```
1                               .REM &
2
3
4                                        IDENTIFICATION
5                                        --------------
6
7                               PRODUCT CODE:  AC-8206E-MC
8
9                               PRODUCT NAME:  CVKAFE0 DRV11 TEST
10
11                              PRODUCT DATE:  JUNE 1982
12
13                              MAINTAINER  :  DIAGNOSTIC ENGINEERING
14
                                AUTHOR      :  J. CARMODY
15
16
17
18
19
20
21
22
23
24                                      COPYRIGHT (C) 1977, 1983 BY
25                                DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS
26                                            ALL RIGHTS RESERVED
27
28                              THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY
29                              BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS
30                              OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE
31                              COPYRIGHT NOTICE.  THIS SOFTWARE OR ANY OTHER COPIES
32                              THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAIL-
33                              ABLE TO ANY OTHER PERSON.  NO TITLE TO AND OWNERSHIP
34                              OF THE SOFTWARE IS HEREBY TRANSFERRED.
35
36                              THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE
37                              WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COM-
38                              MITTMENT BY DIGITAL EQUIPMENT CORPORATION.
39
40                              DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR
41                              RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
42                              NOT SUPPLIED BY DIGITAL.
43
44                      THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:
45
46                              DEC             PDP             UNIBUS          MASSBUS
47                              DECUS           DECTAPE         VAX
48
49
50                                              ---------------------------------
51                                              | D | I | G | I | T | A | L |
52                                              ---------------------------------
53
54
```

```
                        1.      ABSTRACT

                                THIS IS A LOGIC TEST OF THE DRV11. TO ALLOW TESTING OF
                                THE DATA LINES AND INTERRUPTS, A SPECIAL MAINTENANCE CABLE
                                (BC08R) IS USED BY DEFAULT. ALSO, A SPECIAL TEST MODULE
                                IS REQUIRED BY OPTION TO TEST THE NEWDATA RDY AND
                                DATATRANS SIGNALS.
                                NOTE:   THE SPECIAL TEST MODULE IS FOR USE BY IN HOUSE
                                MANUFACTURING ONLY.
                                SEE SECTION 5.2

                                THIS TEST WILL OPERATE ON ONE DRV11. SPECIAL OPERATION-
                                AL PROCEDURES ARE REQUIRED TO OPERATE ON OTHER THAN THE
                                PRIMARY DRV11. SEE SEC. 5.4


                        2.      REQUIREMENTS

                        2.1     EQUIPMENT

                                LSI-11

                                DRV11

                                TEST CABLE  (BC08R)     (BY OPTION)

                                TEST MODULE             (BY OPTION)
                                                        (FOR IN HOUSE MANUFACTURING ONLY)

                        2.2     STORAGE

                                2.2.1  PROGRAM STORAGE - 4K


                        3.      LOADING PROCEDURE

                        3.1     METHOD

                                UNDER XXDP+ OR
                                ABSOLUTE LOADER


                        4.      STARTING PROCEDURE

                                        200 - NORMAL ENTRY TO TEST ONE DEVICE

                                TO LOAD AND EXECUTE
                                -------------------

                                1. UNDER XXDP+ OR LOAD PROGRAM WITH THE ABSOLUTE LOADER.

                                2. IF ANY PROGRAM OPTIONS ARE REQUIRED, SET THE AP-
                                   PROPRIATE BIT IN THE SOFTWARE SWITCH REGISTER AT
                                   LOCATION 422.  (REF   SECTION 5.1)
                                3. START PROGRAM AT 200.
```

```
111                                     4. PROGRAM WILL PRINT "END OF PASS" FOLLOWING EACH PASS.
112
113
114                             4.1     CONTROL SWITCH SETTING
115
116                                     THIS PROGRAM CONTAINS A SOFTWARE SWITCH REGISTER FOR
117                                     OPTION SELECTION. FOR IT TO OPERATE THE OPERATOR MUST
118                                     SELECT THE APPROPRIATE OPTION BY SETTING OR
119                                     RESETTING THE RESPECTIVE BIT IN THE WORD.
120
121                                     TO DO THIS , THE LSI-11 MUST BE IN ODT MODE.
122
123                             4.2     STARTING ADDRESS OR ADDRESSES
124
125                                       200 = START OF TEST--FOR NORMAL TESTING
126
127                             5.      OPERATING PROCEDURE
128
129                                     1. THE PROGRAM WILL CYCLE CONTINUOUSLY UNLESS HALTED
130                                        BY THE OPERATOR, OR SOME ERROR CONDITION.
131                                     2. TO HALT THE PROGRAM, DEPRESS THE BREAK KEY. ODT
132                                        WILL DISPLAY THE PC AT WHICH IT WAS HALTED.
133                                     3. IF NEW OPTIONS ARE TO BE SELECTED IN THE SWR, THEY MUST
134                                        BE SET AT THIS TIME.
135                                     4. CONTINUE THE PROGRAM VIA A "P" OR A "G" COMMAND.
136
137                             5.1 SOFTWARE SWITCH SETTINGS
138
139                                     BIT15 - CONTINUE ON ERROR                   (100000)
140                                     BIT14 - LOOP ON CURRENT ERROR               (040000)
141                                     BIT13 - NOT USED                            (020000)
142                                     BIT12 - NOT USED                            (010000)
143                                     BIT11 - NOT USED                            (004000)
144                                     BIT10 - LOOP ON CURRENT TEST                (002000)
145                                     BIT9 - RUN TEST MODULE                      (001000)
146                                     BIT8 - INHIBIT WRAP CABLE                   (000400)
147                                     BIT7 - NOT USED                             (000200)
148                                     BIT6 - NOT USED                             (000100)
149                                     BIT5 - NOT USED                             (000040)
150                                     BIT4 - NOT USED                             (000020)
151                                     BIT3 - NOT USED                             (000010)
152                                     BIT2 - NOT USED                             (000004)
153                                     BIT1 - NOT USED                             (000002)
154                                     BIT0 - NOT USED                             (000001)
155
156                             5.2 SELECTION OF TEST OPTIONS
157
158                                     1. TO TEST NEWDATA RDY AND DATATRANS SIGNALS, THE
159                                        SPECIAL WRAP MODULE MUST BE INSTALLED. THE OPERATOR
160                                        MUST ALSO SET BIT9 IN THE SWITCH REGISTER (LOC. 422).
161                                        NOTE:   THE SPECIAL MODULE IS FOR USE BY IN HOUSE
162                                        MANUFACTURING ONLY.
163
164                                     2. THIS TEST WILL RUN WITH THE WRAP CABLE BY DEFAULT.
165                                        TO INHIBIT TESTING WITH THE WRAP CABLE, THE OPERATOR
166                                        MUST SET BIT8 IN THE SWITCH REGISTER (LOC. 422).
```

```
167
168                             5.3 WRAP CABLE
169
170                                     THE WRAP CABLE IS REQUIRED TO TEST TRANSFER OF DATA
171                                     INTO AND OUT OF THE INPUT BUFFER, AND THE DEVICE
172                                     INTERRUPTS.
173
174                             NOTE  !!!!!!    THIS DIAGNOSTIC IS APPROXIMATELY 95% EFFECTIVE
175                                             WHEN RUN WITH THE WRAP CABLE, AND APPROXIMATELY
176                                             60-70% EFFECTIVE WHEN RUN WITHOUT IT.
177
178                             5.4 TESTING OTHER DRV11 MODULES
179
180                                     TO TEST A DRV11 NOT ADDRESSED AS 167770, OR
181                                     VECTORED AT 300, THE OPERATOR MUST SUPPLY THE NEW ADDRESSES
182                                     AND VECTORS TO THE PROGRAM BY DEPOSITING THEM AT
183                                     THE LOCATIONS TAGGED BY "RCSR" IN THE BEGINNING OF THE LISTING.
184                                     THE ORDER IS AS FOLLOWS:
185
186                                             RCSR:   CSR ADDRESS
187                                                     OUTPUT BUFFER ADDRESS
188                                                     INPUT BUFFER ADDRESS
189                                                     HIGH BYTE ADDR. OF OUTPUT BUFFER OR
190                                                             (OUTPUT BUFFER ADDR -1)
191                                                     "A" INTERRUPT VECTOR ADDRESS
192                                                     "A" ADDRESS + 2
193                                                     "B" INTERRUPT VECTOR ADDRESS
194                                                     "B" ADDRESS + 2
195
196                             5.5 EXECUTION TIME
197
198                                     TYPICAL RUN TIMES (ONE PASS)
199                                     QUICK VERIFY                    1 SEC.
200                                     WITH WRAP CABLE                10 SEC.
201
202
203                             6.      ERRORS
204
205                             6.1     ERROR REPORTING
206
207                                     ALL ERROR REPORTS WILL BE DONE VIA A HALT WITHIN THE
208                                     PROGRAM. THIS WILL CAUSE ODT TO DISPLAY THE PC+2 OF THE
209                                     ERROR HALT. AT THIS TIME THE OPERATOR MUST REFERENCE
210                                     THE LISTING TO DETERMINE THE ERROR DESCRIPTION.
211                                     THE NUMBER AT TAG $FATAL IN THE APT MAILBOX CONTAINS
212                                     THE ERROR NUMBER AND MAY BE USED TO REFERENCE THE DE-
213                                     SCRIPTION IN THE TABLE OF CONTENTS.
214
215                             6.2     ERROR RECOVERY
216
217
218                                     IN ORDER TO CONTINUE, THE OPERATOR MUST ISSUE A "P" TO
219                                     CONTINUE THE PROGRAM, OR MAY SET THE ERROR LOOP SWITCH
220                                     PRIOR TO CONTINUING.
221
222                             7. IMPORTANT TAGS
```

```
                                FOLLOWING IS A LIST OF IMPORTANT TAGS WITHIN THE LISTING

                                TAG             COMMENT
                                ---             -------
                                $MAIL           START OF THE PROGRAM MAILBOX. MANY CLUES TO
                                                PROBLEMS CAN BE FOUND HERE

                                $FATAL          ERROR NUMBER. USE THE TABLE OF CONTENTS TO
                                                LOCATE THE ERROR INFORMATION AND/OR CODE

                                $TESTN          CURRENT TEST NUMBER

                                $PASS           PASS COUNT OF THE PROGRAM WHEN ERROR WAS
                                                DETECTED OR PROGRAM HALTED

                                $SWREG          SOFTWARE SWITCH REGISTER

                                RCSR            START OF UNIT UNDER TEST ADDRESSES



                                8.      LISTING
                                &


                                        .TITLE CVKAFE
                                        .ENABLE ABS
                                        .NLIST  MD,MC,CND
                                        .LIST   ME


                    000001      X=1
                    000001      N=1
```

```
260
261
262                             ;REVISION       B
263                             ;       1.      CHANGED TEST REQUIRING WRAP CABLE TO BE
264                             ;               INHIBITED BY BIT 8 OF THE SLR
265                             ;       2.      INITIALIZED A & B INTERRUPT VECTOR LOCATIONS
266                             ;               WITH TRAP CATCHER
267
268                             ;REVISION       C
269                             ;       1.      DOCUMENTATION CHANGE ONLY
270
271                             ;REVISION       D
272                             ;       1.      ADDED CODE TO PRINT OUT PROGRAMS TITLE
273                             ;REVISION       E
274                             ;       1.      INCORPORATED PATCHES D1,D2 AND D3
275                             ;               D1 -- "INCORRECT PROGRAM COUNTER ADDRESSING AT
276                             ;                     LOC. 1360 IN TEST 1,CAUSED WRONG ADDRESS
277                             ;                     TO BE LOADED INTO LOC. 4. THIS RESULTS
278                             ;                     IN PROGRAM HALTING AND THE FOLLOWING
279                             ;                     MESSAGE TO BE DISPLAYED TO CONSOLE:
280                             ;                      'ILLEGAL WORD 1'"
281                             ;               D2 -- "PROGRAM OVERPRINTS END PASS REPORT AND
282                             ;                     PROGRAM TITLE."
283                             ;               D3 -- "RELEASED VERSION OF PATCH D2 ON THE MEDIA
284                             ;                     IS INCORRECT CAUSING CPU TO HALT AT 314"
285                             ;       2.      MADE PROGRAM APT COMPATIBLE
286
```

```
                        ;GENERAL REGISTER LOGIC TEST

104000                  HLT=104000
167770                  CSR=167770
001200                  STKPTR=1200
                        ;REGISTER DEFINITIONS
000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          SP=%6
000007                          PC=%7

                        ;SWITCHES

001000                          SW9=1000
002000                          SW10=2000
004000                          SW11=4000
020000                          SW13=20000
040000                          SW14=40000

                        .SBTTL  BASIC DEFINITIONS

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

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

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

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

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

                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

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

