

DRV11

DIAGNOSTIC TEST
MD-11-DVKAF-B

EP-DVKAF-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

The vertical strip on the left contains 15 individual diagnostic test screens. Each screen displays a different set of data, likely representing various engine or system parameters. The data is presented in a structured format, often with headers and columns of values. Some screens also show graphical representations, such as waveforms or bar charts, which are typical for engine diagnostic tools. The overall appearance is that of a multi-page diagnostic manual or a series of test cards.

.REM 2

01-05 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DVKAF-B-D
PRODUCT NAME:	DRV11 TEST
DATE:	OCTOBER 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	R.D. FIORENTINO
REVISED BY:	M. McNALLY

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
1975, 1976

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
SUPPLIED BY IT.
DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164

THIS PROGRAM CONTAINS A SOFTWARE SWITCH REGISTER FOR OPTION SELECTION. FOR IT TO OPERATE THE OPERATOR MUST SELECT THE APPROPRIATE OPTION BY SETTING OR RESETTNG THE RESPECTIVE BIT IN THE WORD.

TO DO THIS , THE LSI-11 MUST BE IN OOT MODE.

4.2 STARTING ADDRESS OR ADDRESSES

200 = START OF TEST--FOR NORMAL TESTING

5. OPERATING PROCEDURE

1. THE PROGRAM WILL CYCLE CONTINUOUSLY UNLESS HALTED BY THE OPERATOR, OR SOME ERROR CONDITION.
2. TO HALT THE PROGRAM, DEPRESS THE BREAK KEY. OOT WILL DISPLAY THE PC AT WHICH IT WAS HALTED.
3. IF NEW OPTIONS ARE TO BE SELECTED IN THE SWR, THEY MUST BE SET AT THIS TIME.
4. CONTINUE THE PROGRAM VIA A "P" OR A "G" COMMAND.

5.1 SOFTWARE SWITCH SETTINGS

BIT15	- CONTINUE ON ERROR	(100000)
BIT14	- LOOP ON CURRENT ERROR	(040000)
BIT13	- NOT USED	(020000)
BIT12	- NOT USED	(010000)
BIT11	- NOT USED	(004000)
BIT10	- LOOP ON CURRENT TEST	(002000)
BIT9	- RUN TEST MODULE	(001000)
BIT8	- INHIBIT WRAP CABLE	(000400)
BIT7	- NOT USED	(000200)
BIT6	- NOT USED	(000100)
BIT5	- NOT USED	(000040)
BIT4	- NOT USED	(000020)
BIT3	- NOT USED	(000010)
BIT2	- NOT USED	(000004)
BIT1	- NOT USED	(000002)
BIT0	- NOT USED	(000001)

5.2 SELECTION OF TEST OPTIONS

1. TO TEST NEWDATA RDY AND DATATRANS SIGNALS, THE SPECIAL WRAP MODULE MUST BE INSTALLED. THE OPERATOR MUST ALSO SET BIT9 IN THE SWITCH REGISTER (LOC. 422).
2. THIS TEST WILL RUN WITH THE WRAP CABLE BY DEFAULT. TO INHIBIT TESTING WITH THE WRAP CABLE, THE OPERATOR MUST SET BIT8 IN THE SWITCH REGISTER (LOC. 422).

5.3 WRAP CABLE

THE WRAP CABLE IS REQUIRED TO TEST TRANSFER OF DATA INTO AND OUT OF THE INPUT BUFFER, AND THE DEVICE INTERRUPTS.

NOTE !!!!! THIS DIAGNOSTIC IS APPROXIMATELY 95% EFFECTIVE

E01

WHEN RUN WITH THE WRAP CABLE, AND APPROXIMATELY
60-70% EFFECTIVE WHEN RUN WITHOUT IT.

5.4 TESTING OTHER DRV11 MODULES

TO TEST A DRV11 NOT ADDRESSED AS 167770, OR
VECTORED AT 300, THE OPERATOR MUST SUPPLY THE NEW ADDRESSES
AND VECTORS TO THE PROGRAM BY DEPOSITING THEM AT
THE LOCATIONS TAGGED BY "RCSR" IN THE BEGINNING OF THE LISTING.
THE ORDER IS AS FOLLOWS:

RCSR: CSR ADDRESS
OUTPUT BUFFER ADDRESS
INPUT BUFFER ADDRESS
HIGH BYTE ADDR. OF OUTPUT BUFFER OR
(OUTPUT BUFFER ADDR -1)
"A" INTERRUPT VECTOR ADDRESS
"A" ADDRESS + 2
"B" INTERRUPT VECTOR ADDRESS
"B" ADDRESS + 2

5.5 EXECUTION TIME

TYPICAL RUN TIMES (ONE PASS)
QUICK VERIFY 1 SEC.
WITH WRAP CABLE 10 SEC.

6. ERRORS

ALL ERROR REPORTS WITHIN THIS TEST ARE IN THE FORM
OF AN ERROR HALT. ON THE LSI-11, A HALT WILL FORCE
ODT TO DISPLAY THE PC+2 OF THE HALT. THIS IS
THE PRIMARY ERROR INDICATOR WITHIN THE PROGRAM.
UPON DETECTION OF AN ERROR, THE PROGRAM WILL PLACE THE
CURRENT ERROR NUMBER AND THE CURRENT TEST IN THE MAILBOX
(SEE IMPORTANT TAGS SEC. 8)
TO DETERMINE THE TYPE OF ERROR, THE OPERATOR MUST REFER-
ENCE THE LISTING.

6.1 ERROR RECOVERY

IN ORDER TO CONTINUE, THE OPERATOR MUST ISSUE A "P" TO
CONTINUE THE PROGRAM, OR MAY SET THE ERROR LOOP SWITCH
PRIOR TO CONTINUING.

6. ERRORS

6.1 ERROR REPORTING

ALL ERROR REPORTS WILL BE DONE VIA A HALT WITHIN THE
PROGRAM. THIS WILL CAUSE ODT TO DISPLAY THE PC+2 OF THE
ERROR HALT. AT THIS TIME THE OPERATOR MUST REFERENCE
THE LISTING TO DETERMINE THE ERROR DESCRIPTION.
THE NUMBER AT TAG SFATAL IN THE APT MAILBOX CONTAINS
THE ERROR NUMBER AND MAY BE USED TO REFERENCE THE DE-
SCRIPTION IN THE TABLE OF CONTENTS.

165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220

GO1

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 7
DVKAFB.P11

265
266
267
268
269
270
271
272

:REVISION
1.
2.

B
CHANGED TEST REQUIRING WRAP CABLE TO BE
INHIBITED BY BIT 8 OF THE SMR
INITIALIZED A & B INTERRUPT VECTOR LOCATIONS
WITH TRAP CATCHER

273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328

104000
167770
001200

000000
000001
000002
000003
000004
000005
000006
000007

001000
002000
004000
020000
040000

001100

000011
000012
000015
000200
177776

177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007

```
;GENERAL REGISTER LOGIC TEST
HLT=104000
CSR=167770
STKPTR=1200
;REGISTER DEFINITIONS
R0=X0
R1=X1
R2=X2
R3=X3
R4=X4
R5=X5
SP=X6
PC=X7

;SWITCHES
SW9=1000
SW10=2000
SW11=4000
SW13=20000
SW14=40000

.MCALL .STYPE, .STRAP, .EQUAT, .SAPTBL, .SAPTHR, .SAPTYPE
.SBTTL BASIC DEFINITIONS

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

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

;#GENERAL PURPOSE REGISTER DEFINITIONS
R0= X0      ;;GENERAL REGISTER
R1= X1      ;;GENERAL REGISTER
R2= X2      ;;GENERAL REGISTER
R3= X3      ;;GENERAL REGISTER
R4= X4      ;;GENERAL REGISTER
R5= X5      ;;GENERAL REGISTER
R6= X6      ;;GENERAL REGISTER
R7= X7      ;;GENERAL REGISTER
.EQUIV R6,SP ;;STACK POINTER
.EQUIV R7,PC ;;PROGRAM COUNTER

;#PRIORITY LEVEL DEFINITIONS
```


000000
000040
000100
000140
000200
000240
000300
000340

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

000000
000040
000100
000140
000200
000240
000300
000340

PR0= 0
PR1= 40
PR2= 100
PR3= 140
PR4= 200
PR5= 240
PR6= 300
PR7= 340

:: PRIORITY LEVEL 0
:: PRIORITY LEVEL 1
:: PRIORITY LEVEL 2
:: PRIORITY LEVEL 3
:: PRIORITY LEVEL 4
:: PRIORITY LEVEL 5
:: PRIORITY LEVEL 6
:: PRIORITY LEVEL 7

::* SWITCH REGISTER* SWITCH DEFINITIONS

SW15= 100000
SW14= 40000
SW13= 20000
SW12= 10000
SW11= 4000
SW10= 2000
SW09= 1000
SW08= 400
SW07= 200
SW06= 100
SW05= 50
SW04= 20
SW03= 10
SW02= 5
SW01= 2
SW00= 1
SW09, SW08, SW07, SW06, SW05, SW04, SW03, SW02, SW01, SW00, SW09, SW08, SW07, SW06, SW05, SW04, SW03, SW02, SW01, SW00

::* DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000
BIT14= 40000
BIT13= 20000
BIT12= 10000
BIT11= 4000
BIT10= 2000
BIT09= 1000
BIT08= 400
BIT07= 200
BIT06= 100
BIT05= 50
BIT04= 20
BIT03= 10
BIT02= 5
BIT01= 2
BIT00= 1
BIT09, BIT8
BIT08, BIT7

38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00

000004
000010
000014
000014
000014
000014
000020
000024
000030
000034
000060
000064
000240

000000
000002
000004
000006
000010
000012
000014
000016
000020
000022
000024
000026
000030
000032
000034
000036
000040
000042
000044
000046
000050
000052
000054
000056
000060
000062
000064
000066
000100
000102
000104

.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
ERRVEC= 4
RESVEC= 10
TBITVEC=14
TRTVEC= 14
BPTVEC= 14
IOTVEC= 20
PMRVEC= 24
EMTVEC= 30
TRAPVEC=34
TKVEC= 60
TPVEC= 64
PIRQVEC=240

.ENABLE ABS
. =0
. +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
. +2
HALT
. +2
HALT
. +2
HALT
. +2
HALT
. +2
HALT
. +2
HALT
. +2
HALT
. +2
HALT
. +2
HALT
. +2
HALT
=100
RTI
. +2

;; TIME OUT AND OTHER ERRORS
;; RESERVED AND ILLEGAL INSTRUCTIONS
;; "T" BIT
;; TRACE TRAP
;; BREAKPOINT TRAP (BPT)
;; INPUT/OUTPUT TRAP (IOT) ##SCOPE##
;; POWER FAIL
;; EMULATOR TRAP (EMT) ##ERROR##
;; "TRAP" TRAP
;; TTY KEYBOARD VECTOR
;; TTY PRINTER VECTOR
;; PROGRAM INTERRUPT REQUEST VECTOR

; RTI FOR POSSIBLE CLOCK INTERRUPT

```

441 000106 000000          HALT
442 000110 000112          .+2
443 000112 000000          HALT
444 000114 000116          .+2
445 000116 000000          HALT
446 000120 000122          .+2
447 000122 000000          HALT
448 000124 000126          .+2
449 000126 000000          HALT
450 000130 000132          .+2
451 000132 000000          HALT
452 000134 000136          .+2
453 000136 000000          HALT
454 000140 000142          .+2
455 000142 000000          HALT
456 000144 000146          .+2
457 000146 000000          HALT
458 000150 000152          .+2
459 000152 000000          HALT
460 000154 000156          .+2
461 000156 000000          HALT
462 000160 000162          .+2
463 000162 000000          HALT
464 000164 000166          .+2
465 000166 000000          HALT
466 000170 000172          .+2
467 000172 000000          HALT
468          000200          .-200
469 000200 005067 000202 CLR SPASS ; CLEAR PASS COUNT
470 000204 005067 000172 CLR SFATAL
471 000210 005067 000170 CLR STESTN
472 000214 000137 001246 JMP @START1 ; INITIAL START
473          000300          .-300 ; DEVICE INTERRUPT VECTORS
474 000300 000302          .+2
475 000302 000000          HALT
476 000304 000306          .+2
477 000306 000000          HALT
478          000400          .-400
479          .SBTTL APT MAILBOX-ETABLE
480
481 ; *****
482 .EVEN
483 000400          SMAIL: ; APT MAILBOX
484 000400 000000          SMSGTY: .WORD  AMSGTY  ;; MESSAGE TYPE CODE
485 000402 000000          SFATAL: .WORD  AFATAL  ;; FATAL ERROR NUMBER
486 000404 000000          STESTN: .WORD  ATESTN  ;; TEST NUMBER
487 000406 000000          SPASS: .WORD  APASS  ;; PASS COUNT
488 000410 000000          SDEVCT: .WORD  ADEVCT  ;; DEVICE COUNT
489 000412 000000          SUNIT: .WORD  AUNIT  ;; I/O UNIT NUMBER
490 000414 000000          SMSGAD: .WORD  AMSGAD  ;; MESSAGE ADDRESS
491 000416 000000          SMSGLG: .WORD  AMSGLG  ;; MESSAGE LENGTH
492 000420          SETABLE: ; APT ENVIRONMENT TABLE
493 000420          000          SENV: .BYTE  AENV  ;; ENVIRONMENT BYTE
494 000421          000          SENVM: .BYTE  AENVM  ;; ENVIRONMENT MODE BITS
495 000422 000000          SSWREG: .WORD  ASWREG  ;; APT SWITCH REGISTER
496 000424 000000          SUSWR: .WORD  AUSWR  ;; USER SWITCHES

```

L01

497 000426 000000
498
499
500
501
502
503
504 000430
505
506
507
508
509
510
511 000430
512 000024
513 000024 000200
514 000044
515 000044 000430
516 000430
517
518
519
520
521 000430
522 000430 000000
523 000432 000400
524 000434 000010
525 000436 000010
526 000440 000000
527 000442 000014
528 001200
529 000410
530 000402
531 000426
532
533
534
535 001200 167770
536 001202 167772
537 001204 167774
538 001206 167773
539 001210 000300
540 001212 000302
541 001214 000304
542 001216 000306
543
544
545
546 001220 167770
547 001222 167772
548 001224 167774
549 001226 167773
550
551 001230 000300
552 001232 000302

```
SCPUOP: .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, P00=07, Q=10
*   BIT 10=REAL TIME CLOCK
*   BIT  9=FLOATING POINT PROCESSOR
*   BIT  8=MEMORY MANAGEMENT
*
SETEND:
.MEXIT
.SBTTL  APT PARAMETER BLOCK

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

SAPTHD:
SHIGTS: .WORD  0      ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
SMBADR: .WORD  SMAIL  ;; ADDRESS OF APT MAILBOX (BITS 0-15)
STSTM:  .WORD  10     ;; RUN TIM OF LONGEST TEST
SPASTM: .WORD  10     ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
SUNITH:  .WORD  0      ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        .WORD  SETEND-SMAIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)
        .=1200

DEVcnt=$DEVCT
ERRNUM=$FATAL
OPTION=$SCPUOP

;THIS TABLE CONTAINS INITIAL REGISTER AND VECTOR ADDRESSES

RCSR:   CSR
        CSR+2
        CSR+4
        CSR+3
RCSR1:  300
        302
        304
        306

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

DRCSR:  167770      ;; ADDRESS OF DR11-C STATUS REGISTER
DR0BUF: 167772     ;; ADDRESS OF DR OUTPUT BUFFER REG.
DR1BUF: 167774     ;; ADDRESS OF DR INPUT BUFFER REG.
DRBHIO: 167773     ;; HIGH BYTE OF OUTPUT BUFFER REG.

DRVECA: 300        ;; INTERRUPT VECTOR OF UNIT UNDER TEST
DRLVLA: 302
```

