

# DRV11

DIAGNOSTIC TEST  
MD-11-DVKAF-B

EP-DVKAF-B-DL-A

NOV 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN U.S.A.

Screen 1	Screen 2	Screen 3
Screen 4	Screen 5	Screen 6
Screen 7	Screen 8	Screen 9
Screen 10	Screen 11	Screen 12
Screen 13	Screen 14	Screen 15
Screen 16	Screen 17	Screen 18
Screen 19	Screen 20	Screen 21
Screen 22	Screen 23	Screen 24
Screen 25	Screen 26	Screen 27
Screen 28	Screen 29	Screen 30
Screen 31	Screen 32	Screen 33
Screen 34	Screen 35	Screen 36
Screen 37	Screen 38	Screen 39
Screen 40	Screen 41	Screen 42
Screen 43	Screen 44	Screen 45
Screen 46	Screen 47	Screen 48
Screen 49	Screen 50	Screen 51
Screen 52	Screen 53	Screen 54
Screen 55	Screen 56	Screen 57
Screen 58	Screen 59	Screen 60
Screen 61	Screen 62	Screen 63
Screen 64	Screen 65	Screen 66
Screen 67	Screen 68	Screen 69
Screen 70	Screen 71	Screen 72
Screen 73	Screen 74	Screen 75
Screen 76	Screen 77	Screen 78
Screen 79	Screen 80	Screen 81
Screen 82	Screen 83	Screen 84
Screen 85	Screen 86	Screen 87
Screen 88	Screen 89	Screen 90
Screen 91	Screen 92	Screen 93
Screen 94	Screen 95	Screen 96
Screen 97	Screen 98	Screen 99
Screen 100	Screen 101	Screen 102







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 EP OR 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

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

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  
OOT 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 OOT 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.

6.2 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.

8. 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

10. LISTING

```
.ENABLE ABS
.NLIST MD,MC,CND
.LIST ME
```

000001  
 000001

X=1  
 M=1

-----  
 0  
 1  
 2  
 3  
 4  
 5  
 6  
 7  
 8  
 9  
 A  
 B  
 C  
 D  
 E  
 F  
 G  
 H  
 I  
 J  
 K  
 L  
 M  
 N  
 O  
 P  
 Q  
 R  
 S  
 T  
 U  
 V  
 W  
 X  
 Y  
 Z  
 [  
 \  
 ]  
 ^  
 \_  
 `

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 SWR  
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
CSA=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
.SBTL 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
DSWR= 177570 ;;HARDWARE SWITCH REGISTER
DOISP= 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
```



33  
34  
35  
36  
37  
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

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

```

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= 40
SW04= 20
SW03= 10
SW02= 4
SW01= 2
SW00= 1
.EQUIV SW09, SW15
.EQUIV SW08, SW14
.EQUIV SW07, SW13
.EQUIV SW06, SW12
.EQUIV SW05, SW11
.EQUIV SW04, SW10
.EQUIV SW03, SW09
.EQUIV SW02, SW08
.EQUIV SW01, SW07
.EQUIV SW00, SW06
  
```

..\* 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= 40
BIT04= 20
BIT03= 10
BIT02= 4
BIT01= 2
BIT00= 1
.EQUIV BIT09, BIT9
.EQUIV BIT08, BIT8
  
```



```

441 000106 000000
442 000110 000112
443 000112 000000
444 000114 000116
445 000116 000000
446 000120 000122
447 000122 000000
448 000124 000126
449 000126 000000
450 000130 000132
451 000132 000000
452 000134 000136
453 000136 000000
454 000140 000142
455 000142 000000
456 000144 000146
457 000146 000000
458 000150 000152
459 000152 000000
460 000154 000156
461 000156 000000
462 000160 000162
463 000162 000166
464 000166 000000
465 000170 000172
466 000172 000000
467 000174 000200
468 000176 000200
469 000200 005067
470 000204 005067
471 000210 005067
472 000214 000137
473 000300 000300
474 000300 000302
475 000302 000000
476 000304 000306
477 000306 000000
478 000400 000400

```

```

000202
000172
000170
001246

```

```

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
.=200
CLR
CLR
CLR
JMP
.=300
.+2
HALT
.+2
HALT
.=400

