

DRV11

DRV11 TEST
CVKAFCO

AH-8207C-MC

MAR 1978

COPYRIGHT © 75-78

digital

FICHE 1 OF 1

MADE IN USA

This image shows a vertical strip of microfilm on the left side of the page. The strip contains approximately 15 frames of data, each appearing as a small grid or table of characters. The data is too small and faint to be transcribed accurately, but it appears to be organized in a structured format, possibly a test log or data table. The rest of the page is a dark, mostly blank area with some faint, illegible markings.

B01

EOF1CQK PDBS8Q411 00010000 I080228 I CATION PDP10 411 64HDR1CVKAF CSEQ 00010000 780223

SEQ 0001

PRODUCT CODE: AC-8206C-MC
PRODUCT NAME: CVKAFCD DRV11 TEST
PRODUCT DATE: FEB 1978
MAINTAINER: DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

1. ABSTRACT

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

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

2. REQUIREMENTS

2.1 EQUIPMENT

LSI-11

DRV11

TEST CABLE (BCOBR) (BY OPTION)

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

2.2 STORAGE

2.2.1 PROGRAM STORAGE - 4K

3. LOADING PROCEDURE

3.1 METHOD

ABSOLUTE LOADER

4. STARTING PROCEDURE

200 - NORMAL ENTRY TO TEST ONE DEVICE

TO LOAD AND EXECUTE

1. LOAD PROGRAM WITH THE ABSOLUTE LOADER.
2. IF ANY PROGRAM OPTIONS ARE REQUIRED, SET THE APPROPRIATE BIT IN THE SOFTWARE SWITCH REGISTER AT LOCATION 422. (REF. SECTION 5.1)
3. START PROGRAM AT 200.
4. PROGRAM WILL PRINT "END OF PASS" FOLLOWING EACH PASS.

4.1 CONTROL SWITCH SETTING

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 ODT 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. ODT 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).
NOTE: THE SPECIAL MODULE IS FOR USE BY IN HOUSE MANUFACTURING ONLY.
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 WHEN RUN WITH THE WRAP CABLE, AND APPROXIMATELY

60-70% EFFECTIVE WHEN RUN WITHOUT IT.

SEQ 0004

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 REFERENCE 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 \$FATAL IN THE APT MAILBOX CONTAINS THE ERROR NUMBER AND MAY BE USED TO REFERENCE THE DESCRIPTION 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

<u>TAG</u>	<u>COMMENT</u>
\$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

22
23
24
25
26
27
28
29
30
31
32
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

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
000006
000007

000000

```

;GENERAL REGISTER LOGIC TEST
HLT=104000
CSR=167770
STKPTR=1200
;REGISTER DEFINITIONS
R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7

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

.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
CRLF= 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
DDISP= 177570 ;;HARDWARE DISPLAY REGISTER

;*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0 ;;GENERAL REGISTER
R1= %1 ;;GENERAL REGISTER
R2= %2 ;;GENERAL REGISTER
R3= %3 ;;GENERAL REGISTER
R4= %4 ;;GENERAL REGISTER
R5= %5 ;;GENERAL REGISTER
R6= %6 ;;GENERAL REGISTER
R7= %7 ;;GENERAL REGISTER
SP= %6 ;;STACK POINTER
PC= %7 ;;PROGRAM COUNTER

;*PRIORITY LEVEL DEFINITIONS
PRO= 0 ;;PRIORITY LEVEL 0
  
```

78 000040
79 000100
80 000140
81 000200
82 000240
83 000300
84 000340
85

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

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

86
87 100000
88 040000
89 020000
90 010000
91 004000
92 002000
93 001000
94 000400
95 000200
96 000100
97 000040
98 000020
99 000010
100 000004
101 000002
102 000001
103
104
105
106
107
108
109
110
111
112
113

::*"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,SW9
.EQUIV SW08,SW8
.EQUIV SW07,SW7
.EQUIV SW06,SW6
.EQUIV SW05,SW5
.EQUIV SW04,SW4
.EQUIV SW03,SW3
.EQUIV SW02,SW2
.EQUIV SW01,SW1
.EQUIV SW00,SW0

114
115 100000
116 040000
117 020000
118 010000
119 004000
120 002000
121 001000
122 000400
123 000200
124 000100
125 000040
126 000020
127 000010
128 000004
129 000002
130 000001
131
132
133

::*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
.EQUIV BIT07,BIT7

```

134 .EQUIV BIT06,BIT6
135 .EQUIV BIT05,BIT5
136 .EQUIV BIT04,BIT4
137 .EQUIV BIT03,BIT3
138 .EQUIV BIT02,BIT2
139 .EQUIV BIT01,BIT1
140 .EQUIV BIT00,BIT0
  
```

```

141
142 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
143 000004 ERRVEC= 4 ; TIME OUT AND OTHER ERRORS
144 000010 RESVEC= 10 ; RESERVED AND ILLEGAL INSTRUCTIONS
145 000014 TBITVEC=14 ; "T" BIT
146 000014 TRTVEC= 14 ; TRACE TRAP
147 000014 BPTVEC= 14 ; BREAKPOINT TRAP (BPT)
148 000020 IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) **SCOPE**
149 000024 PWRVEC= 24 ; POWER FAIL
150 000030 EMTVEC= 30 ; EMULATOR TRAP (EMT) **ERROR**
151 000034 TRAPVEC=34 ; "TRAP" TRAP
152 000060 TKVEC= 60 ; TTY KEYBOARD VECTOR
153 000064 TPVEC= 64 ; TTY PRINTER VECTOR
154 000240 PIRQVEC=240 ; PROGRAM INTERRUPT REQUEST VECTOR
  
```

```

155 .ENABLE ABS
156 .=0
157 000000 000002 .+2
158 000002 000000 HALT
159 000004 000006 .+2
160 000006 000000 HALT
161 000010 000012 .+2
162 000012 000000 HALT
163 000014 000016 .+2
164 000016 000000 HALT
165 000020 000022 .+2
166 000022 000000 HALT
167 000024 000026 .+2
168 000026 000000 HALT
169 000030 000032 .+2
170 000032 000000 HALT
171 000034 000036 .+2
172 000036 000000 HALT
173 000040 000042 .+2
174 000042 000000 HALT
175 000044 000046 .+2
176 000046 000000 HALT
177 000050 000052 .+2
178 000052 000000 HALT
179 000054 000056 .+2
180 000056 000000 HALT
181 000060 000062 .+2
182 000062 000000 HALT
183 000064 000066 .+2
184 000066 000000 HALT
185 000100 000100 .=100
186 000100 000102 102
187 000102 000002 RTI
188 000104 000106 .+2
189 000106 000000 HALT
  