MO1

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 13
 DVKAFB.P11 APT PARAMETER BLOCK

```

553 001234 000304      DRVECB: 304      ; INTERRUPT VECTOR
554 001236 000306      DRVLVB: 306
555 001240 000000      XORFLG: 0
556
557 001242 000000      COUNT: 0        ; COUNT LOCATION
558 001244 000240      PL: 240         ; PRIORITY LEVEL
559
560 001246 012706 001200  START1: MOV      #STKPTR, SP
561 001252 000137 001256      JMP      @#START
562
563 001256 012700 001200  ; INITIALIZE ADDRESS AND VECTORS
564 001262 012701 001220  START:  MOV      #RCSR, R0      ; GET ADDRESS OF UNIT UNDER TEST
565
566 001266 012737 004574 000024  MOV      #PFAIL, @#24
567 001274 012021      MOV      (R0)+, (R1)+      ; LOAD INITIAL TEST ADDRESSES
568 001276 012021      MOV      (R0)+, (R1)+
569 001300 012021      MOV      (R0)+, (R1)+
570 001302 012021      MOV      (R0)+, (R1)+
571 001304 012021      MOV      (R0)+, (R1)+
572 001306 012021      MOV      (R0)+, (R1)+
573 001310 012021      MOV      (R0)+, (R1)+
574
575
576 001312      ; DOES RESET CLEAR REGISTER?
577 001312 012767 000001 177064  TST1:
578 001320 016705 177674      LPI:  MOV      #1, $STESTN      ; MOVE TEST NUMBER TO MAILBOX
579 001324 106427 000200      MOV      DRCSR, R5          ; GET ADDRESS OF STATUS REGISTER
580 001330 016737 000056 000004  MTPS      #200            ; TURN OFF INTERRUPTS
581 001336 012777 177777 177656  MOV      ERR1, @#4          ; SET TIME OUT TRAP VECTOR
582 001344 000005      MOV      @-1, @DROBUF      ; PRESET OUTPUT BUFFER
583 001346 017700 177650      RESET
584 001352 001450      MOV      @DROBUF, R0       ; CLEAR DATA REGISTER
585 001354 032767 040000 177040  BEQ      CON              ; GET RESULT OF RESET
586 001362 001356      BIT      #BIT14, $SWREG    ; CHECK FOR LOOP ON ERROR
587 001364 012767 000001 177010  BNE      LPI              ; GO TO LOOP ERROR
588 001372 012767 000001 177000  MOV      #1, $FATAL
589 001400 005767 177016      MOV      #1, $MSGTY        ; MOVE ERROR NUM TO MAILBOX
590 001404 100401      TST      $SWREG           ; CHECK FOR HALT ON ERROR
591 001406 000000      BMI     $S                ; CONTINUE IF SET
592 001410      HALT                    ; <DATA REG DID NOT CLEAR>
593 001410 000431      $S:  BR      CON
594 001412      ERR1:
595 001412 032767 040000 177002  BIT      #BIT14, $SWREG    ; CHECK FOR LOOP ON ERROR
596 001420 001337      BNE      LPI              ; GO TO LOOP ERROR
597 001422 012767 000002 176752  MOV      #2, $FATAL
598 001430 012767 000001 176742  MOV      #1, $MSGTY        ; MOVE ERROR NUM TO MAILBOX
599 001436 005767 176760      TST      $SWREG           ; CHECK FOR HALT ON ERROR
600 001442 100401      BMI     $S                ; CONTINUE IF SET
601 001444 000000      HALT                    ; <TIME WHEN ADDRESSING PLU>
602 001446      $S:
603 001446 032767 002000 176746  BIT      #BIT10, $SWREG    ; CHECK FOR LOOP ON TEST
604 001454 001316      BNE      TST1            ; GO TO LOOP ON TEST
605 001456 000407      BR      TST2
606 001460 012706 001200  $S:  MOV      #STKPTR, SP      ; RESET STACK POINTER
607 001464 012737 000006 000004  MOV      #6, @#4          ; RESTORE TIME OUT TRAP
608 001472 000765      BR      $S
  
```

NO1

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 14
 DVKAFB.P11 ERROR 2 TIME WHEN ADDRESSING PLU

```

609 001474 000772
610
611
612
613
614 001476
615 001476 012767 000002 176700
616 001504 032767 001000 176710
617 001512 001505
618 001514 012706 001200
619 001520 000005
620
621
622 001522 012777 031460 177472
623 001530 000240
624 001532 000240
625 001534 017700 177464
626 001540 032700 000001
627 001544 001016
628 001546 032767 040000 176646
629 001554 001350
630 001556 012767 000003 176616
631 001564 012767 000001 176606
632 001572 005767 176624
633 001576 100401
634 001600 000000
635 001602
636 001602 032700 000002
637 001606 001016
638 001610 032767 040000 176604
639 001616 001371
640 001620 012767 000004 176554
641 001626 012767 000001 176544
642 001634 005767 176562
643 001640 100401
644 001642 000000
645 001644
646 001644 000005
647 001646 000240
648 001650 000240
649 001652 017700 177346
650 001656 005700
651 001660 001416
652 001662 032767 040000 176532
653 001670 001365
654 001672 012767 000005 176502
655 001700 012767 000001 176472
656 001706 005767 176510
657 001712 100401
658 001714 000000
659 001716
660 001716 032767 002000 176476
661 001724 001264
662 001726
663 001726 012767 000003 176450
664 001734 012777 177777 177260

```

CON: BR .-12
 ; TEST "NEWDATA RDY" AND "DATATRANS" SIGNALS IN PLU
 ; NOTE***** THE PLU TEST MODULE MUST BE INSTALLED
 ; TO EXECUTE THIS TEST

TST2:
 MOV #2,STESTN ; MOVE TEST NUMBER TO MAILBOX
 BIT #BIT9,SSWREG ;
 BEQ TST3 ; SKIP TEST IF NOT SELECTED
 MOV #STKPTR,SP ; SET UP STACK POINTER
 RESET ; CLEAR EVERYTHING
 ; THIS RESET SHOULD INITIALIZE THE
 ; SIGNAL LATCHES IN THE TEST MODULE
 MOV #31460,JDROBUF ; PRIME THE LATCHES

MOV #31460,JDROBUF ; TIMING ALLOWANCE
 MOV JDROBUF,RD ; GET DATA
 BIT #BIT0,RD ; CHECK DATA TRANS SIG
 BNE T2CON ; CONTINUE IF PRESENT
 BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
 BNE TST2 ; GO TO LOOP ERROR
 MOV #3,SFATAL
 MOV #1,SMSGTY ; MOVE ERROR NUM TO MAILBOX
 TST SSWREG ; CHECK FOR HALT ON ERROR
 BMI IS ; CONTINUE IF SET
 HALT ; <NO DATA TRANS SIGNAL>

IS:
 T2CON: BIT #BIT1,RD ; CHECK NEW DATA RDY SIGNAL
 BNE T2CN1 ; CONTINUE IF OK
 BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
 BNE T2CON ; GO TO LOOP ERROR
 MOV #4,SFATAL
 MOV #1,SMSGTY ; MOVE ERROR NUM TO MAILBOX
 TST SSWREG ; CHECK FOR HALT ON ERROR
 BMI IS ; CONTINUE IF SET
 HALT ; <NO NEW DATA RDY SIGNAL>

IS:
 T2CN1: RESET ; CLEAR EVERYTHING
 NOP
 NOP
 MOV JDROBUF,RD ; CHECK SIGNAL LATCHES
 TST RD ; SHOULD BE CLEAR
 BEQ IS ; CONTINUE IF CLEAR
 BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
 BNE T2CN1 ; GO TO LOOP ERROR
 MOV #5,SFATAL
 MOV #1,SMSGTY ; MOVE ERROR NUM TO MAILBOX
 TST SSWREG ; CHECK FOR HALT ON ERROR
 BMI IS ; CONTINUE IF SET
 HALT ; <SIGNALS DID NOT CLEAR>

IS:
 BIT #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
 BNE TST2 ; GO TO LOOP ON TEST

TST3:
 MOV #3,STESTN ; MOVE TEST NUMBER TO MAILBOX
 MOV #-1,JDROBUF ; ALL ONES TO REGISTER

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 15
 DVKAFB.P11 ERROR 5 SIGNALS DID NOT CLEAR

```

665 001742 017700 177254      MOV    @DROBUF,RO
666 001746 022700 177777      CMP    @-1,RO
667 001752 001416      BEQ    IS
668 001754 032767 040000 176440      BIT    @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
669 001762 001361      BNE    TST3          ; GO TO LOOP ERROR
670 001764 012767 000006 176410      MOV    @6,$FATAL
671 001772 012767 000001 176400      MOV    @1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
672 002000 005767 176416      TST    SSWREG        ; CHECK FOR HALT ON ERROR
673 002004 100401      BMI    IS           ; CONTINUE IF SET
674 002006 000000      HALT                ; <REGISTER WILL NOT HOLD ALL ONES>
675 002010      IS:
676 002010 032767 002000 176404      BIT    @BIT10,SSWREG ; CHECK FOR LOOP ON TEST
677 002016 001343      BNE    TST3          ; GO TO LOOP ON TEST
678
679      ;WRAP CABLE MUST BE INSTALLED TO EXECUTE THIS TEST
680 002020      TST4:
681 002020 012767 000004 176356      MOV    @4,$TESTN    ; MOVE TEST NUMBER TO MAILBOX
682 002026 032767 000400 176366      BIT    @BIT8,SSWREG
683 002034 001031      BNE    TST5          ; SKIP TEST IF NOT SELECTED
684 002036 012777 177777 177156      MOV    @-1,@DROBUF
685 002044 000005      RESET              ; SET DATA TO ALL ONES
686 002046 005777 177152      TST    @DRIBUF      ; REGISTER SHOULD CLEAR
687 002052 001416      BEQ    IS
688 002054 032767 040000 176340      BIT    @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
689 002062 001356      BNE    TST4          ; GO TO LOOP ERROR
690 002064 012767 000007 176310      MOV    @7,$FATAL
691 002072 012767 000001 176300      MOV    @1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
692 002100 005767 176316      TST    SSWREG        ; CHECK FOR HALT ON ERROR
693 002104 100401      BMI    IS           ; CONTINUE IF SET
694 002106 000000      HALT                ; <REGISTER DID NOT CLEAR BY RESET>
695 002110      IS:
696 002110 032767 002000 176304      BIT    @BIT10,SSWREG ; CHECK FOR LOOP ON TEST
697 002116 001340      BNE    TST4          ; GO TO LOOP ON TEST
698
699      TST5:
700 002120 012767 000005 176256      MOV    @5,$TESTN    ; MOVE TEST NUMBER TO MAILBOX
701 002126 012777 052525 177066      MOV    @52525,@DROBUF ; LOAD TEST DATA INTO BUFFER
702 002134 017700 177062      MOV    @DROBUF,RO   ; COPY DATA FROM BUFFER TO RO
703 002140 022700 052525      CMP    @52525,RO     ; COMPARE DATA
704 002144 001416      BEQ    IS           ; BR IF DATA IS CORRECT
705 002146 032767 040000 176246      BIT    @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
706 002154 001361      BNE    TST5          ; GO TO LOOP ERROR
707 002156 012767 000010 176216      MOV    @10,$FATAL
708 002164 012767 000001 176206      MOV    @1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
709 002172 005767 176224      TST    SSWREG        ; CHECK FOR HALT ON ERROR
710 002176 100401      BMI    IS           ; CONTINUE IF SET
711 002200 000000      HALT                ; <INCORRECT DATA IN REG>
712 002202      IS:
713 002202 032767 002000 176212      BIT    @BIT10,SSWREG ; CHECK FOR LOOP ON TEST
714 002210 001343      BNE    TST5          ; GO TO LOOP ON TEST
715
716      TST6:
717 002212 012767 000006 176164      MOV    @6,$TESTN    ; MOVE TEST NUMBER TO MAILBOX
718 002220 012777 125252 176774      MOV    @125252,@DROBUF ; LOAD TEST DATA INTO BUFFER
719 002226 017700 176770      MOV    @DROBUF,RO   ; COPY DATA FROM BUFFER TO RO
720 002232 022700 125252      CMP    @125252,RO     ; COMPARE DATA
    
```

