;BRANCH IF YES
```

M5

```
(1)  016538  006316                    ASL     (SP)            ;;NO, SHIFT PRESENT
(1)  016532  006316                    ASL     (SP)            ;;   CHAR OVER TO MAKE
(1)  016534  006316                    ASL     (SP)            ;;   ROOM FOR NEW ONE.
(1)  016536  005266  000002    168$:   INC     2(SP)           ;;KEEP COUNT OF CHAR
(1)  016542  056616  177776            BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
(1)  016546  000712                    BR      7$              ;;GET THE NEXT ONE
(2)                                 ;;**************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
(1)                                 ;*                              ;;WITH PARITY BIT STRIPPED OFF
(1)                                 ;
(1)
(1)  016550  011646    SRDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
(1)  016552  016666  000004  000002    MOV     4(SP),2(SP)     ;;SAVE THE PS
(1)  016560  105777  162356    1$:     TSTB    @#TKS           ;;WAIT FOR
(1)  016564  100375                    BPL     1$              ;;A CHARACTER
(1)  016566  117766  162352  000004    MOVB    @#TKB,4(SP)     ;;READ THE TTY
(1)  016574  042766  177600  000004    BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
(1)  016602  026627  000004  000140    CMP     4(SP),#140      ;;IS IT UPPER CASE?
(1)  016610  002407                    BLT     2$              ;;BRANCH IF YES
(1)  016612  026627  000004  000175    CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
(1)  016620  003003                    BGT     2$              ;;BRANCH IF YES
(1)  016622  042766  000040  000004    BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
(1)  016630  000002            2$:     RTI                     ;;GO BACK TO USER
(2)                                 ;;**************************************************************
(1)                                 ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
(1)                                 ;*CALL:
(1)                                 ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
(1)                                 ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
(1)                                 ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
(1)
(1)  016632  010346    SRDLIN: MOV     R3,-(SP)        ;;SAVE R3
(1)  016634  012703  016740    1$:     MOV     #TTYIN,R3       ;;GET ADDRESS
(1)  016640  022703  016750    2$:     CMP     #TTYIN+8.,R3    ;;BUFFER FULL?
(1)  016644  101405                    BLOS    4$              ;;BR IF YES
(1)  016646  104406                    RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
(1)  016650  112613                    MOVB    (SP)+,(R3)      ;;GET CHARACTER
(1)  016652  122713  000177    10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
(1)  016656  001003                    BNE     3$              ;;SKIP IF NOT
(1)  016660  104400  001170    4$:     TYPE    ,#QUES          ;;TYPE A '?'
(1)  016664  000763                    BR      1$              ;;CLEAR THE BUFFER AND LOOP
(1)  016666  111337  016736    3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
(1)  016672  104400  016736            TYPE    ,9$
(1)  016676  122723  000015            CMPB    #15,(R3)+       ;;CHECK FOR RETURN
(1)  016702  001356                    BNE     2$              ;;LOOP IF NOT RETURN
(1)  016704  105063  177777            CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
(1)  016710  104400  001172            TYPE    ,#LF            ;;TYPE A LINE FEED
(1)  016714  012603                    MOV     (SP)+,R3        ;;RESTORE R3
(1)  016716  011646                    MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1)  016720  016666  000004  000002    MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
(1)  016726  012766  016740  000004    MOV     #TTYIN,4(SP)
(1)  016734  000002                    RTI                     ;;RETURN
(1)  016736     000            9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
(1)  016737     000                    .BYTE   0               ;;TERMINATOR
```

```
   (1)  016740  000010                    STTYIN: .BLKB   8.                  ;;RESERVE 8 BYTES FOR TTY INPUT
   (1)  016750  052536  005015     000    SCNTLU: .ASCIZ  /^U/<15><12>        ;;CONTROL "U"
   (1)  016755     136  006507  000012    SCNTLG: .ASCIZ  /^G/<15><12>        ;;CONTROL "G"
   (1)  016762  005015  053523  020122    SMSWR:  .ASCIZ  <15><12>/SWR = /
   (1)  016770  020075     000
   (1)  016773     040  047040  053505    SMNEW:  .ASCIZ  /  NEW = /
   (1)  017000  036440  000040
  1500
   (1)                                    .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
   (1)
   (2)                                    ;;**************************************************************
   (1)                                    ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
   (1)                                    ;*CHANGE IT TO BINARY.
   (1)                                    ;*CALL:
   (1)                                    ;*      RDOCT                   ;;READ AN OCTAL NUMBER
   (1)                                    ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
   (1)                                    ;*                              ;;HIGH ORDER BITS ARE IN $NIOCT
   (1)
   (1)  017004  011646                    $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
   (1)  017006  016666  000004  000002            MOV     4(SP),2(SP)     ;;INPUT NUMBER
   (3)  017014  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
   (3)  017016  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
   (3)  017020  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
   (1)  017022  104407                    18:     RDLIN                   ;;READ AN ASCIZ LINE
   (1)  017024  012600                            MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
   (1)  017026  005001                            CLR     R1              ;;CLEAR DATA WORD
   (1)  017030  005002                            CLR     R2
   (1)  017032  112046                    28:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
   (1)  017034  001412                            BEQ     38              ;;IF ZERO GET OUT
   (1)  017036  006301                            ASL     R1              ;;*2
   (1)  017040  006102                            ROL     R2
   (1)  017042  006301                            ASL     R1              ;;*4
   (1)  017044  006102                            ROL     R2
   (1)  017046  006301                            ASL     R1              ;;*8
   (1)  017050  006102                            ROL     R2
   (1)  017052  042716  177770                    BIC     #^C7,(SP)       ;;STRIP THE ASCII JUNK
   (1)  017056  062601                            ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
   (1)  017060  000764                            BR      28              ;;LOOP
   (1)  017062  005726                    38:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
   (1)  017064  010166  000012                    MOV     R1,12(SP)       ;;SAVE THE RESULT
   (1)  017070  010237  017104                    MOV     R2,$NIOCT
   (3)  017074  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
   (3)  017076  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
   (3)  017100  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
   (1)  017102  000002                            RTI                     ;;RETURN
   (1)  017104  000000                    $NIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
```

```
 1582
 1583
  (1)                                   .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
  (1)
  (2)                                   ;;**********************************************************************
  (1)                                   ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
  (1)                                   ;*ERROR IS TO BE REPORTED, IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
  (1)                                   ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
  (1)
  (1)  017106                           SERRTYP:
  (1)  017106  104400  001171                   TYPE    ,SCRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  017112  010046                           MOV     R0,-(SP)        ;;SAVE R0
  (1)  017114  005000                           CLR     R0              ;;PICKUP THE ITEM INDEX
  (1)  017116  153700  001114                   BISB    @#SITEMB,R0
  (1)  017122  001004                           BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
  (1)                                                                   ;;TYPE THE PC OF THE ERROR
  (2)  017124  013746  001116                   MOV     SERRPC,-(SP)    ;;SAVE SERRPC FOR TYPEOUT
  (2)                                                                   ;;ERROR ADDRESS
  (2)  017130  104401                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  (1)  017132  000426                           BR      6$              ;;GET OUT
  (1)  017134  005300                   1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
  (1)  017136  006300                           ASL     R0              ;;      WORK FOR THE ERROR TABLE
  (1)  017140  006300                           ASL     R0
  (1)  017142  006300                           ASL     R0
  (1)  017144  062700  001320                   ADD     #SERRTB,R0      ;;FORM TABLE POINTER
  (1)  017150  012037  017160                   MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
  (1)  017154  001404                           BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
  (1)  017156  104400                           TYPE                    ;;TYPE THE "ERROR MESSAGE"
  (1)  017160  000000                   2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
  (1)  017162  104400  001171                   TYPE    ,SCRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  017166  012037  017176           3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
  (1)  017172  001404                           BEQ     5$              ;;SKIP TYPEOUT IF 0
  (1)  017174  104400                           TYPE                    ;;TYPE THE "DATA HEADER"
  (1)  017176  000000                   4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
  (1)  017200  104400  001171                   TYPE    ,SCRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  017204  011000                   5$:     MOV     (R0),R0         ;;PICKUP "DATA TABLE" POINTER
  (1)  017206  001004                           BNE     7$              ;;GO TYPE THE DATA
  (1)  017210  012600                   6$:     MOV     (SP)+,R0        ;;RESTORE R0
  (1)  017212  104400  001171                   TYPE    ,SCRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
  (1)  017216  000207                           RTS     PC              ;;RETURN
  (2)  017220                           7$:
  (2)  017220  013046                           MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
  (2)  017222  104401                           TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
  (1)  017224  005710                           TST     (R0)            ;;IS THERE ANOTHER NUMBER?
  (1)  017226  001770                           BEQ     6$              ;;BR IF NO
  (1)  017230  104400  017236                   TYPE    ,8$             ;;TYPE TWO(2) SPACES
  (1)  017234  000771                           BR      7$              ;;LOOP
  (1)  017236  020040      000          8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
  (1)           017242                          .EVEN
```