```

```

$PASS
$FATAL
$TESTN
$START1

```

```

; CLEAR PASS COUNT
; INITIAL START
; DEVICE INTERRUPT VECTORS

```

.SBTTL APT MAILBOX-ETABLE

```

; *****
.EVEN
$MAIL:
$MSGTY: .WORD   $MSGTY
$FATAL: .WORD   $FATAL
$TESTN: .WORD   $TESTN
$PASS:  .WORD   $PASS
$DEVCT: .WORD   $DEVCT
$UNIT:  .WORD   $UNIT
$MSGAD: .WORD   $MSGAD
$MSGLG: .WORD   $MSGLG
$ETABLE:
$ENV:   .BYTE   $ENV
$ENVH:  .BYTE   $ENVH
$SWREG: .WORD   $SWREG
$USWR:  .WORD   $USWR

;: APT MAILBOX
;: MESSAGE TYPE CODE
;: FATAL ERROR NUMBER
;: TEST NUMBER
;: PASS COUNT
;: DEVICE COUNT
;: I/O UNIT NUMBER
;: MESSAGE ADDRESS
;: MESSAGE LENGTH
;: APT ENVIRONMENT TABLE
;: ENVIRONMENT BYTE
;: ENVIRONMENT MODE BITS
;: APT SWITCH REGISTER
;: 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

$ETEND:
.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.
$APTHDR ;; POINT TO APT HEADER BLOCK
.=.SX ;; RESET LOCATION COUNTER

;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.

$APTHD:
$SHFTS: .WORD 0 ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBAOR: .WORD $MAIL ;; ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT: .WORD 10 ;; RUN TIM OF LONGEST TEST
$PASTM: .WORD 10 ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$SUNIT: .WORD 0 ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
      .WORD $ETEND-$MAIL/2 ;; LENGTH MAILBOX-ETABLE (WORDS)
      =1200

DEVcnt=$DEVC7
ERRNUM=$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
DRBUF: 167772 ;; ADDRESS OF DR OUTPUT BUFFER REG.
DRIBUF: 167774 ;; ADDRESS OF DR INPUT BUFFER REG.
DRSHIO: 167773 ;; HIGH BYTE OF OUTPUT BUFFER REG.

DRVECA: 300 ;; INTERRUPT VECTOR OF UNIT UNDER TEST
DALVLA: 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 ;INITIALIZE ADDRESS AND VECTORS
562
563 001256 012700 001200 START: MOV #RCSR, R0 ;GET ADDRESS OF UNIT UNDER TEST
564 001262 012701 001220 MOV #DRCSA, R1
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 ;DOES RESET CLEAR REGISTER?
576 001312 TST1:
577 001312 012767 000001 177064 LPI: MOV #1, $TESTN ; MOVE TEST NUMBER TO MAILBOX
578 001320 016705 177674 MOV DRCSR, R5 ; GET ADDRESS OF STATUS REGISTER
579 001324 106427 000200 MTPS #200 ; TURN OFF INTERRUPTS
580 001330 016737 000056 000004 MOV ERR1, @#4 ; SET TIME OUT TRAP VECTOR
581 001336 012777 177777 177656 MOV @-1, @OROBUF ; PRESET OUTPUT BUFFER
582 001344 000005 RESET ; CLEAR DATA REGISTER
583 001346 017700 177650 MOV @OROBUF, R0 ; GET RESULT OF RESET
584 001352 001450 BEQ CON
585 001354 032767 040000 177040 BIT #BIT14, $$WREG ; CHECK FOR LOOP ON ERROR
586 001362 001356 BNE LPI ; GO TO LOOP ERROR
587 001364 012767 000001 177010 MOV #1, $FATAL
588 001372 012767 000001 177000 MOV #1, $MSGTY ; MOVE ERROR NUM TO MAILBOX
589 001400 005767 177016 TST $$WREG ; CHECK FOR HALT ON ERROR
590 001404 100401 BMI IS ; CONTINUE IF SET
591 001406 000000 HALT ; <DATA REG DID NOT CLEAR>
592
593 001410 IS:
594 001410 000431 BR CON
595 001412 ERR1:
596 001412 032767 040000 177002 BIT #BIT14, $$WREG ; CHECK FOR LOOP ON ERROR
597 001420 001337 BNE LPI ; GO TO LOOP ERROR
598 001422 012767 000002 176752 MOV #2, $FATAL
599 001430 012767 000001 176742 MOV #1, $MSGTY ; MOVE ERROR NUM TO MAILBOX
600 001436 005767 176760 TST $$WREG ; CHECK FOR HALT ON ERROR
601 001442 100401 BMI IS ; CONTINUE IF SET
602 001444 000000 HALT ; <TIME WHEN ADDRESSING PLU>
603
604 001446 IS:
605 001446 032767 002000 176746 BIT #BIT10, $$WREG ; CHECK FOR LOOP ON TEST
606 001454 001316 BNE TST1 ; GO TO LOOP ON TEST
607 001456 000407 BR TST2
608 001460 012706 001200 SS: MOV #STKPTR, SP ; RESET STACK POINTER
609 001464 012737 000006 000004 MOV #6, @#4 ; RESTORE TIME OUT TRAP
610 001472 000765 BR IS

```

# 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,$TESTN ; MOVE TEST NUMBER TO MAILBOX
BIT #BIT9,$SWREG
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,$ORIBUF ; PRIME THE LATCHES

; TIMING ALLOWANCE
MOV $ORIBUF,R0 ; GET DATA
BIT #BIT0,R0 ; CHECK DATA TRANS SIG
BNE T2CON ; CONTINUE IF PRESENT
BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
BNE TST2 ; GO TO LOOP ERROR
MOV #3,$FATAL
MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
TST $SWREG ; CHECK FOR HALT ON ERROR
BMI $S ; CONTINUE IF SET
; <NO DATA TRANS SIGNAL>

IS:
T2CON: BIT #BIT1,R0 ; CHECK NEW DATA RDY SIGNAL
BNE T2CN1 ; CONTINUE IF OK
BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
BNE T2CON ; GO TO LOOP ERROR
MOV #4,$FATAL
MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
TST $SWREG ; CHECK FOR HALT ON ERROR
BMI $S ; CONTINUE IF SET
; <NO NEW DATA RDY SIGNAL>

IS:
T2CN1: RESET ; CLEAR EVERYTHING
NOP
NOP
MOV $ORIBUF,R0 ; CHECK SIGNAL LATCHES
TST R0 ; SHOULD BE CLEAR
BEQ $S ; CONTINUE IF CLEAR
BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
BNE T2CN1 ; GO TO LOOP ERROR
MOV #5,$FATAL
MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
TST $SWREG ; CHECK FOR HALT ON ERROR
BMI $S ; CONTINUE IF SET
; <SIGNALS DID NOT CLEAR>