```
                                .EQUIV  BIT06,BIT6
                                .EQUIV  BIT05,BIT5
                                .EQUIV  BIT04,BIT4
                                .EQUIV  BIT03,BIT3
                                .EQUIV  BIT02,BIT2
                                .EQUIV  BIT01,BIT1
                                .EQUIV  BIT00,BIT0

                                ;*BASIC "CPU" TRAP VECTOR ADDRESSES
        000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
        000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
        000014                  TBITVEC=14              ;;"T" BIT
        000014                  TRTVEC= 14              ;;TRACE TRAP
        000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
        000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
        000024                  PWRVEC= 24              ;;POWER FAIL
        000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
        000034                  TRAPVEC=34              ;;"TRAP" TRAP
        000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
        000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
        000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
                                        .ENABLE ABS
        000000                          .=0
000000  000002                          .+2
000002  000000                          HALT
000004  000006                          .+2
000006  000000                          HALT
000010  000012                          .+2
000012  000000                          HALT
000014  000016                          .+2
000016  000000                          HALT
000020  000022                          .+2
000022  000000                          HALT
000024  000026                          .+2
000026  000000                          HALT
000030  000032                          .+2
000032  000000                          HALT
000034  000036                          .+2
000036  000000                          HALT
000040  000042                          .+2
000042  000000                          HALT
000044  000046                          .+2
000046  000000                          HALT
000050  000052                          .+2
000052  000000                          HALT
000054  000056                          .+2
000056  000000                          HALT
000060  000062                          .+2
000062  000000                          HALT
000064  000066                          .+2
000066  000000                          HALT
        000100                          .-100
000100  000102                          102
000102  000002                          RTI             ; RTI FOR POSSIBLE CLOCK INTERRUPT
000104  000106                          .+2
000106  000000                          HALT
```

```
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .+2
                                                HALT
                                                .=200
000200  005067  000202                          CLR     $PASS           ; CLEAR PASS COUNT
000204  005067  000172                          CLR     $FATAL
000210  005067  000170                          CLR     $TESTN
000214  000137  001246                          JMP     @#START1        ;INITIAL START
        000300                                  .=300                   ;DEVICE INTERRUPT VECTORS
000300  000302                                  .+2
000302  000000                                  HALT
000304  000306                                  .+2
000306  000000                                  HALT
        000400                                  .=400
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
000400                          $MAIL:                          ;;APT MAILBOX
000400  000000                  $MSGTY: .WORD   AMSGTY   ;;MESSAGE TYPE CODE
000402  000000                  $FATAL: .WORD   AFATAL   ;;FATAL ERROR NUMBER
000404  000000                  $TESTN: .WORD   ATESTN   ;;TEST NUMBER
000406  000000                  $PASS:  .WORD   APASS    ;;PASS COUNT
000410  000000                  $DEVCT: .WORD   ADEVCT   ;;DEVICE COUNT
000412  000000                  $UNIT:  .WORD   AUNIT    ;;I/O UNIT NUMBER
000414  000000                  $MSGAD: .WORD   AMSGAD   ;;MESSAGE ADDRESS
000416  000000                  $MSGLG: .WORD   AMSGLG   ;;MESSAGE LENGTH
000420                          $ETABLE:                        ;;APT ENVIRONMENT TABLE
000420      000                 $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
000421      000                 $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
000422  000000                  $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
000424  000000                  $USWR:  .WORD   AUSWR    ;;USER SWITCHES
000426  000000                  $CPUOP: .WORD   ACPUOP   ;;CPU TYPE,OPTIONS
```

```
                                ;*                      BITS 15-11=CPU TYPE
                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                                ;*                              11/70=06,PDQ=07,Q=10
                                ;*                      BIT 10=REAL TIME CLOCK
                                ;*                      BIT  9=FLOATING POINT PROCESSOR
                                ;*                      BIT  8=MEMORY MANAGEMENT
000430                          $ETEND:
                                .MEXIT
                                .SBTTL  APT PARAMETER BLOCK

                                ;;*****************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;*****************************************************************
        000430                          .$X=.   ;;SAVE CURRENT LOCATION
        000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                          200     ;;FOR APT START UP
        000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  000430                          $APTHDR ;;POINT TO APT HEADER BLOCK
        000430                          .=.$X   ;;RESET LOCATION COUNTER
                                ;;*****************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

000430                          $APTHD:
000430  000000                  $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
000432  000400                  $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
000434  000010                  $TSTM:  .WORD   10      ;;RUN TIM OF LONGEST TEST
000436  000010                  $PASTM: .WORD   10      ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
000440  000000                  $UNITM: .WORD   0       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
000442  000014                          .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
        001200                          .=1200
        000410                  DEVCNT=$DEVCT
        000402                  ERRNUM=$FATAL
        000426                  OPTION=$CPUOP

                                ;THIS TABLE CONTAINS INITIAL REGISTER AND VECTOR ADDRESSES

001200  167770                  RCSR:   CSR
001202  167772                          CSR+2
001204  167774                          CSR+4
001206  167773                          CSR+3
001210  000300                  RCSR1:  300
001212  000302                          302
001214  000304                          304
001216  000306                          306

                                ;THIS TABLE CONTAINS REGISTER AND VECTOR ADDRESSES OF THE DR11-C UNDER TEST

001220  167770                  DRCSR:  167770                  ;ADDRESS OF DR11-C STATUS REGISTER
001222  167772                  DROBUF: 167772                  ;ADDRESS OF DR OUTPUT BUFFER REG.
001224  167774                  DRIBUF: 167774                  ;ADDRESS OF DR INPUT BUFFER REG.
001226  167773                  DRBHIO: 167773                  ;HIGH BYTE OF OUTPUT BUFFER REG.