```
1585
1586
 (1)                                    .SBTTL  POWER DOWN AND UP ROUTINES
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;POWER DOWN ROUTINE
 (1)  017242  012737  017406  000024    $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
 (1)  017250  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
 (3)  017256  010046                            MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 (3)  017260  010146                            MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 (3)  017262  010246                            MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 (3)  017264  010346                            MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 (3)  017266  010446                            MOV     R4,-(SP)        ;;PUSH R4 ON STACK
 (3)  017270  010546                            MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 (3)  017272  017746  161640                    MOV     @#PSW,-(SP)     ;;PUSH @#PSW ON STACK
 (1)  017276  010637  017412                    MOV     SP,$SAVR6       ;;SAVE SP
 (1)  017302  012737  017314  000024            MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
 (1)  017310  000000                            HALT
 (1)  017312  000776                            BR      .-2             ;;HANG UP
 (1)
 (2)                                    ;;*****************************************************************
 (1)                                    ;POWER UP ROUTINE
 (1)  017314  012737  017406  000024    $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
 (1)  017322  013706  017412                    MOV     $SAVR6,SP       ;;GET SP
 (1)  017326  005037  017412                    CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
 (1)  017332  005237  017412            1$:     INC     $SAVR6          ;;WAIT FOR THE INC
 (1)  017336  001375                            BNE     1$              ;;OF  WORD
 (3)  017340  012677  161572                    MOV     (SP)+,@#PSW     ;;POP STACK INTO @#PSW
 (3)  017344  012605                            MOV     (SP)+,R5        ;;POP STACK INTO R5
 (3)  017346  012604                            MOV     (SP)+,R4        ;;POP STACK INTO R4
 (3)  017350  012603                            MOV     (SP)+,R3        ;;POP STACK INTO R3
 (3)  017352  012602                            MOV     (SP)+,R2        ;;POP STACK INTO R2
 (3)  017354  012601                            MOV     (SP)+,R1        ;;POP STACK INTO R1
 (3)  017356  012600                            MOV     (SP)+,R0        ;;POP STACK INTO R0
 (1)  017360  012737  017242  000024            MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
 (1)  017366  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;PRIO:7
 (1)  017374  104400                            TYPE                    ;;REPORT THE POWER FAILURE
 (1)  017376  017414                    $PWRMG: .WORD   PWRMSG          ;;POWER FAIL MESSAGE POINTER
 (1)  017400  012716                            MOV     (PC)+,(SP)      ;;RESTART AT BEGIN1
 (1)  017402  001602                    $PWRAD: .WORD   BEGIN1          ;;RESTART ADDRESS
 (1)  017404  000002                            RTI
 (1)  017406  000000                    $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
 (1)  017410  000776                            BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
 (1)  017412  000000                    $SAVR6: 0                       ;;PUT THE SP HERE
1587  017414  005015  042522  052123    PWRMSG: .ASCIZ  <15><12>/RESTARTING AFTER A POWER FAILURE/<15><12><12>
      017422  051101  044524  043516
      017430  040440  052106  051105
      017436  040440  050040  053517
      017444  051105  043040  044501
      017452  052514  042522  005015
      017460  000012
1588                                    .EVEN
```

```
1590
1591
 (1)                                .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)                                ;*STYPOS---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)                                ;*STYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 (1)                                ;*STYPOS OR STYPOC
 (1)                                ;*CALL:
 (1)                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 (1)                                ;*      TYPON                   ;;CALL FOR TYPEOUT
 (1)                                ;*
 (1)                                ;*STYPOC---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)  017462  017646  000000  STYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 (1)  017466  116637  000001  017785  MOVB    1(SP),$0FILL    ;;LOAD ZERO FILL SWITCH
 (1)  017474  112637  017787          MOVB    (SP)+,$0MODE+1  ;;NUMBER OF DIGITS TO TYPE
 (1)  017500  062716  000002          ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 (1)  017504  000406                  BR      STYPON
 (1)  017506  112737  000001  017785  STYPOC: MOVB    #1,$0FILL       ;;SET THE ZERO FILL SWITCH
 (1)  017514  112737  000006  017787          MOVB    #6,$0MODE+1     ;;SET FOR SIX(6) DIGITS
 (1)  017522  112737  000005  017784  STYPON: MOVB    #5,$0CNT        ;;SET THE ITERATION COUNT
 (1)  017530  010346                  MOV     R3,-(SP)        ;;SAVE R3
 (1)  017532  010446                  MOV     R4,-(SP)        ;;SAVE R4
 (1)  017534  010546                  MOV     R5,-(SP)        ;;SAVE R5
 (1)  017536  113704  017787          MOVB    $0MODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 (1)  017542  005404                  NEG     R4
 (1)  017544  062704  000006          ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 (1)  017550  110437  017786          MOVB    R4,$0MODE       ;;SAVE IT FOR USE
 (1)  017554  113704  017785          MOVB    $0FILL,R4       ;;SET THE ZERO FILL SWITCH
 (1)  017560  016605  000012          MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 (1)  017564  005003                  CLR     R3              ;;CLEAR THE OUTPUT WORD
 (1)  017566  006105          18:     ROL     R5              ;;ROTATE MSB INTO "C"
 (1)  017570  000404                  BR      38              ;;GO DO MSB
 (1)  017572  006105          28:     ROL     R5              ;;FORM THIS DIGIT
 (1)  017574  006105                  ROL     R5
 (1)  017576  006105                  ROL     R5
 (1)  017600  010503                  MOV     R5,R3
 (1)  017602  006103          38:     ROL     R3              ;;GET LSB OF THIS DIGIT
 (1)  017604  105337  017786          DECB    $0MODE          ;;TYPE THIS DIGIT?
 (1)  017610  100016                  BPL     78              ;;BR IF NO
 (1)  017612  042703  177770          BIC     #177770,R3      ;;GET RID OF JUNK
 (1)  017616  001002                  BNE     48              ;;TEST FOR 0
```

```
   (1)  017620  005704                        TST     R4              ;;SUPPRESS THIS 0?
   (1)  017622  001403                        BEQ     5$              ;;BR IF YES
   (1)  017624  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
   (1)  017626  052703  000060               BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
   (1)  017632  052703  000040        5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
   (1)  017636  110337  017702               MOVB    R3,6$           ;;SAVE FOR TYPING
   (1)  017642  104400  017702               TYPE    ,6$             ;;GO TYPE THIS DIGIT
   (1)  017646  105337  017704        7$:     DECB    $OCNT           ;;COUNT BY 1
   (1)  017652  003347                        BGT     2$              ;;BR IF MORE TO DO
   (1)  017654  002402                        BLT     6$              ;;BR IF DONE
   (1)  017656  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
   (1)  017660  000744                        BR      2$              ;;GO DO THE LAST DIGIT
   (1)  017662  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
   (1)  017664  012604                        MOV     (SP)+,R4        ;;RESTORE R4
   (1)  017666  012603                        MOV     (SP)+,R3        ;;RESTORE R3
   (1)  017670  016666  000002  000004        MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
   (1)  017676  012616                        MOV     (SP)+,(SP)
   (1)  017700  000002                        RTI                     ;;RETURN
   (1)  017702     000                6$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
   (1)  017703     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
   (1)  017704     000                $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
   (1)  017705     000                $BFILL: .BYTE   0               ;;ZERO FILL SWITCH
   (1)  017706  000000                $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 1592
   (1)                                .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)  017710  105737  001155        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
   (1)  017714  100002                        BPL     1$              ;;BR IF YES
   (1)  017716  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
   (1)  017720  000430                        BR      3$              ;;LEAVE
   (1)  017722  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
   (1)  017724  017600  000002               MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
   (1)  017730  122737  000001  001214        CMPB    @APTENV,$ENV    ;;RUNNING IN APT MODE
   (1)  017736  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
   (1)  017740  132737  000100  001215        BITB    @APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
   (1)  017746  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
   (1)  017750  010037  017760               MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
   (1)  017754  004737  020200               JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
   (1)  017760  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
   (1)  017762  132737  000040  001215  62$:  BITB    @APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
   (1)  017770  001003                        BNE     6$              ;;YES,SKIP TYPE OUT
```

F6

```
(1)  020772  112046              28$:    MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  020774  001005                      BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
(1)  020776  005726                      TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
(1)  021000  012600              60$:    MOV     (SP)+,R0        ;;RESTORE R0
(1)  021002  062716  000002      3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
(1)  021006  000002                      RTI                     ;;RETURN
(1)  021010  122716  000011      4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
(1)  021014  001430                      BEQ     8$
(1)  021016  122716  000200              CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
(1)  021022  001006                      BNE     5$
(1)  021024  005726                      TST     (SP)+           ;;POP  <CR><LF> EQUIV
(1)  021026  104400                      TYPE                    ;;TYPE A CR AND LF
(1)  021030  001171                      $CRLF
(1)  021032  105037  021166              CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
(1)  021036  000755                      BR      2$              ;;GET NEXT CHARACTER
(1)  021040  004737  021122      5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
(1)  021044  123726  001154      6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
(1)  021050  001350                      BNE     2$              ;;IF NO GO GET NEXT CHAR.
(1)  021052  013746  001152              MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
(1)                                                              ;;AND THE NULL CHAR.
(1)  021056  105366  000001      7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
(1)  021062  002770                      BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  021064  004737  021122              JSR     PC,$TYPEC       ;;GO TYPE A NULL
(1)  021070  105337  021166              DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
(1)  021074  000770                      BR      7$              ;;LOOP
(1)
(1)                              ;HORIZONTAL TAB PROCESSOR
(1)
(1)  021076  112716  000040      8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
(1)  021102  004737  021122      9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
(1)  021106  132737  000007  021166      BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
(1)  021114  001372                      BNE     9$              ;;TAB STOP
(1)  021116  005726                      TST     (SP)+           ;;POP SPACE OFF STACK
(1)  021120  000724                      BR      2$              ;;GET NEXT CHARACTER
(1)  021122  105777  161020      $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
(1)  021126  100375                      BPL     $TYPEC
(1)  021130  116677  000002  161012      MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  021136  122766  000015  000002      CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
(1)  021144  001003                      BNE     1$              ;;BRANCH IF NO
(1)  021146  105037  021166              CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
(1)  021152  000406                      BR      $TYPEX          ;;EXIT
(1)  021154  122766  000012  000002  1$: CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
(1)  021162  001402                      BEQ     $TYPEX          ;;BRANCH IF YES
(1)  021164  105227                      INCB    (PC)+           ;;COUNT THE CHARACTER
(1)  021166  000000              $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
(1)  021170  000207              $TYPEX: RTS     PC
(1)
1593
(1)                              .SBTTL  APT COMMUNICATIONS ROUTINE
(1)
(2)                              ;;**************************************************************
(1)  021172  112737  000001  021436  $ATV1: MOVB    #1,$PFLG        ;;TO REPORT FATAL ERROR
(1)  021200  112737  000001  021434  $ATV3: MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
(1)  021206  000403                      BR      $ATVC
(1)  021210  112737  000001  021436  $ATV4: MOVB    #1,$PFLG        ;;TO ONLY REPORT FATAL ERROR
(2)  021216                      $ATVC:
```