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

TST3:
MOV #3,$TESTN ; MOVE TEST NUMBER TO MAILBOX
MOV #-1,$ORIBUF ; 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	20R0BUF, R0	
666	001746	022700	177777		CMP	R-1, R0	
667	001752	001416			BEG	IS	
668	001754	032767	040000	176440	BIT	8BIT14, \$SWREG	; CHECK FOR LOOP ON ERROR
669	001762	001361			BNE	TST3	; GO TO LOOP ERROR
670	001764	012767	000006	176410	MOV	86, \$FATAL	
671	001772	012767	000001	176400	MOV	81, \$MSGTY	; MOVE ERROR NUM TO MAILBOX
672	002000	005767	176416		TST	\$SWREG	; 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						
676	002010	032767	002000	176404	BIT	8BIT10, \$SWREG	; CHECK FOR LOOP ON TEST
677	002016	001343			BNE	TST3	; GO TO LOOP ON TEST
678							
679							
680	002020						
681	002020	012767	000004	176356	MOV	84, \$TESTN	; MOVE TEST NUMBER TO MAILBOX
682	002026	032767	000400	176366	BIT	8BIT8, \$SWREG	
683	002034	001031			BNE	TST5	; SKIP TEST IF NOT SELECTED
684	002036	012777	177777	177156	MOV	R-1, 20R0BUF	
685	002044	000005			RESET		; SET DATA TO ALL ONES
686	002046	005777	177152		TST	20R16UF	; REGISTER SHOULD CLEAR
687	002052	001416			BEG	IS	
688	002054	032767	040000	176340	BIT	8BIT14, \$SWREG	; CHECK FOR LOOP ON ERROR
689	002062	001356			BNE	TST4	; GO TO LOOP ERROR
690	002064	012767	000007	176310	MOV	87, \$FATAL	
691	002072	012767	000001	176300	MOV	81, \$MSGTY	; MOVE ERROR NUM TO MAILBOX
692	002100	005767	176316		TST	\$SWREG	; 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						
696	002110	032767	002000	176304	BIT	8BIT10, \$SWREG	; CHECK FOR LOOP ON TEST
697	002116	001340			BNE	TST4	; GO TO LOOP ON TEST
698							
699	002120						
700	002120	012767	000005	176256	MOV	85, \$TESTN	; MOVE TEST NUMBER TO MAILBOX
701	002126	012777	052525	177066	MOV	852525, 20R0BUF	; LOAD TEST DATA INTO BUFFER
702	002134	017700	177062		MOV	20R0BUF, R0	; COPY DATA FROM BUFFER TO R0
703	002140	022700	052525		CMP	852525, R0	; COMPARE DATA
704	002144	001416			BEG	IS	; BR IF DATA IS CORRECT
705	002146	032767	040000	176246	BIT	8BIT14, \$SWREG	; CHECK FOR LOOP ON ERROR
706	002154	001361			BNE	TST5	; GO TO LOOP ERROR
707	002156	012767	000010	176216	MOV	810, \$FATAL	
708	002164	012767	000001	176206	MOV	81, \$MSGTY	; MOVE ERROR NUM TO MAILBOX
709	002172	005767	176224		TST	\$SWREG	; CHECK FOR HALT ON ERROR
710	002176	100401			BMI	IS	; CONTINUE IF SET
711	002200	000000			HALT		; (INCORRECT DATA IN REG)
712	002202						
713	002202	032767	002000	176212	BIT	8BIT10, \$SWREG	; CHECK FOR LOOP ON TEST
714	002210	001343			BNE	TST5	; GO TO LOOP ON TEST
715							
716	002212						
717	002212	012767	000006	176164	MOV	86, \$TESTN	; MOVE TEST NUMBER TO MAILBOX
718	002220	012777	125252	176774	MOV	8125252, 20R0BUF	; LOAD TEST DATA INTO BUFFER
719	002226	017700	176770		MOV	20R0BUF, R0	; COPY DATA FROM BUFFER TO R0
720	002232	022700	125252		CMP	8125252, R0	; COMPARE DATA