```

; RTI FOR POSSIBLE CLOCK INTERRUPT

```

190 000110 000112 .+2
191 000112 000000 HALT
192 000114 000116 .+2
193 000116 000000 HALT
194 000120 000122 .+2
195 000122 000000 HALT
196 000124 000126 .+2
197 000126 000000 HALT
198 000130 000132 .+2
199 000132 000000 HALT
200 000134 000136 .+2
201 000136 000000 HALT
202 000140 000142 .+2
203 000142 000000 HALT
204 000144 000146 .+2
205 000146 000000 HALT
206 000150 000152 .+2
207 000152 000000 HALT
208 000154 000156 .+2
209 000156 000000 HALT
210 000160 000162 .+2
211 000162 000000 HALT
212 000164 000166 .+2
213 000166 000000 HALT
214 000170 000172 .+2
215 000172 000000 HALT
216 000200 000200 .+200
217 000200 005067 000202 CLR $PASS ; CLEAR PASS COUNT
218 000204 005067 000172 CLR $FATAL
219 000210 005067 000170 CLR $TESTN
220 000214 000137 001246 JMP $START1 ; INITIAL START
221 000300 000300 .+300 ; DEVICE INTERRUPT VECTORS
222 000300 000302 .+2
223 000302 000000 HALT
224 000304 000306 .+2
225 000306 000000 HALT
226 000400 000400 .+400
227 .SBTTL APT MAILBOX-ETABLE
228
229 ; *****
230 .EVEN
231 000400 $MAIL: ; APT MAILBOX
232 000400 000000 $MSGTY: .WORD AMSGTY ; MESSAGE TYPE CODE
233 000402 000000 $FATAL: .WORD AFATAL ; FATAL ERROR NUMBER
234 000404 000000 $TESTN: .WORD ATESTN ; TEST NUMBER
235 000406 000000 $PASS: .WORD APASS ; PASS COUNT
236 000410 000000 $DEVCT: .WORD ADEVCT ; DEVICE COUNT
237 000412 000000 $UNIT: .WORD AUNIT ; I/O UNIT NUMBER
238 000414 000000 $MSGAD: .WORD AMSGAD ; MESSAGE ADDRESS
239 000416 000000 $MSGLG: .WORD AMGLG ; MESSAGE LENGTH
240 000420 $ETABLE: ; APT ENVIRONMENT TABLE
241 000420 000 $ENV: .BYTE AENV ; ENVIRONMENT BYTE
242 000421 000 $ENVM: .BYTE AENVM ; ENVIRONMENT MODE BITS
243 000422 000000 $SWREG: .WORD ASWREG ; APT SWITCH REGISTER
244 000424 000000 $USWR: .WORD AUSWR ; USER SWITCHES
245 000426 000000 $CPUOP: .WORD ACPUOP ; CPU TYPE, OPTIONS

```

246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
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

000430

000024

000044

000430

000430 000000

000432 000400

000434 000010

000436 000010

000440 000000

000442 000014

001200

000410

000402

000426

001200 167770

001202 167772

001204 167774

001206 167773

001210 000300

001212 000302

001214 000304

001216 000306

001220 167770

001222 167772

001224 167774

001226 167773

001230 000300

001232 000302

001234 000304

BITS 15-11=CPU TYPE
11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
11/70=06,PDQ=07,Q=10
BIT 10=REAL TIME CLOCK
BIT 9=FLOATING POINT PROCESSOR
BIT 8=MEMORY MANAGEMENT

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

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

\$APTHD:
\$HIBTS: .WORD 0 ; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
\$MBADR: .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)
\$UNITM: .WORD 0 ; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
=1200 : SETEND-\$MAIL/2 ; LENGTH MAILBOX-ETABLE(WORDS)
DEVCNT=\$DEVCT
ERRNUM=\$FATAL
OPTION=\$CPUOP

;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
DROBUF: 167772 ; ADDRESS OF DR OUTPUT BUFFER REG.
DRIBUF: 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
DRVECB: 304 ; INTERRUPT VECTOR

NO1

CVKAFC MACY11 30A(1052) 22-DEC-77 11:54 PAGE 8
 CVKAFC.P11 22-DEC-77 11:52 APT PARAMETER BLOCK

SEQ 0013

```

302 001236 000306 DRLVLB: 306
303 001240 000000 XORFLG: 0
304
305 001242 000000 COUNT: 0 ;COUNT LOCATION
306 001244 000240 PL: 240 ;PRIORITY LEVEL
307
308 001246 012706 001200 START1: MOV #STKPTR,SP
309 001252 000137 001256 JMP @START ;INITIALIZE ADDRESS AND VECTORS
310 ;INITIALIZE ADDRESS AND VECTORS
311 001256 012700 001200 START: MOV #RCSR,R0 ;GET ADDRESS OF UNIT UNDER TEST
312 001262 012701 001220 MOV #DRCSR,R1
313
314 001266 012737 004574 000024 MOV #PFAIL,@#24
315 001274 012021 MOV (R0)+,(R1)+ ;LOAD INITIAL TEST ADDRESSES
316 001276 012021 MOV (R0)+,(R1)+
317 001300 012021 MOV (R0)+,(R1)+
318 001302 012021 MOV (R0)+,(R1)+
319 001304 012021 MOV (R0)+,(R1)+
320 001306 012021 MOV (R0)+,(R1)+
321 001310 012021 MOV (R0)+,(R1)+
322
323 ;DOES RESET CLEAR REGISTER?
324 001312 TST1:
325 001312 012767 000001 177064 LP1: MOV #1,$TESTN ; MOVE TEST NUMBER TO MAILBOX
326 001320 016705 177674 MOV DRCSR,R5 ; GET ADDRESS OF STATUS REGISTER
327 001324 106427 000200 MTPS #200 ; TURN OFF INTERRUPTS
328 001330 016737 000056 000004 MOV ERR1,@#4 ; SET TIME OUT TRAP VECTOR
329 001336 012777 177777 177656 MOV #-1,@DROBUF ; PRESET OUTPUT BUFFER
330 001344 000005 RESET ; CLEAR DATA REGISTER
331 001346 017700 177650 MOV @DROBUF,R0 ; GET RESULT OF RESET
332 001352 001450 BEQ CON
333 001354 032767 040000 177040 BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
334 001362 001356 BNE LP1 ; GO TO LOOP ERROR
335 001364 012767 000001 177010 MOV #1,$FATAL
336 001372 012767 000001 177000 MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
337 001400 005767 177016 TST $SWREG ; CHECK FOR HALT ON ERROR
338 001404 100401 BMI $ ; CONTINUE IF SET
339 001406 000000 HALT ; <DATA REG DID NOT CLEAR>
340 001410 1$:
341 001410 000431 BR CON
342 001412 ERR1:
343 001412 032767 040000 177002 BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
344 001420 001337 BNE LP1 ; GO TO LOOP ERROR
345 001422 012767 000002 176752 MOV #2,$FATAL
346 001430 012767 000001 176742 MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
347 001436 005767 176760 TST $SWREG ; CHECK FOR HALT ON ERROR
348 001442 100401 BMI $ ; CONTINUE IF SET
349 001444 000000 HALT ; <TIME WHEN ADDRESSING PLU>
350 001446 1$:
351 001446 032767 002000 176746 BIT #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
352 001454 001316 BNE TST1 ; GO TO LOOP ON TEST
353 001456 000407 BR TST2
354 001460 012706 001200 5$: MOV #STKPTR,SP ; RESET STACK POINTER
355 001464 012737 000006 000004 MOV #6,@#4 ; RESTORE TIME OUT TRAP
356 001472 000765 BR $
357 001474 000772 CON: BR .-12