```
(3)  020216  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
(3)  020220  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
(1)  020222  105737  020434                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
(1)  020226  001450                        BEQ     5$              ;;IF NOT!  BR
(1)  020230  122737  000001  001214        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
(1)  020236  001031                        BNE     3$              ;;IF NOT!  BR
(1)  020240  132737  000100  001215        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
(1)  020246  001425                        BEQ     3$              ;;IF NOT!  BR
(1)  020250  017600  000004                MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
(1)  020254  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
(1)  020262  005737  001174          1$:   TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
(1)  020266  001375                        BNE     1$              ;;IF NOT!  WAIT
(1)  020270  010037  001210                MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
(1)  020274  105720                  2$:   TSTB    (R0)+           ;;FIND END OF MESSAGE
(1)  020276  001376                        BNE     2$
(1)  020300  163700  001210                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
(1)  020304  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
(1)  020306  010037  001212                MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
(1)  020312  012737  000004  001174        MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
(1)  020320  000413                        BR      5$
(1)  020322  017637  000004  020346  3$:   MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
(1)  020330  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
(3)  020336  013746  177776                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
(1)  020342  004737  017710                JSR     PC,$TYPE        ;;CALL TYPE MACRO
(1)  020346  000000                  4$:   .WORD   0
(1)  020350                          5$:
(1)  020350  105737  020436          10$:  TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
(1)  020354  001416                        BEQ     12$             ;;IF NOT!  BR
(1)  020356  005737  001214                TST     $ENV            ;;RUNNING UNDER APT?
(1)  020362  001413                        BEQ     12$             ;;IF NOT!  BR
(1)  020364  005737  001174          11$:  TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
(1)  020370  001375                        BNE     11$             ;;IF NOT!  WAIT
(1)  020372  017637  000004  001176        MOV     @4(SP),$FATAL   ;;GET ERROR #
(1)  020400  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
(1)  020406  005237  001174                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
(1)  020412  105037  020436          12$:  CLRB    $FFLG           ;;CLEAR FATAL FLAG
(1)  020416  105037  020435                CLRB    $LFLG           ;;CLEAR LOG FLAG
(1)  020422  105037  020434                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
(3)  020426  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
(3)  020430  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
(1)  020432  000207                        RTS     PC              ;;RETURN
(1)  020434  000                    $MFLG:  .BYTE   0               ;;MESSG. FLAG
(1)  020435  000                    $LFLG:  .BYTE   0               ;;LOG FLAG
(1)  020436  000                    $FFLG:  .BYTE   0               ;;FATAL FLAG
(1)                                        .EVEN
(1)                                 APTSIZE=200
(1)                                 APTENV=001
(1)                                 APTSPOOL=100
(1)                                 APTCSUP=040
1594
(1)                                 .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
```

```
                                        ;+GO TO THAT ROUTINE.

020440  010046          STRAP:  MOV     R0,-(SP)        ;;SAVE R0
020442  016600  000002          MOV     2(SP),R0        ;;GET TRAP ADDRESS
020446  005740                  TST     -(R0)           ;;BACKUP BY 2
020450  111000                  MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
020452  006300                  ASL     R0              ;;POSITION FOR INDEXING
020454  016000  020462          MOV     STRPAD(R0),R0   ;;INDEX TO TABLE
020460  000200                  RTS     R0              ;;GO TO ROUTINE


                        .SBTTL  TRAP TABLE

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

                        ;       ROUTINE
                        ;       -------
020462                  STRPAD:
020462  017710                  .STYPE  ;;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
020464  017506                  .STYPOC ;;CALL=TYPOC     TRAP+1(104401)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
020466  017462                  .STYPOS ;;CALL=TYPOS     TRAP+2(104402)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
020470  017522                  .STYPON ;;CALL=TYPON     TRAP+3(104403)  TYPE OCTAL NUMBER (AS PER LAST CALL)
020472  013452                  .STYPDS ;;CALL=TYPDS     TRAP+4(104404)  TYPE DECIMAL NUMBER (WITH SIGN)
020474  016312                  .SCKSWR ;;CALL=CKSWR     TRAP+5(104405)  TEST FOR CHANGE IN SOFT-SWR
020476  016550                  .SRDCHR ;;CALL=RDCHR     TRAP+6(104406)  TTY TYPEIN CHARACTER ROUTINE
020500  016632                  .SRDLIN ;;CALL=RDLIN     TRAP+7(104407)  TTY TYPEIN STRING ROUTINE
020502  017004                  .SRDOCT ;;CALL=RDOCT     TRAP+10(104410) READ AN OCTAL NUMBER FROM TTY
 1595           000001                  .END
```

```
ABASE = 170400            140#     26
ACDW1 = 000000            26
ACDW2 = 000000            26
ACPUOP= 000000            26
ADBRL   001510            1200   1319#   1322    1325    1327
ADCS    001516            1260    210     233   1145#   1147   1153#   1155   1161#   1163   1169#   1171   1177#   1179
                          1185#   1187   1193#   1195   1201#   1203   1210#   1211#   1212   1220   1222#   1224   1232#
                          1234#   1235   1242   1252#   1254   1258   1262   1267   1275#   1276#   1277   1291#   1294#
                          1298    1304#  1308#   1313   1330#   1334#   1340#   1346#   1350#   1355#   1367#   1373#   1375
                          1382#   1385   1390#   1392   1395   1399#   1408   1419   1562   1563
ADDBR   001520            1270    217    1231    1250   1260   1280   1293   1331   1347   1403   1573
ADDW0 = 000000            26
ADDW1 = 000000            26
ADDW10= 000000            26
ADDW11= 000000            26
ADDW12= 000000            26
ADDW13= 000000            26
ADDW14= 000000            26
ADDW15= 000000            26
ADDW2 = 000000            26
ADDW3 = 000000            26
ADDW4 = 000000            26
ADDW5 = 000000            26
ADDW6 = 000000            26
ADDW7 = 000000            26
ADDW8 = 000000            26
ADDW9 = 000000            26
ADEVCT= 000000            26
ADEVM = 000000            26
ADINT   001536            1390    212    1332#   1349#   1359#   1368#
ADINT1  001540            1400   1368    1369#
AENV  = 000000            26
AENVM = 000000            26
AFATAL= 000000            26
AMADR1= 000000            26
AMADR2= 000000            26
AMADR3= 000000            26
AMADR4= 000000            26
AMAMS1= 000000            26
AMAMS2= 000000            26
AMAMS3= 000000            26
AMAMS4= 000000            26
AMSBAD= 000000            26
AMSGLG= 000000            26
AMSGTY= 000000            26
AMTYP1= 000000            26
AMTYP2= 000000            26
AMTYP3= 000000            26
AMTYP4= 000000            26
APASS = 000000            26
APRIOR= 000200            160     86
APTCSU= 000040           1502   19030
APTENV= 000001           1570   1502    19030
APTSIZ= 000200            166   19030
APTSPD= 000100           1502   19030
ARBADD  001552            1400   1402#   211     215    1450#   1454#   1572
```

```
ARBVCT  001554          1490    1830    214    1451#   1455#
ASWREG# 000000          26
ATESTN# 000000          26
AUNIT # 000000          26
AUSWR # 000000          26
AVECT1# 000340          150     26
AVECT2# 000000          26
BEGIN   001576          19      1610
BEGIN1  001602          20      1630    1586
BIT0  # 000001          130     360     377     651     678     717     971     1079    1092    1233    1244    1252    1269
                        1334    1350    1390
BIT00 # 000001          130
BIT01 # 000002          130
BIT02 # 000004          130
BIT03 # 000010          130
BIT04 # 000020          130
BIT05 # 000040          130
BIT06 # 000100          130
BIT07 # 000200          130
BIT08 # 000400          130     1577
BIT09 # 001000          130     1577    1578
BIT1  # 000002          130     368     464     465     549     589     619     639     699     715
BIT10 # 002000          130     837     838     1048    1117    1185    1186    1281    1373
BIT11 # 004000          130     457     458     466     467     486     487     505     506     519     520     536     537
                        702     703     844     845     1117    1193    1194    1373    1577
BIT12 # 010000          130     851     852     1048    1049
BIT13 # 020000          130     1261    1282    1373    1578
BIT14 # 040000          130     489     558     1577
BIT15 # 100000          130     424     567
BIT2  # 000004          130     376     549     604     619     798     799     988     989     1007    1018    1108
BIT3  # 000010          130     384     549     638     639     812     813     1019    1020    1027    1038    1108
BIT4  # 000020          130     368     368     376     384     393     401     409     417     424     433     465     589
                        604     619     638     639     1145    1146    1292    1294    1382
BIT5  # 000040          130     1153    1154    1218    1219    1222    1386    1388    1310
BIT6  # 000100          130     393     558     699     715     821     822     1079    1092    1117    1161    1162    1334
                        1350    1382
BIT7  # 000200          130     417     465     567     589     604     619     638     639     798     799     805     813
                        822     838     838     845     981     988     1018    1019    1038    1189    1118    1243    1253
                        1386
BIT8  # 000400          130     401     549     1169    1178    1373
BIT9  # 001000          130     829     838     1117    1177    1178    1289    1218    1222    1373
BPTVEC# 000014          130
BRLEV1  001572          1560    698#    691#    701     1071#   1072#   1078    1325#   1326#   1333
BRLEV2  001574          1570    692#    716     1073#   1091    1327#   1348
BUFADR  013162          1427#   1445#   1574
BUFNUM  013142          1413    1428    1436#
BYPASS  013166          246     1448#
BYPAS1  013244          1453    1458#   1468
CKBRL   001512          1210    683#    686     698     692
CKDLY   014864          1497#   1515    1528#
CKSWR # 104405          1577    1594#
CNTDEV  014860          1495#   1512    1526#
COUNT   001562          1520    485#    488#    504#    507#    518#    521#    535#    538#
CR    # 000015          130     1592
CRLF  # 000200          130     1592
CSR     001524          1300    252#    253     268#    261     268#    269     276#    277     286#    287     297#    299
```

K6