WRAP CABLE MUST BE INSTALLED TO EXECUTE THIS TEST

```

721 002236 001416          BEQ      18          ; 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     18          ; CONTINUE IF SET
728 002272 000000          HALT                    ; <INCORRECT DATA IN REG>
729 002274
730 002274 032767 002000 176120 15:      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          TST7:
735 002304 012767 000007 176072  MOV     #7,$TESTN   ; MOVE TEST NUMBER TO MAILBOX
736 002312 010502  BUFTST: MOV     R5,R2     ; GET ADDRESS OF DRCSR
737 002314 005722          TST     (R2)+       ; R2=ADDRESS OF OUTPUT BUFFER REG.
738 002316 005003          CLR     R3          ; INITIALIZE DATA REGISTER
739 002320 010312  LP7:      MOV     R3,(R2)   ; SEND THE DATA
740 002322 021203          CMP     (R2),R3    ; CHECK THE RECEIVED DATA
741 002324 001004          BNE     55         ; ERROR IF NOT THE SAME
742 002326 005203          INC     R3         ; INCREMENT THE DATA
743 002330 105703          TSTB   R3          ; CHECK FOR END OF DATA
744 002332 001417          BEQ     18         ; CONTINUE IF END
745 002334 000771          BR     LP7         ; GO TO SEND DATA IF NOT
746 002336
747 002336 032767 040000 176056 55:      BIT      #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
748 002344 001365          BNE     LP7        ; GO TO LOOP ERROR
749 002346 012767 000012 175026  MOV     #12,$FATAL
750 002354 012767 000001 176016  MOV     #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
751 002362 005767 176034          TST     SSWREG      ; CHECK FOR HALT ON ERROR
752 002366 100401          BMI     18         ; CONTINUE IF SET
753 002370 000000          HALT                    ; <DATA INCORRECT IN REG>
754 002372
755 002372 032767 002000 176022 15:      BIT      #BIT10,SSWREG ; CHECK FOR LOOP ON TEST
756 002400 001341          BNE     TST7        ; GO TO LOOP ON TEST
757          ;TEST THAT BYTE REFERENCE TO DR0BUF AFFECT PROPER BYTE ONLY
758
759          TST10:
760 002402 012767 000010 175774  MOV     #10,$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     18
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     18         ; CONTINUE IF SET
772 002466 000000          HALT                    ; <LOW BYTE FAILED TO CLEAR>
773 002470
774 002470 032767 002000 175724 15:      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, \$OROBUF	; SET ALL ONES IN BUFFER
780	002514	105077	176506			CLR3	\$OROB10	; CLEAR HIGH BYTE
781	002520	017700	176476			MOV	\$OROBUF, 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	012000	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

# E02

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 18  
 DVKAFB.P11 ERROR 14 HIGH BYTE FAILED TO CLEAR

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	#20R0BUF	: CLEAR OUTPUT BUFFER
799	002620	105077	176402			CLRB	#20R0H10	: CLEAR HIGH BYTE
800	002624	105277	176376		T12CON:	INCB	#20R0H10	: INCREMENT HIGH BYTE
801	002630	105264	000001		38:	INCB	I(R4)	: INCREMENT COMPARISON DATA
802	002634	027714	176362			CMP	#20R0BUF,(R4)	: COMPARE DATA
803	002640	001004				BNE	6S	: BRANCH ON ERROR
804	002644	105764	000001		48:	TSTB	I(R4)	: FINISHED?
805	002648	001417				BEQ	1S	: YES
806	002654	000765				BR	38	: CONTINUE TESTING
807	002658				68:			
808	002664	032767	040000	175542		BIT	#BIT14,SSWREG	: CHECK FOR LOOP ON ERROR
809	002670	001346				BNE	TST12	: GO TO LOOP ERROR
810	002674	012767	000015	175512		MOV	#15,\$FATAL	
811	002678	012767	000001	175502		MOV	#1,\$MSGTY	: MOVE ERROR NUM TO MAILBOX
812	002684	005767	175520			TST	SSWREG	: CHECK FOR HALT ON ERROR
813	002690	100401				BMI	1S	: CONTINUE IF SET
814	002694	000000				HALT		: <DATA INCORRECT IN REG>
815	002700				18:			
816	002706	032767	002000	175506		BIT	#BIT10,SSWREG	: CHECK FOR LOOP ON TEST
817	002712	001330				BNE	TST12	: GO TO LOOP ON TEST
818	002718	000401				BR	TST13	
819	002724	000000						
820						T12DAT: .WORD	0	
821	002728					: CONTROL STATUS REGISTER (DRCSR) TESTS.		
822	002722	012767	000013	175454		TST13:		
823	002726	005015				MOV	#13,\$TESTN	: MOVE TEST NUMBER TO MAILBOX
824	002730	011500				CLR	(R5)	: CLEAR STATUS REGISTER
825	002734	032700	000143			MOV	(R5),R0	: COPY DATA
826	002738	001416				BIT	#143,R0	: VERIFY THAT IE AND CS# BITS ARE CLEAR
827	002742	032767	040000	175452		BEQ	T13CON	: IF YES, CONTINUE
828	002746	001364				BIT	#BIT14,SSWREG	: CHECK FOR LOOP ON ERROR
829	002750	012767	000016	175422		BNE	TST13	: GO TO LOOP ERROR
830	002754	012767	000001	175412		MOV	#16,\$FATAL	
831	002758	005767	175430			MOV	#1,\$MSGTY	: MOVE ERROR NUM TO MAILBOX
832	002762	100401				TST	SSWREG	: CHECK FOR HALT ON ERROR
833	002766	000000				BMI	1S	: CONTINUE IF SET
834	002770					HALT		: <STATUS REG DID NOT CLEAR>
835	002774				18:			
836	002778	012715	000140			T13CON:		
837	002782	011500				MOV	#140,\$RS	: INTERRUPT ENABLE FOR A+B
838	002786	032700	000140			MOV	\$RS,R0	
839	002790	001016				BIT	#140,R0	: INTERRUPT ENABLE #1TS SET?
840	002794	032767	040000	175402		BNE	1S	: CONTINUE IF YES
841	002798	001366				BIT	#BIT14,SSWREG	: CHECK FOR LOOP ON ERROR
842	002802	012767	000017	175352		BNE	T13CON	: GO TO LOOP ERROR
843	002806	012767	000001	175342		MOV	#17,\$FATAL	
844	002810	005767	175360			MOV	#1,\$MSGTY	: MOVE ERROR NUM TO MAILBOX
845	002814	100401				TST	SSWREG	: CHECK FOR HALT ON ERROR
846	002818	000000				BMI	1S	: CONTINUE IF SET
847	002822					HALT		: <ENABLE BITS NOT ON>
848	002826				18:			
849	002830	032767	002000	175346		BIT	#BIT10,SSWREG	: CHECK FOR LOOP ON TEST
850	002834	001322				BNE	TST13	: GO TO LOOP ON TEST
851	002838							
852	002842					TST14:		
853	002846	012767	000014	175320		MOV	#14,\$TESTN	: MOVE TEST NUMBER TO MAILBOX

852	003064	012715	000140		MOV	#140,DRS	: SET INTERRUPT ENABLE FLOPS	
853	003070	000005			RESET		: CLEAR THE FLOPS	
854	003072	011500			MOV	DRS,RO	: COPY CONTENTS OF DRCSR TO RO	
855	003074	032700	000140		BIT	#140,RO	: TEST INTERRUPT ENABLE BITS	
856	003100	001416			BEG	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,\$FATAL		
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,\$TESTN	: MOVE TEST NUMBER TO MAILBOX
870	003154	052715	000001		BIS	#1,DRS	: SHOULD SET REQ A ALSO	
871	003160	032715	000201		BIT	#201,DRS	: VERIFY THAT REQ A IS SET	
872	003164	001402			BEG	SS	: FLAG ERROR MESSAGE IF NO	
873	003166	005015			CLR	DRS	: 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,\$FATAL		
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 REQ DID NOT SET >	
883								
884	003226				IS:			

```

003226 012767 000016 175150 TST16: MOV #16, $TESTN ; MOVE TEST NUMBER TO MAILBOX
003226 052715 000002 BIS #2, $RS ; SHOULD SET REG B
003234 032715 100002 BIT #100002, $RS ; VERIFY THAT REG B IS SET
003244 001402 BEQ $S ; FLAG ERROR MESSAGE IF NO
003246 005015 CLR $RS ; CLEAR STATUS REGISTER
003250 000416 BR ; GO TO NEXT TEST

003276 032767 040000 175142 58: BIT #BIT14, $SWREG ; CHECK FOR LOOP ON ERROR
003282 001362 BNE TST16 ; GO TO LOOP ERROR
003286 012767 000022 175112 MOV #22, $FATAL
003290 012767 000001 175102 MOV #1, $MSGTY ; MOVE ERROR NUM TO MAILBOX
003294 005767 175120 TST $SWREG ; CHECK FOR HALT ON ERROR
003300 100401 BMI ; CONTINUE IF SET
003304 000300 HALT ; <B REG DID NOT SET>

003306 012767 000017 175070 TST17: MOV #17, $TESTN ; MOVE TEST NUMBER TO MAILBOX
003314 106427 000340 MTPS #340 ; LOCK OUT INTERRUPTS
003320 052715 177777 BIS #1, $RS ; LOAD ALL ONES IN STATUS REGISTER
003324 022715 100343 CMP #100343, (RS) ; VERIFY THAT ALL WRITE BITS ARE SET IN DRCSR
003330 001416 BEQ T17CON ; BR IF SET
003332 032767 040000 175062 BIT #BIT14, $SWREG ; CHECK FOR LOOP ON ERROR
003340 001362 BNE TST17 ; GO TO LOOP ERROR
003342 012767 000023 175032 MOV #23, $FATAL
003350 012767 000001 175022 MOV #1, $MSGTY ; MOVE ERROR NUM TO MAILBOX
003356 005767 175040 TST $SWREG ; CHECK FOR HALT ON ERROR
003362 100401 BMI ; CONTINUE IF SET
003364 000000 HALT ; <INCORRECT DATA IN REG>