.MAIN, MACY11 27(732) 04-OCT-76 14:37 PAGE 16
 DVKAFB.P11 ERROR 10 INCORRECT DATA IN REG

```

721 002236 001416          BEQ      IS          ; BR IF DATA IS CORRECT
722 002240 032767 040000 176154  BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
723 002246 001361          BNE      TST6        ; GO TO LOOP ERROR
724 002250 012767 000011 176124  MOV      #11,$FATAL
725 002256 012767 000001 176114  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
726 002264 005767 176132          TST      SSWREG      ; CHECK FOR HALT ON ERROR
727 002270 100401          BMI      IS          ; CONTINUE IF SET
728 002272 000000          HALT                    ; <INCORRECT DATA IN REG>
729 002274
730 002274 032767 002000 176120  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
731 002302 001343          BNE      TST6        ; GO TO LOOP ON TEST
732
733          ;TEST RELIABILITY OF DR11-C OUTPUT BUFFER REGISTER
734 002304          TST7:
735 002304 012767 000007 176072  MOV      #7,$TESTN    ; MOVE TEST NUMBER TO MAILBOX
736 002312 010502          BUFTST: MOV      R5,R2     ; GET ADDRESS OF DRCSR
737 002314 005722          TST      (R2)+        ; R2=ADDRESS OF OUTPUT BUFFER REG.
738 002316 005003          CLR      R3          ; INITIALIZE DATA REGISTER
739 002320 010312          LP7:    MOV      R3,(R2)   ; SEND THE DATA
740 002322 021203          CMP      (R2),R3     ; CHECK THE RECEIVED DATA
741 002324 001004          BNE      SS          ; ERROR IF NOT THE SAME
742 002326 005203          INC      R3          ; INCREMENT THE DATA
743 002330 105703          TSTB    R3          ; CHECK FOR END OF DATA
744 002332 001417          BEQ      IS          ; CONTINUE IF END
745 002334 000771          BR      LP7         ; GO TO SEND DATA IF NOT
746 002336
747 002336 032767 040000 176056  SS:      BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
748 002344 001365          BNE      LP7         ; GO TO LOOP ERROR
749 002346 012767 000012 176026  MOV      #12,$FATAL
750 002354 012767 000001 176016  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
751 002362 005767 176034          TST      SSWREG      ; CHECK FOR HALT ON ERROR
752 002366 100401          BMI      IS          ; CONTINUE IF SET
753 002370 000000          HALT                    ; <DATA INCORRECT IN REG>
754 002372
755 002372 032767 002000 176022  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
756 002400 001341          BNE      TST7        ; GO TO LOOP ON TEST
757
758          ;TEST THAT BYTE REFERENCE TO DR0BUF AFFECT PROPER BYTE ONLY
759 002402          TST10:
760 002402 012767 000010 175774  MOV      #10,$TESTN   ; MOVE TEST NUMBER TO MAILBOX
761 002410 012777 177777 176604  TAG:    MOV      #1,DR0BUF ; SET ALL ONES IN BUFFER
762 002416 105077 176600          CLR      DR0BUF      ; CLEAR LOW BYTE
763 002422 017700 176574          MOV      DR0BUF,R0   ; COPY DATA
764 002426 022700 177400          CMP      #177400,R0  ; VERIFY THAT LOW BYTE IS CLEAR
765 002432 001416          BEQ      IS          ;
766 002434 032767 040000 175760  BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
767 002442 001362          BNE      TAG        ; GO TO LOOP ERROR
768 002444 012767 000013 175730  MOV      #13,$FATAL
769 002452 012767 000001 175720  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
770 002460 005767 175736          TST      SSWREG      ; CHECK FOR HALT ON ERROR
771 002464 100401          BMI      IS          ; CONTINUE IF SET
772 002466 000000          HALT                    ; <LOW BYTE FAILED TO CLEAR>
773 002470
774 002470 032767 002000 175724  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
775 002476 001341          BNE      TST10       ; GO TO LOOP ON TEST
776

```


.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 17
 DVKAFB.P11 ERROR 13 LOW BYTE FAILED TO CLEAR

777	002500				TST11:			
778	002500	012767	000011	175676		MOV	#11, \$TESTN	; MOVE TEST NUMBER TO MAILBOX
779	002506	012777	177777	176506		MOV	#-1, \$DROBUF	; SET ALL ONES IN BUFFER
780	002514	105077	176506			CLRB	\$DROHIO	; CLEAR HIGH BYTE
781	002520	017700	176476			MOV	\$DROBUF, R0	
782	002524	022700	000377			CMP	#377, R0	; VERIFY THAT HIGH BYTE IS CLEAR
783	002530	001416				BEQ	IS	
784	002532	032767	040000	175662		BIT	#BIT14, \$SWREG	; CHECK FOR LOOP ON ERROR
785	002540	001357				BNE	TST11	; GO TO LOOP ERROR
786	002542	012767	000014	175632		MOV	#14, \$FATAL	
787	002550	012767	000001	175622		MOV	#1, \$MSGTY	; MOVE ERROR NUM TO MAILBOX
788	002556	005767	175640			TST	\$SWREG	; CHECK FOR HALT ON ERROR
789	002562	100401				BMI	IS	; CONTINUE IF SET
790	002564	000000				HALT		; <HIGH BYTE FAILED TO CLEAR>
791	002566				IS:			
792	002566	032767	032070	175626		BIT	#BIT10, \$SWREG	; CHECK FOR LOOP ON TEST
793	002574	001341				BNE	TST11	; GO TO LOOP ON TEST
794	002576				TST12:			
795	002576	012767	000012	175600		MOV	#12, \$TESTN	; MOVE TEST NUMBER TO MAILBOX

```

796 002604 005067 000110 CLR T12DAT ; CLEAR DATA LOCATION
797 002610 012704 002720 MOV @T12DAT,R4 ; STORE ADDRESS OF DATA LOCATION
798 002614 005077 176402 CLR @ROBUF ; CLEAR OUTPUT BUFFER
799 002620 105077 176402 T12CON: CLRB @RSHIO ; CLEAR HIGH BYTE
800 002624 105277 176376 38: INCB @RSHIO ; INCREMENT HIGH BYTE
801 002630 105264 000001 INCB I(R4) ; INCREMENT COMPARISON DATA
802 002634 027714 176362 CMP @ROBUF,(R4) ; COMPARE DATA
803 002640 001004 BNE 6S ; BRANCH ON ERROR
804 002642 105764 000001 48: TSTB I(R4) ; FINISHED?
805 002646 001417 BEQ 1S ; YES
806 002650 000765 BR 38 ; CONTINUE TESTING
807 002652 000000
808 002654 032767 040000 175542 68: BIT @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
809 002660 001346 BNE TST12 ; GO TO LOOP ERROR
810 002662 012767 000015 175512 MOV @15,$FATAL
811 002670 012767 000001 175502 MOV @1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
812 002676 005767 175520 TST SSWREG ; CHECK FOR HALT ON ERROR
813 002702 100401 BHI 1S ; CONTINUE IF SET
814 002704 000000 HALT ; <DATA INCORRECT IN REG>
815 002706 032767 002000 175506 18: BIT @BIT10,SSWREG ; CHECK FOR LOOP ON TEST
816 002714 001330 BNE TST12 ; GO TO LOOP ON TEST
817 002716 000401 BR TST13
818 002720 000000
819
820 T12DAT: .WORD 0
821 ; CONTROL STATUS REGISTER (DRCR) TESTS.
822 002722 012767 000013 175454 TST13: MOV @13,$TESTN ; MOVE TEST NUMBER TO MAILBOX
823 002730 005015 CLR (R5) ; CLEAR STATUS REGISTER
824 002732 011500 MOV (R5),R0 ; COPY DATA
825 002734 032700 000143 BIT @143,R0 ; VERIFY THAT IE AND CSN BITS ARE CLEAR
826 002740 001416 BEQ T13CON ; IF YES, CONTINUE
827 002742 032767 040000 175452 BIT @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
828 002750 001364 BNE TST13 ; GO TO LOOP ERROR
829 002752 012767 000016 175422 MOV @16,$FATAL
830 002760 012767 000001 175412 MOV @1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
831 002766 005767 175430 TST SSWREG ; CHECK FOR HALT ON ERROR
832 002772 100401 BHI 1S ; CONTINUE IF SET
833 002774 000000 HALT ; <STATUS REG DID NOT CLEAR>
834 002776 012715 000140 18: BIT @140,$RS ; INTERRUPT ENABLE FOR A+B
835 003002 011500 MOV @RS,R0
836 003004 032700 000140 BIT @140,R0 ; INTERRUPT ENABLE BITS SET?
837 003010 001016 BNE 1S ; CONTINUE IF YES
838 003012 032767 040000 175402 BIT @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
839 003020 001364 BNE T13CON ; GO TO LOOP ERROR
840 003022 012767 000017 175352 MOV @17,$FATAL
841 003030 012767 000001 175342 MOV @1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
842 003036 005767 175360 TST SSWREG ; CHECK FOR HALT ON ERROR
843 003042 100401 BHI 1S ; CONTINUE IF SET
844 003044 000000 HALT ; <ENABLE BITS NOT ON>
845 003046 032767 002000 175346 18: BIT @BIT10,SSWREG ; CHECK FOR LOOP ON TEST
846 003054 001322 BNE TST13 ; GO TO LOOP ON TEST
847
848
849
850 TST14:
851 003056 012767 000014 175320 MOV @14,$TESTN ; MOVE TEST NUMBER TO MAILBOX