```
                     305*    313*    323*    331*    340*    351*    443*    463*    475*    495*    515*    532*    576*
                     698*    714*   1469*   1493*   1522*   1565
CBC      001534      137#     236     306     314     324     332     341     353     447     479     498     523     540
                     596      611     657     676    1411    1528    1567
CBR      001522      129#     361*    362     369*    370     377*    378     385*    386     394*    395     402*    403
                     410*     411     418*    419     425*    426     432*    435     442*    444*    452     457*    458*
                     462*     464*    466*    467*    468     474*    477*    486*    487*    494*    496*    505*    506*
                     514*     517*    519*    520*    531*    534*    536*    537*    549*    552     558*    561     567*
                     570      577*    580     591     606     621     632     641     650     651*    669     670*    697*
                     699*     702*    703*    713*    715*    717*   1305*   1460*   1470*   1472    1492*   1516*   1517*
                    1521*    1564    1566
DDISP = 177570       130       26     186
DELAY    001564     1530    1471*   1475*
DM1      014662       30    1547#
DM10     015115       72    1553#
DM11     015153       78    1554#
DM12     015213       84    1555#
DM13     015254       90    1556#
DM14     015315       96    1557#
DM15     015334      102    1558#
DM16     015364      109    1559#
DM17     015425      115    1560#
DM2      014722       36    1548#
DM3      014740       42    1549#
DM4      015000       48    1550#
DM5      015040       54    1551#
DM6      015055       60       66    1552#
DISPLA   001140      260     186*   1577*   1578*
DISPRE   000174      190     186
DSWR = 177570        130       26     186
DT1      015466       31    1562#
DT11     015552       79    1568#
DT12     015564       85    1569#
DT13     015576       91    1570#
DT14     015610       97    1571#
DT15     015616      103    1572#
DT16     015626      110    1573#
DT17     015640      116    1574#
DT2      015500       37    1563#
DT3      015506       43    1564#
DT4      015520       49    1565#
DT5      015532       55    1566#
DT6      015540       61       67      73    1567#
EMTVEC=  000030      130     186*
EM1      014066       29    1530#
EM10     014370       71    1537#
EM11     014423       77    1539#
EM12     014457       83    1540#
EM13     014503       89    1541#
EM14     014527       95    1542#
EM15     014555      101    1543#
EM16     014576      108    1545#
EM17     014630      114    1546#
EM2      014126       35    1531#
EM3      014155       41    1532#
EM4      014214       47    1533#
```

```
EMS     014253            93    15340
EM6     014301            99    15350
EM7     014341            65    15360
ERRVEC* 000004           130     186*   1577*
GN8   * ****** U          19     282     283     287    1594
HT    * 000011           130    1592
IOTVEC* 000020           130     186*
KWIV    001542          1420     696*    711*    738*    738*
KWIV8   001544          1430     712*    738     739*
LF    * 000012           130    1592
MINCNT  014062          1496*   1507    1509    1527*
NBEXT   001560          1510     184*   1448    1452*   1456*
NMBEXT  001556          1500     179*    184     204    1456
PC    =%000007           13#     598*    605*    620*    631*    640*    649*    668*   1389*   1460*   1498*   1499*   1581*
                        1523*   1578*   1579*   1583*   1586    1592*   1593*
PIRQ  = 177772           130
PIRQVE* 000240           130
PR0   = 000000           130
PR1   = 000040           130
PR2   = 000100           130
PR3   = 000140           130
PR4   = 000200           130
PR5   = 000240           130
PR6   = 000300           130
PR7   = 000340           130
PS    = 177776           130
PSW   = 177776           130     781*    716*    732*   1075*   1078*   1088*   1091*   1100*   1333*   1348*   1361*
PWRMSG  017414          1586    1587#
PWRVEC* 000024           130     186*   1586*
RATE    014056           588*    603*    618*    629*    638*    647*    666*   1387*   1470    1494*   1516    1525#
RBE8    001606           162     1640
RBE81   001714          1850
RBEG2   002170          1870    1659
RDCHR * 104406          1579    1594#
RDLIN * 104407          1588    1594#
RDOCT * 104410          1594#
REPEAT  013676           742     752     761     770     788    1492#
RESVEC* 000010           130
R0    =%000000           130     161*    163*    196     210*    211*    212     214*    215     217*    219*    223     225*
                         700*    704*    718*    719*    731*    733*    969*    979*    987*   1008*   1018*   1036*   1046*
                        1059*   1077*   1088*   1098*   1093*   1231*   1239*   1240*   1250*   1251*   1256*   1268*   1293*
                        1295*   1296*   1318*   1311*   1331*   1335*   1336*   1347*   1351*   1352*   1360*   1362*   1411*
                        1412*   1415*   1419*   1424    1427    1429    1458*   1460*   1479*   1500    1502    1503*   1504*
                        1506*   1512    1515*   1518*   1520*   1588*   1583*   1586*   1592*   1593*   1594*
R1    =%000001           130     189*    190*    191*    192     210*    219     220     221     224*    225     226     227
                        1413*   1415    1420*   1423    1429    1479*   1588*   1586*   1593*
R2    =%000002           130     167*    169     170*    188*    198     192*    193     194    1414*   1416*   1421*   1430*
                        1479*   1588*   1586*
R3    =%000003           130     168*    171*    174     178*    179    1479*   1579*   1586*   1591*
R4    =%000004           130    1586*   1591*
R5    =%000005           130     742*    752*    761*    770*    788*   1479*   1494    1495    1496    1518*   1511    1513*
                        1586*   1591*
R6    =%000006           130     186*
R7    =%000007           130
SP    =%000006           130     165*    173     186*    204*    243     708     723     737    1086    1105    1341    1356
                        1366    1468*   1479*   1577*   1578*   1579*   1588*   1583*   1586*   1591*   1592*   1593*   1594*
```

```
STACK = 001100              130#    165     186
STKLMT= 177774              130#
SWITCH  001570              155#
SWR     001136              260#    186#    1044    1281    1577    157A    1579#   1586#
SWREG   000176              190#    186     1579
SW0   = 000001              130#
SW00  = 000001              130#
SW01  = 000002              130#
SW02  = 000004              130#
SW03  = 000010              130#
SW04  = 000020              130#
SW05  = 000040              130#
SW06  = 000100              130#
SW07  = 000200              130#
SW08  = 000400              130#
SW09  = 001000              130#
SW1   = 000002              130#
SW10  = 002000              130#
SW11  = 004000              130#
SW12  = 010000              130#
SW13  = 020000              130#
SW14  = 040000              130#
SW15  = 100000              130#
SW2   = 000004              130#
SW3   = 000010              130#
SW4   = 000020              130#
SW5   = 000040              130#
SW6   = 000100              130#
SW7   = 000200              130#
SW8   = 000400              130#
SW9   = 001000              130#
TBITVE= 000014              130#
TEMP    001566              154#    434#    439#    445#    446     459#    1047#   1057#   1220#   1221#   1223#
TKVEC = 000060              130#
TPVEC = 000064              130#
TRAPVE= 000034              130#    186#
TRTVEC= 000014              130#
TST1    002514              230#
TST10   003120              300     303#
TST100  007510              918     921#
TST101  007546              926     930#
TST102  007604              935     938#
TST103  007670              951#
TST104  007762              963     967#
TST105  010054              976     978     985#
TST106  010202              995     1004    1007    1016#
TST107  010326              1026    1033    1035    1043#
TST11   003156              308     311#
TST110  010420              1045    1053    1056    1064#
TST111  010476              1074#
TST112  010566              1084    1087#
TST113  010704              1098    1103    1107#
TST114  010752              1113    1116#
TST115  011020              1122    1125#
TST116  011066              1131    1134#
TST117  011134              1140    1144#
```