003366 042715 000003 175014 T17CON: BIC #3, $RS ; CLEAR CSR BITS
003372 032715 000140 BIT #140, $RS ; TEST INTERRUPT ENABLE BITS
003376 001016 BNE ; CONTINUE IF STILL SET
003400 032767 040000 175014 BIT #BIT14, $SWREG ; CHECK FOR LOOP ON ERROR
003406 001362 BNE T17CON ; GO TO LOOP ERROR
003410 012767 000024 174764 MOV #24, $FATAL
003416 012767 000001 174754 MOV #1, $MSGTY ; MOVE ERROR NUM TO MAILBOX
003424 005767 174772 TST $SWREG ; CHECK FOR HALT ON ERROR
003430 100401 BMI ; CONTINUE IF SET
003432 000000 HALT ; <WRONG BITS SET>

003434 032767 002000 174760 18: BIT #BIT10, $SWREG ; CHECK FOR LOOP ON TEST
003442 001321 BNE TST17 ; GO TO LOOP ON TEST

003444 012767 000020 174732 TST20: MOV #20, $TESTN ; MOVE TEST NUMBER TO MAILBOX
003452 106427 000340 MTPS #340 ; LOCK OUT INTERRUPTS
003456 052715 000003 BIS #3, $RS ; SET CSR BITS
003462 000005 RESET ; SHOULD CLEAR INTERRUPT ENABLE FLOPS
003464 032715 000140 BIT #140, $RS ; VERIFY THAT FLOPS ARE CLEARED
003470 001416 BEQ $S ; BR IF YES
003472 032767 040000 174722 BIT #BIT14, $SWREG ; CHECK FOR LOOP ON ERROR
003480 001361 BNE TST20 ; GO TO LOOP ERROR
003486 012767 000025 174672 MOV #25, $FATAL
003490 012767 000001 174662 MOV #1, $MSGTY ; MOVE ERROR NUM TO MAILBOX
  
```



# H02

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 21  
 DVKAFB.P11 ERROR 25 RESET DID NOT CLEAR BITS

```

941 003516 005767 174700      TST    $SWREG      ; CHECK FOR HALT ON ERROR
942 003522 100401      BMI    1$         ; CONTINUE IF SET
943 003524 000000      HALT                ; <RESET DID NOT CLEAR BITS>
944 003528
945 003528 032767 002000 174666 1$: BIT    $BIT10,$SWREG ; 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
951 003536      TST21:
952 003536 012767 000021 174640      MOV    $21,$TESTN ; MOVE TEST NUMBER TO MAILBOX
953 003544 032767 000400 174650      BIT    $BIT8,$SWREG
954 003552 001402      BEQ    LP21
955 003554 000167 000710      JMP    TST1999    ; DO TESTS IF NOT INHIBITED
956 003560 005015      LP21: CLR    $CS      ; IF INHIBITED
957 003564 105715      INC    $CS      ; CLEAR STATUS REGISTER
958 003566 100416      TSTB  $CS      ; SET CSRD
959 003570 032767 040000 174624      BMI    1$         ; CHECK FOR REG A FLAG
960 003576 001357      BIT    $BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
961 003600 012767 000026 174574      BNE    TST21     ; GO TO LOOP ERROR
962 003606 012767 000001 174564      MOV    $26,$FATAL
963 003614 005767 174602      MOV    $1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
964 003620 100401      TST    $SWREG    ; CHECK FOR HALT ON ERROR
965 003622 000000      BMI    1$         ; CONTINUE IF SET
966 003624      HALT                ; <BIT0 DID NOT SET BIT7>
967 003624 032767 002000 174570 1$: BIT    $BIT10,$SWREG ; CHECK FOR LOOP ON TEST
968 003632 001341      BNE    TST21     ; GO TO LOOP ON TEST
969
  
```

```

970 00734          TST22:
971 00 34 012767 000022 174542  MOV      #22,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
972 003642 012715 000022  MOV      #2,$RS      ; SET CSRI
973 003646 005715 000022  TST      #RS      ; CHECK FOR REQ B FLAG
974 00 50 100416 000022  BMI      #S      ; BR IF SET
975 00 2 032767 040000 174542  BIT      #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
976 00 30 001365 000022  BNE     TST22      ; GO TO LOOP ERROR
977 003662 012767 000027 174512  MOV      #27,$FATAL
978 003670 012767 000001 174502  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
979 003676 005767 174520  TST      $SWREG      ; CHECK FOR HALT ON ERROR
980 003702 100401 000000  BMI      #S      ; CONTINUE IF SET
981 003704 000000 000000  HALT                    ; <BIT1 DID NOT SET BIT15>
982 003706
983 003706 032767 002000 174506  1S:  BIT      #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
984 003714 001347 000000  BNE     TST22      ; GO TO LOOP ON TEST
985 003716
986 003716 012767 000023 174460  TST23: MOV      #23,$TESTN ; MOVE TEST NUMBER TO MAILBOX
987 003724 012715 000003  MOV      #3,$RS      ; CSRI AND CSRI
988 003730 011500 000000  MOV      (RS),R0     ; COPY DATA
989 003732 022700 100203  CMP      #100203,R0  ; COMPARE DATA
990 003736 001416 000000  BEQ      #S      ; BR IF GOOD DATA IS RECEIVED
991 003740 032767 040000 174454  BIT      #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
992 003746 001363 000000  BNE     TST23      ; GO TO LOOP ERROR
993 003750 012767 000030 174424  MOV      #30,$FATAL
994 003756 012767 000001 174414  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
995 003764 005767 174432  TST      $SWREG      ; CHECK FOR HALT ON ERROR
996 003770 100401 000000  BMI      #S      ; CONTINUE IF SET
997 003772 000000 000000  HALT                    ; <INCORRECT BITS SET IN REG>
998 003774
999 003774 032767 002000 174420  1S:  BIT      #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
1000 004002 001345 000000  BNE     TST23      ; GO TO LOOP ON TEST
1001
1002
1003 004004          :CAN WE RAISE INTERRUPT "A"
1004 004004 012767 000024 174372  TST24: MOV      #24,$TESTN ; MOVE TEST NUMBER TO MAILBOX
1005 004012 106427 000340  MTPS    #340        ; LOCK OUT INTERRUPTS
1006 004016 012706 001200  MOV      #STKPTR,SP  ; INITIALIZE STACK POINTER
1007 004022 012777 004044 175200  MOV      #48,$ORVECA ; INTERRUPT RETURN POINTER
1008 004030 012715 000101  MOV      #101,$RS    ; INTERRUPT ENABLE AND CSRI
1009 004034 106427 000000  MTPS    #0          ; ALLOW INTERRUPTS
1010 004040 00 240 000000  NOP
1011 004042 000402 000000  BR      #S          ; NO INTERRUPT
1012
1013 004044 005015 000000  4S:  CLR      #RS      ; CLEAR INTERRUPT ENABLE
1014 004046 000416 000000  BR      #S          ; GO TO NEXT TEST
1015 004050
1016 0040 0 032767 040000 174344  5S:  BIT      #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
1017 0040 6 001352 000000  BNE     TST24      ; GO TO LOOP ERROR
1018 0040 0 012767 000031 174314  MOV      #31,$FATAL
1019 0040 6 012767 000001 174304  MOV      #1,$MSGTY    ; MOVE ERROR NUM TO MAILBOX
1020 004074 005767 174322  TST      $SWREG      ; CHECK FOR HALT ON ERROR
1021 004100 10 401 000000  BMI      #S      ; CONTINUE IF SET
1022 004102 000000 000000  HALT                    ; <NO A INTERRUPT>
1023 004104
1024 004104 032767 002000 174310  1S:  BIT      #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
1025 004112 001334 000000  BNE     TST24      ; GO TO LOOP ON TEST

```