```

B02

CVKAF C MACY11 30A(1052) 22-DEC-77 11:54 PAGE 9
 CVKAF C.P11 22-DEC-77 11:52 ERROR 2

TIME WHEN ADDRESSING PLU

SEQ 0014

```

; TEST "NEWDATA RDY" AND "DATATRANS" SIGNALS IN PLU
; NOTE***** THE PLU TEST MODULE MUST BE INSTALLED
; TO EXECUTE THIS TEST
358
359
360
361
362
363 001476 012767 000002 176700
364 001504 032767 001000 176710
365 001512 001505
366 001514 012706 001200
367 001520 000005
368
369
370 001522 012777 031460 177472
371 001530 000240
372 001532 000240
373 001534 017700 177464
374 001540 032700 000001
375 001544 001016
376 001546 032767 040000 176646
377 001554 001350
378 001556 012767 000003 176616
379 001564 012767 000001 176606
380 001572 005767 176624
381 001576 100401
382 001600 000000
383 001602
384 001602 032700 000002
385 001606 001016
386 001610 032767 040000 176604
387 001616 001371
388 001620 012767 000004 176554
389 001626 012767 000001 176544
390 001634 005767 176562
391 001640 100401
392 001642 000000
393 001644
394 001644 000005
395 001646 000240
396 001650 000240
397 001652 017700 177346
398 001656 005700
399 001660 001416
400 001662 032767 040000 176532
401 001670 001365
402 001672 012767 000005 176502
403 001700 012767 000001 176472
404 001706 005767 176510
405 001712 100401
406 001714 000000
407 001716
408 001716 032767 002000 176476
409 001724 001264
410 001726
411 001726 012767 000003 176450
412 001734 012777 177777 177260
413 001742 017700 177254

; 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
; PRIME THE LATCHES
MOV #31460,$DROBUF
NOP
NOP
MOV $DROBUF,R0 ; GET DATA
BIT #BIT0,R0 ; CHECK DATA TRANS SIG
BNE T2CON ; CONTINUE IF PRESENT
BIT #BIT14,$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 IS ; CONTINUE IF SET
HALT ; <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 IS ; CONTINUE IF SET
HALT ; <NO NEW DATA RDY SIGNAL>

IS:
T2CN1: RESET ; CLEAR EVERYTHING
NOP
NOP
MOV $DROBUF,R0 ; CHECK SIGNAL LATCHES
TST R0 ; SHOULD BE CLEAR
BEQ IS ; CONTINUE IF CLEAR
BIT #BIT14,$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 IS ; CONTINUE IF SET
HALT ; <SIGNALS DID NOT CLEAR>

IS:
BIT #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
BNE TST3 ; GO TO LOOP ON TEST

TST3: MOV #3,$TESTN ; MOVE TEST NUMBER TO MAILBOX
MOV #-1,$DROBUF ; ALL ONES TO REGISTER
MOV $DROBUF,R0