001230  000300                  DRVECA: 300                     ;INTERRUPT VECTOR OF UNIT UNDER TEST
001232  000302                  DRLVLA: 302
001234  000304                  DRVECB: 304                     ;INTERRUPT VECTOR
```

```
001236  000306                          DR.VLB: 306
001240  000000                          XORFLG: 0

001242  000000                          COUNT:  0                       ;COUNT LOCATION
001244  000240                          PL:     240                     ;PRIORITY LEVEL

001246  012706  001200                  START1: MOV     #STKPTR,SP
001252  000137  001256                          JMP     @#START
                                        ;+**************************************************************************

                                        ;+              ROUTINE TO PRINT DIAGNOSTIC'S TITLE IF  ALLOWED BY  APT MODE

                                        ;  ALL CODE ADDED FOR TITLE PRINTOUT IS NOTED
                                        ; WITH   A ;+ IN THE  COMMENT SECTION.
                                        ;+**************************************************************************
001256  132767  000040  177135          START:  BITB  #40, $ENVM        ;+ WILL APT ALLOW PRINTING?
001264  001013                                  BNE AROUND              ;+ NO BRANCH AROUND
001266  012700  004616                          MOV #TITLE1,R0          ;+ GET STARTING ADDRESS OF TITLE MSG
001272  105737  177564                  LOOP1:  TSTB @#TPS              ;+ CHECK IF TERMINAL READY
001276  100375                                  BPL     LOOP1           ;+ LOOP IF NOT
001300  112037  177566                          MOVB (R0)+,@#TPB        ;+ PRINT A CHARACTER
001304  001372                                  BNE LOOP1               ;+ BRANCH BACK IF NOT DONE PRINTING
001306  105737  177564                  LOOP:   TSTB @#TPS
001312  100375                                  BPL LOOP
                                        ;+**************************************************************************

                                        ; INITIALIZE ADDRESS AND VECTORS

001314  012700  001200                  AROUND: MOV #RCSR, R0           ; GET ADDRESS OF UNIT UNDER TEST
001320  012701  005000                          MOV #5000, R1
001324  077101                              1$: SOB R1,1$
001326  012701  001220                          MOV #DRCSR,R1

001332  012737  004672  000024                  MOV     #PFAIL,@#24
001340  012021                                  MOV     (R0)+,(R1)+     ;LOAD INITIAL TEST ADDRESSES
001342  012021                                  MOV     (R0)+,(R1)+
001344  012021                                  MOV     (R0)+,(R1)+
001346  012021                                  MOV     (R0)+,(R1)+
001350  012021                                  MOV     (R0)+,(R1)+
001352  012021                                  MOV     (R0)+,(R1)+
001354  012021                                  MOV     (R0)+,(R1)+

                                        ;DOES RESET CLEAR REGISTER?
001356                                  TST1:
001356  012767  000001  177020                  MOV     #1,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
001364  016705  177630                  LP1:    MOV     DRCSR,R5        ;GET ADDRESS OF STATUS REGISTER
001370  106427  000200                          MTPS    #200            ;TURN OFF INTERRUPTS
001374  012737  001456  000004                  MOV     #ERR1,@#4               ;SET TIME OUT TRAP VECTOR
001402  012777  177777  177612                  MOV     #-1,@DROBUF     ;PRESET OUTPUT BUFFER
001410  000005                                  RESET                   ;CLEAR DATA REGISTER
001412  017700  177604                          MOV     @DROBUF,R0      ;GET RESULT OF RESET
001416  001450                                  BEQ     CON
001420  032767  040000  176774                  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
001426  001356                                  BNE     LP1             ; GO TO LOOP ERROR
001430  012767  000001  176744                  MOV     #1,$FATAL
001436  012767  000001  176734                  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
```

```
623  001444  005767  176752          TST     $SWREG          ; CHECK FOR HALT ON ERROR
624  001450  100401                  BMI     1$              ; CONTINUE IF SET
625  001452  000000                  HALT            ;<DATA REG DID NOT CLEAR>
626  001454                  1$:
627  001454  000431                  BR      CON
628  001456                  ERR1:
629  001456  032767  040000  176736  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
630  001464  001337                  BNE     LP1             ; GO TO LOOP ERROR
631  001466  012767  000002  176706  MOV     #2,$FATAL
632  001474  012767  000001  176676  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
633  001502  005767  176714          TST     $SWREG          ; CHECK FOR HALT ON ERROR
634  001506  100401                  BMI     1$              ; CONTINUE IF SET
635  001510  000000                  HALT            ;<TIME WHEN ADDRESSING PLU>
636  001512                  1$:
637  001512  032767  002000  176702  BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
638  001520  001316                  BNE     TST1            ; GO TO LOOP ON TEST
639  001522  000407                  BR      TST2
640  001524  012706  001200  5$:     MOV     #STKPTR,SP      ;RESET STACK POINTER
641  001530  012737  000006  000004  MOV     #6,@#4          ;RESTORE TIME OUT TRAP
642  001536  000765                  BR      1$
643  001540  000772          CON:    BR      .-12
644                          ; TEST "NEWDATA RDY" AND "DATATRANS" SIGNALS IN PLU
645                          ;  NOTE****** THE PLU TEST MODULE MUST BE INSTALLED
646                          ;               TO EXECUTE THIS TEST
647                          ;
648  001542                  TST2:
649  001542  012767  000002  176634  MOV     #2,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
650  001550  032767  001000  176644  BIT     #BIT9,$SWREG
651  001556  001505                  BEQ     TST3            ; SKIP TEST IF NOT SELECTED
652  001560  012706  001200          MOV     #STKPTR,SP              ; SET UP STACK POINTER
653  001564  000005                  RESET                           ; CLEAR EVERYTHING
654                                           ; THIS RESET SHOULD INITIALIZE THE
655                                           ; SIGNAL LATCHES IN THE TEST MODULE
656  001566  012777  031460  177426  MOV     #31460,@DROBUF          ; PRIME THE LATCHES
657  001574  000240                  NOP
658  001576  000240                  NOP             ; TIMING ALLOWANCE
659  001600  017700  177420          MOV     @DRIBUF,R0              ; GET DATA
660  001604  032700  000001          BIT     #BIT0,R0                ; CHECK DATA TRANS SIG
661  001610  001016                  BNE     T2CON           ; CONTINUE IF PRESENT
662  001612  032767  040000  176602  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
663  001620  001350                  BNE     TST2            ; GO TO LOOP ERROR
664  001622  012767  000003  176552  MOV     #3,$FATAL
665  001630  012767  000001  176542  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
666  001636  005767  176560          TST     $SWREG          ; CHECK FOR HALT ON ERROR
667  001642  100401                  BMI     1$              ; CONTINUE IF SET
668  001644  000000                  HALT            ;<NO DATA TRANS SIGNAL>
669  001646                  1$:
670  001646  032700  000002  T2CON:  BIT     #BIT1,R0        ; CHECK NEW DATA RDY SIGNAL
671  001652  001016                  BNE     T2CN1           ; CONTINUE IF OK
672  001654  032767  040000  176540  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
673  001662  001371                  BNE     T2CON           ; GO TO LOOP ERROR
674  001664  012767  000004  176510  MOV     #4,$FATAL
675  001672  012767  000001  176500  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
676  001700  005767  176516          TST     $SWREG          ; CHECK FOR HALT ON ERROR
677  001704  100401                  BMI     1$              ; CONTINUE IF SET
678  001706  000000                  HALT            ;<NO NEW DATA RDY SIGNAL>
```