```
T$T12   003214            316     3210
T$T120  011172           1149    11520
T$T121  011230           1157    11600
T$T122  011266           1165    11680
T$T123  011324           1173    11760
T$T124  011362           1181    11840
T$T125  011420           1189    11920
T$T126  011456           1197    12000
T$T127  011514           1205    12080
T$T13   003252            326     3290
T$T130  011556           1214    12180
T$T131  011632           1226    12290
T$T132  011736           1245    12490
T$T133  012062           1270    12740
T$T134  012140           1282    1284    12900
T$T135  012222           1300    13030
T$T136  012320           1315    13180
T$T137  012376           13290
T$T14   003310            334     3370
T$T140  012462           1339    13450
T$T141  012616           1358    1365    13710
T$T142  012654           1376    13810
T$T143  012712           1386    13890
T$T144  012756           1396    13980
T$T145  013030           1406    14090
T$T146  013164           1432    14470
T$T15   003366            3490
T$T16   003426            354     3590
T$T17   003464            364     3670
T$T2    002612            241     2500
T$T20   003522            372     3750
T$T21   003560            380     3830
T$T22   003616            388     3920
T$T23   003654            397     4000
T$T24   003712            405     4080
T$T25   003750            413     4160
T$T26   004006            421     4230
T$T27   004044            428     4310
T$T3    002650            255     2580
T$T30   004116           4410
T$T31   004242            456     4610
T$T32   004324            470     4730
T$T33   004440           4930
T$T34   004550           5130
T$T35   004654            525     5300
T$T36   004760            542     5480
T$T37   005026            554     5570
T$T4    002706            263     2660
T$T40   005074            563     5660
T$T41   005142            572     5750
T$T42   005216            582     5870
T$T43   005314            598     6020
T$T44   005412            613     6170
T$T45   005462            623     6280
T$T46   005532            634     6370
T$T47   005602            643     6460
```

```
TST5    002744            271     2748
TST50   005706            659     6658
TST51   006012            678     6828
TST52   006072            6958
TST53   006162            707     7108
TST54   006316            724     736     7418
TST55   006358            748     7518
TST56   006402            758     7688
TST57   006434            767     7698
TST6    003002            279     2838
TST60   006466            776     7798
TST61   006528            786     7898
TST62   006568            794     7978
TST63   006658            808     8118
TST64   006706            816     8288
TST65   006744            825     8288
TST66   007002            833     8368
TST67   007048            841     8438
TST7    003068            2958
TST70   007076            848     8508
TST71   007148            8598
TST72   007176            864     8678
TST73   007234            872     8758
TST74   007272            888     8838
TST75   007338            888     8928
TST76   007414            9058
TST77   007452            918     9138
TYPD8 = 104404           1468    15948
TYPE  = 104408            282     283     287     1468    1479    1578    1579    1583    1586    1591    1592    15948
TYPDC = 104481           1579    1583    15948
TYPDN = 104483           15948
TYPDS = 104482            285    15948
UPCNT   013376            598     605     628     631     648     649     668    1389    1468#
UPCNTA  013424           14728    1476
UPCNTB  013458           1473    14778
VCBRL   001514            1228    1065*   1068    1071    1073
VCIV    001546            1458    1076*   1085*   1089*   1099*   1184*
VCIVS   001558            1468    1085    1184
VCSTAT  001526            1328     792    798*    888     884*    886     812*    814     821*    823     829*    831     837*
                           839     844*    846     851*    853     857*    971*    972     977     982     989*    998     999
                          1886    1811    1828*   1821    1829    1834    1839    1848*   1849*   1858    1855    1879*   1882*
                          1892*   1896*   1181*   1188*   1111    1117*   1128    1568    1571
VCXREG  001538            1338     861*    862     869*    878     877*    878     885*    886     895*    896     952*    961
                           998*   1126*   1129    1569
VCYREG  001532            1348     987*    988     915*    916     923*    924     932*    933     941*    942     954*    955
                          1028*   1135*   1138    1578
SAPTMD  001888             258
SARG1   002146            1868
SASTAT* ******  U         1593
SATVC   020216            15938
SATV1   020172            15938
SATV3   020288            1592    15938
SATV4   020218            1578    15938
SBASE   001258             268     167     182     1454
SBDADR  001122             268
SBDDAT  001126             268     186     233*    235     236     238*    253*    254     261*    262     269*    278     277*
```

C7

```
                    278     287#    288     299#    306#    307     314#    315     324#    325     332#    333     341#
                    342     353#    362#    363     370#    371     378#    379     386#    387     395#    396     403#
                    404     411#    412     419#    428     426#    427     435#    436     447#    448     452#    468#
                    469     479#    488     498#    499     523#    524     548#    541     552#    553     561#    562
                    570#    571     578#    588#    581     591#    592     596#    597     606#    607     611#    612
                    621#    622     632#    633     641#    642     650#    652     657#    658     669#    671     676#
                    677     792#    793     888#    881     886#    887     814#    815     823#    824     831#    832
                    839#    848     846#    847     853#    854     862#    863     878#    871     878#    879     886#
                    887     896#    897     988#    989     916#    917     924#    925     933#    934     942#    943
                    955#    956     961#    962     972#    973     982#    998#    991     999#    1888    1811#   1821#
                   1822    1829#   1838    1839#   1111#   1112    1128#   1121    1129#   1138    1138#   1139    1147#
                   1148    1155#   1156    1163#   1164    1171#   1172    1179#   1188    1187#   1188    1195#   1196
                   1283#   1284    1212#   1213    1224#   1225    1235#   1236    1242#   1244    1258#   1262#   1263
                   1267#   1269    1288#   1283    1298#   1299    1313#   1314    1375#   1385#   1395#   1483#   1485
                   1424#   1425    1582#   1584    1587    1562    1564    1565    1567    1568    1569    1578    1572
                   1573    1574
SCDW1   881254      268
SCDW2   881256      268
SCMARC  828166     15928#
SCKBWR  816312     1579#   1594
SCMTAG  881188      268     186
SCM1    = 888882    268
SCM2    = 888884    268
SCM3    = 888882    268
SCNTLG  816755     1579#
SCNTLU  816758     1579#
SCPUOP  881222      268
SCRLF   881171      268    1578    1579    1583    1592
SDBLK   813666     1479#
SDDW8   881268      268
SDDW1   881262      268
SDDW18  881384      268
SDDW11  881386      268
SDDW12  881318      268
SDDW13  881312      268
SDDW14  881314      268
SDDW15  881316      268
SDDW2   881264      268
SDDW3   881266      268
SDDW4   881278      268
SDDW5   881272      268
SDDW6   881274      268
SDDW7   881276      268
SDDW8   881388      268
SDDW9   881382      268
SDEVCT  881284      268
SDEVM   881252      268
SDOAGN  813352     1468#
SDTBL   813656     1479#
SENDAD  813342       23    1468#   1578
SENDCT  813318      186    1468#
SENDMG  813361     1468#
SENULL  813356     1468#
SENV    881214      268    1578    1592    1593
SENVM   881215      268     186    1592    1593
SEOP    813254     1457    1468#
```

```
SEOPCT  013302            186*  1460#
SERFLG  001103            260   1577*  1578*
SERMAX  001115            260    186*  1577*
SERROR  016132            186   1578#
SERRPC  001116            260   1562   1563   1564   1565   1566   1567   1568   1569   1570   1571   1572   1573
                         1574   1578*  1583
SERRTB  001320            260   1583
SERRTV  017106           1578   1583#
SERTTL  001112            260   1578*
SESCAP  001166            260    186*  1577*  1578
SETABL  001214            260
SETEND  001320             25    260
SFATAL  001176            260   1593*
SFFLG   020436           1593#*
SFILLC  001154            260   1592
SFILLS  001153            260   1592
SGDADR  001120            260
SGDDAT  001124            260    251*   252    254    259*   260    262    267*   268    270    275*   276    278
                          285*   286    288    292*   296*   304*   305    307    312*   313    315    322*   323
                          325    330*   331    333    339*   340    342    346*   350*   360*   361    363    368*
                          369    371    376*   377    379    384*   385    387    393*   394    396    401*   402
                          404    409*   410    412    417*   418    420    424*   425    427    433*   446*   448
                          454*   465*   476*   478*   488    490*   497*   499    509*   516*   524    533*   541
                          558*   553    559*   562    568*   571    581    589*   592    595*   597    604*   607
                          618*   612    619*   622    630*   633    639*   642    648*   652    656*   658    667*
                          671    675*   677    798*   793    799*   801    805*   807    813*   815    822*   824
                          830*   832    838*   840    845*   847    852*   854    860*   861    863    868*   869
                          871    876*   877    879    884*   885    887    894*   895    897    901*   902    906*
                          907    909    914*   915    917    922*   923    925    931*   932    934    940*   941
                          943    947*   948    953*   954    956    960*   962    970*   981*   988*   997*  1010*
                         1019*  1027*  1038*  1109*  1112   1118*  1121   1127*  1130   1136*  1139   1146*  1148
                         1154*  1156   1162*  1164   1170*  1172   1178*  1180   1186*  1188   1194*  1196   1202*
                         1204   1209*  1213   1219*  1225   1233*  1234   1236   1243*  1253*  1261*  1268*  1279*
                         1283   1292*  1306*  1314   1372*  1383*  1391*  1404*  1405   1423*  1425   1508*  1583
                         1589*  1562   1564   1565   1567   1568   1569   1570   1573   1574
SGET42  013332           1460#
SHD   = 000000             12
SHIBTS  001000             250
SHIOCT  017104           1588#*
SICNT   001104             260  1577*
SILLUP  017406           1586#
SITEMB  001114             260  1578*  1583
SLF     001172             260  1578   1579   1592
SLFLG   020435           1593#*
SLPADR  001106             260   186*  1577*
SLPERR  001110             260   186*   284*   338*   893*   939*  1577*  1578
SMADR1  001226             260
SMADR2  001232             260
SMADR3  001236             260
SMADR4  001242             260
SMAIL   001174              25    260    186    1577   1578   1592
SMAMS1  001224             260
SMAMS2  001230             260
SMAMS3  001234             260
SMAMS4  001240             260
SMBADR  001002             250
```