```

F02

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 19
 DVKAFB.P11 ERROR 17 ENABLE BITS NOT ON

852	003064	012715	000140		MOV	#140,ARS	: SET INTERRUPT ENABLE FLOPS	
853	003070	000005			RESET		: CLEAR THOSE FLOPS	
854	003072	011500			MOV	ARS,RO	: COPY CONTENTS OF DRCSR TO RO	
855	003074	032700	000140		BIT	#140,RO	: TEST INTERRUPT ENABLE BITS	
856	003100	001416			BEQ	IS	: BR IF CLEARED	
857	003102	032767	040000	175312	BIT	#BIT14,SSWREG	: CHECK FOR LOOP ON ERROR	
858	003110	001362			BNE	TST14	: GO TO LOOP ERROR	
859	003112	012767	000020	175262	MOV	#20,SFATAL		
860	003120	012767	000001	175252	MOV	#1,MSGTY	: MOVE ERROR NUM TO MAILBOX	
861	003126	005767	175270		TST	SSWREG	: CHECK FOR HALT ON ERROR	
862	003132	100401			BMI	IS	: CONTINUE IF SET	
863	003134	000000			HALT		: < INTERRUPT ENABLE DID NOT CLEAR >	
864	003136							
865	003136	032767	002000	175256	BIT	#BIT10,SSWREG	: CHECK FOR LOOP ON TEST	
866	003144	001344			BNE	TST14	: GO TO LOOP ON TEST	
867								
868	003146							
869	003146	012767	000015	175230	TST15:	MOV	#15,STESTN	: MOVE TEST NUMBER TO MAILBOX
870	003154	052715	000001		BIS	#1,ARS	: SHOULD SET REG A ALSO	
871	003160	032715	000201		BIT	#201,ARS	: VERIFY THAT REG A IS SET	
872	003164	001402			BEQ	SS	: FLAG ERROR MESSAGE IF NO	
873	003166	005015			CLR	ARS	: CLEAR STATUS REGISTER	
874	003170	000416			BR	IS	: GO TO NEXT TEST	
875	003172							
876	003172	032767	040000	175222	BIT	#BIT14,SSWREG	: CHECK FOR LOOP ON ERROR	
877	003200	001362			BNE	TST15	: GO TO LOOP ERROR	
878	003202	012767	000021	175172	MOV	#21,SFATAL		
879	003210	012767	000001	175162	MOV	#1,MSGTY	: MOVE ERROR NUM TO MAILBOX	
880	003216	005767	175200		TST	SSWREG	: CHECK FOR HALT ON ERROR	
881	003222	100401			BMI	IS	: CONTINUE IF SET	
882	003224	000000			HALT		: < A REG DID NOT SET >	
883								
884								

```

885 003226          012767 000016 175150 TST16:  MOV      #16,STESTN      ; MOVE TEST NUMBER TO MAILBOX
886 003226          052715 000002          BIT      #2,ARS        ; SHOULD SET REG B
887 003234          032715 100002          BIT      #100002,ARS   ; VERIFY THAT REG B IS SET
888 003240          001402          BEQ      SS           ; FLAG ERROR MESSAGE IF NO
889 003244          005015          CLR      ARS          ; CLEAR STATUS REGISTER
890 003246          000416          BR       IS           ; GO TO NEXT TEST
891 003250          032767 040000 175142 SS:      BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
892 003252          001362          BNE     TST16        ; GO TO LOOP ERROR
893 003254          012767 000022 175112  MOV     #22,$FATAL
894 003256          012767 000001 175102  MOV     #1,$MSGTY
895 003270          005767 175120          TST     SSWREG        ; MOVE ERROR NUM TO MAILBOX
896 003276          100401          BMI     IS           ; CHECK FOR HALT ON ERROR
897 003302          000000          HALT                    ; CONTINUE IF SET
898 003306          012767 000017 175070 TST17:  MOV     #17,STESTN    ; MOVE TEST NUMBER TO MAILBOX
899 003306          106427 000340          MTPS   #340          ; LOCK OUT INTERRUPTS
900 003314          052715 177777          BIS     #1,ARS        ; LOAD ALL ONES IN STATUS REGISTER
901 003320          022715 100343          CMP     #100343,(RS)  ; VERIFY THAT ALL WRITE BITS ARE SET IN DRCSR
902 003324          001416          BEQ     T17CON        ; BR IF SET
903 003330          032767 040000 175062  BIT     #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
904 003332          001362          BNE     TST17        ; GO TO LOOP ERROR
905 003340          012767 000023 175032  MOV     #23,$FATAL
906 003342          012767 000001 175022  MOV     #1,$MSGTY
907 003350          005767 175040          TST     SSWREG        ; MOVE ERROR NUM TO MAILBOX
908 003356          100401          BMI     IS           ; CHECK FOR HALT ON ERROR
909 003362          000000          HALT                    ; CONTINUE IF SET
910 003364          042715 000003          T17CON: BIC     #3,ARS        ; CLEAR CSR BITS
911 003366          032715 000140          BIT     #140,ARS      ; TEST INTERRUPT ENABLE BITS
912 003372          001016          BNE     IS           ; CONTINUE IF STILL SET
913 003376          032767 040000 175014  BIT     #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
914 003400          001367          BNE     T17CON        ; GO TO LOOP ERROR
915 003406          012767 000024 174764  MOV     #24,$FATAL
916 003410          012767 000001 174754  MOV     #1,$MSGTY
917 003416          005767 174772          TST     SSWREG        ; MOVE ERROR NUM TO MAILBOX
918 003424          100401          BMI     IS           ; CHECK FOR HALT ON ERROR
919 003428          000000          HALT                    ; CONTINUE IF SET
920 003430          032767 002000 174760  BIT     #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
921 003432          001321          BNE     TST17        ; GO TO LOOP ON TEST
922 003434          012767 000020 174732 TST20:  MOV     #20,STESTN    ; MOVE TEST NUMBER TO MAILBOX
923 003436          106427 000340          MTPS   #340          ; LOCK OUT INTERRUPTS
924 003438          052715 000003          BIS     #3,ARS        ; SET CSR BITS
925 003440          000005          RESET                    ; SHOULD CLEAR INTERRUPT ENABLE FLOPS
926 003442          032715 000140          BIT     #140,ARS      ; VERIFY THAT FLOPS ARE CLEARED
927 003444          001416          BEQ     IS           ; BR IF YES
928 003446          032767 040000 174722  BIT     #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
929 003448          001361          BNE     TST20        ; GO TO LOOP ERROR
930 003450          012767 000025 174672  MOV     #25,$FATAL
931 003452          012767 000001 174662  MOV     #1,$MSGTY
932 003454          000001          HALT                    ; MOVE ERROR NUM TO MAILBOX