```

414	001746	022700	177777		CMP	#-1,RO	
415	001752	001416			BEQ	IS	
416	001754	032767	040000	176440	BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
417	001762	001361			BNE	TST3	; GO TO LOOP ERROR
418	001764	012767	000006	176410	MOV	#6,\$FATAL	
419	001772	012767	000001	176400	MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
420	002000	005767	176416		TST	\$SWREG	; CHECK FOR HALT ON ERROR
421	002004	100401			BMI	IS	; CONTINUE IF SET
422	002006	000000			HALT		; <REGISTER WILL NOT HOLD ALL ONES>
423	002010						
424	002010	032767	002000	176404	1S:	BIT	#BIT10,\$SWREG ; CHECK FOR LOOP ON TEST
425	002016	001343			BNE	TST3	; GO TO LOOP ON TEST
426							
427							
428	002020						
429	002020	012767	000004	176356	↑ST4:	MOV	#4,\$TESTN ; MOVE TEST NUMBER TO MAILBOX
430	002026	032767	000400	176366	BIT	#BIT8,\$SWREG	
431	002034	001031			BNE	TST5	; SKIP TEST IF NOT SELECTED
432	002036	012777	177777	177156	MOV	#-1,\$DROBUF	
433	002044	000005			RESET		; SET DATA TO ALL ONES
434	002046	005777	177152		TST	\$DROBUF	; REGISTER SHOULD CLEAR
435	002052	001416			BEQ	IS	
436	002054	032767	040000	176340	BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
437	002062	001356			BNE	TST4	; GO TO LOOP ERROR
438	002064	012767	000007	176310	MOV	#7,\$FATAL	
439	002072	012767	000001	176300	MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
440	002100	005767	176316		TST	\$SWREG	; CHECK FOR HALT ON ERROR
441	002104	100401			BMI	IS	; CONTINUE IF SET
442	002106	000000			HALT		; <REGISTER DID NOT CLEAR BY RESET>
443	002110						
444	002110	032767	002000	176304	1S:	BIT	#BIT10,\$SWREG ; CHECK FOR LOOP ON TEST
445	002116	001340			BNE	TST4	; GO TO LOOP ON TEST
446							
447	002120						
448	002120	012767	000005	176256	TST5:	MOV	#5,\$TESTN ; MOVE TEST NUMBER TO MAILBOX
449	002126	012777	052525	177066	MOV	#52525,\$DROBUF	; LOAD TEST DATA INTO BUFFER
450	002134	017700	177062		MOV	\$DROBUF,RO	; COPY DATA FROM BUFFER TO RO
451	002140	022700	052525		CMP	#52525,RO	; COMPARE DATA
452	002144	001416			BEQ	IS	; BR IF DATA IS CORRECT
453	002146	032767	040000	176246	BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
454	002154	001361			BNE	TST5	; GO TO LOOP ERROR
455	002156	012767	000010	176216	MOV	#10,\$FATAL	
456	002164	012767	000001	176206	MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
457	002172	005767	176224		TST	\$SWREG	; CHECK FOR HALT ON ERROR
458	002176	100401			BMI	IS	; CONTINUE IF SET
459	002200	000000			HALT		; <INCORRECT DATA IN REG>
460	002202						
461	002202	032767	002000	176212	1S:	BIT	#BIT10,\$SWREG ; CHECK FOR LOOP ON TEST
462	002210	001343			BNE	TST5	; GO TO LOOP ON TEST
463							
464	002212						
465	002212	012767	000006	176164	TST6:	MOV	#6,\$TESTN ; MOVE TEST NUMBER TO MAILBOX
466	002220	012777	125252	176774	MOV	#125252,\$DROBUF	; LOAD TEST DATA INTO BUFFER
467	002226	017700	176770		MOV	\$DROBUF,RO	; COPY DATA FROM BUFFER TO RO
468	002232	022700	125252		CMP	#125252,RO	; COMPARE DATA
469	002236	001416			BEQ	IS	; BR IF DATA IS CORRECT

```

470 002240 032767 040000 176154 BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
471 002246 001361 BNE TST6 ; GO TO LOOP ERROR
472 002250 012767 000011 176124 MOV #11,$FATAL
473 002256 012767 000001 176114 MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
474 002264 005767 176132 TST $SWREG ; CHECK FOR HALT ON ERROR
475 002270 100401 BMI 1$ ; CONTINUE IF SET
476 002272 000000 HALT ; <INCORRECT DATA IN REG>
477 1$:
478 002274 032767 002000 176120 BIT #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
479 002302 001343 BNE TST6 ; GO TO LOOP ON TEST
480
481 ;TEST RELIABILITY OF DR11-C OUTPUT BUFFER REGISTER
482 TST7:
483 002304 012767 000007 176072 BUFTST: MOV #7,$TESTN ; MOVE TEST NUMBER TO MAILBOX
484 002312 010502 MOV R5,R2 ; GET ADDRESS OF DRCSR
485 002314 005722 TST (R2)+ ; R2=ADDRESS OF OUTPUT BUFFER REG.
486 002316 005003 CLR R3 ; INITIALIZE DATA REGISTER
487 002320 010312 LP7: MOV R3,(R2) ; SEND THE DATA
488 002322 021203 CMP (R2),R3 ; CHECK THE RECEIVED DATA
489 002324 001004 BNE 5$ ; ERROR IF NOT THE SAME
490 002326 005203 INC R3 ; INCREMENT THE DATA
491 002330 105703 TSTB R3 ; CHECK FOR END OF DATA
492 002332 001417 BEQ 1$ ; CONTINUE IF END
493 002334 000771 BR LP7 ; GO TO SEND DATA IF NOT
494 5$:
495 002336 032767 040000 176056 BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
496 002344 001365 BNE LP7 ; GO TO LOOP ERROR
497 002346 012767 000012 176026 MOV #12,$FATAL
498 002354 012767 000001 176016 MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
499 002362 005767 176034 TST $SWREG ; CHECK FOR HALT ON ERROR
500 002366 100401 BMI 1$ ; CONTINUE IF SET
501 002370 000000 HALT ; <DATA INCORRECT IN REG>
502 1$:
503 002372 032767 002000 176022 BIT #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
504 002400 001341 BNE TST7 ; GO TO LOOP ON TEST
505 ;TEST THAT BYTE REFERENCE TO DROBUF AFFECT PROPER BYTE ONLY
506
507 TST10:
508 002402 012767 000010 175774 TAG: MOV #10,$TESTN ; MOVE TEST NUMBER TO MAILBOX
509 002410 012777 177777 MOV #-1,DROBUF ; SET ALL ONES IN BUFFER
510 002416 105077 176600 CLRB DROBUF ; CLEAR LOW BYTE
511 002422 017700 176574 MOV DROBUF,R0 ; COPY DATA
512 002426 022700 177400 CMP #177400,R0 ; VERIFY THAT LOW BYTE IS CLEAR
513 002432 001416 BEQ 1$
514 002434 032767 040000 175760 BIT #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
515 002442 001362 BNE TAG ; GO TO LOOP ERROR
516 002444 012767 000013 175730 MOV #13,$FATAL
517 002452 012767 000001 175720 MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
518 002460 005767 175736 TST $SWREG ; CHECK FOR HALT ON ERROR
519 002464 100401 BMI 1$ ; CONTINUE IF SET
520 002466 000000 HALT ; <LOW BYTE FAILED TO CLEAR>
521 1$:
522 002470 032767 002000 175724 BIT #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
523 002476 001341 BNE TST10 ; GO TO LOOP ON TEST
524
525 TST11:
    
```

E02

CVKAFB MACY11 30A(1052) 22-DEC-77 11:54 PAGE 12
 CVKAFB.P11 22-DEC-77 11:52 ERROR 13

SEQ 0017

LOW BYTE FAILED TO CLEAR

526	002500	012767	000011	175676		MOV	#11,\$TESTN	; MOVE TEST NUMBER TO MAILBOX
527	002506	012777	177777	176506		MOV	#-1,@DROBUF	; SET ALL ONES IN BUFFER
528	002514	105077	176506			CLRB	@DRBHIO	; CLEAR HIGH BYTE
529	002520	017700	176476			MOV	@DROBUF,R0	
530	002524	022700	000377			CMP	#377,R0	; VERIFY THAT HIGH BYTE IS CLEAR
531	002530	001416				BEQ	1\$	
532	002532	032767	040000	175662		BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
533	002540	001357				BNE	TST11	; GO TO LOOP ERROR
534	002542	012767	000014	175632		MOV	#14,\$FATAL	
535	002550	012767	000001	175622		MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
536	002556	005767	175640			TST	\$SWREG	; CHECK FOR HALT ON ERROR
537	002562	100401				BMI	1\$; CONTINUE IF SET
538	002564	000000				HALT		; <HIGH BYTE FAILED TO CLEAR>
539	002566				1\$:			
540	002566	032767	002000	175626		BIT	#BIT10,\$SWREG	; CHECK FOR LOOP ON TEST
541	002574	001341				BNE	TST11	; GO TO LOOP ON TEST
542	002576				TST12:			
543	002576	012767	000012	175600		MOV	#12,\$TESTN	; MOVE TEST NUMBER TO MAILBOX