```

1026
1027
1028 004114
1029 004114 012767 000C25 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 018777 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"  
 †ST25:  
 MOV #25, STESTN ; MOVE TEST NUMBER TO MAILBOX  
 MOV #STKPTR, SP ; INITIALIZE STACK POINTER  
 MTPS #340 ; LOCK OUT INTERRUPTS  
 MOV #48, @ORVECB ; INTERRUPT RETURN POINTER  
 MOV #42, @RS ; IE AND CSRI  
 MTPS #0 ; ALLOW INTERRUPTS  
 NOP  
 BR SS ; NO INTERRUPT  
 CLR @RS ; CLEAR INTERRUPT ENABLE  
 BR IS ; GO TO NEXT TEST  
 SS:  
 BIT #BIT14, \$SWREG ; CHECK FOR LOOP ON ERROR  
 BNE TST25 ; GO TO LOOP ERROR  
 MOV #32, \$FATAL  
 MOV #1, \$MSGTY ; MOVE ERROR NUM TO MAILBOX  
 TST \$SWREG ; CHECK FOR HALT ON ERROR  
 BMI IS ; CONTINUE IF SET  
 HALT ; <NO B INTERRUPT>  
 IS:  
 BIT #BIT10, \$SWREG ; CHECK FOR LOOP ON TEST  
 BNE TST25 ; GO TO LOOP ON TEST  
 : TEST FOR INTERRUPT FROM DEVICE  
 †ST26:  
 MOV #26, STESTN ; MOVE TEST NUMBER TO MAILBOX  
 MOV @ORLVLA, R2 ; SAVE INTERRUPT PSH  
 LP26: MOV PL, @ORLVLA ; LOCK OUT SUCCESSIVE INTERRUPTS  
 MOV #STKPTR, SP ; INITIALIZE STACK POINTER  
 MOV #18, @ORVECA ; INTERRUPT RETURN POINTER  
 MOV #101, @RS ; SET INTERRUPT ENABLE-AND CSRI  
 MTPS #0 ; ALLOW INTERRUPTS  
 NOP  
 SS:  
 BIT #BIT14, \$SWREG ; CHECK FOR LOOP ON ERROR  
 BNE TST26 ; GO TO LOOP ERROR  
 MOV #33, \$FATAL  
 MOV #1, \$MSGTY ; MOVE ERROR NUM TO MAILBOX  
 TST \$SWREG ; CHECK FOR HALT ON ERROR  
 BMI IS ; CONTINUE IF SET  
 HALT ; <NO DEVICE INTERRUPT>  
 IS:  
 BIT #BIT10, \$SWREG ; CHECK FOR LOOP ON TEST  
 BNE LP26 ; GO TO LOOP ON TEST  
 CLR @RS ; CLEAR INTERRUPT ENABLE  
 MOV R2, @ORLVLA ; RESTORE INTERRUPT PSH  
 : PLU WRAP TEST  
 †ST27:  
 MOV #27, STESTN ; MOVE TEST NUMBER TO MAILBOX  
 CLR R0 ; SET UP STARTING DATA  
 WLOOP: MOV R0, @ORIBUF ; SEND DATA  
 CMP @ORIBUF, R0 ; CHECK THE DATA  
 BNE SS ; ERROR IF NOT RIGHT  
 INC R0 ; 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		35:	CMP	#31460, R0		: ; CHECK FOR TEST MODULE CODE
1084	004374	001411				BEG	49		
1085	004376	022700	031461			CMP	#31461, R0		
1086	004402	001406				BEG	45		
1087	004404	022700	031462			CMP	#31462, R0		
1088	004410	001403				BEG	45		
1089	004412	022700	031463			CMP	#31463, R0		
1090	004416	001355				BNE	ML00P		
1091	004420	022700			45:	INC	R0		
1092	004422	000762				BR	35		: ; RECHECK DATA CODE
1093	004424				55:				
1094	004424	032767	040000	173770		BIT	#BIT14, \$SWREG		: ; CHECK FOR LOOP ON ERROR
1095	004432	001347				BNE	ML00P		: ; GO TO LOOP ERROR
1096	004434	012767	000034	173740		MOV	#34, \$FATAL		
1097	004442	012767	000001	173730		MOV	#1, \$MSGTY		: ; MOVE ERROR NUM TO MAILBOX
1098	004450	005767	173746			TST	\$SWREG		: ; CHECK FOR HALT ON ERROR
1099	004454	100401				BMI	15		: ; CONTINUE IF SET
1100	004456	000300				HALT			: ; (<WRAP DATA DID NOT COMPARE>)
1101	004460				15:				
1102	004460	032767	0C2000	173734		BIT	#BIT10, \$SWREG		: ; CHECK FOR LOOP ON TEST
1103	004466	001325				BNE	TST27		: ; GO TO LOOP ON TEST

