

*PEM -

IDENTIFICATION

PRODUCT CODE: AC-E800E-MC
PRODUCT NAME: CXADAE0 AD01-D MODULE
PRODUCT DATE: SEPTEMBER 1978
MAINTAINER: DEC/X11 SUPPORT GROUP

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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1973, 1978 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

ADA IS A IOMOD THAT USES PROGRAMMED INTERRUPTS. IT PERFORMS A WAS-15 TEST ON CHANNELS ZERO THROUGH THREE. ONE CONVERSION FOR EACH CHANNEL IS LOADED INTO A TABLE FOLLOWED BY A SECOND CONVERSION FOR EACH CHANNEL. A SECOND TABLE IS LOADED INTO A SECOND TABLE. TWO TABLES ARE THEN COMPARED TO INSURE THAT THE TWO CONVERSIONS AGREE WITHIN THE COUNT SPREAD VALUE. THE CONVERSIONS ARE DONE AT A GAIN OF X8 WHICH DICTATES THAT EACH CHANNEL (0-3) MUST HAVE A CONSTANT DC VOLTAGE INPUT OF LESS THAN 1.25 VOLTS.

2. REQUIREMENTS

HARDWARE: ONE AD01-D ANALOG/DIGITAL CONVERTER OR EQUIVALENT.

STORAGE:: ADA REQUIRES:

1. DECIMAL WORDS: 259
2. OCTAL WORDS: 0403
3. OCTAL BYTES: 1006

3. PASS DEFINITION

ONE PASS OF THE ADA MODULE CONSISTS OF 100. ITERATIONS OF THE BASIC TEST WHICH RESULTS IN 800. CONVERSIONS (800 UNIPUS DATA TRANSFERS)

4. EXECUTION TIME

ADA RUNNING ALONE ON A PDP11/05 PROCESSOR TAKES APPROXIMATELY 1 MINUTE PER PASS.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 176770, VECTOR: 130, BR1: 5, DEVCNT: 1 SRI: 0

REQUIRED PARAMETERS:

NONE FOR STANDARD AD01-D CONVERTER.
4001FY "SRI": AS PER CONVERTER BIT LENGTH
(AT CNF TIME OR RUN TIME)
SRI: =0 IS THE DEFAULT FOR 10 BITS (SPREAD OF 1)
SRI: =1 FOR 11 OR 12 BIT MODELS (SPREAD OF 2)
SRI: =2 FOR 14 BIT MODELS (SPREAD OF 4)

6. DEVICE/OPTION SET-UP

INSURE DC VOLTAGE INPUT TO CHAN. 0-3 IS LESS THAN 1.25 VDC

7. MODULF OPERATION

TEST SEQUENCE:

- A. READ RESULTS OF FOUR CONVERSIONS CHAN. 0-3 INTO TABLE A
- B. READ RESULTS OF FOUR CONVERSIONS CHAN. 0-3 INTO TABLE B
- C. COMPARE DATA IN TABLE A TO DATA IN TABLE B REPORT ERRORS
- D. REREAD A THROUGH C 100 TIMES
- E. REPORT END OF PASS AND REPEAT A THROUGH D

8. OPERATION OPTIONS

USER CAN MODIFY LOCATION ST+2 TO VARY THE NO. OF ITERATIONS PER PASS.

9. NON-STANDARD PRINTOUTS

NONE - ALL PRINTOUTS HAVE STANDARD FORMATS AS DESCRIBED IN THE DEC/X11 DOCUMENT

FADA DEC/X11 AD01 EXERCISER MODULE

000000 IOMOD <ADAE > 176770,130,5,10000,44
000000 MODULE 140000,ADAE 176770,130,5,10000,44
; TITLE ADAE DEC/X11 SYSTEM EXERCISER MODULE
; DOXCOM VERSION 6 23-MAY-78
.LIST BIN

000000 RECFM:
000000 MODNAM: - ASCII /ADAE / ;MODULE NAME.
000005 XFLAG: - BYTE OPEN ;USED TO KEEP TRACK OF WBOFF USAGE
000016 ADDR: 176770+0 ;1ST DEVICE ADDR.
000018 VECFOR: 130+0 ;1ST DEVICE VFCFOR.
000012 BR1: - BYTE PRTY5+0 ;2ND RR LEVEL.
000013 BR2: - BYTE PRTY+0 ;3RD RR LEVEL.
000014 DIVD1: +1 ;DEVICE INDICATOR 1.
000020 SR1: OPEN ;SWITCH REGISTER 1
000020 SR2: OPEN ;SWITCH REGISTER 2
000022 SR3: OPEN ;SWITCH REGISTER 3
000024 SR4: OPEN ;SWITCH REGISTER 4