```

```

941 003516 005767 174700      TST      SSWREG      ; CHECK FOR HALT ON ERROR
942 003522 100401      BMI      IS          ; CONTINUE IF SET
943 003524 000000      HALT                    ; <RESET DID NOT CLEAR BITS>
944 003526
945 013526 032767 002000 174666  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
946 003534 001343      BNE      TST20       ; GO TO LOOP ON TEST
947
948 ;NOTE: THE WRAP CABLE MUST BE INSTALLED TO EXECUTE
949 ;TESTS 21-27
950 003536
951 003538 012767 000021 174640  TST21:  MOV      #21,STESTN   ; MOVE TEST NUMBER TO MAILBOX
952 003544 032767 000400 174650  BIT      #BIT10,SSWREG ; DO TESTS IF NOT INHIBITED
953 003550 001402      BEQ      LP21        ; IF INHIBITED
954 003554 000167 000710      JMP      TST999      ; CLEAR STATUS REGISTER
955 003560 005015      CLR      #R5         ; SET CSRD
956 003566 005215      INC      #R5         ; CHECK FOR REG A FLAG
957 003570 032767 040000 174624  TSTB    #R5         ; BR IF SET
958 003576 001357      BMI      IS          ; CHECK FOR LOOP ON ERROR
959 003600 012767 000026 174574  BNE      TST21       ; GO TO LOOP ERROR
960 003606 012767 000001 174564  MOV      #26,$FATAL  ; MOVE ERROR NUM TO MAILBOX
961 003614 005767 174602      MOV      #1,$MSGTY   ; CHECK FOR HALT ON ERROR
962 003620 100401      TST      SSWREG      ; CONTINUE IF SET
963 003622 000000      BMI      IS          ; <BIT0 DID NOT SET BIT7>
964 003624
965 003624 032767 002000 174570  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
966 003632 001341      BNE      TST21       ; GO TO LOOP ON TEST
967
968
969

```

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 22
 DVKAFB.P11 ERROR 26 BIT0 DID NOT SET BIT7

```

970 003634          TST22:
971 003634 012767 000022 174542  MOV      #22,STESTN      ; MOVE TEST NUMBER TO MAILBOX
972 003642 012715 000002          MOV      #2,RS        ; SET CSRI
973 003646 005715          TST      RS          ; CHECK FOR REG B FLAG
974 003650 100416          BMI     IS          ; BR IF SET
975 003652 032767 040000 174542  BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
976 003660 001365          BNE     TST22       ; GO TO LOOP ERROR
977 003662 012767 000027 174512  MOV      #27,$FATAL
978 003670 012767 000001 174502  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
979 003676 005767 174520          TST      SSWREG      ; CHECK FOR HALT ON ERROR
980 003702 100401          BMI     IS          ; CONTINUE IF SET
981 003704 000000          HALT                    ; <BIT1 DID NOT SET BIT15>
982 003706
983 003706 032767 002000 174506  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
984 003714 001347          BNE     TST22       ; GO TO LOOP ON TEST
985 003716
986 003716 012767 000023 174460  TST23:  MOV      #23,STESTN      ; MOVE TEST NUMBER TO MAILBOX
987 003724 012715 000003          MOV      #3,RS        ; CSRI AND CSRI
988 003730 011500          MOV      (RS),R0      ; COPY DATA
989 003732 022700 100203          CMP     #100203,R0    ; COMPARE DATA
990 003736 001416          BEQ     IS          ; BR IF GOOD DATA IS RECEIVED
991 003740 032767 040000 174454  BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
992 003746 001363          BNE     TST23       ; GO TO LOOP ERROR
993 003750 012767 000030 174424  MOV      #30,$FATAL
994 003756 012767 000001 174414  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
995 003764 005767 174432          TST      SSWREG      ; CHECK FOR HALT ON ERROR
996 003770 100401          BMI     IS          ; CONTINUE IF SET
997 003772 000000          HALT                    ; <INCORRECT BITS SET IN REG>
998 003774
999 003774 032767 002000 174420  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
1000 004002 001345          BNE     TST23       ; GO TO LOOP ON TEST
1001
1002
1003          ;CAN WE RAISE INTERRUPT "A"
1004 004004          TST24:
1005 004012 106427 000340          MOV      #24,STESTN      ; MOVE TEST NUMBER TO MAILBOX
1006 004016 012706 001200          MTPS   #340          ; LOCK OUT INTERRUPTS
1007 004022 012777 004044 175200  MOV      #45,ORVECA     ; INITIALIZE STACK POINTER
1008 004030 012715 000101          MOV      #101,RS        ; INTERRUPT RETURN POINTER
1009 004034 106427 000000          MTPS   #0            ; INTERRUPT ENABLE AND CSRI
1010 004040 000240          NOP
1011 004042 000402          BR      SS          ; ALLOW INTERRUPTS
1012
1013 004044 005015          4S:      CLR     RS          ; NO INTERRUPT
1014 004046 000416          BR      IS          ; CLEAR INTERRUPT ENABLE
1015 004050          5S:
1016 004050 032767 040000 174344  BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
1017 004056 001352          BNE     TST24       ; GO TO LOOP ERROR
1018 004060 012767 000031 174314  MOV      #31,$FATAL
1019 004066 012767 000001 174304  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
1020 004074 005767 174322          TST      SSWREG      ; CHECK FOR HALT ON ERROR
1021 004100 100401          BMI     IS          ; CONTINUE IF SET
1022 004102 000000          HALT                    ; <NO A INTERRUPT>
1023 004104
1024 004104 032767 002000 174310  IS:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
1025 004112 001334          BNE     TST24       ; GO TO LOOP ON TEST
    
```

```

1026
1027
1028 004114
1029 004114 012767 000025 174262
1030 004122 012706 001200
1031 004126 106427 000340
1032 004132 012777 004154 175074
1033 004140 012715 000042
1034 004144 106427 000000
1035 004150 000240
1036 004152 000402
1037 004154 005015
1038 004156 000416
1039 004160
1040 004160 032767 040000 174234
1041 004166 001352
1042 004170 012767 000032 174204
1043 004176 012767 000001 174174
1044 004204 005767 174212
1045 004210 100401
1046 004212 000000
1047 004214
1048 004214 032767 002000 174200
1049 004222 001334
1050
1051
1052 004224
1053 004224 012767 000026 174152
1054 004232 017702 174774
1055 004236 016777 175002 174766
1056 004244 012706 001200
1057 004250 012777 004324 174752
1058 004256 012715 000101
1059 004262 106427 000000
1060 004266 000240
1061 004270
1062 004270 032767 040000 174124
1063 004276 001352
1064 004300 012767 000033 174074
1065 004306 012767 000001 174064
1066 004314 005767 174102
1067 004320 100401
1068 004322 000000
1069 004324
1070 004324 032767 002000 174070
1071 004332 001341
1072 004334 005015
1073 004336 010277 174670
1074
1075 004342
1076 004342 012767 000027 174034
1077 004350 005000
1078 004352 010077 174644
1079 004356 027700 174642
1080 004362 001020
1081 004364 005200

:RAISE INTERRUPT "B"
TST25:
MOV #25,STESTN ; MOVE TEST NUMBER TO MAILBOX
MOV #STKPTR,SP ; INITIALIZE STACK POINTER
MTPS #340 ; LOCK OUT INTERRUPTS
MOV #48,ADRVECB ; INTERRUPT RETURN POINTER
MOV #42,ARS ; IE AND CSRI
MTPS #0 ; ALLOW INTERRUPTS
NOP
BR SS ; NO INTERRUPT
CLR ARS ; CLEAR INTERRUPT ENABLE
BR IS ; GO TO NEXT TEST
SS:
BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
BNE TST25 ; GO TO LOOP ERROR
MOV #32,SFATAL
MOV #1,MSGTY ; MOVE ERROR NUM TO MAILBOX
TST SSWREG ; CHECK FOR HALT ON ERROR
BMI IS ; CONTINUE IF SET
HALT ; <NO B INTERRUPT>
IS:
BIT #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
BNE TST25 ; GO TO LOOP ON TEST

:TEST FOR INTERRUPT FROM DEVICE
TST26:
MOV #26,STESTN ; MOVE TEST NUMBER TO MAILBOX
MOV ADRVLA,R2 ; SAVE INTERRUPT PSM
LP26: MOV PL,ADRVLA ; LOCK OUT SUCCESSIVE INTERRUPTS
MOV #STKPTR,SP ; INITIALIZE STACK POINTER
MOV #18,ADRVECA ; INTERRUPT RETURN POINTER
MOV #101,ARS ; SET INTERRUPT ENABLE-AND CSRI
MTPS #0 ; ALLOW INTERRUPTS
NOP
SS:
BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
BNE TST26 ; GO TO LOOP ERROR
MOV #33,SFATAL
MOV #1,MSGTY ; MOVE ERROR NUM TO MAILBOX
TST SSWREG ; CHECK FOR HALT ON ERROR
BMI IS ; CONTINUE IF SET
HALT ; <NO DEVICE INTERRUPT>
IS:
BIT #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
BNE LP26 ; GO TO LOOP ON TEST
CLR ARS ; CLEAR INTERRUPT ENABLE
MOV R2,ADRVLA ; RESTORE INTERRUPT PSM

:PLU WRAP TEST
TST27:
MOV #27,STESTN ; MOVE TEST NUMBER TO MAILBOX
CLR RO ; SET UP STARTING DATA
MLOOP: MOV RO,ADRBUF ; SEND DATA
CMP ADRIBUF,RO ; CHECK THE DATA
BNE SS ; ERROR IF NOT RIGHT
INC RO ; CHANGE DATA
  