```
SMFLG   020434          1593#*
SMNEW   016773          1579#
SMSGAD  001210            26#   1593*
SMSGLG  001212            26#   1593*
SMSGTY  001174            26#   1593*
SMSWR   016762          1579#
SMTYP1  001225            26#
SMTYP2  001231            26#
SMTYP3  001235            26#
SMTYP4  001241            26#
SMXCNT  016130          1577#
SNULL   001152            26#   1592
SNWTST* 000001           23#0   25#0   25#0   26#0   27#0   28#0   29#0   30#0   31#0   32#0   32#0   33#0   34#0
                         35#0   36#7   37#5   38#0   39#2   40#0   40#8   41#6   42#3   43#1   44#1   46#1   47#3
                         49#3   51#3   53#8   54#8   55#7   56#6   57#5   58#7   60#2   61#7   62#8   63#7   64#6
                         66#5   68#2   69#5   71#0   74#1   75#1   76#0   76#9   77#9   78#9   79#7   81#1   82#8
                         82#8   83#6   84#3   85#8   85#9   86#7   87#5   88#3   89#2   90#5   91#3   92#1   93#8
                         93#8   95#1   96#7   98#5  101#6  104#3  106#4  107#4  108#7  110#7  111#6  112#5  113#4
                        114#4  115#2  116#8  116#8  117#6  118#4  119#2  120#8  120#8  121#8  122#9  124#9  127#4
                        129#8  130#3  131#8  132#9  134#5  135#1  136#1  137#9  139#8  140#9  144#7
SOCNT   017704          1591#*
SOMODE  017706          1591#*
SOVER   016114          1577#
SPASS   001202            26#    186*   1460*   1577
SPASTH  001006            25#
SPRIOR  001246            26#
SPWRAD  017402          1586#
SPWRDN  017242           186    1586#
SPWRMG  017376          1586#
SPWRUP  017314          1586#
SQUES   001170            26#   1578    1579    1592
SRDCHR  016550          1579#   1594
SRDDEC* ****** U        1594
SRDLIN  016632          1579#   1594
SRDOCT  017004          1588#   1594
SRDSZ = 000010          1579#
SREGAD  001156            26#
SREG0   001160            26#
SREG1   001162            26#
SRTNAD  013354          1460#
SR2A  = ****** U        1594
SSAVRE* ****** U        1594
SSAVR6  017412          1586#*
SSCOPE  015652           186   1577#
SSETUP* 000037          105#    186   1460   1577   1578
STUP  = 177777          105#
SSVLAD  016060          1577#
SSVPC = 000210            23#
SSWR  = 165400             5#     12     18     26    186    230    250    258    266    274    283    295    303
                         311    321    329    337    349    359    367    375    383    392    400    408    416
                         423    431    441    461    473    493    513    530    548    557    566    575    587
                         602    617    628    637    646    665    682    695    710    741    751    760    769
                         779    789    797    811    820    828    836    843    858    859    867    875    883
                         892    905    913    921    930    938    951    967    985   1016   1043   1064   1074
                        1087   1107   1116   1125   1134   1144   1152   1160   1168   1176   1184   1192   1200
                        1208   1218   1229   1249   1274   1290   1303   1318   1329   1345   1371   1381   1389
```

```
                    1398    1409    1447    1468    1577    1578    1586
SSWREG  001216       268     186
SSWRMK= 000000        18    1577
STESTN  001208       268    1577#
STIMES  001164       268     186#    230#    283#    295#    337#    349#    431#    441#    473#    493#    513#    530#
                     548#    557#    566#    575#    587#    602#    617#    628#    637#    646#    665#    682#    701#
                     751#    760#    769#    779#    789#    892#    938#   1064#   1074#   1087#   1107#   1116#   1125#
                    1134#   1290#   1303#   1318#   1329#   1345#   1371#   1381#   1389#   1398#   1409#   1468#   1577#
STKB    001144       268    1579
STKS    001142       268    1579
STN     000147        68      12    2308     241    2508     255    2588     263    2668     271    2748     279     283#
                     295#     308     303#     308     311#     316     321#     326     329#     334     337#     349#     354
                     359#     364     367#     372     375#     380     383#     388     392#     397     400#     405     408#
                     413     4168     421     4238     428     431#     441#     456     461#     478     473#     493#     513#
                     525     5308     542     5488     554     557#     563     566#     572     575#     582     587#     598
                     602#     613     617#     623     628#     634     637#     643     646#     659     665#     678     682#
                     695#     707     710#     724     736     741#     748     751#     758     760#     767     769#     776
                     779#     786     789#     794     797#     808     811#     816     820#     825     828#     833     836#
                     841     843#     848     850#     859#     864     867#     872     875#     888     883#     888     892#
                     985#     918     913#     918     921#     926     938#     935     938#     951#     963     967#     976
                    1064#   1074#   1084    1087#   1098    1103    1107#   1113    1116#   1122    1125#   1131    1134#
                    1148    1144#   1149    1152#   1157    1168#   1165    1168#   1173    1176#   1181    1184#   1189
                    1192#   1197    1208#   1205    1208#   1214    1218#   1226    1229#   1245    1249#   1278    1274#
                    1282    1284    1298#   1308    1303#   1315    1318#   1329#   1339    1345#   1358    1365    1371#
                    1376    1381#   1386    1389#   1396    1398#   1406    1409#   1432    1447#
STPB    001150       268    1592#
STPFLG  001155       268    1592
STPS    001146       268    1592
STRAP   020440       186    1594#
STRP    000011      1594#
STRPAD  020462      1594#
STSTM   001004        25#
STSTNM  001102       268    1468#   1577#   1578
STTYIN  016748      1579#
STYPBN= ****** U    1594
STYPDS  013452      1479#   1594
STYPE   017710      1592#   1593    1594
STYPEC  020122      1579    1592#
STYPEX  020170      1592#
STYPOC  017506      1591#   1594
STYPON  017522      1591#   1594
STYPOS  017462      1591#   1594
SUNIT   001206        26#
SUNITM  001010        25#
SUSWR   001220        26#
SVECT1  001244       268     183    1455
SVECT2  001245       268
SXTSTR  015664      1577#
SSGET4= 000000      1468#
SBFILL  017705      1591#
SABCAT= ****** U    1577    1578
.       028504       198     238     248     258     268     186     283#    684     687     728     734    1066    1069
                    1081    1094    1241    1328    1323    1337    1353    1363    1393    1468    1479#   1577    1578
                    1579#   1583#   1586    1592    1593#
.SASTA= ****** U    1593
```

.SX = 001000 250

```
COMMEN    130
ENDCOM    130
ERROR     130    245    256    264    272    280    298    381    389    317    327    335    344    355    365
          373    381    389    398    406    414    422    429    438    450    455    471    482    501    526
          543    555    564    573    583    594    599    609    614    624    635    644    654    668    673
          679    706    722    735    747    749    757    759    766    768    775    777    785    787    795
          803    809    817    826    834    842    849    856    865    873    881    889    899    911    919
          927    936    945    958    964    975    983    993   1003   1012   1024   1032   1040   1052   1061
         1083   1097   1102   1114   1123   1132   1141   1150   1158   1166   1174   1182   1190   1198   1206
         1215   1227   1238   1246   1259   1265   1271   1285   1301   1316   1338   1357   1364   1377   1387
         1397   1407   1428
ESCAPE    130
GETPRI    130
MULT      130
NEWTST    130    230    250    258    266    274    283    295    303    311    321    329    337    349    359
          367    375    383    392    400    408    416    423    431    441    461    473    493    513    530
          548    557    566    575    587    602    617    628    637    646    665    682    695    710    741
          751    760    769    779    789    797    811    820    828    836    843    850    859    867    875
          883    892    905    913    921    930    938    951    967    985   1016   1043   1064   1074   1087
         1107   1116   1125   1134   1144   1152   1160   1168   1176   1184   1192   1200   1208   1218   1229
         1249   1274   1290   1303   1318   1329   1345   1371   1381   1389   1398   1409   1447
POP       130   1479   1580   1586   1593
PUSH      130   1479   1580   1586   1593
REPORT    130
SCOPE     130    230    250    258    266    274    283    295    303    311    321    329    337    349    359
          367    375    383    392    400    408    416    423    431    441    461    473    493    513    530
          548    557    566    575    587    602    617    628    637    646    665    682    695    710    741
          751    760    769    779    789    797    811    820    828    836    843    850    859    867    875
          883    892    905    913    921    930    938    951    967    985   1016   1043   1064   1074   1087
         1107   1116   1125   1134   1144   1152   1160   1168   1176   1184   1192   1200   1208   1218   1229
         1249   1274   1290   1303   1318   1329   1345   1371   1381   1389   1398   1409   1447   1468
SETPRI    130
SETTRA   1594#
SETUP     130    186
SKIP      130    241    255    263    271    279    289    300    308    316    326    334    343    354    364
          372    380    388    397    405    413    421    428    437    449    453    456    470    481    500
          525    542    554    563    572    582    593    598    608    613    623    634    643    653    659
          672    678    707    721    724    736    748    758    767    776    786    794    802    808    816
          825    833    841    848    855    864    872    880    888    898    910    918    926    935    944
          957    963    974    976    978    992    995   1002   1004   1007   1009   1023   1026   1031   1033
         1035   1037   1045   1051   1053   1056   1058   1060   1084   1095   1098   1103   1113   1122   1131
         1140   1149   1157   1165   1173   1181   1189   1197   1205   1214   1226   1237   1245   1264   1270
         1282   1284   1300   1315   1339   1354   1358   1365   1376   1386   1396   1406   1426   1432
SLASH     130
SPACE     130
STARS     130     23     25     26    230    250    258    266    274    283    295    303    311    321    329
          337    349    359    367    375    383    392    400    408    416    423    431    441    461    473
          493    513    530    548    557    566    575    587    602    617    628    637    646    665    682
          695    710    741    751    760    769    779    789    797    811    820    828    836    843    850
          859    867    875    883    892    905    913    921    930    938    951    967    985   1016   1043
         1064   1074   1087   1107   1116   1125   1134   1144   1152   1160   1168   1176   1184   1192   1200
         1208   1218   1229   1249   1274   1290   1303   1318   1329   1345   1371   1381   1389   1398   1409
         1447   1468   1479   1577   1578   1579   1580   1583   1586   1591   1592   1593   1594
SWRBU     130   1868
TRNTRP   1594#
TYPBIN    130
```

DZARAB.P11 CROSS REFERENCE TABLE -- MACRO NAMES