1104									
1105									
1106	004470					TST999:			
1107	004470	005237	000406			INC	@#PASS	:	INCREMENT PASS COUNT
1108	004474	132767	000040	173717		BITB	#40,SEVM	:	WILL APT ALLOW PRINTING?
1109	004502	001010				BNE	ACT	:	NO
1110	004504	012700	004554			MOV	#MSG,RO	:	GET MESSAGE ADDRESS
1111	004510	105777	000056		WAIT:	TSTB	@TPS	:	CHECK IF TTY READY
1112	004514	100375				BPL	WAIT	:	IF NOT
1113	004516	112077	000026			MOVB	(RO)+,@TPB	:	PRINT THE CHARACTER
1114	004522	001372				BNE	WAIT	:	NEXT IF NOT DONE
1115	004524	013700	000042		ACT:	MOV	@#42,RO	:	CHECK ACT
1116	004530	001405				BEQ	GOAGIN	:	KEEP GOING
1117	004532	000005				RESET		:	
1118	004534	004710			SENDAD:	JSR	PC,(RO)	:	ACT HOOKS
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	:	

# M02

.MAIN. MACY11 27(732) 04-OCT-76 14:37 PAGE 26  
 DV:AFB.P11 ERROR 34 WRAP DATA DID NOT COMPARE

```

1129
1130           ;ENTER HERE FOR POWER FAIL
1131
1132 004574 010046 PFAIL: MOV %0,-(6) ;SAVE REGISTER OR STACK
1133 004576 010146 MOV %1,-(6) ;WHEN POWERING DOWN
1134 004600 010246 MOV %2,-(6)
1135 004602 010346 MOV %3,-(6)
1136 004604 010446 MOV %4,-(6)
1137 004606 010546 MOV %5,-(6)
1138 004610 016746 173210 MOV 24,-(6)
1139 004614 010637 004630 MOV %6,@SAVR6 ;STORE STACK POSITION
1140 004620 012737 004632 000024 #RESTAR,@24
1141 004626 000000 HALT ;HALT ON POWER DOWN NORMAL
1142 004630 000000 SAVR6: 0 ;STACK IS SAVED HERE
1143 004632 016706 177772 RESTAR:MOV SAVR6,%6 ;RESTORE REGISTER OFF STACK
1144 004636 012667 173162 MOV (6)+,%24 ;WHEN POWERING UP
1145 004642 012605 MOV (6)+,%5
1146 004644 012604 MOV (6)+,%4
1147 004646 012603 MOV (6)+,%3
1148 004650 012602 MOV (6)+,%2
1149 004652 012601 MOV (6)+,%1
1150 004654 012600 MOV (6)+,%0
1151 004656 000137 001256 JMP @START
1152 000001 .END
  
```



ABASE	=	000000	482	
ACDW1	=	000000	482	
ACDW2	=	000000	482	
ACPUOP	=	000000	482	497
ACT	=	004524	1103	1115#
ADWD	=	000000	482	
ADW1	=	000000	482	
ADW10	=	000000	482	
ADW11	=	000000	482	
ADW12	=	000000	482	
ADW13	=	000000	482	
ADW14	=	000000	482	
ADW15	=	000000	482	
ADW2	=	000000	482	
ADW3	=	000000	482	
ADW4	=	000000	482	
ADW5	=	000000	482	
ADW6	=	000000	482	
ADW7	=	000000	482	
ADW8	=	000000	482	
ADW9	=	000000	482	
ADEVCT	=	000000	482	488
ADEVH	=	000000	482	
RENV	=	000000	482	483
RENVH	=	000000	482	494
RFATAL	=	000000	482	485
AMADR1	=	000000	482	
AMADR2	=	000000	482	
AMADR3	=	000000	482	
AMADR4	=	000000	482	
AMAMS1	=	000000	482	
AMAMS2	=	000000	482	
AMAMS3	=	000000	482	
AMAMS4	=	000000	482	
AMSGAD	=	000000	482	490
AMSGLC	=	000000	482	491
AMSGTY	=	000000	482	484
AMTYP1	=	000000	482	
AMTYP2	=	000000	482	
AMTYP3	=	000000	482	
AMTYP4	=	000000	482	
APASS	=	000700	482	487
APRIOR	=	000000	482	
ASWREG	=	000000	482	495
ATESTN	=	000000	482	486
AUNIT	=	000000	482	489
AUSUR	=	000000	482	196
AVECTJ	=	000000	482	
AVECT2	=	000000	482	
BIT0	=	000001	392#	626
BIT10	=	000001	382#	392
BIT01	=	000002	381#	351
BIT02	=	000004	380#	390
BIT03	=	000010	379#	389
BIT04	=	000020	378#	388
BIT05	=	000040	377#	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	399#																
BIT4 =	000020	368#																
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												
DOISP =	177570	317#																
DEVcnt =	007410	529#																
DRBHIO	001226	549#	780#	799#	800#													
DRCSR	001220	546#	564	578														
DP1BL	001224	548#	625	649	686	1079												
DRLVLA	001232	552#	1054	1055#	1073#													
DRLVLB	001236	554#																
DROBLF	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#															
DSWR =	177570	313#																
EMTVEC =	000030	402#																
ERRMUP =	000402	530#																
ERRVEC =	000004	395#																
EPR1	001412	580	594#															
GORGIN	004544	1116	1122#															
HLT =	104000	275#																
HT =	000011	305#																
IOTVEC =	000020	400#																
LF =	000012	306#																
LP1	001220	578#	576	596														
LP21	002760	953	955#															
LP26	001236	1055#	1071															
LP7	002720	739#	745	748														
MSG	004554	1110	1125#															
N =	000035	262#	587	588#	597	599#	630	631#	640	641#	654	655#	670	671#				