```

K02

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 24
 DVKAFB.P11 ERROR 33 NO DEVICE INTERRUPT

1082	004366	001434				BEG	15		:NEXT TEST IF END
1083	004370	022700	031460		3S:	CMP	831460,R0		: CHECK FOR TEST MODULE CODE
1084	004374	001411				BEG	45		
1085	004376	022700	031461			CMP	831461,R0		
1086	004402	001406				BEG	45		
1087	004404	022700	031462			CMP	831462,R0		
1088	004410	001403				BEG	45		
1089	004412	022700	031463			CMP	831463,R0		
1090	004416	001355				BNE	MLOOP		
1091	004420	005200			4S:	INC	R0		
1092	004422	000762				BR	3S		: RECHECK DATA CODE
1093	004424				5S:				
1094	004424	032767	040000	173770		BIT	8BIT14,SSWREG		: CHECK FOR LOOP ON ERROR
1095	004432	001347				BNE	MLOOP		: GO TO LOOP ERROR
1096	004434	012767	000034	173740		MOV	834,SFATAL		
1097	004442	012767	000001	173730		MOV	81,MSGTY		: MOVE ERROR NUM TO MAILBOX
1098	004450	005767	173746			TST	SSWREG		: CHECK FOR HALT ON ERROR
1099	004454	100401				BMI	15		: CONTINUE IF SET
1100	004456	000000				HALT			: <WRAP DATA DID NOT COMPARE>
1101	004460				1S:				
1102	004460	032767	002000	173734		BIT	8BIT10,SSWREG		: CHECK FOR LOOP ON TEST
1103	004466	001325				BNE	TST27		: GO TO LOOP ON TEST

1104									
1105									
1106	004470					TST999:	INC	@#SPASS	: INCREMENT PASS COUNT
1107	004470	005237	000406				BITB	#40, SENVM	: WILL APT ALLOW PRINTING?
1108	004474	132767	000040	173717			BNE	ACT	: NO
1109	004502	001010					MOV	#MSG, RD	: GET MESSAGE ADDRESS
1110	004504	012700	004554			WAIT:	TSTB	@TPS	: CHECK IF TTY READY
1111	004510	105777	000056				BPL	WAIT	: IF NOT
1112	004514	100375					MOV	(RD)+, @TPB	: PRINT THE CHARACTER
1113	004516	112077	000026				BNE	WAIT	: NEXT IF NOT DONE
1114	004522	001372				ACT:	MOV	@#42, RD	: CHECK ACT
1115	004524	013700	000042				BEQ	GOAGIN	: KEEP GOING
1116	004530	001405					RESET		
1117	004532	000005				SENDAD:	JSR	PC, (RD)	: ; ACT HOOKS
1118	004534	004710					NOP		
1119	004536	000240					NOP		
1120	004540	000240					NOP		
1121	004542	000240					NOP		
1122	004544	000167	174506			GOAGIN:	JMP	START	: ; DO ANOTHER PASS
1123	004550	177566				TPB:	.WORD	177566	
1124	004552	177777				PASSPT:	-1		
1125	004554	047105	020104	043117		MSG:	.ASCIZ	.END OF PASS.<15><12>	
1126	004562	050040	051501	006523					
1127	004570	000012							
1128	004572	177564				TPS:	.WORD	177564	

```

1129
1130
1131
1132 004574 010046
1133 004576 010146
1134 004600 010246
1135 004602 010346
1136 004604 010446
1137 004606 010546
1138 004610 016746 173210
1139 004614 010637 004630
1140 004620 012737 004632 000024
1141 004626 000000
1142 004630 000000
1143 004632 016706 177772
1144 004636 012667 173162
1145 004642 012605
1146 004644 012604
1147 004646 012603
1148 004650 012602
1149 004652 012601
1150 004654 012600
1151 004656 000137 001256
1152 000001

;ENTER HERE FOR POWER FAIL
PFAIL: MOV %0,-(6) ;SAVE REGISTER OR STACK
        MOV %1,-(6) ;WHEN POWERING DOWN
        MOV %2,-(6)
        MOV %3,-(6)
        MOV %4,-(6)
        MOV %5,-(6)
        MOV 24,-(6)
        MOV %6,@SAVR6 ;STORE STACK POSITION
        HALT @RESTAR,@24 ;HALT ON POWER DOWN NORMAL
SAVR6: 0 ;STACK IS SAVED HERE
RESTAR:MOV SAVR6,%6 ;RESTORE REGISTER OFF STACK
        MOV (6)+,%4 ;WHEN POWERING UP
        MOV (6)+,%5
        MOV (6)+,%4
        MOV (6)+,%3
        MOV (6)+,%2
        MOV (6)+,%1
        MOV (6)+,%0
        JMP @START
        .END
    
```

ABASE =	000000	482	
ACDW1 =	000000	482	
ACDW2 =	000000	482	
ACPUOP =	000000	482	497
ACT =	004524	1109	11158
ADDND =	000000	482	
ADDW1 =	000000	482	
ADDW10 =	000000	482	
ADDW11 =	000000	482	
ADDW12 =	000000	482	
ADDW13 =	000000	482	
ADDW14 =	000000	482	
ADDW15 =	000000	482	
ADDW2 =	000000	482	
ADDW3 =	000000	482	
ADDW4 =	000000	482	
ADDW5 =	000000	482	
ADDW6 =	000000	482	
ADDW7 =	000000	482	
ADDW8 =	000000	482	
ADDW9 =	000000	482	
ADEVCT =	000000	482	488
ADEVN =	000000	482	
RENV =	000000	482	493
RENVN =	000000	482	494
AFATAL =	000000	482	485
AMADR1 =	000000	482	
AMADR2 =	000000	482	
AMADR3 =	000000	482	
AMADR4 =	000000	482	
AMANS1 =	000000	482	
AMANS2 =	000000	482	
AMANS3 =	000000	482	
AMANS4 =	000000	482	
AMSGAD =	000000	482	490
AMSLC =	000000	482	491
AMSGTY =	000000	482	484
AMTYP1 =	000000	482	
AMTYP2 =	000000	482	
AMTYP3 =	000000	482	
AMTYP4 =	000000	482	
APASS =	000000	482	487
APRIOR =	000000	482	
ASWREG =	000000	482	495
ATESTN =	000000	482	486
AUNIT =	000000	482	489
AUSMR =	000000	482	496
AVECT1 =	000000	482	
AVECT2 =	000000	482	
BIT0 =	000001	3928	626
BIT10 =	000001	3828	392
BIT01 =	000002	3818	351
BIT02 =	000004	3808	390
BIT03 =	000010	3798	389
BIT04 =	000020	3788	388
BIT05 =	000040	3778	387

BIT06 =	000100	376#	386															
BIT07 =	000200	375#	385															
BIT08 =	000400	374#	384															
BIT09 =	001000	373#	383															
BIT1 =	000002	391#	636															
BIT10 =	002000	372#	603	660	676	696	713	730	755	774	792	816	847	865				
		927	945	967	983	999	1024	1048	1070	1102								
BIT11 =	004000	371#																
BIT12 =	010000	370#																
BIT13 =	020000	369#																
BIT14 =	040000	368#	585	595	628	638	652	668	688	705	722	747	766	784				
		808	827	839	857	876	893	908	919	937	959	975	991	1016				
		1040	1062	1094														
BIT15 =	100000	367#																
BIT2 =	000004	390#																
BIT3 =	000010	389#																
BIT4 =	000020	388#																
BIT5 =	000040	387#																
BIT6 =	000100	386#																
BIT7 =	000200	385#																
BIT8 =	000400	384#	682	952														
BIT9 =	001000	383#	616															
BPTVEC =	000014	399#																
BUFTST	002312	736#																
CON	001474	584	593	609#														
COUNT	001242	557#																
CR =	000015	307#																
CRLF =	000200	308#																
CSR =	167770	276#	535	536	537	538												
DDISP =	177570	315#																
DEVCNT =	000410	529#																
DRBHIO	001226	549#	780#	799#	800#													
DRCSR	001220	546#	564	578														
DRIBUF	001224	548#	625	649	686	1079												
DALVLA	001232	552#	1054	1055#	1073#													
DALVLB	001236	554#																
DROBUF	001222	547#	581#	583	622#	664#	665	684#	701#	702	718#	719	761#	762#				
		763	779#	781	798#	802	1078#											
DRVECA	001230	551#	1007#	1057#														
DRVECB	001234	553#	1032#															
DSMR =	177570	313#																
EMTVEC =	000030	402#																
ERRNUM =	000402	530#																
ERRVEC =	000004	395#																
ERR1	001412	580	594#															
GOAGIN	004544	1116	1122#															
HLT =	104000	275#																
HT =	000011	305#																
IOTVEC =	000020	400#																
LF =	000012	306#																
LP1	001320	578#	586	596														
LP21	003560	953	955#															
LP26	004236	1055#	1071															
LP7	002320	739#	745	748														
HSG	004554	1110	1125#															
N =	000035	262#	587	588#	597	598#	630	631#	640	641#	654	655#	670	671#				