```
TYPDEC     13#   1460
TYPNAM     13#
TYPNUM     13#
TYPOCS     13#
TYPOCT     13#   1579   1583
TYPTXT     13#    282    283    287
SSCHRE     26#
SSCHTH     26#
SSESCA     13#
SSNEXT     13#    238    250    258    266    274    283    295    303    311    321    329    337    349    359
          367    375    383    392    400    408    416    423    431    441    461    473    493    513    530
          548    557    566    575    587    602    617    628    637    646    665    682    695    710    741
          751    760    769    779    789    797    811    820    828    836    843    850    859    867    875
          883    892    905    913    921    930    938    951    967    985   1016   1043   1064   1074   1087
         1107   1116   1125   1134   1144   1152   1160   1168   1176   1184   1192   1200   1208   1218   1229
         1249   1274   1290   1303   1318   1329   1345   1371   1381   1389   1398   1409   1447
SSSET      1594#
SSSETH     1868
SSSKIP     13#    241    255    263    271    279    300    308    316    326    334    354    364    372    388
          388    397    405    413    421    428    456    470    525    542    554    563    572    582    598
          613    623    634    643    659    678    707    724    736    748    758    767    776    786    794
          808    816    825    833    841    848    864    872    888    888    910    918    926    935    963
          976    978    995   1004   1007   1026   1033   1035   1045   1053   1054   1084   1098   1103   1113
         1122   1131   1140   1149   1157   1165   1173   1181   1189   1197   1205   1214   1226   1245   1270
         1282   1284   1300   1315   1339   1350   1365   1376   1386   1396   1406   1432
.EQUAT      7#     13
.MEADE      7#     12
.SETUP      8#    105
.SWRMI      9#     18
.SWRLO     18#
.SACTI     18#     23
.SAPTB     18#    268
.SAPTH     18#     25
.SAPTY     18#   1593
.SCATC      7#     19
.SCHTA      7#     26
.SEOP       7#   1468
.SERRO      7#   1578
.SERRY      9#   1583
.SPARM      8#
.SPOWE      8#   1586
.SRDOC      9#   1588
.SREAD      8#   1579
.SSAVE      8#
.SSCOP      8#   1577
.SSPAC      8#
.SSWDO      8#
.STRAP      8#   1594
.STYPD      9#   1479
.STYPE      7#     8#   1592
.STYPO      7#   1591
```

DZARAB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADD      170   219   225   238  1412  1450  1451  1475  1479  1579  1580  1583  1591  1592  1593
ASL     1579  1580  1583  1594
ASLB    1479
ASR     1593
BCC     1479
BEQ      186   197   208   237   255   263   271   279   289   308   318   326   334   343
         354   364   372   388   388   397   405   413   421   428   449   481   508   525   542
         554   563   572   582   593   598   608   613   623   634   643   653   659   672   678
         794   802   808   816   825   833   841   848   855   864   872   880   888   898   918
         918   926   935   944   957   963  1045  1113  1122  1131  1140  1149  1157  1165  1173
        1181  1189  1197  1205  1214  1226  1237  1245  1278  1282  1284  1315  1376  1386  1396
        1406  1426  1449  1460  1577  1578  1579  1580  1583  1591  1592  1593
BGE     1508  1577
BGT     1460  1479  1579  1591
BHI     1577
BIC      458   467   487   506   528   537   651   670   683   703  1065  1108  1319  1460  1579
        1588  1591
BIS      457   466   486   505   519   536   702   717  1049  1294  1308  1334  1358  1479  1579
        1591
BISB    1583
BIT     1044  1244  1269  1281  1577  1578
BITB     186  1592  1593
BLE      748   758   767   776   786
BLOS    1579
BLT     1479  1579  1591  1592
BMI      478   978   992  1007  1023  1035  1056  1255  1473  1479
BNE      175   186   195   213   216   222   228   293   347   440   460   489   491   508   510
         522   539   684   687   705   720   734   983   949   988  1009  1037  1058  1060  1066
        1069  1081  1094  1241  1257  1297  1312  1328  1323  1337  1353  1363  1417  1431  1476
        1479  1519  1577  1578  1579  1583  1586  1591  1592  1593
BPL      437   453   974  1002  1031  1051  1264  1278  1308  1393  1401  1479  1505  1578  1579
        1591  1592
BR       162   172   202   283   287   239   241   456   707   721   724   736   976   995  1004
        1026  1033  1053  1084  1095  1098  1183  1339  1354  1358  1365  1432  1453  1479  1577
        1578  1579  1580  1583  1586  1591  1592  1593
CLR      161   168   181   186   191   285   339   432   434   442   443   445   462   474   475
         476   478   494   495   497   514   515   531   532   697   713   732   739   804   857
         894   948   998  1028  1047  1082  1096  1181  1232  1268  1275  1291  1304  1305  1338
        1348  1346  1355  1361  1367  1369  1372  1383  1391  1404  1468  1468  1479  1492  1493
        1521  1522  1577  1579  1580  1583  1586  1591
CLRB    1211  1223  1479  1577  1579  1592  1593
CMP      173   186   194   212   215   221   227   236   243   254   262   270   278   288   307
         315   329   333   342   363   371   379   387   396   404   412   420   427   448   488
         499   524   541   553   562   571   581   592   597   607   612   622   633   642   652
         658   671   677   686   708   723   737   793   801   807   815   824   832   840   847
         854   863   871   879   887   897   902   909   917   925   934   943   948   956   962
        1068  1086  1105  1112  1121  1138  1139  1148  1156  1164  1172  1180  1188  1196  1204
        1213  1225  1236  1283  1314  1322  1341  1356  1366  1405  1425  1429  1474  1479  1587
        1512  1577  1578  1579
CMPB    1577  1578  1579  1592  1593
DEC      178   488   507   521   538   704   719   733   979  1008  1036  1057  1059  1088  1093
        1248  1311  1336  1352  1362  1416  1438  1452  1468  1518  1583
DECB    1591  1592
EMT       13
HALT      19   176   685   688  1067  1078  1321  1324  1578  1586  1592
INC      171   439   901   947  1221  1256  1276  1296  1468  1479  1517  1577  1578  1579  1586
```