```
679  001710                                1$:
680  001710  000005                        T2CN1:  RESET                   ; CLEAR EVERYTHING
681  001712  000240                                NOP
682  001714  000240                                NOP
683  001716  017700  177302                        MOV     @DRIBUF,R0      ; CHECK SIGNAL LATCHES
684  001722  005700                                TST     R0              ; SHOULD BE CLEAR
685  001724  001416                                BEQ     1$              ; CONTINUE IF CLEAR
686  001726  032767  040000  176466                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
687  001734  001365                                BNE     T2CN1           ; GO TO LOOP ERROR
688  001736  012767  000005  176436                MOV     #5,$FATAL
689  001744  012767  000001  176426                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
690  001752  005767  176444                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
691  001756  100401                                BMI     1$              ; CONTINUE IF SET
692  001760  000000                                HALT            ;<SIGNALS DID NOT CLEAR>
693  001762                                1$:
694  001762  032767  002000  176432                BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
695  001770  001264                                BNE     TST2            ; GO TO LOOP ON TEST
696  001772                                TST3:
697  001772  012767  000003  176404                MOV     #3,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
698  002000  012777  177777  177214                MOV     #-1,@DROBUF     ;ALL ONES TO REGISTER
699  002006  017700  177210                        MOV     @DROBUF,R0
700  002012  022700  177777                        CMP     #-1,R0
701  002016  001416                                BEQ     1$
702  002020  032767  040000  176374                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
703  002026  001361                                BNE     TST3            ; GO TO LOOP ERROR
704  002030  012767  000006  176344                MOV     #6,$FATAL
705  002036  012767  000001  176334                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
706  002044  005767  176352                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
707  002050  100401                                BMI     1$              ; CONTINUE IF SET
708  002052  000000                                HALT            ;<REGISTER WILL NOT HOLD ALL ONES>
709  002054                                1$:
710  002054  032767  002000  176340                BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
711  002062  001343                                BNE     TST3            ; GO TO LOOP ON TEST
712
713                                        ;WRAP CABLE MUST BE INSTALLED TO EXECUTE THIS TEST
714  002064                                TST4:
715  002064  012767  000004  176312                MOV     #4,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
716  002072  032767  000400  176322                BIT     #BIT8,$SWREG
717  002100  001031                                BNE     TST5            ;SKIP TEST IF NOT SELECTED
718  002102  012777  177777  177112                MOV     #-1,@DROBUF
719  002110  000005                                RESET                   ;SET DATA TO ALL ONES
720  002112  005777  177106                        TST     @DRIBUF         ;REGISTER SHOULD CLEAR
721  002116  001416                                BEQ     1$
722  002120  032767  040000  176274                BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
723  002126  001356                                BNE     TST4            ; GO TO LOOP ERROR
724  002130  012767  000007  176244                MOV     #7,$FATAL
725  002136  012767  000001  176234                MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
726  002144  005767  176252                        TST     $SWREG          ; CHECK FOR HALT ON ERROR
727  002150  100401                                BMI     1$              ; CONTINUE IF SET
728  002152  000000                                HALT            ;<REGISTER DID NOT CLEAR BY RESET>
729  002154                                1$:
730  002154  032767  002000  176240                BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
731  002162  001340                                BNE     TST4            ; GO TO LOOP ON TEST
732
733  002164                                TST5:
734  002164  012767  000005  176212                MOV     #5,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
```

```
CVKAFE  MACY11 30(1046)  22-NOV-82  10:43  PAGE 16
CVKAFE.P11    22-NOV-82 10:37          ERROR   7       REGISTER DID NOT CLEAR BY RESET                                  SEQ 0015

   735  002172  012777  052525  177022          MOV     #52525,@DROBUF  ;LOAD TEST DATA INTO BUFFER
   736  002200  017700  177016                  MOV     @DROBUF,R0      ;COPY DATA FROM BUFFER TO R0
   737  002204  022700  052525                  CMP     #52525,R0       ;COMPARE DATA
   738  002210  001416                          BEQ     1$              ;BR IF DATA IS CORRECT
   739  002212  032767  040000  176202          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
   740  002220  001361                          BNE     TST5            ; GO TO LOOP ERROR
   741  002222  012767  000010  176152          MOV     #10,$FATAL
   742  002230  012767  000001  176142          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
   743  002236  005767  176160                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
   744  002242  100401                          BMI     1$              ; CONTINUE IF SET
   745  002244  000000                          HALT            ;<INCORRECT DATA IN REG>
   746  002246                          1$:
   747  002246  032767  002000  176146          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
   748  002254  001343                          BNE     TST5            ; GO TO LOOP ON TEST
   749
   750  002256                          TST6:
   751  002256  012767  000006  176120          MOV     #6,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
   752  002264  012777  125252  176730          MOV     #125252,@DROBUF ;LOAD TEST DATA INTO BUFFER
   753  002272  017700  176724                  MOV     @DROBUF,R0      ;COPY DATA FROM BUFFER TO R0
   754  002276  022700  125252                  CMP     #125252,R0      ;COMPARE DATA
   755  002302  001416                          BEQ     1$              ;BR IF DATA IS CORRECT
   756  002304  032767  040000  176110          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
   757  002312  001361                          BNE     TST6            ; GO TO LOOP ERROR
   758  002314  012767  000011  176060          MOV     #11,$FATAL
   759  002322  012767  000001  176050          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
   760  002330  005767  176066                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
   761  002334  100401                          BMI     1$              ; CONTINUE IF SET
   762  002336  000000                          HALT            ;<INCORRECT DATA IN REG>
   763  002340                          1$:
   764  002340  032767  002000  176054          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
   765  002346  001343                          BNE     TST6            ; GO TO LOOP ON TEST
   766
   767                                  ;TEST RELIABILITY OF DR11-C OUTPUT BUFFER REGISTER
   768  002350                          TST7:
   769  002350  012767  000007  176026          MOV     #7,$TESTN       ; MOVE TEST NUMBER TO MAILBOX
   770  002356  010502                  BUFTST: MOV     R5,R2           ;GET ADDRESS OF DRCSR
   771  002360  005722                          TST     (R2)+           ;R2=ADDRESS OF OUTPUT BUFFER REG.
   772  002362  005003                          CLR     R3              ; INITIALIZE DATA REGISTER
   773  002364  010312                  LP7:    MOV     R3,(R2)         ; SEND THE DATA
   774  002366  021203                          CMP     (R2),R3         ; CHECK THE RECEIVED DATA
   775  002370  001004                          BNE     5$              ; ERROR IF NOT THE SAME
   776  002372  005203                          INC     R3              ; INCREMENT THE DATA
   777  002374  105703                          TSTB    R3              ; CHECK FOR END OF DATA
   778  002376  001417                          BEQ     1$              ; CONTINUE IF END
   779  002400  000771                          BR      LP7             ; GO TO SEND DATA IF NOT
   780  002402                          5$:
   781  002402  032767  040000  176012          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
   782  002410  001365                          BNE     LP7             ; GO TO LOOP ERROR
   783  002412  012767  000012  175762          MOV     #12,$FATAL
   784  002420  012767  000001  175752          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
   785  002426  005767  175770                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
   786  002432  100401                          BMI     1$              ; CONTINUE IF SET
   787  002434  000000                          HALT            ;<DATA INCORRECT IN REG>
   788  002436                          1$:
   789  002436  032767  002000  175756          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
   790  002444  001341                          BNE     TST7            ; GO TO LOOP ON TEST
```

```
                                        ;TEST THAT BYTE REFERENCE TO DROBUF AFFECT PROPER BYTE ONLY
                                        TST10:
002446                                  MOV     #10,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
002446  012767  000010  175730          MOV     #-1,@DROBUF     ;SET ALL ONES IN BUFFER
002454  012777  177777  176540  TAG:    CLRB    @DROBUF         ;CLEAR LOW BYTE
002462  105077  176534                  MOV     @DROBUF,R0      ;COPY DATA
002466  017700  176530                  CMP     #177400,R0      ;VERIFY THAT LOW BYTE IS CLEAR
002472  022700  177400                  BEQ     1$
002476  001416                          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
002500  032767  040000  175714          BNE     TAG             ; GO TO LOOP ERROR
002506  001362                          MOV     #13,$FATAL
002510  012767  000013  175664          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
002516  012767  000001  175654          TST     $SWREG          ; CHECK FOR HALT ON ERROR
002524  005767  175672                  BMI     1$              ; CONTINUE IF SET
002530  100401                          HALT            ;<LOW BYTE FAILED TO CLEAR>
002532  000000                  1$:
002534                          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
002534  032767  002000  175660          BNE     TST10           ; GO TO LOOP ON TEST
002542  001341
                                        TST11:
002544                                  MOV     #11,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
002544  012767  000011  175632          MOV     #-1,@DROBUF     ;SET ALL ONES IN BUFFER
002552  012777  177777  176442          CLRB    @DRBHIO         ;CLEAR HIGH BYTE
002560  105077  176442                  MOV     @DROBUF,R0
002564  017700  176432                  CMP     #377,R0         ;VERIFY THAT HIGH BYTE IS CLEAR
002570  022700  000377                  BEQ     1$
002574  001416                          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
002576  032767  040000  175616          BNE     TST11           ; GO TO LOOP ERROR
002604  001357                          MOV     #14,$FATAL
002606  012767  000014  175566          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
002614  012767  000001  175556          TST     $SWREG          ; CHECK FOR HALT ON ERROR
002622  005767  175574                  BMI     1$              ; CONTINUE IF SET
002626  100401                          HALT            ;<HIGH BYTE FAILED TO CLEAR>
002630  000000                  1$:
002632                          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
002632  032767  002000  175562          BNE     TST11           ; GO TO LOOP ON TEST
002640  001341
                                        TST12:
002642                                  MOV     #12,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
002642  012767  000012  175534
```