F02

CVKAFC MACY11 30A(1052) 22-DEC-77 11:54 PAGE 13
 CVKAFC.P11 22-DEC-77 11:52 ERROR 14

HIGH BYTE FAILED TO CLEAR

SEQ 0018

544	002604	005067	000110		CLR	T12DAT	; CLEAR DATA LOCATION
545	002610	012704	002720		MOV	#T12DAT,R4	; STORE ADDRESS OF DATA LOCATION
546	002614	005077	176402		CLR	QDRBUF	; CLEAR OUTPUT BUFFER
547	002620	105077	176402		CLR	QDRBHIO	; CLEAR HIGH BYTE
548	002624	105277	176376		INCB	QDRBHIO	; INCREMENT HIGH BYTE
549	002630	105264	000001		INCB	1(R4)	; INCREMENT COMPARISON DATA
550	002634	027714	176362		CMP	QDRBUF,(R4)	; COMPARE DATA
551	002640	001004			BNE	6S	; BRANCH ON ERROR
552	002642	105764	000001		TSTB	1(R4)	; FINISHED?
553	002646	001417			BEQ	1S	; YES
554	002650	000765			BR	3S	; CONTINUE TESTING
555	002652						
556	002652	032767	040000	175542	BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
557	002660	001346			BNE	TST12	; GO TO LOOP ERROR
558	002662	012767	000015	175512	MOV	#15,\$FATAL	
559	002670	012767	000001	175502	MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
560	002676	005767	175520		TST	\$SWREG	; CHECK FOR HALT ON ERROR
561	002702	100401			BMI	1S	; CONTINUE IF SET
562	002704	000000			HALT		; <DATA INCORRECT IN REG>
563	002706						
564	002706	032767	002000	175506	BIT	#BIT10,\$SWREG	; CHECK FOR LOOP ON TEST
565	002714	001330			BNE	TST12	; GO TO LOOP ON TEST
566	002716	000401			BR	TST13	
567	002720	000000					
568					T12DAT: .WORD	0	
569	002722				CONTROL STATUS REGISTER (DRCSR) TESTS.		
570	002722	012767	000013	175454	TST13:		
571	002730	005015			MOV	#13,\$TESTN	; MOVE TEST NUMBER TO MAILBOX
572	002732	011500			CLR	(R5)	; CLEAR STATUS REGISTER
573	002734	032700	000143		MOV	(R5),R0	; COPY DATA
574	002740	001416			BIT	#143,R0	; VERIFY THAT IE AND CSR BITS ARE CLEAR
575	002742	032767	040000	175452	BEQ	T13CON	; IF YES, CONTINUE
576	002750	001364			BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
577	002752	012767	000016	175422	BNE	TST13	; GO TO LOOP ERROR
578	002760	012767	000001	175412	MOV	#16,\$FATAL	
579	002766	005767	175430		MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
580	002772	100401			TST	\$SWREG	; CHECK FOR HALT ON ERROR
581	002774	000000			BMI	1S	; CONTINUE IF SET
582	002776				HALT		; <STATUS REG DID NOT CLEAR>
583	002776	012715	000140		1S:		
584	003002	011500			T13CON:		
585	003004	032700	000140		MOV	#140,QR5	; INTERRUPT ENABLE FOR A+B
586	003010	001016			MOV	QR5,R0	
587	003012	032767	040000	175402	BIT	#140,R0	; INTERRUPT ENABLE BITS SET?
588	003020	001366			BNE	1S	; CONTINUE IF YES
589	003022	012767	000017	175352	BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
590	003030	012767	000001	175342	BNE	T13CON	; GO TO LOOP ERROR
591	003036	005767	175360		MOV	#17,\$FATAL	
592	003042	100401			MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
593	003044	000000			TST	\$SWREG	; CHECK FOR HALT ON ERROR
594	003046				BMI	1S	; CONTINUE IF SET
595	003046	032767	002000	175346	HALT		; <ENABLE BITS NOT ON>
596	003054	001322			1S:		
597							
598	003056				TST14:		
599	003056	012767	000014	175320	BIT	#BIT10,\$SWREG	; CHECK FOR LOOP ON TEST
					BNE	TST13	; GO TO LOOP ON TEST
					MOV	#14,\$TESTN	; MOVE TEST NUMBER TO MAILBOX

G02

CVKAF C MACY11 30A(1052) 22-DEC-77 11:54 PAGE 14
 CVKAF C.P11 22-DEC-77 11:52 ERROR 17

ENABLE BITS NOT ON

SEQ 0019

600	003064	012715	000140		MOV	#140,DR5	; SET INTERRUPT ENABLE FLOPS
601	003070	000005			RESET		; CLEAR THOSE FLOPS
602	003072	011500			MOV	DR5,RO	; COPY CONTENTS OF DRCSR TO RO
603	003074	032700	000140		BIT	#140,RO	; TEST INTERRUPT ENABLE BITS
604	003100	001416			BEQ	1\$; BR IF CLEARED
605	003102	032767	040000	175312	BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
606	003110	001362			BNE	TST14	; GO TO LOOP ERROR
607	003112	012767	000020	175262	MOV	#20,\$FATAL	
608	003120	012767	000001	175252	MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
609	003126	005767	175270		TST	\$SWREG	; CHECK FOR HALT ON ERROR
610	003132	100401			BMI	1\$; CONTINUE IF SET
611	003134	000000			HALT		; < INTERRUPT ENABLE DID NOT CLEAR >
612	003136						
613	003136	032767	002000	175256	BIT	#BIT10,\$SWREG	; CHECK FOR LOOP ON TEST
614	003144	001344			BNE	TST14	; GO TO LOOP ON TEST
615							
616	003146						
617	003146	012767	000015	175230	MOV	#15,\$TESTN	; MOVE TEST NUMBER TO MAILBOX
618	003154	052715	000001		BIS	#1,DR5	; SHOULD SET REQ A ALSO
619	003160	032715	000201		BIT	#201,DR5	; VERIFY THAT REQ A IS SET
620	003164	001402			BEQ	5\$; FLAG ERROR MESSAGE IF NO
621	003166	005015			CLR	DR5	; CLEAR STATUS REGISTER
622	003170	000416			BR	1\$; GO TO NEXT TEST
623	003172						
624	003172	032767	040000	175222	BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR
625	003200	001362			BNE	TST15	; GO TO LOOP ERROR
626	003202	012767	000021	175172	MOV	#21,\$FATAL	
627	003210	012767	000001	175162	MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX
628	003216	005767	175200		TST	\$SWREG	; CHECK FOR HALT ON ERROR
629	003222	100401			BMI	1\$; CONTINUE IF SET
630	003224	000000			HALT		; < A REQ DID NOT SET >
631	003226						
632							