K7

```
        1591    1593
INCB     292     346     459     498     589    1577    1578    1592
IOT       13
JMP       19      28     198     201     246    1457    1459    1460
JSR      598     605     620     631     640     649     668     742     752     761     770     780    1389    1460    1498
        1499    1501    1578    1579    1592    1593
MOV      163     165     166     167     179     180     182     183     184     186     188     189     190     192     284
         210     211     214     217     218     224     230     232     233     240     244     251     252     253     259
         260     261     267     268     269     275     276     277     283     284     286     287     295     296     297
         299     304     305     306     312     313     314     322     323     324     330     331     332     337     338
         340     341     349     350     351     353     360     361     362     368     369     370     376     377     378
         384     385     386     393     394     395     401     402     403     409     410     411     417     418     419
         424     425     426     431     433     435     441     444     446     447     454     463     464     465     468
         473     477     479     485     493     496     498     504     513     516     517     518     523     530     533
         534     535     540     548     549     550     552     557     558     559     561     566     567     568     570
         575     576     577     578     580     587     588     589     591     595     596     602     603     604     606
         610     611     617     618     619     621     628     629     630     632     637     638     639     641     646
         647     648     650     656     657     665     666     667     669     675     676     682     690     692     696
         698     699     700     701     711     712     714     715     716     718     730     731     738     741     751
         760     769     779     789     790     792     798     799     800     805     806     812     813     814     821
         822     823     829     830     831     837     838     839     844     845     846     851     852     853     860
         861     862     868     869     870     876     877     878     884     885     886     892     893     895     896
         906     907     908     914     915     916     922     923     924     931     932     933     938     939     941
         942     952     953     954     955     960     961     969     970     971     972     981     982     987     988
         989     998     997     999    1010    1011    1018    1019    1020    1021    1027    1029    1038    1039    1046
        1048    1064    1071    1073    1074    1078    1076    1077    1078    1079    1085    1087    1088    1089    1098
        1091    1098    1099    1104    1107    1108    1109    1111    1116    1117    1118    1120    1125    1126    1127
        1129    1134    1135    1136    1138    1145    1146    1147    1153    1154    1155    1161    1162    1163    1169
        1170    1171    1177    1178    1179    1185    1186    1187    1193    1194    1195    1201    1202    1203    1209
        1210    1212    1219    1220    1222    1224    1231    1233    1234    1235    1239    1242    1243    1250    1251
        1252    1253    1258    1260    1261    1262    1267    1279    1280    1290    1292    1293    1295    1298    1303
        1306    1307    1310    1313    1318    1325    1327    1329    1331    1332    1333    1335    1345    1347    1348
        1349    1351    1359    1360    1368    1371    1373    1375    1381    1382    1385    1389    1390    1395    1398
        1399    1403    1409    1411    1413    1414    1415    1419    1420    1421    1423    1424    1427    1454    1455
        1456    1458    1468    1469    1470    1471    1479    1494    1495    1496    1497    1500    1502    1503    1509
        1515    1516    1520    1577    1578    1579    1580    1583    1586    1591    1592    1593    1594
MOVB     186     452    1479    1577    1578    1579    1580    1591    1592    1593    1594
NEG     1479    1506    1591
NOP      209    1081    1460
RESET    164     185     187     298     352     551     560     569     579     693     791    1110    1119    1120    1137
        1374    1384    1394    1402    1410    1460
ROL     1588    1591
RTI      186    1479    1577    1578    1579    1580    1586    1591    1592
RTS     1477    1510    1513    1523    1583    1592    1593    1594
SUB      691    1072    1326    1479    1584    1578    1593
TRAP    1594
TST      169     174     186     193     196     199     220     223     226     235     436    1448    1479    1511    1577
        1578    1579    1580    1583    1591    1592    1593    1594
TSTB     469     973     977     991    1000    1006    1022    1030    1034    1050    1055    1254    1263    1277    1299
        1392    1400    1472    1479    1577    1579    1592    1593
.ASCII    26
.ASCIZ    26     202     203     207    1460    1530    1531    1532    1533    1534    1535    1536    1537    1539    1540
        1541    1542    1543    1545    1546    1547    1548    1549    1550    1551    1552    1553    1554    1555    1556
        1557    1558    1559    1560    1579    1583    1587
.BLKB   1579
.BLKW   1479
```

```
.BYTE     26  1460  1578  1579  1591  1593
.ENABL     4
.END    1595
.ENDC     12    13    18    19    23    25    26   185   186   202   203   207   230   241   250
         255   258   263   266   271   274   279   283   289   295   300   303   308   311   316
         321   326   329   334   337   343   349   354   359   364   367   372   375   380   383
         388   392   397   400   405   408   413   416   421   423   428   431   437   441   449
         453   456   461   470   473   481   493   500   513   525   530   542   548   554   557
         563   566   572   575   582   587   593   598   602   608   613   617   623   628   634
         637   643   646   653   659   665   672   678   682   695   707   710   721   724   736
         741   748   751   758   760   767   769   776   779   786   789   794   797   802   808
         811   816   820   825   828   833   836   841   843   848   850   855   859   864   867
         872   875   880   883   888   892   898   905   910   913   918   921   926   930   935
         938   944   951   957   963   967   974   976   978   985   992   995  1002  1004  1007
        1009  1016  1023  1026  1031  1033  1035  1037  1043  1045  1051  1053  1056  1058  1060
        1064  1074  1084  1087  1095  1098  1103  1107  1113  1116  1122  1125  1131  1134  1140
        1144  1149  1152  1157  1160  1165  1168  1173  1176  1181  1184  1189  1192  1197  1200
        1205  1208  1214  1218  1226  1229  1237  1245  1249  1264  1270  1274  1282  1284  1290
        1300  1303  1315  1318  1329  1339  1345  1354  1358  1365  1371  1376  1381  1386  1389
        1396  1398  1406  1409  1426  1432  1447  1460  1479  1577  1578  1579  1588  1583  1586
        1591  1592  1593  1594
.EQUIV    13
.EVEN     26   202   203   207  1561  1583  1588  1593
.IF       12    13    18    19    23    25    26   185   186   202   203   207   230   241   250
         255   258   263   266   271   274   279   283   289   295   300   303   308   311   316
         321   326   329   334   337   343   349   354   359   364   367   372   375   380   383
         388   392   397   400   405   408   413   416   421   423   428   431   437   441   449
         453   456   461   470   473   481   493   500   513   525   530   542   548   554   557
         563   566   572   575   582   587   593   598   602   608   613   617   623   628   634
         637   643   646   653   659   665   672   678   682   695   707   710   721   724   736
         741   748   751   758   760   767   769   776   779   786   789   794   797   802   808
         811   816   820   825   828   833   836   841   843   848   850   855   859   864   867
         872   875   880   883   888   892   898   905   910   913   918   921   926   930   935
         938   944   951   957   963   967   974   976   978   985   992   995  1002  1004  1007
        1009  1016  1023  1026  1031  1033  1035  1037  1043  1045  1051  1053  1056  1058  1060
        1064  1074  1084  1087  1095  1098  1103  1107  1113  1116  1122  1125  1131  1134  1140
        1144  1149  1152  1157  1160  1165  1168  1173  1176  1181  1184  1189  1192  1197  1200
        1205  1208  1214  1218  1226  1229  1237  1245  1249  1264  1270  1274  1282  1284  1290
        1300  1303  1315  1318  1329  1339  1345  1354  1358  1365  1371  1376  1381  1386  1389
        1396  1398  1406  1409  1426  1432  1447  1460  1479  1577  1578  1579  1588  1583  1586
        1591  1592  1593  1594
.IFF      13    18    23    25    26   186   230   241   250   255   258   263   266   271   274
         279   283   289   295   300   303   308   311   316   321   326   329   334   337   343
         349   354   359   364   367   372   375   380   383   388   392   397   400   405   408
         413   416   421   423   428   431   437   441   449   453   456   461   470   473   481
         493   500   513   525   530   542   548   554   557   563   566   572   575   582   587
         593   598   602   608   613   617   623   628   634   637   643   646   653   659   665
         672   678   682   695   707   710   721   724   736   741   748   751   758   760   767
         769   776   779   786   789   794   797   802   808   811   816   820   825   828   833
         836   841   843   848   850   855   859   864   867   872   875   880   883   888   892
         898   905   910   913   918   921   926   930   935   938   944   951   957   963   967
         974   976   978   985   992   995  1002  1004  1007  1009  1016  1023  1026  1031  1033
        1035  1037  1043  1045  1051  1053  1056  1058  1060  1064  1074  1084  1087  1095  1098
        1103  1107  1113  1116  1122  1125  1131  1134  1140  1144  1149  1152  1157  1160  1165
        1168  1173  1176  1181  1184  1189  1192  1197  1200  1205  1208  1214  1218  1226  1229
        1237  1245  1249  1264  1270  1274  1282  1284  1290  1300  1303  1315  1318  1329  1339
```

```
        1345    1354    1358    1365    1371    1376    1381    1386    1389    1396    1398    1406    1409    1426    1432
        1447    1460    1479    1577    1578    1579    1580    1583    1586    1591    1592    1593    1594
.IFT     202     203     207    1577    1578    1579    1580
.IFTF    202     203     207    1577    1578    1579    1580
.IIF      12      18      19      26     186    1460    1577    1578    1579    1583    1592    1594
.IRP     105     230     250     258     266     274     283     295     303     311     321     329     337     349     359
         367     375     383     392     400     408     416     423     431     441     461     473     493     513     530
         548     557     566     575     587     602     617     628     637     646     665     682     695     710     741
         751     760     769     779     789     797     811     820     828     836     843     850     859     867     875
         883     892     905     913     921     930     938     951     967     985    1016    1043    1064    1074    1087
        1107    1116    1125    1134    1144    1152    1160    1168    1176    1184    1192    1200    1208    1218    1229
        1249    1274    1290    1303    1318    1329    1345    1371    1381    1389    1398    1409    1447    1460    1577
        1578    1586    1593
.LIST       2      11      13      18      19      26     105     186     202     203     207     230     250     258     266
         274     283     295     303     311     321     329     337     349     359     367     375     383     392     400
         408     416     423     431     441     461     473     493     513     530     548     557     566     575     587
         602     617     628     637     646     665     682     695     710     741     751     760     769     779     789
         797     811     820     828     836     843     850     859     867     875     883     892     905     913     921
         930     938     951     967     985    1016    1043    1064    1074    1087    1107    1116    1125    1134    1144
        1152    1160    1168    1176    1184    1192    1200    1208    1218    1229    1249    1274    1290    1303    1318
        1329    1345    1371    1381    1389    1398    1409    1447    1460    1577    1578    1579    1594
.MACRO     18      26     186    1594
.MCALL      7       8       9      10      13      26     186
.NLIST      1       3      13      18      19      26     105     186     202     203     207     230     250     258     266
         274     283     295     303     311     321     329     337     349     359     367     375     383     392     400
         408     416     423     431     441     461     473     493     513     530     548     557     566     575     587
         602     617     628     637     646     665     682     695     710     741     751     760     769     779     789
         797     811     820     828     836     843     850     859     867     875     883     892     905     913     921
         930     938     951     967     985    1016    1043    1064    1074    1087    1107    1116    1125    1134    1144
        1152    1160    1168    1176    1184    1192    1200    1208    1218    1229    1249    1274    1290    1303    1318
        1329    1345    1371    1381    1389    1398    1409    1447    1460    1577    1578    1579    1594
.PAGE      26
.REPT      19      26
.SBTTL     13      18      19      23      25      26     230     250     258     266     274     283     295     303     311
         321     329     337     349     359     367     375     383     392     400     408     416     423     431     441
         461     473     493     513     530     548     557     566     575     587     602     617     628     637     646
         665     682     695     710     741     751     760     769     779     789     797     811     820     828     836
         843     850     859     867     875     883     892     905     913     921     930     938     951     967     985
        1016    1043    1064    1074    1087    1107    1116    1125    1134    1144    1152    1160    1168    1176    1184
        1192    1200    1208    1218    1229    1249    1274    1290    1303    1318    1329    1345    1371    1381    1389
        1398    1409    1447    1460    1479    1577    1578    1579    1580    1583    1586    1591    1592    1593    1594
.TITLE     12
.WORD      19      23      25      26     206    1460    1580    1583    1586    1591    1592    1593

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 @DZARAB,DZARAB/CRF=DZARAB.P11
 RUN-TIME:  79  48  8 SECONDS
 RUN-TIME RATIO:  416/136=3.0
 CORE USED:  25K  (50 PAGES)
```