```
 830  002650  005067  000110                  CLR     T12DAT          ;CLEAR DATA LOCATION
 831  002654  012704  002764                  MOV     #T12DAT,R4      ;STORE ADDRESS OF DATA LOCATION
 832  002660  005077  176336                  CLR     @DROBUF         ;CLEAR OUTPUT BUFFER
 833  002664  105077  176336          T12CON: CLRB    @DRBHIO         ;CLEAR HIGH BYTE
 834  002670  105277  176332          3$:     INCB    @DRBHIO         ;INCREMENT HIGH BYTE
 835  002674  105264  000001                  INCB    1(R4)           ;INCREMENT COMPARISON DATA
 836  002700  027714  176316                  CMP     @DROBUF,(R4)    ;COMPARE DATA
 837  002704  001004                          BNE     6$              ;BRANCH ON ERROR
 838  002706  105764  000001          4$:     TSTB    1(R4)           ;FINISHED?
 839  002712  001417                          BEQ     1$              ;YES
 840  002714  000765                          BR      3$              ;CONTINUE TESTING
 841  002716                          6$:
 842  002716  032767  040000  175476          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 843  002724  001346                          BNE     TST12           ; GO TO LOOP ERROR
 844  002726  012767  000015  175446          MOV     #15,$FATAL
 845  002734  012767  000001  175436          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 846  002742  005767  175454                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
 847  002746  100401                          BMI     1$              ; CONTINUE IF SET
 848  002750  000000                          HALT            ;<DATA INCORRECT IN REG>
 849  002752                          1$:
 850  002752  032767  002000  175442          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 851  002760  001330                          BNE     TST12           ; GO TO LOOP ON TEST
 852  002762  000401                          BR      TST13
 853  002764  000000                  T12DAT: .WORD   0
 854                                  ;CONTROL STATUS REGISTER (DRCSR) TESTS.
 855  002766                          TST13:
 856  002766  012767  000013  175410          MOV     #13,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 857  002774  005015                          CLR     (R5)            ;CLEAR STATUS REGISTER
 858  002776  011500                          MOV     (R5),R0         ;COPY DATA
 859  003000  032700  000143                  BIT     #143,R0         ;VERIFY THAT IE AND CSR BITS ARE CLEAR
 860  003004  001416                          BEQ     T13CON          ;IF YES, CONTINUE
 861  003006  032767  040000  175406          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 862  003014  001364                          BNE     TST13           ; GO TO LOOP ERROR
 863  003016  012767  000016  175356          MOV     #16,$FATAL
 864  003024  012767  000001  175346          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 865  003032  005767  175364                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
 866  003036  100401                          BMI     1$              ; CONTINUE IF SET
 867  003040  000000                          HALT            ;<STATUS REG DID NOT CLEAR>
 868  003042                          1$:
 869  003042  012715  000140          T13CON: MOV     #140,@R5        ;INTERRUPT ENABLE FOR A+B
 870  003046  011500                          MOV     @R5,R0
 871  003050  032700  000140                  BIT     #140,R0         ;INTERRUPT ENABLE BITS SET?
 872  003054  001016                          BNE     1$              ;CONTINUE IF YES
 873  003056  032767  040000  175336          BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 874  003064  001366                          BNE     T13CON          ; GO TO LOOP ERROR
 875  003066  012767  000017  175306          MOV     #17,$FATAL
 876  003074  012767  000001  175276          MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 877  003102  005767  175314                  TST     $SWREG          ; CHECK FOR HALT ON ERROR
 878  003106  100401                          BMI     1$              ; CONTINUE IF SET
 879  003110  000000                          HALT            ;<ENABLE BITS NOT ON>
 880  003112                          1$:
 881  003112  032767  002000  175302          BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 882  003120  001322                          BNE     TST13           ; GO TO LOOP ON TEST
 883
 884  003122                          TST14:
 885  003122  012767  000014  175254          MOV     #14,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
```

```
003130 012715 000140                MOV     #140,@R5        ;SET INTERRUPT ENABLE FLOPS
003134 000005                       RESET                   ;CLEAR THOSE FLOPS
003136 011500                       MOV     @R5,R0          ;COPY CONTENTS OF DRCSR TO R0
003140 032700 000140                BIT     #140,R0         ;TEST INTERRUPT ENABLE BITS
003144 001416                       BEQ     1$              ;BR IF CLEARED
003146 032767 040000 175246         BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003154 001362                       BNE     TST14           ; GO TO LOOP ERROR
003156 012767 000020 175216         MOV     #20,$FATAL
003164 012767 000001 175206         MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003172 005767 175224                TST     $SWREG          ; CHECK FOR HALT ON ERROR
003176 100401                       BMI     1$              ; CONTINUE IF SET
003200 000000                       HALT            ;<INTERRUPT ENABLE DID NOT CLEAR>
003202                      1$:
003202 032767 002000 175212         BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
003210 001344                       BNE     TST14           ; GO TO LOOP ON TEST

003212                      TST15:
003212 012767 000015 175164         MOV     #15,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
003220 052715 000001                BIS     #1,@R5          ;SHOULD SET REQ A ALSO
003224 032715 000201                BIT     #201,@R5        ;VERIFY THAT REQ A IS SET
003230 001402                       BEQ     5$              ;FLAG ERROR MESSAGE IF NO
003232 005015                       CLR     @R5             ;CLEAR STATUS REGISTER
003234 000416                       BR      1$              ;GO TO NEXT TEST
003236                      5$:
003236 032767 040000 175156         BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003244 001362                       BNE     TST15           ; GO TO LOOP ERROR
003246 012767 000021 175126         MOV     #21,$FATAL
003254 012767 000001 175116         MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003262 005767 175134                TST     $SWREG          ; CHECK FOR HALT ON ERROR
003266 100401                       BMI     1$              ; CONTINUE IF SET
003270 000000                       HALT            ;<A REQ DID NOT SET>
003272                      1$:

```

```
 919  003272                                TST16:
 920  003272  012767  000016  175104              MOV     #16,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 921  003300  052715  000002                      BIS     #2,@R5          ;SHOULD SET REQ B
 922  003304  032715  100002                      BIT     #100002,@R5     ;VERIFY THAT REQ B IS SET
 923  003310  001402                              BEQ     5$              ;FLAG ERROR MESSAGE IF NO
 924  003312  005015                              CLR     @R5             ;CLEAR STATUS REGISTER
 925  003314  000416                              BR      1$              ;GO TO NEXT TEST
 926  003316                                5$:
 927  003316  032767  040000  175076              BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 928  003324  001362                              BNE     TST16           ; GO TO LOOP ERROR
 929  003326  012767  000022  175046              MOV     #22,$FATAL
 930  003334  012767  000001  175036              MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 931  003342  005767  175054                      TST     $SWREG          ; CHECK FOR HALT ON ERROR
 932  003346  100401                              BMI     1$              ; CONTINUE IF SET
 933  003350  000000                              HALT            ;<B REQ DID NOT SET>
 934  003352                                1$:
 935
 936  003352                                TST17:
 937  003352  012767  000017  175024              MOV     #17,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 938  003360  106427  000340                      MTPS    #340            ;LOCK OUT INTERRUPTS
 939  003364  052715  177777                      BIS     #-1,@R5         ;LOAD ALL ONES IN STATUS REGISTER
 940  003370  022715  100343                      CMP     #100343,(R5)    ;VERIFY THAT ALL WRITE BITS ARE SET IN DRCSR
 941  003374  001416                              BEQ     T17CON          ;BR IF SET
 942  003376  032767  040000  175016              BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 943  003404  001362                              BNE     TST17           ; GO TO LOOP ERROR
 944  003406  012767  000023  174766              MOV     #23,$FATAL
 945  003414  012767  000001  174756              MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 946  003422  005767  174774                      TST     $SWREG          ; CHECK FOR HALT ON ERROR
 947  003426  100401                              BMI     1$              ; CONTINUE IF SET
 948  003430  000000                              HALT            ;<INCORRECT DATA IN REG>
 949  003432                                1$:
 950  003432  042715  000003            T17CON:  BIC     #3,@R5          ;CLEAR CSR BITS
 951  003436  032715  000140                      BIT     #140,@R5        ;TEST INTERRUPT ENABLE BITS
 952  003442  001016                              BNE     1$              ;CONTINUE IF STILL SET
 953  003444  032767  040000  174750              BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 954  003452  001367                              BNE     T17CON          ; GO TO LOOP ERROR
 955  003454  012767  000024  174720              MOV     #24,$FATAL
 956  003462  012767  000001  174710              MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 957  003470  005767  174726                      TST     $SWREG          ; CHECK FOR HALT ON ERROR
 958  003474  100401                              BMI     1$              ; CONTINUE IF SET
 959  003476  000000                              HALT            ;<WRONG BITS SET>
 960  003500                                1$:
 961  003500  032767  002000  174714              BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 962  003506  001321                              BNE     TST17           ; GO TO LOOP ON TEST
 963
 964  003510                                TST20:
 965  003510  012767  000020  174666              MOV     #20,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 966  003516  106427  000340                      MTPS    #340            ;LOCK OUT INTERRUPTS
 967  003522  052715  000003                      BIS     #3,@R5          ;SET CSR BITS
 968  003526  000005                              RESET                   ;SHOULD CLEAR INTERRUPT ENABLE FLOPS
 969  003530  032715  000140                      BIT     #140,@R5        ;VERIFY THAT FLOPS ARE CLEARED
 970  003534  001416                              BEQ     1$              ;BR IF YES
 971  003536  032767  040000  174656              BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 972  003544  001361                              BNE     TST20           ; GO TO LOOP ERROR
 973  003546  012767  000025  174626              MOV     #25,$FATAL
 974  003554  012767  000001  174616              MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
```

```
003562 005767 174634                     TST     $SWREG          ; CHECK FOR HALT ON ERROR
003566 100401                            BMI     1$              ; CONTINUE IF SET
003570 000000                            HALT            ;<RESET DID NOT CLEAR BITS>
003572                           1$:
003572 032767 002000 174622              BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
003600 001343                            BNE     TST20           ; GO TO LOOP ON TEST