S109	=	001000	345	355		
S11	=	000002	363			
S110	=	002000	291	344		
S111	=	004000	293	343		
S112	=	010000	343			
S113	=	020000	293	341		
S114	=	040000	294	340		
S115	=	100000	339			
S116	=	000004	362			
S117	=	000010	361			
S118	=	000020	360			
S119	=	000040	359			
S120	=	000100	358			
S121	=	000200	357			
S122	=	000400	356			
S123	=	001000	290	355		
TAC	=	002410	761	767		
TBITVE	=	000014	397			
TKVEC	=	000060	404			
TPB	=	004550	1113	1123		
TPS	=	004572	1111	1128		
TPVEC	=	000064	405			
TRAPVE	=	000034	403			
TRTVEC	=	000014	398			
TST1	=	001312	576	604		
TST10	=	002402	759	775		
TST11	=	002500	777	785		
TST12	=	002578	794	809		793
TST13	=	002722	818	821		817
TST14	=	003056	850	858		828
TST15	=	003146	868	877		866
TST16	=	003226	885	894		
TST17	=	003306	902	909		928
TST2	=	001476	605	614		628
TST20	=	003444	930	938		946
TST21	=	003536	950	960		968
TST22	=	003634	970	976		984
TST23	=	003716	985	992		1000
TST24	=	004004	1003	1017		1025
TST25	=	004114	1028	1041		1049
TST26	=	004224	1052	1063		
TST27	=	004342	1075	1103		
TST3	=	001726	617	662		669
TST4	=	002020	680	689		697
TST5	=	002120	687	699		706
TST6	=	002212	716	723		731
TST7	=	002304	734	756		
TST999	=	004470	954	1106		
T12CON	=	002620	799			
T12DRT	=	002720	796	797		819
T13CON	=	002776	826	835		840
T17CON	=	003366	907	916		920
T2CN1	=	001644	637	646		653
T2CON	=	001602	627	636		639
WAIT	=	004510	1111	1112		1114
WLOOP	=	004352	1078	1090		1095

.SPONE	18	
.SRAND	18	
.SRDDE	18	
.SRDOC	18	
.SREAD	18	
.SR2RZ	18	
.SSAVE	18	
.SSB2D	18	
.SSB2O	18	
.SSCOP	18	
.SSIZE	18	
.SSUPR	18	
.STRAP	18	2968
.STYPB	18	
.STYPD	18	
.STYPE	18	2968
.STYPO	18	
.S4OCA	18	
.1170	18	

H03

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 37
 DVKAFB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

BEQ	584 907 916	617 936	651 953	667 990	687 1082	704 1084	721 1086	744 1088	765 1116	783	805	826	856	872	889
BIC	870	887	905	933											
BIS	585 705 839 945	595 713 847 952	603 722 855 959	616 730 857 967	626 747 865 975	628 755 871 983	636 766 876 991	638 774 888 999	652 784 893 1016	660 792 908 1024	668 808 917 1040	676 816 919 1048	682 825 927 1062	688 827 935 1070	696 837 937 1094
BITB	1102														
BMI	590 862	600 881	633 898	643 913	657 924	673 942	693 958	710 964	727 974	752 980	771 996	789 1021	813 1045	832 1067	844 1099
BNE	596 714 840 994	596 723 848 992	604 731 858 1000	627 741 866 1017	629 748 877 1025	637 756 894 1041	639 767 909 1049	653 775 918 1063	661 785 920 1071	669 793 928 1080	677 803 938 1090	683 809 946 1095	689 817 960 1103	697 828 968 1109	706 838 976 1114
BPL	1112														
BR	593	605	608	609	745	806	818	874	891	1011	1014	1036	1038	1092	
CLR	469	470	471	738	796	798	823	873	890	955	1013	1037	1072	1077	
CLRB	762	780	799												
CMP	666	703	720	740	764	782	802	906	989	1079	1083	1085	1087	1089	
ENT	301														
HALT	410 443 591 863 742	412 445 601 882 956	414 447 634 899 1081	416 449 644 914 1091	418 451 658 925 1107	420 453 674 943	422 455 694 965	424 457 711 981	426 459 728 997	428 461 753 1022	430 463 772 1046	432 465 790 1068	434 467 814 1100	436 475 833 1141	441 477 845
INC															
INCB	800	801	1081	1091	1107										
IOT	302														
JMP	472	561	954	1122	1151										
JSR	1118														
MOV	560 583 649 707 768 835 903 986 1042 1110 1147 1113	563 587 654 708 769 836 910 987 1043 1115 1148	564 588 655 717 778 841 911 988 1053 1132 1149	566 597 663 718 779 842 921 993 1054 1133 1150	567 598 664 719 781 851 922 994 1055 1134	568 606 665 724 786 852 931 1004 1056 1135	569 607 670 725 787 854 939 1006 1057 1136	570 615 671 735 795 859 940 1007 1058 1137	571 618 681 736 797 860 951 1008 1064 1138	572 622 684 739 799 869 961 1018 1065 1139	573 625 690 749 810 878 962 1019 1073 1140	577 630 691 750 822 879 971 1029 1076 1143	578 631 700 760 824 886 972 1030 1078 1144	580 640 701 761 829 895 977 1032 1096 1145	581 641 702 763 830 896 978 1033 1097 1146
MOV8															
MIPS	579	904	932	1005	1009	1031	1034	1059							
NOP	623	624	647	648	1010	1035	1060	1119	1120	1121					
RESET	582	619	646	685	853	934	1117								
RTI	439														
TST	589 812 1066	599 831 1098	632 843	642 861	650 880	656 897	672 912	686 923	692 941	709 963	726 973	737 979	751 995	770 1020	788 1044
TSTB	743	804	957	1111											
.ASCIZ	1125														
.BYTE	493	494													
.ENABL	1	256	407												
.END	1152														
.ENDC	301	393	407	482	506	509	511	518	587	591	593	597	601	605	630

	634	636	640	644	646	654	658	662	670	674	678	690	694	698	707
	711	715	724	728	732	749	753	757	768	772	776	785	790	794	810
	814	818	829	833	835	841	845	849	859	863	867	878	882	884	895
	899	901	910	914	916	921	925	929	939	943	947	961	965	969	977
	981	985	993	997	1001	1018	1022	1026	1042	1046	1050	1064	1068	1072	1096
.EQUIV	1100	1104													
	301	302	310	325	326	355	356	357	358	359	360	361	362	363	364
	383	384	385	386	387	388	389	390	391	392					
.EVEN	482														
.IF	299	365	393	481	504	506	508	510	517	585	589	593	595	599	603
	628	632	636	638	642	646	652	656	660	668	672	676	688	692	696
	705	709	713	722	726	730	747	751	755	766	770	774	784	788	792
	808	812	816	827	831	835	839	843	847	857	861	865	876	880	884
	893	897	901	908	912	916	919	923	927	937	941	945	959	963	967
	975	979	983	991	995	999	1016	1020	1024	1040	1044	1048	1062	1066	1070
.IFF	1094	1098	1102												
.IIF	299	482	509	511	518										
.LIST	482														
	655	258	407	482	578	587	588	597	598	616	630	631	640	641	654
	750	664	670	671	682	690	691	701	707	708	718	724	725	736	749
	852	761	768	769	779	786	787	796	810	811	823	829	830	841	842
	939	859	860	870	878	879	887	895	896	904	910	911	921	922	932
	1042	940	952	961	962	972	977	978	987	993	994	1005	1018	1019	1030
.MACRO	1043	1043	1054	1064	1065	1077	1096	1097							
.MCALL	299	264	265												
.MEXIT	505	407													
.MLIST	655														
	750	257	407	482	578	587	588	597	598	616	630	631	640	641	654
	852	664	670	671	682	690	691	701	707	708	718	724	725	736	749
	939	761	768	769	779	786	787	796	810	811	823	829	830	841	842
	1042	859	860	870	878	879	887	895	896	904	910	911	921	922	932
.PAGE	265	940	952	961	962	972	977	978	987	993	994	1005	1018	1019	1030
.REN	1	1043	1054	1064	1065	1077	1096	1097							
.REPT	409	273	885	970	1104										
.SBTTL	297	440													
	810	479	506	587	597	630	640	654	670	690	707	724	749	768	786
	1096	829	841	859	878	895	910	921	939	961	977	993	1018	1042	1064
.WORD	484	485	486	487	488	489	490	491	495	496	497	522	523	524	525
	526	527	819	1123	1128										

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

#, DVKAFB.SEQ/SOL/CRF/PAGNUM/NL: TOC=SYSMAC.CO, DVKAFB.P11
RUN-TIME: 24 27 2 SECONDS
RUN-TIME RATIO: 202/55=3.6
CORE USED: 33K (65 PAGES)