H02

CVKAF C MACY11 30A(1052) 22-DEC-77 11:54 PAGE 15
 CVKAF C.P11 22-DEC-77 11:52 ERROR 21

A REQ DID NOT SET

SEQ 0020

633	003226					TST16:			
634	003226	012767	000016	175150		MOV	#16,\$TESTN	; MOVE TEST NUMBER TO MAILBOX	
635	003234	052715	000002			BIS	#2,\$RS	; SHOULD SET REQ B	
636	003240	032715	100002			BIT	#100002,\$RS	; VERIFY THAT REQ B IS SET	
637	003244	001402				BEQ	\$S	; FLAG ERROR MESSAGE IF NO	
638	003246	005015				CLR	\$PS	; CLEAR STATUS REGISTER	
639	003250	000416				BR	\$S	; GO TO NEXT TEST	
640	003252					\$S:			
641	003252	032767	040000	175142		BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR	
642	003260	001362				BNE	TST16	; GO TO LOOP ERROR	
643	003262	012767	000022	175112		MOV	#22,\$FATAL		
644	003270	012767	000001	175102		MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX	
645	003276	005767	175120			TST	\$SWREG	; CHECK FOR HALT ON ERROR	
646	003302	100401				BMI	\$S	; CONTINUE IF SET	
647	003304	000000				HALT		; <B REQ DID NOT SET>	
648	003306					\$S:			
649						TST17:			
650	003306					MOV	#17,\$TESTN	; MOVE TEST NUMBER TO MAILBOX	
651	003306	012767	000017	175070		MTPS	#340	; LOCK OUT INTERRUPTS	
652	003314	106427	000340			BIS	#-1,\$RS	; LOAD ALL ONES IN STATUS REGISTER	
653	003320	052715	177777			CMP	#100343,(\$RS)	; VERIFY THAT ALL WRITE BITS ARE SET IN DRCSR	
654	003324	022715	100343			BEQ	T17CON	; BR IF SET	
655	003330	001416				BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR	
656	003332	032767	040000	175062		BNE	TST17	; GO TO LOOP ERROR	
657	003340	001362				MOV	#23,\$FATAL		
658	003342	012767	000023	175032		MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX	
659	003350	012767	000001	175022		TST	\$SWREG	; CHECK FOR HALT ON ERROR	
660	003356	005767	175040			BMI	\$S	; CONTINUE IF SET	
661	003362	100401				HALT		; <INCORRECT DATA IN REG>	
662	003364	000000				\$S:			
663	003366					T17CON:			
664	003366	042715	000003			BIC	#3,\$RS	; CLEAR CSR BITS	
665	003372	032715	000140			BIT	#140,\$RS	; TEST INTERRUPT ENABLE BITS	
666	003376	001016				BNE	\$S	; CONTINUE IF STILL SET	
667	003400	032767	040000	175014		BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR	
668	003406	001367				BNE	T17CON	; GO TO LOOP ERROR	
669	003410	012767	000024	174764		MOV	#24,\$FATAL		
670	003416	012767	000001	174754		MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX	
671	003424	005767	174772			TST	\$SWREG	; CHECK FOR HALT ON ERROR	
672	003430	100401				BMI	\$S	; CONTINUE IF SET	
673	003432	000000				HALT		; <WRONG BITS SET>	
674	003434					\$S:			
675	003434	032767	002000	174760		BIT	#BIT10,\$SWREG	; CHECK FOR LOOP ON TEST	
676	003442	001321				BNE	TST17	; GO TO LOOP ON TEST	
677						TST20:			
678	003444					MOV	#20,\$TESTN	; MOVE TEST NUMBER TO MAILBOX	
679	003444	012767	000020	174732		MTPS	#340	; LOCK OUT INTERRUPTS	
680	003452	106427	000340			BIS	#3,\$RS	; SET CSR BITS	
681	003456	052715	000003			RESET		; SHOULD CLEAR INTERRUPT ENABLE FLOPS	
682	003462	000005				BIT	#140,\$RS	; VERIFY THAT FLOPS ARE CLEARED	
683	003464	032715	000140			BEQ	\$S	; BR IF YES	
684	003470	001416				BIT	#BIT14,\$SWREG	; CHECK FOR LOOP ON ERROR	
685	003472	032767	040000	174722		BNE	TST20	; GO TO LOOP ERROR	
686	003500	001361				MOV	#25,\$FATAL		
687	003502	012767	000025	174672		MOV	#1,\$MSGTY	; MOVE ERROR NUM TO MAILBOX	
688	003510	012767	000001	174662		MOV			

RESET DID NOT CLEAR BITS

SEQ 0021

```

689 003516 005767 174700          TST  $SWREG          ; CHECK FOR HALT ON ERROR
690 003522 100401          BMI  1$             ; CONTINUE IF SET
691 003524 000000          HALT                ; <RESET DID NOT CLEAR BITS>
692 003526          1$:
693 003526 032767 002000 174666      BIT  #BIT10,$SWREG   ; CHECK FOR LOOP ON TEST
694 003534 001343          BNE  TST20         ; GO TO LOOP ON TEST
695
696          ;NOTE: THE WRAP CABLE MUST BE INSTALLED TO EXECUTE
697          ;TESTS 21-27
698 003536          †ST21:
699 003536 012767 000021 174640      MOV  #21,$TESTN     ; MOVE TEST NUMBER TO MAILBOX
700 003544 032767 000400 174650      BIT  #BIT8,$SWREG
701 003552 001402          BEQ  LP21           ; DO TESTS IF NOT INHIBITED
702 003554 000167 000710          JMP  TST999        ; IF INHIBITED
703 003560 005015          LP21: CLR  @R5           ; CLEAR STATUS REGISTER
704 003562 005215          INC  @R5           ; SET CSRO
705 003564 105715          TSTB @R5          ; CHECK FOR REQ A FLAG
706 003566 100416          BMI  1$             ; BR IF SET
707 003570 032767 040000 174624      BIT  #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
708 003576 001357          BNE  TST21         ; GO TO LOOP ERROR
709 003600 012767 000026 174574      MOV  #26,$FATAL
710 003606 012767 000001 174564      MOV  #1,$MSGTY     ; MOVE ERROR NUM TO MAILBOX
711 003614 005767 174602          TST  $SWREG        ; CHECK FOR HALT ON ERROR
712 003620 100401          BMI  1$             ; CONTINUE IF SET
713 003622 000000          HALT                ; <BIT0 DID NOT SET BIT7>
714 003624          1$:
715 003624 032767 002000 174570      BIT  #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
716 003632 001341          BNE  TST21         ; GO TO LOOP ON TEST
717
    
```