                                 ;NOTE: THE WRAP CABLE MUST BE INSTALLED TO EXECUTE
                                 ;       TESTS 21-27
003602                           TST21:
003602 012767 000021 174574              MOV     #21,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
003610 032767 000400 174604              BIT     #BIT8,$SWREG
003616 001402                            BEQ     LP21            ;DO TESTS IF NOT INHIBITED
003620 000167 000710                     JMP     TST999          ;IF INHIBITED
003624 005015                    LP21:   CLR     @R5             ;CLEAR STATUS REGISTER
003626 005215                            INC     @R5             ;SET CSR0
003630 105715                            TSTB    @R5             ;CHECK FOR REQ A FLAG
003632 100416                            BMI     1$              ;BR IF SET
003634 032767 040000 174560              BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
003642 001357                            BNE     TST21           ; GO TO LOOP ERROR
003644 012767 000026 174530              MOV     #26,$FATAL
003652 012767 000001 174520              MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
003660 005767 174536                     TST     $SWREG          ; CHECK FOR HALT ON ERROR
003664 100401                            BMI     1$              ; CONTINUE IF SET
003666 000000                            HALT            ;<BIT0 DID NOT SET BIT7>
003670                           1$:
003670 032767 002000 174524              BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
003676 001341                            BNE     TST21           ; GO TO LOOP ON TEST
```

```
CVKAFE  MACY11 30(1046)  22-NOV-82  10:43  PAGE 22
CVKAFE.P11    22-NOV-82 10:37          ERROR   26      BIT0 DID NOT SET BIT7                                   SEQ 0021

 1004  003700                                  TST22:
 1005  003700  012767  000022  174476                  MOV     #22,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 1006  003706  012715  000002                          MOV     #2,@R5          ;SET CSR1
 1007  003712  005715                                  TST     @R5             ;CHECK FOR REQ B FLAG
 1008  003714  100416                                  BMI     1$              ;BR IF SET
 1009  003716  032767  040000  174476                  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1010  003724  001365                                  BNE     TST22           ; GO TO LOOP ERROR
 1011  003726  012767  000027  174446                  MOV     #27,$FATAL
 1012  003734  012767  000001  174436                  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1013  003742  005767  174454                          TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1014  003746  100401                                  BMI     1$              ; CONTINUE IF SET
 1015  003750  000000                                  HALT            ;<BIT1 DID NOT SET BIT15>
 1016  003752                                  1$.
 1017  003752  032767  002000  174442                  BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 1018  003760  001347                                  BNE     TST22           ; GO TO LOOP ON TEST
 1019  003762                                  TST23:
 1020  003762  012767  000023  174414                  MOV     #23,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 1021  003770  012715  000003                          MOV     #3,@R5          ;CSR0 AND CSR1
 1022  003774  011500                                  MOV     (R5),R0         ;COPY DATA
 1023  003776  022700  100203                          CMP     #100203,R0      ;COMPARE DATA
 1024  004002  001416                                  BEQ     1$              ;BR IF GOOD DATA IS RECEIVED
 1025  004004  032767  040000  174410                  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1026  004012  001363                                  BNE     TST23           ; GO TO LOOP ERROR
 1027  004014  012767  000030  174360                  MOV     #30,$FATAL
 1028  004022  012767  000001  174350                  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1029  004030  005767  174366                          TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1030  004034  100401                                  BMI     1$              ; CONTINUE IF SET
 1031  004036  000000                                  HALT            ;<INCORRECT BITS SET IN REG>
 1032  004040                                  1$:
 1033  004040  032767  002000  174354                  BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 1034  004046  001345                                  BNE     TST23           ; GO TO LOOP ON TEST
 1035
 1036                                  ;CAN WE RAISE INTERRUPT "A"
 1037  004050                                  TST24:
 1038  004050  012767  000024  174326                  MOV     #24,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
 1039  004056  106427  000340                          MTPS    #340            ;LOCK OUT INTERRUPTS
 1040  004062  012706  001200                          MOV     #STKPTR,SP      ;INITIALIZE STACK POINTER
 1041  004066  012777  004110  175134                  MOV     #4$,@DRVECA     ;INTERRUPT RETURN POINTER
 1042  004074  012715  000101                          MOV     #101,@R5        ;INTERRUPT ENABLE AND CSR0
 1043  004100  106427  000000                          MTPS    #0              ;ALLOW INTERRUPTS
 1044  004104  000240                                  NOP
 1045  004106  000402                                  BR      5$              ;NO INTERRUPT
 1046
 1047  004110  005015                          4$:     CLR     @R5             ; CLEAR INTERRUPT ENABLE
 1048  004112  000416                                  BR      1$              ;GO TO NEXT TEST
 1049  004114                                  5$:
 1050  004114  032767  040000  174300                  BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
 1051  004122  001352                                  BNE     TST24           ; GO TO LOOP ERROR
 1052  004124  012767  000031  174250                  MOV     #31,$FATAL
 1053  004132  012767  000001  174240                  MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
 1054  004140  005767  174256                          TST     $SWREG          ; CHECK FOR HALT ON ERROR
 1055  004144  100401                                  BMI     1$              ; CONTINUE IF SET
 1056  004146  000000                                  HALT            ;<NO A INTERRUPT>
 1057  004150                                  1$:
 1058  004150  032767  002000  174244                  BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
 1059  004156  001334                                  BNE     TST24           ; GO TO LOOP ON TEST
```

```
1060
1061                                        ;RAISE INTERRUPT "B"
1062  004160                                TST25:
1063  004160  012767  000025  174216              MOV     #25,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
1064  004166  012706  001200                      MOV     #STKPTR,SP      ;INITIALIZE STACK POINTER
1065  004172  106427  000340                      MTPS    #340            ;LOCK OUT INTERRUPTS
1066  004176  012777  004220  175030              MOV     #4$,@DRVECB     ;INTERRUPT RETURN POINTER
1067  004204  012715  000042                      MOV     #42,@R5         ;IE AND CSR1
1068  004210  106427  000000                      MTPS    #0              ;ALLOW INTERRUPTS
1069  004214  000240                              NOP
1070  004216  000402                              BR      5$              ;NO INTERRUPT
1071  004220  005015                      4$:     CLR     @R5             ;CLEAR INTERRUPT ENABLE
1072  004222  000416                              BR      1$              ;GO TO NEXT TEST
1073  004224                                5$:
1074  004224  032767  040000  174170              BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
1075  004232  001352                              BNE     TST25           ; GO TO LOOP ERROR
1076  004234  012767  000032  174140              MOV     #32,$FATAL
1077  004242  012767  000001  174130              MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
1078  004250  005767  174146                      TST     $SWREG          ; CHECK FOR HALT ON ERROR
1079  004254  100401                              BMI     1$              ; CONTINUE IF SET
1080  004256  000000                              HALT            ;<NO B INTERRUPT>
1081  004260                                1$:
1082  004260  032767  002000  174134              BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
1083  004266  001334                              BNE     TST25           ; GO TO LOOP ON TEST
1084
1085                                        ;TEST FOR INTERRUPT FROM DEVICE
1086  004270                                TST26:
1087  004270  012767  000026  174106              MOV     #26,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
1088  004276  017702  174730                      MOV     @DRLVLA,R2      ;SAVE INTERRUPT PSW
1089  004302  016777  174736  174722       LP26:  MOV     PL,@DRLVLA      ;LOCK OUT SUCCESSIVE INTERRUPTS
1090  004310  012706  001200                      MOV     #STKPTR,SP      ;INITIALIZE STACK POINTER
1091  004314  012777  004370  174706              MOV     #1$,@DRVECA     ;INTERRUPT RETURN POINTER
1092  004322  012715  000101                      MOV     #101,@R5        ;SET INTERRUPT ENABLE-AND CSR0
1093  004326  106427  000000                      MTPS    #0              ;ALLOW INTERRUPTS
1094  004332  000240                              NOP
1095  004334                                5$:
1096  004334  032767  040000  174060              BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
1097  004342  001352                              BNE     TST26           ; GO TO LOOP ERROR
1098  004344  012767  000033  174030              MOV     #33,$FATAL
1099  004352  012767  000001  174020              MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
1100  004360  005767  174036                      TST     $SWREG          ; CHECK FOR HALT ON ERROR
1101  004364  100401                              BMI     1$              ; CONTINUE IF SET
1102  004366  000000                              HALT            ;<NO DEVICE INTERRUPT>
1103  004370                                1$:
1104  004370  032767  002000  174024              BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
1105  004376  001341                              BNE     LP26            ; GO TO LOOP ON TEST
1106  004400  005015                              CLR     @R5             ;CLEAR INTERRUPT ENABLE
1107  004402  010277  174624                      MOV     R2,@DRLVLA      ;RESTORE INTERRUPT PSW
1108                                        ; PLU WRAP TEST
1109  004406                                TST27:
1110  004406  012767  000027  173770              MOV     #27,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
1111  004414  005000                              CLR     R0              ;SET UP STARTING DATA
1112  004416  010077  174600               WLOOP: MOV     R0,@DROBUF      ; SEND DATA
1113  004422  027700  174576                      CMP     @DRIBUF,R0      ; CHECK THE DATA
1114  004426  001020                              BNE     5$              ; ERROR IF NOT RIGHT
1115  004430  005200                              INC     R0              ; CHANGE DATA
```