SW09	001000	345	355		
SW1	000002	363			
SW10	002000	291	344		
SW11	004000	283	343		
SW12	010000	342			
SW13	020000	293	341		
SW14	040000	294	340		
SW15	100000	339			
SW16	000004	362			
SW17	000010	361			
SW18	000020	360			
SW19	000040	359			
SW20	000100	358			
SW21	000200	357			
SW22	000400	356			
SW23	001000	290	355		
TR6	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	793	
TST12	002576	794	809	817	
TST13	002722	818	821	828	848
TST14	003056	850	858	866	
TST15	003146	878	877		
TST16	003226	885	894		
TST17	003306	902	909	928	
TST2	001476	605	614	628	661
TST20	003444	920	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	677
TST4	002020	690	689	697	
TST5	002120	687	699	706	714
TST6	002212	716	723	731	
TST7	002304	734	756		
TST999	004470	954	1106		
T12CON	000000	799			
T12OAT	000000	796	797	819	
T13CON	002776	826	835	840	
T17CON	003306	907	916	920	
T2CN1	001644	637	648	653	
T2CON	001602	627	636	639	
WBIT	004510	1111	1112	1114	
WLOOP	004352	1078	1090	1095	







.SPOHE	10	
.SRAND	10	
.SRODE	10	
.SRODC	10	
.SREAO	10	
.SR227	10	
.SSAVE	10	
.SSR20	10	
.SSB20	10	
.SSCOP	10	
.SSIZE	10	
.SSUPR	10	
.STRAP	10	2960
.STYPB	10	
.STYPD	10	
.STYPE	10	2960
.STYPO	10	
.S40CA	10	
.1170	10	

BEO	594	617	651	667	687	704	721	744	765	783	805	826	856	872	889
BIC	907	936	953	990	1082	1084	1086	1088	1116						
EIS	916														
BIT	870	887	905	933											
	535	535	603	616	626	628	636	638	652	660	668	676	682	688	696
	703	713	722	730	747	755	766	774	784	792	808	816	825	827	837
	833	847	855	857	865	871	876	888	893	908	917	919	927	935	937
	943	952	959	967	975	983	991	999	1016	1024	1040	1048	1062	1070	1094
BITB	1102														
BTI	1103														
BTI	533	600	633	643	657	673	693	710	727	752	771	789	813	832	844
BNE	530	881	898	913	924	942	958	964	974	980	996	1021	1045	1067	1099
	586	536	604	627	629	637	639	653	661	669	677	683	689	697	706
	714	723	731	741	748	756	767	775	785	793	803	809	817	828	838
	870	878	878	866	877	894	909	918	920	928	933	946	960	963	976
	984	992	1000	1017	1025	1041	1049	1063	1071	1080	1090	1095	1103	1109	1114
BR	1112														
BR	592	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	
CLR	780	780	799												
COMP	666	703	720	740	764	782	802	906	989	1079	1083	1085	1087	1089	
ENT	301														
HLT	410	412	414	416	418	420	422	424	426	428	430	432	434	436	441
	443	445	447	449	451	453	455	457	459	461	463	465	467	475	477
	591	601	634	644	658	674	694	711	728	753	772	790	814	833	845
	863	882	899	914	925	943	965	981	997	1022	1046	1068	1100	1141	
	742	801	1081	1091	1107										
IN	870														
IN	301														
ICT	472	561	954	1122	1151										
JMP	1118														
JSR	543	564	566	567	568	569	570	571	572	573	577	578	580	581	
MOV	587	588	537	548	606	607	615	618	622	625	630	631	640	641	
	649	654	655	663	664	665	670	671	681	684	690	691	700	701	702
	707	708	717	718	719	724	725	725	736	739	749	750	760	761	763
	768	769	778	779	781	785	787	795	797	810	811	822	824	829	830
	835	835	841	842	851	852	854	859	860	869	878	879	886	895	896
	903	910	911	921	922	931	939	940	951	961	972	971	972	977	978
	986	987	988	993	994	1004	1006	1007	1008	1018	1019	1029	1030	1032	1033
	1042	1043	1053	1054	1055	1066	1057	1058	1064	1065	1073	1076	1078	1096	1097
	1110	1115	1132	1133	1134	1135	1136	1137	1138	1139	1140	1143	1144	1145	1146
	1147	1148	1149	1150											
MOV8	1113														
MPS	579	904	932	1005	1009	1031	1034	1059							
NOP	623	624	647	648	1010	1035	1060	1119	1120	1121					
RESET	522	619	646	685	853	934	1117								
RTI	439														
TST	589	599	632	642	650	656	672	686	692	709	726	737	751	770	788
	812	831	843	861	880	897	912	923	941	963	973	979	995	1020	1044
	1066	1098													
TSTB	743	804	957	1111											
.ASCIZ	1125														
.BYTE	493	494													
.EMOVL	1	256	407												
.EXD	1152														
.EMOC	301	393	407	482	506	509	511	518	587	591	593	597	601	605	630

	634	635	640	644	646	654	658	662	670	674	678	690	694	698	707
	711	715	724	728	732	749	753	757	768	772	776	786	790	794	810
	814	818	829	833	835	841	845	849	859	863	867	878	882	884	835
	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	612	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	891	834
	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														
	1	258	407	482	578	587	588	597	598	616	630	631	640	641	654
	655	664	670	671	682	690	691	701	707	708	718	724	725	736	749
	750	761	768	769	779	786	787	796	810	811	823	829	830	841	842
	832	859	860	870	878	879	887	895	896	904	910	911	921	922	932
	939	940	952	961	962	972	977	978	987	993	994	1005	1018	1019	1030
.MACRO	1042	1043	1054	1064	1065	1077	1096	1097							
.MCALL	1	264	265												
.MEXIT	298	407													
.MLIST	505														
	1	257	407	482	578	587	588	597	598	616	630	631	640	641	654
	655	664	670	671	682	690	691	701	707	708	718	724	725	736	749
	750	761	768	769	779	786	787	796	810	811	823	829	830	841	842
	852	859	860	870	878	879	887	895	896	904	910	911	921	922	932
	939	940	952	961	962	972	977	978	987	993	994	1005	1018	1019	1030
.PAGE	1042	1043	1054	1064	1065	1077	1096	1097							
.REM	265	273	885	970	1104										
.REPT	1														
.SBTTL	409	440	506	587	597	630	640	654	670	690	707	724	749	768	786
	297	479	841	859	878	835	910	921	939	961	977	993	1018	1042	1064
	810	829													
.WORD	1096														
	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/ML: TOC=SYSMAC.CO, DVKAFB.P11  
 RUN-TIME: 24 27 2 SECONDS  
 RUN-TIME RATIO: 202/55=3.6  
 CORE USED: 33K (65 PAGES)