K02

```

774
775      ;RAISE INTERRUPT "B"
776      †ST25:
777      004114 012767 000025 174262      MOV      #25,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
778      004122 012706 001200              MOV      #STKPTR,SP      ; INITIALIZE STACK POINTER
779      004126 106427 000340              MTPS     #340            ; LOCK OUT INTERRUPTS
780      004132 012777 004154 175074      MOV      #45,@DRVECB    ; INTERRUPT RETURN POINTER
781      004140 012715 000042              MOV      #42,@RS        ; IE AND CSRI
782      004144 106427 000000              MTPS     #0              ; ALLOW INTERRUPTS
783      004150 000240
784      004152 000402
785      004154 005015      4$:      CLR      @RS            ; NO INTERRUPT
786      004156 000416              BR       1$              ; CLEAR INTERRUPT ENABLE
787      004160
788      004160 032767 040000 174234      5$:      BIT      #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
789      004166 001352              BNE     TST25            ; GO TO LOOP ERROR
790      004170 012767 000032 174204      MOV      #32,$FATAL
791      004176 012767 000001 174174      MOV      #1,$MSGTY      ; MOVE ERROR NUM TO MAILBOX
792      004204 005767 174212              TST     $SWREG           ; CHECK FOR HALT ON ERROR
793      004210 100401              BMI     1$              ; CONTINUE IF SET
794      004212 000000              HALT                    ; <NO B INTERRUPT>
795      004214
796      004214 032767 002000 174200      1$:      BIT      #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
797      004222 001334              BNE     TST25            ; GO TO LOOP ON TEST
798
799      ;TEST FOR INTERRUPT FROM DEVICE
800      †ST26:
801      004224 012767 000026 174152      MOV      #26,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
802      004232 017702 174774              MOV      @DRLVLA,R2      ; SAVE INTERRUPT PSW
803      004236 016777 175002 174766      LP26:   MOV      PL,@DRLVLA      ; LOCK OUT SUCCESSIVE INTERRUPTS
804      004244 012706 001200              MOV      #STKPTR,SP      ; INITIALIZE STACK POINTER
805      004250 012777 004324 174752      MOV      #15,@DRVECA    ; INTERRUPT RETURN POINTER
806      004256 012715 000101              MOV      #101,@RS        ; SET INTERRUPT ENABLE-AND CSRO
807      004262 106427 000000              MTPS     #0              ; ALLOW INTERRUPTS
808      004266 000240
809      004270
810      004270 032767 040000 174124      5$:      BIT      #BIT14,$SWREG ; CHECK FOR LOOP ON ERROR
811      004276 001352              BNE     TST26            ; GO TO LOOP ERROR
812      004300 012767 000033 174074      MOV      #33,$FATAL
813      004306 012767 000001 174064      MOV      #1,$MSGTY      ; MOVE ERROR NUM TO MAILBOX
814      004314 005767 174102              TST     $SWREG           ; CHECK FOR HALT ON ERROR
815      004320 100401              BMI     1$              ; CONTINUE IF SET
816      004322 000000              HALT                    ; <NO DEVICE INTERRUPT>
817      004324
818      004324 032767 002000 174070      1$:      BIT      #BIT10,$SWREG ; CHECK FOR LOOP ON TEST
819      004332 001341              BNE     LP26             ; GO TO LOOP ON TEST
820      004334 005015              CLR      @RS            ; CLEAR INTERRUPT ENABLE
821      004336 010277 174670              MOV      R2,@DRLVLA      ; RESTORE INTERRUPT PSW
822
823      ; PLU WRAP TEST
824      †ST27:
825      004342 012767 000027 174034      MOV      #27,$TESTN      ; MOVE TEST NUMBER TO MAILBOX
826      004350 005000              CLR      RO              ; SET UP STARTING DATA
827      004352 010077 174644      WLOOP:  MOV      RO,@DROBUF      ; SEND DATA
828      004356 027700 174642      CMP      @DRIBUF,RO      ; CHECK THE DATA
829      004362 001020              BNE     5$              ; ERROR IF NOT RIGHT
830      004364 005200              INC      RO              ; CHANGE DATA

```

L02

NO DEVICE INTERRUPT

SEQ 0024

830	004366	001434				BEQ	1\$; NEXT TEST IF END
831	004370	022700	031460	3\$:		CMP	#31460,R0		; CHECK FOR TEST MODULE CODE
832	004374	001411				BEQ	4\$		
833	004376	022700	031461			CMP	#31461,R0		
834	004402	001406				BEQ	4\$		
835	004404	022700	031462			CMP	#31462,R0		
836	004410	001403				BEQ	4\$		
837	004412	022700	031463			CMP	#31463,R0		
838	004416	001355				BNE	WLOOP		
839	004420	005200		4\$:		INC	R0		
840	004422	000762				BR	3\$; RECHECK DATA CODE
841	004424			5\$:					
842	004424	032767	040000	173770		BIT	#BIT14,\$SWREG		; CHECK FOR LOOP ON ERROR
843	004432	001347				BNE	WLOOP		; GO TO LOOP ERROR
844	004434	012767	000034	173740		MOV	#34,\$FATAL		
845	004442	012767	000001	173730		MOV	#1,\$MSGTY		; MOVE ERROR NUM TO MAILBOX
846	004450	005767	173746			TST	\$SWREG		; CHECK FOR HALT ON ERROR
847	004454	100401				BMI	1\$; CONTINUE IF SET
848	004456	000000				HALT			; <WRAP DATA DID NOT COMPARE>
849	004460				1\$:				
850	004460	032767	002000	173734		BIT	#BIT10,\$SWREG		; CHECK FOR LOOP ON TEST
851	004466	001325				BNE	TST27		; GO TO LOOP ON TEST

M02

CVKAF C MACY11 30A(1052) 22-DEC-77 11:54 PAGE 20
 CVKAF C.P11 22-DEC-77 11:52 ERROR 34

WRAP DATA DID NOT COMPARE

SEQ 0025

852										
853										
854	004470					TST999:	INC	@#SPASS		; INCREMENT PASS COUNT
855	004470	005237	000406				BITB	#40,\$ENVM		; WILL APT ALLOW PRINTING?
856	004474	132767	000040	173717			BNE	ACT		; NO
857	004502	001010					MOV	#MSG,RO		; GET MESSAGE ADDRESS
858	004504	012700	004554				MOV	@TPS		; CHECK IF TTY READY
859	004510	105777	000056			WAIT:	TSTB	@TPS		; CHECK IF TTY READY
860	004514	100375					BPL	WAIT		; IF NOT
861	004516	112077	000026				MOV	(RO)+,@TPB		; PRINT THE CHARACTER
862	004522	001372					BNE	WAIT		; NEXT IF NOT DONE
863	004524	013700	000042			ACT:	MOV	@#42,RO		; CHECK ACT
864	004530	001405					BEQ	GOAGIN		; KEEP GOING
865	004532	000005					RESET			
866	004534	004710				\$ENDAD:	JSR	PC,(RO)		; ACT HOOKS
867	004536	000240					NOP			
868	004540	000240					NOP			
869	004542	000240					NOP			
870	004544	000167	174506			GOAGIN:	JMP	START		; DO ANOTHER PASS
871	004550	177566				TPB:	.WORD	177566		
872	004552	177777				PASSPT:	-1			
873	004554	047105	020104	043117		MSG:	.ASCIZ	.END OF PASS.<15><12>		
874	004562	050040	051501	006523						
875	004570	000012								
876	004572	177564				TPS:	.WORD	177564		