```
1116  004432  001434                        BEQ     1$              ;NEXT TEST IF END
1117  004434  022700  031460          3$:   CMP     #31460,R0       ; CHECK FOR TEST MODULE CODE
1118  004440  001411                        BEQ     4$
1119  004442  022700  031461                CMP     #31461,R0
1120  004446  001406                        BEQ     4$
1121  004450  022700  031462                CMP     #31462,R0
1122  004454  001403                        BEQ     4$
1123  004456  022700  031463                CMP     #31463,R0
1124  004462  001355                        BNE     WLOOP
1125  004464  005200                  4$:   INC     R0
1126  004466  000762                        BR      3$              ; RECHECK DATA CODE
1127  004470                            5$:
1128  004470  032767  040000  173724        BIT     #BIT14,$SWREG   ; CHECK FOR LOOP ON ERROR
1129  004476  001347                        BNE     WLOOP           ; GO TO LOOP ERROR
1130  004500  012767  000034  173674        MOV     #34,$FATAL
1131  004506  012767  000001  173664        MOV     #1,$MSGTY       ; MOVE ERROR NUM TO MAILBOX
1132  004514  005767  173702                TST     $SWREG          ; CHECK FOR HALT ON ERROR
1133  004520  100401                        BMI     1$              ; CONTINUE IF SET
1134  004522  000000                        HALT            ;<WRAP DATA DID NOT COMPARE>
1135  004524                            1$:
1136  004524  032767  002000  173670        BIT     #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
1137  004532  001325                        BNE     TST27           ; GO TO LOOP ON TEST
```

```
1138
1139
1140  004534                                  TST999:
1141  004534  005237  000406                          INC     @#$PASS         ; INCREMENT PASS COUNT
1142  004540  132767  000040  173653                  BITB    #40,$ENVM       ; WILL APT ALLOW PRINTING?
1143  004546  001010                                  BNE     ACT             ; NO
1144  004550  012700  004654                          MOV     #MSG,R0         ; GET MESSAGE ADDRESS
1145  004554  105737  177564          WAIT:   TSTB    @#TPS           ; CHECK IF TTY READY
1146  004560  100375                                  BPL     WAIT            ; IF NOT
1147  004562  112037  177566                          MOVB    (R0)+,@#TPB     ; PRINT THE CHARACTER
1148  004566  001372                                  BNE     WAIT            ; NEXT IF NOT DONE
1149  004570  013700  000042          ACT:    MOV     @#42,R0         ; CHECK ACT
1150  004574  001405                                  BEQ     GOAGIN          ; KEEP GOING
1151  004576  000005                                  RESET
1152  004600  004710                  $ENDAD: JSR     PC,(R0)         ; ACT HOOKS
1153  004602  000240                                  NOP
1154  004604  000240                                  NOP
1155  004606  000240                                  NOP
1156  004610  000167  174500          GOAGIN: JMP      AROUND ;* DO ANOTHER PASS
1157                                                  ;* ON PASSES >1 THE TITLE PRINTOUT WILL BE SUPPRESSED
1158                                          .EVEN
1159  004614                                  STORE:
1160          177564                          TPS=177564
1161          177566                          TPB=177566
1162  004614  177777                          PASSPT: -1
1163  004616  053103  040513  026506          TITLE1:.ASCIZ .CVKAF-E0   DRV11 DIAGNOSTIC. <15><12>
1164  004624  030105  020040  042040
1165  004632  053122  030461  042040
1166  004640  040511  047107  051517
1167  004646  044524  006503  000012
1168  004654  047105  020104  043117  MSG:    .ASCIZ  .END OF PASS.<15><12>
1169  004662  050040  051501  006523
1170  004670  000012
1171                                          .EVEN
```

```
1172
1173                                    ;ENTER HERE FOR POWER FAIL
1174
1175  004672  010046                    PFAIL:  MOV     %0,-(6)         ;SAVE REGISTER OR STACK
1176  004674  010146                            MOV     %1,-(6)         ;WHEN POWERING DOWN
1177  004676  010246                            MOV     %2,-(6)
1178  004700  010346                            MOV     %3,-(6)
1179  004702  010446                            MOV     %4,-(6)
1180  004704  010546                            MOV     %5,-(6)
1181  004706  016746  173112                    MOV     24,-(6)
1182  004712  010637  004726                    MOV     %6,@#SAVR6      ;STORE STACK POSITION
1183  004716  012737  004730  000024            MOV     #RESTAR,@#24
1184  004724  000000                            HALT                    ;HALT ON POWER DOWN NORMAL
1185  004726  000000                    SAVR6:  0                       ;STACK IS SAVED HERE
1186  004730  016706  177772            RESTAR:MOV      SAVR6,%6        ;RESTORE REGISTER OFF STACK
1187  004734  012667  173064                    MOV     (6)+,24         ;WHEN POWERING UP
1188  004740  012605                            MOV     (6)+,%5
1189  004742  012604                            MOV     (6)+,%4
1190  004744  012603                            MOV     (6)+,%3
1191  004746  012602                            MOV     (6)+,%2
1192  004750  012601                            MOV     (6)+,%1
1193  004752  012600                            MOV     (6)+,%0
1194  004754  000137  001256                    JMP     @#START
1195          000001                            .END
```

```
ABASE = 000000            495
ACDW1 = 000000            495
ACDW2 = 000000            495
ACPUOP= 000000            495     510
ACT     004570           1143    1149#
ADDW0 = 000000            495
ADDW1 = 000000            495
ADDW10= 000000            495
ADDW11= 000000            495
ADDW12= 000000            495
ADDW13= 000000            495
ADDW14= 000000            495
ADDW15= 000000            495
ADDW2 = 000000            495
ADDW3 = 000000            495
ADDW4 = 000000            495
ADDW5 = 000000            495
ADDW6 = 000000            495
ADDW7 = 000000            495
ADDW8 = 000000            495
ADDW9 = 000000            495
ADEVCT= 000000            495     501
ADEVM = 000000            495
AENV  = 000000            495     506
AENVM = 000000            495     507
AFATAL= 000000            495     498
AMADR1= 000000            495
AMADR2= 000000            495
AMADR3= 000000            495
AMADR4= 000000            495
AMAMS1= 000000            495
AMAMS2= 000000            495
AMAMS3= 000000            495
AMAMS4= 000000            495
AMSGAD= 000000            495     503
AMSGLG= 000000            495     504
AMSGTY= 000000            495     497
AMTYP1= 000000            495
AMTYP2= 000000            495
AMTYP3= 000000            495
AMTYP4= 000000            495
APASS = 000000            495     500
APRIOR= 000000            495
AROUND  001314            583     595#   1156
ASWREG= 000000            495     508
ATESTN= 000000            495     499
AUNIT = 000000            495     502
AUSWR = 000000            495     509
AVECT1= 000000            495
AVECT2= 000000            495
BIT0  = 000001            405#    660
BIT00 = 000001            395#    405
BIT01 = 000002            394#    404
BIT02 = 000004            393#    403
BIT03 = 000010            392#    402
BIT04 = 000020            391#    401
```

```
BIT05 = 000040            390#    400
BIT06 = 000100            389#    399
BIT07 = 000200            388#    398
BIT08 = 000400            387#    397
BIT09 = 001000            386#    396
BIT1  = 000002            404#    670
BIT10 = 002000            385#    637     694     710     730     747     764     789     808     826     850     881     899
                          961     979    1001    1017    1033    1058    1082    1104    1136
BIT11 = 004000            384#
BIT12 = 010000            383#
BIT13 = 020000            382#
BIT14 = 040000            381#    619     629     662     672     686     702     722     739     756     781     800     818
                          842     861     873     891     910     927     942     953     971     993    1009    1025    1050
                         1074    1096    1128
BIT15 = 100000            380#
BIT2  = 000004            403#
BIT3  = 000010            402#
BIT4  = 000020            401#
BIT5  = 000040            400#
BIT6  = 000100            399#
BIT7  = 000200            398#
BIT8  = 000400            397#    716     986
BIT9  = 001000            396#    650
BPTVEC= 000014            412#
BUFTST  002356            770#
CON     001540            618     627     643#
COUNT   001242            570#
CR    = 000015            320#
CRLF  = 000200            321#
CSR   = 167770            290#    548     549     550     551
DDISP = 177570            327#
DEVCNT= 000410            542#
DRBHIO  001226            562#    814*    833*    834*
DRCSR   001220            559#    598     612
DRIBUF  001224            561#    659     683     720    1113
DRLVLA  001232            565#   1088    1089*   1107*
DRLVLB  001236            567#
DROBUF  001222            560#    615*    617     656*    698*    699     718*    735*    736     752*    753     795*    796*
                          797     813*    815     832*    836    1112*
DRVECA  001230            564#   1041*   1091*
DRVECB  001234            566#   1066*
DSWR  = 177570            326#
EMTVEC= 000030            415#
ERRNUM= 000402            543#
ERRVEC= 000004            408#
ERR1    001456            614     628#
GOAGIN  004610           1150    1156#
HLT   = 104000            289#
HT    = 000011            318#
IOTVEC= 000020            413#
LF    = 000012            319#
LOOP    001306            589#    590
LOOP1   001272            585#    586     588
LP1     001364            612#    620     630
LP21    003624            987     989#
LP26    004302           1089#   1105
```