N02

CVKAF0 MACY11 30A(1052) 22-DEC-77 11:54 PAGE 21
CVKAF0.P11 22-DEC-77 11:52 ERROR 34

WRAP DATA DID NOT COMPARE

SEQ 0026

```

877
878
879
880 004574 010046
881 004576 010146
882 004600 010246
883 004602 010346
884 004604 010446
885 004606 010546
886 004610 016746 173210
887 004614 010637 004630
888 004620 012737 004632 000024
889 004626 000000
890 004630 000000
891 004632 016706 177772
892 004636 012667 173162
893 004642 012605
894 004644 012604
895 004646 012603
896 004650 012602
897 004652 012601
898 004654 012600
899 004656 000137 001256
900 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
MOV @RESTAR,@#24
HALT ;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	230	
ACDW1 = 000000	230	
ACDW2 = 000000	230	
ACPUOP = 000000	230	245
ACT = 004524	857	863#
ADDW0 = 000000	230	
ADDW1 = 000000	230	
ADDW10 = 000000	230	
ADDW11 = 000000	230	
ADDW12 = 000000	230	
ADDW13 = 000000	230	
ADDW14 = 000000	230	
ADDW15 = 000000	230	
ADDW2 = 000000	230	
ADDW3 = 000000	230	
ADDW4 = 000000	230	
ADDW5 = 000000	230	
ADDW6 = 000000	230	
ADDW7 = 000000	230	
ADDW8 = 000000	230	
ADDW9 = 000000	230	
ADEVCT = 000000	230	236
ADEVN = 000000	230	
AENV = 000000	230	241
AENVN = 000000	230	242
AFATAL = 000000	230	233
AMADR1 = 000000	230	
AMADR2 = 000000	230	
AMADR3 = 000000	230	
AMADR4 = 000000	230	
AMAMS1 = 000000	230	
AMAMS2 = 000000	230	
AMAMS3 = 000000	230	
AMAMS4 = 000000	230	
AMSGAD = 000000	230	238
AMSGLG = 000000	230	239
AMSGTY = 000000	230	232
AMTYP1 = 000000	230	
AMTYP2 = 000000	230	
AMTYP3 = 000000	230	
AMTYP4 = 000000	230	
APASS = 000000	230	235
APRIOR = 000000	230	
ASWREG = 000000	230	243
ATESTN = 000000	230	234
AUNIT = 000000	230	237
AUSWR = 000000	230	244
AVECT1 = 000000	230	
AVECT2 = 000000	230	
BIT0 = 000001	140#	374
BIT00 = 000001	130#	140
BIT01 = 000002	129#	139
BIT02 = 000004	128#	138
BIT03 = 000010	127#	137
BIT04 = 000020	126#	136
BIT05 = 000040	125#	135

C03

CVKAF C MACY11 30A(1052) 22-DEC-77 11:54 PAGE 24
 CVKAF C.P11 22-DEC-77 11:52 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0028

BIT06 = 000100	124#	134																
BIT07 = 000200	123#	133																
BIT08 = 000400	122#	132																
BIT09 = 001000	121#	131																
BIT1 = 000002	139#	384																
BIT10 = 002000	120#	351	408	424	444	461	478	503	522	540	564	595	613					
	675	693	715	731	747	772	796	818	850									
BIT11 = 004000	119#																	
BIT12 = 010000	118#																	
BIT13 = 020000	117#																	
BIT14 = 040000	116#	333	343	376	386	400	416	436	453	470	495	514	532					
	556	575	587	605	624	641	656	667	685	707	723	739	764					
	788	810	842															
BIT15 = 100000	115#																	
BIT2 = 000004	138#																	
BIT3 = 000010	137#																	
BIT4 = 000020	136#																	
BIT5 = 000040	135#																	
BIT6 = 000100	134#																	
BIT7 = 000200	133#																	
BIT8 = 000400	132#	430	700															
BIT9 = 001000	131#	364																
BPTVEC = 000014	147#																	
BUFTST = 002312	484#																	
CON = 001474	332	341	357#															
COUNT = 001242	305#																	
CR = 000015	55#																	
CRLF = 000200	56#																	
CSR = 167770	25#	283	284	285	286													
DDISP = 177570	62#																	
DEVCNT = 000410	277#																	
DRBHIO = 001226	297#	528*	547*	548*														
DRCSR = 001220	294#	312	326															
DRIBUF = 001224	296#	373	397	434	827													
DRLVLA = 001232	300#	802	803*	821*														
DRLVLB = 001236	302#																	
DROBUF = 001222	295#	329*	331	370*	412*	413	432*	449*	450	466*	467	509*	510*					
	511	527*	529	546*	550	826*												
DRVECA = 001230	299#	755*	805*															
DRVECB = 001234	301#	780*																
DSWR = 177570	61#																	
EMTVEC = 000030	150#																	
ERRNUM = 000402	278#																	
ERRVEC = 000004	143#																	
ERR1 = 001412	328	342#																
GOAGIN = 004544	864	870#																
HLT = 104000	24#																	
HT = 000011	53#																	
IOTVEC = 000020	148#																	
LF = 000012	54#																	
LP1 = 001320	326#	334	344															
LP21 = 003560	701	703#																
LP26 = 004236	803#	819																
LP7 = 002320	487#	493	496															
MSG = 004554	858	873#																
N = 000035	9#	335	336#	345	346#	378	379#	388	389#	402	403#	418	419#					

G03

CVKAFC MACY11 30A(1052) 22-DEC-77 11:54 PAGE 29
 CVKAFC.P11 22-DEC-77 11:52 CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0032

COMMEN	155#														
ENDCOM	155#														
ERNUM	11#	335	345	378	388	402	418	438	455	472	497	516	534	558	577
	589	607	626	643	658	669	687	709	725	741	766	790	812	844	
ERR	11#	333	342	376	386	400	416	436	453	470	494	514	532	555	575
	587	605	623	640	656	667	685	707	723	739	763	787	809	841	
ERROR	49#														
ESCAPE	155#														
GETPRI	155#														
GETSWR	155#														
MULT	155#														
NEWTST	155#														
POP	155#														
PUSH	155#														
REPORT	155#														
SCOPE	50#														
SEQ	12#														
	650	324	362	410	428	447	464	482	507	525	542	569	598	616	633
	678	698	718	733	751	776	800	823							
SETPRI	155#														
SETUP	155#														
SKIP	155#														
SLASH	155#														
SPACE	155#														
STARS	155#	229	256	258	265										
SWRSU	155#														
TYPBIN	155#														
TYPDEC	155#														
TYPNAM	155#														
TYPNUM	155#														
TYPOCS	155#														
TYPOCT	155#														
TYPTXT	155#														
STNUM	12#														
	651	325	363	411	429	448	465	483	508	526	543	570	599	617	634
	679	699	719	734	752	777	801	824							
\$\$ESCA	155#														
\$\$NEWT	155#														
\$\$SKIP	155#														
.EQUAT	45#														
.SAPT8	45#	227													
.SAPTH	45#	254													
.SAPTY	45#														
.STRAP	45#														
.STYPE	45#														

. ABS. 004662 000

ERRORS DETECTED: 0

CVKAFC.BIN, CVKAFC.LST/CRF/SOL/NL: TOC=CVKAFC.P11
 RUN-TIME: 8 2 .4 SECONDS
 RUN-TIME RATIO: 66/11=5.7
 CORE USED: 13K (25 PAGES)

H03