```
.P7     002364            773#    779     782
MSG     004654           1144    1168#
N     = 000035            257#    621     622#    631     632#    664     665#    674     675#    688     689#    704     705#
                          724     725#    741     742#    758     759#    783     784#    802     803#    820     821#    844
                          845#    863     864#    875     876#    893     894#    912     913#    929     930#    944     945#
                          955     956#    973     974#    995     996#   1011    1012#   1027    1028#   1052    1053#   1076
                         1077#   1098    1099#   1130    1131#
OPTION= 000426            544#
PASSPT  004614           1162#
PFAIL   004672            600    1175#
PIRQ  = 177772            325#
PIRQVE= 000240            419#
PL      001244            571#   1089
PR0   = 000000            342#
PR1   = 000040            343#
PR2   = 000100            344#
PR3   = 000140            345#
PR4   = 000200            346#
PR5   = 000240            347#
PR6   = 000300            348#
PR7   = 000340            349#
PS    = 177776            322#    323
PSW   = 177776            323#
PWRVEC= 000024            414#
RCSR    001200            548#    595
RCSR1   001210            552#
RESTAR  004730           1183    1186#
RESVEC= 000010            409#
SAVR6   004726           1182*   1185#   1186
STACK = 001100            313#
START   001256            574     582#   1194
START1  001246            485     573#
STKLMT= 177774            324#
STKPTR= 001200            291#    573     640     652    1040    1064    1090
STORE   004614           1159#
SW0   = 000001            377#
SW00  = 000001            367#    377
SW01  = 000002            366#    376
SW02  = 000004            365#    375
SW03  = 000010            364#    374
SW04  = 000020            363#    373
SW05  = 000040            362#    372
SW06  = 000100            361#    371
SW07  = 000200            360#    370
SW08  = 000400            359#    369
SW09  = 001000            358#    368
SW1   = 000002            376#
SW10  = 002000            305#    357#
SW11  = 004000            306#    356#
SW12  = 010000            355#
SW13  = 020000            307#    354#
SW14  = 040000            308#    353#
SW15  = 100000            352#
SW2   = 000004            375#
SW3   = 000010            374#
SW4   = 000020            373#
```

```
SW5    = 000040            372#
SW6    = 000100            371#
SW7    = 000200            370#
SW8    = 000400            369#
SW9    = 001000            304#    368#
TAG      002454            795#    801
TBITVE= 000014             410#
TITLE1   004616            584    1163#
TKVEC  = 000060            417#
TPB    = 177566            587*   1147*   1161#
TPS    = 177564            585     589    1145    1160#
TPVEC  = 000064            418#
TRAPVE= 000034             416#
TRTVEC= 000014             411#
TST1     001356            610#    638
TST10    002446            793#    809
TST11    002544            811#    819     827
TST12    002642            828#    843     851
TST13    002766            852     855#    862     882
TST14    003122            884#    892     900
TST15    003212            902#    911
TST16    003272            919#    928
TST17    003352            936#    943     962
TST2     001542            639     648#    663     695
TST20    003510            964#    972     980
TST21    003602            984#    994    1002
TST22    003700           1004#   1010    1018
TST23    003762           1019#   1026    1034
TST24    004050           1037#   1051    1059
TST25    004160           1062#   1075    1083
TST26    004270           1086#   1097
TST27    004406           1109#   1137
TST3     001772            651     696#    703     711
TST4     002064            714#    723     731
TST5     002164            717     733#    740     748
TST6     002256            750#    757     765
TST7     002350            768#    790
TST999   004534            988    1140#
T12CON   002664            833#
T12DAT   002764            830*    831     853#
T13CON   003042            860     869#    874
T17CON   003432            941     950#    954
T2CN1    001710            671     680#    687
T2CON    001646            661     670#    673
WAIT     004554           1145#   1146    1148
WLOOP    004416           1112#   1124    1129
X      = 000030            256#    611     612#    649     650#    697     698#    715     716#    734     735#    751     752#
                            769     770#    794     795#    812     813#    829     830#    856     857#    885     886#    903
                            904#    920     921#    937     938#    965     966#    985     986#   1005    1006#   1020    1021#
                           1038    1039#   1063    1064#   1087    1088#   1110    1111#
XORFLG   001240            568#
$APTHD   000430            528     534#
$CPUOP   000426            510#    544
$DEVCT   000410            501#    542
$ENDAD   004600           1152#
$ENV     000420            506#
```

```
$ENVM   000421   507#  582   1142
$ETABL  000420   505#
$ETEND  000430   517#  540
$FATAL  000402   483*  498#  543   621*  631*  664*  674*  688*  704*  724*  741*  758*  783*
                 802*  820*  844*  863*  875*  893*  912*  929*  944*  955*  973*  995*  1011*
                 1027* 1052* 1076* 1098* 1130*
$HIBTS  000430   535#
$MAIL   000400   496#  536   540
$MBADR  000432   536#
$MSGAD  000414   503#
$MSGLG  000416   504#
$MSGTY  000400   497#  622*  632*  665*  675*  689*  705*  725*  742*  759*  784*  803*  821*
                 845*  864*  876*  894*  913*  930*  945*  956*  974*  996*  1012* 1028* 1053*
                 1077* 1099* 1131*
$PASS   000406   482*  500#  1141*
$PASTM  000436   538#
$SWREG  000422   508#  619   623   629   633   637   650   662   666   672   676   686   690
                 694   702   706   710   716   722   726   730   739   743   747   756   760
                 764   781   785   789   800   804   808   818   822   826   842   846   850
                 861   865   873   877   881   891   895   899   910   914   927   931   942
                 946   953   957   961   971   975   979   986   993   997   1001  1009  1013
                 1017  1025  1029  1033  1050  1054  1058  1074  1078  1082  1096  1100  1104
                 1128  1132  1136
$TESTN  000404   484*  499#  611*  649*  697*  715*  734*  751*  769*  794*  812*  829*  856*
                 885*  903*  920*  937*  965*  985*  1005* 1020* 1038* 1063* 1087* 1110*
$TSTM   000434   537#
$UNIT   000412   502#
$UNITM  000440   539#
$USWR   000424   509#
.       = 004760 421#  422   424   426   428   430   432   434   436   438   440   442   444
                 446   448   450#  453   455   457   459   461   463   465   467   469   471
                 473   475   477   479   481#  486#  487   489   491#  524   525#  527#  529#
                 541#  643
.$x     000430   524#  529
```

```
COMMEN      1#    420#
ENDCOM      1#    420#
ERNUM     259#    621    631    664    674    688    704    724    741    758    783    802    820    844    863
          875     893    912    929    944    955    973    995   1011   1027   1052   1076   1098   1130
ERR       259#    619    628    662    672    686    702    722    739    756    780    800    818    841    861
          873     891    909    926    942    953    971    993   1009   1025   1049   1073   1095   1127
ERROR     314#
ESCAPE      1#    420#
GETPRI      1#    420#
GETSWR      1#    420#
MULT        1#    420#
NEWTST      1#    420#
POP         1#    420#
PUSH        1#    420#
REPORT      1#    420#
SCOPE     315#
SEQ       260#    610    648    696    714    733    750    768    793    811    828    855    884    902    919
          936     964    984   1004   1019   1037   1062   1086   1109
SETPRI      1#    420#
SETUP       1#    420#
SKIP        1#    420#
SLASH       1#    420#
SPACE     420#
STARS       1#    420#    494    521    523    530
SWRSU       1#    420#
TYPBIN      1#    420#
TYPDEC      1#    420#
TYPNAM      1#    420#
TYPNUM      1#    420#
TYPOCS      1#    420#
TYPOCT      1#    420#
TYPTXT      1#    420#
$TNUM     260#    611    649    697    715    734    751    769    794    812    829    856    885    903    920
          937     965    985   1005   1020   1038   1063   1087   1110
$$ESCA      1#    420#
$$NEWT      1#    420#
$$SKIP      1#    420#
.EQUAT      1#    310#
.HEADE      1#
.KT11       1#
.SETUP      1#
.SWRHI      1#
.$ACT1      1#
.$APTB      1#    310#    492
.$APTH      1#    310#    519
.$APTY      1#    310#
.$ASTA      1#
.$CATC      1#
.$CMTA      1#
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#
.$ERRO      1#
.$ERRT      1#
.$MULT      1#
```

CVKAFE MACY11 30(1046) 22-NOV-82 10:43 PAGE 35
CVKAFE.P11   22-NOV-82 10:37   CROSS REFERENCE TABLE -- MACRO NAMES SEQ 0032

```
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    310#
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#    310#
.$TYPO      1#
.$40CA      1#
.1170       1#


. ABS.  004760     000


 ERRORS DETECTED:  0

 CVKAFE.BIN,CVKAFE.LST/CRF/SOL/NL:TOC=SYSMAC.SML,CVKAFE.P11
 RUN-TIME:  9  7  .5 SECONDS
 RUN-TIME RATIO: 91/18=4.9
 CORE USED:  35K  (70 PAGES)
```