000026 STAT: 140000 ;STATUS WORD.
000030 INIT: START ;MODULE START ADDR.
000032 SPPOINT: MODSP ;MODULE STACK POINTER.
000034 PASCNT: 0 ;PASS COUNTER.
000036 ITCNT: 10000. ;# OF ITERATIONS PER PASS=10000.
000040 ICPNT: 0 ;LOC TO COUNT ITERATIONS
000042 SPCFNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044 HRDCVNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046 SOPPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000050 HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052 SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
000054 RANUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056 CONFIG: 0 ;RESERVED FOR MONITOR USE
000058 RES1: 0 ;RESERVED FOR MONITOR USE
000062 SVR2: 0 ;RESERVED FOR MONITOR USE
000064 SVR0: OPEN ;LOC TO SAVE R0.
000066 SVR1: OPEN ;LOC TO SAVE R1.
000070 SVR2: OPEN ;LOC TO SAVE R2.
000072 SVR3: OPEN ;LOC TO SAVE R3.
000074 SVR4: OPEN ;LOC TO SAVE R4.
000076 SVR5: OPEN ;LOC TO SAVE R5.
000100 SVR6: OPEN ;LOC TO SAVE R6.
000102 CSRA: OPEN ;ADDR OF CURRENT CSR.
000104 SBADR: OPEN ;ADDR OF GOOD DATA, OR
000106 WASADP: OPEN ;CONTENTS OF CSR.
000108 ASTAT: OPEN ;ADDR OF BAD DATA OR
000106 ASRTYP: OPEN ;STATUS REG CONTENTS.
000106 ASRTYP: OPEN ;TYPE OF ERROR
000110 AWAS: OPEN ;EXPECTED DATA.
000112 WDR: OPEN ;LOCAL DATA.
000114 RSTRT: RESTRT ;RESTART ADDRESS AFTER END OF PASS
000116 WDR0: OPEN ;WORDS TO MEMORY PER ITERATION
000120 INTR: OPEN ;WORDS FROM MEMORY PER ITERATION
;# OF INTERRUPTS PER ITERATION

000122 000044 IDNUM: 44 ;MODULE IDENTIFICATION NUMBER=44
000040 ;MODULE STACK STARTS HERE.
;REPT SPSTZ
;LIST 0
;LIST
;ENDR
000224 MODSP:

;SOME MODULE VARIABLES
ADCSR: OPEN
ADCSRO: OPEN
ADDRR: OPEN
;MODULE INITIALIZATION
200 START: MOV #8, WOTO ;#8. WORDS TO MEM PER ITERATION
201 MOV #8, INTR ;#8. INTERRUPTS
202
203 000224 000000 RESTRT: MOV ADDR, R5 ;GET THE FIRST ADDRESS
204 000226 000000 MOV R5, ADCSR ;SET UP THE CSR ADDRESS
205 000230 000000 TSTR (5)+ ;CALCULATE ADDR ADDRESS
206 ;ADD +1 FOR DATA BUFFER ADDRESS
207 ;SET UP HI BYTE OF CSR ADDRESSING
208 ;GET THE ADDR ADDRESS
209 ;SET UP THE ADDR ADDRESS
210 ;GET THE VECTOR ADDRESS
211 ;POINT INTERRUPTS TO AD01
212 ;SET UP THE PRIORITY LEVEL
213 000246 016705 177534 ST: MOV ADCSR, CSRA ;MOV CSR ADDRESS TO CSRA
214 000248 010557 177746 MOV #ADTBLA, TRPTR ;LOAD POINTER TO TABLE
215 000260 010567 177742 INC #134, ADCSR ;SET FLAG TO INDICATE USING A TABLE
216 000264 105725 ;ENABLE INTERRUPTS, SET GAIN (130=X8,
217 000266 010567 177732 ;120=X110-X2,100=X1)
218 000272 016700 177512 BIT #1, SR1 ;TEST SR1 SET FOR I10R 12 BITS
219 000276 012720 000416 BR 15, SPREAD ;NO CHECK FOR 14 BITS
220 000302 016720 177504 BR 35 ;YES LOAD PROPER SPREAD VAL'VE
221 000306 016767 177712 ;CONTINUE
222 000314 016767 177712 ;14 BIT DEVICF?
223 000322 095767 000430 ;NO GO LOAD DEFAULT
224 000326 112777 177670 ;YES LOAD PROPER SPREAD VAL'VE
225 ;CONTINUE
226 000334 032767 000001 15: BIT #2, SR1 ;TEST SR2 SET FOR I10R 12 BITS
227 000342 001404 BR 15, SPREAD ;NO CHECK FOR 14 BITS
228 000344 012767 000002 000400 BR 35 ;YES LOAD PROPER SPREAD VAL'VE
229 000352 00413 ;CONTINUE
230 000354 032767 000002 177434 15: BIT #2, SR1 ;TEST SR2 SET FOR I10R 12 BITS
231 000362 001404 BR 15, SPREAD ;NO CHECK FOR 14 BITS
232 000364 012767 000010 000360 BR 35 ;YES LOAD PROPER SPREAD VAL'VE
233 000372 00403 ;CONTINUE
234 000374 012767 000001 000350 25: MOV #1, SPREAD ;MAINTAIN COUNT SPREAD OF 1
235 000402 105077 177620 CLR ADCSRO ;START CONVERSION VIA DATOB TO HIGH BYTE
236 000406 005067 000346 CLR COUNT
237 000412 104400 000000 EXITS, BGRIN
238 ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
239
240 ;INTERRUPT SERVICE ROUTINES
241 AD01D:
242 ;-----
243 ;TRQS, BEGIN, IS ; QUEUE UP TO CONTINUE AT IS AND RTI
244 ;-----
245 000424 005777 177574 15: TST #ADCSR ;CHECK FOR ERROR
246 000430 000010 BPL 25 ;BRANCH IF NONE
247 000432 017767 177566 177442 MOV #ADCSR, ACSR ;MOV CONTENTS OF CSP TO ACSR

```
248 000440 005067 177442 CLR ERRTPY ;UNKNOWN ERROR TYPE
249 *****
250 000444 104405 000000 000000 HDRFRS,BEGIN, NULL ;ERROR BIT IN CSR IS SET
251 *****
252 000452 017777 177552 000274 2S: MOV @ADDR,@TRPTR ;LOAD DATA BUFFER INTO TABLE
253 000460 026727 000270 000774 CMP TRPTR,#ADTBLA+6 ;CHECK FOR TABLE A FILLED
254 000466 001004 BNE AD1 ;BRANCH IF NOT FILLED
255 000470 012767 000776 000256 MOV #ADTBLB,TRPTR ;LOAD POINTER TO TABLE B
256 000476 004422 BDC ;
257 000500 026727 000250 001004 AD1: CMP TRPTR,#ADTBLB+6 ;CHECK FOR TABLE B BEING FULL
258 000506 001004 BNE AD2 ;BRANCH IF NOT FULL
259 000510 012767 000766 000236 MOV #ADTBLA,TRPTR ;LOAD POINTER TO POINT RACK AT TABLE A
260 000516 004412 BDC ;
261 000520 062767 000002 000226 AD2: ADD #2,TRPTR ;MOVE THE TABLE POINTER TO NEXT WORD
262 000526 005267 000226 CHOUT ;ERROR MAY BE SET SO USE A MOVE TO CLEAR
263 000532 116777 000222 177466 MOV#R,CHOUT,@ADCSRO ;THE MOVE THAT DOES IT I THINK
264 000540 104400 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
265 000544 005767 000206 ADCK: TST ADTEMP ;READY TO START TABLE B ??
266 000550 001053 BNE ;BR IF YES
267
268 ;ROUTINE TO CHECK DATA FOR ACCURACY OF CONVERSION
269
270 000552 012767 000766 000202 MOV #ADTBLA,POINTA ;LOAD POINTER TO TABLE A
271 000556 012767 000776 000176 MOV #ADTBLB,POINTB ;LOAD POINTER TO TABLE B
272 000560 017767 000170 000162 ADLOOP: MOV @POINTA,ADTEMP ;PUT A TABLE DATA IN ADTEMP
273 000574 167767 000164 000154 SUB @POINTB,ADTEMP ;SUBTRACT SECOND READING ON THE SAME CHANNEL
274 000602 066767 000144 000146 ADD SPKPAD,ADTEMP ;TEST FOR DIFFERENCE TO EXCEED C.S.
275 000610 100004 BPL 1S ;BR IF OK
276 000612 004767 000102 JSR PC,DATERR ;GO LOAD ASR,AWAS, AND ACSR
277 *****
278 000616 104404 000000 DATERR,BEGIN ;DATA ERROR!!!
279 *****
280 000622 166767 000124 000126 1S: SUB SPREAD,ADTEMP ;FIND ORIGINAL DIFFERENCE
281 000630 166767 000116 000120 SUB SPREAD,ADTEMP ;TEST FOR DIFFERENCE TO EXCEED C.S..
282 000636 100404 BMT 2S ;BRANCH IF OK
283 000640 004767 000054 JSR PC,DATERR ;GO LOAD ASR,AWAS, AND ACSR
284 *****
285 000644 104404 000000 DATERR,BEGIN ;DATA ERROR!!!
286 *****
287 000650 062767 000002 000104 2S: ADD #2,POINTA ;MOVE POINTER A
288 000656 062767 000002 000100 ADD #2,POINTB ;MOVE POINTER B
289 000664 026727 000072 000776 CMP POINTA,#ADTBLA+10 ;END OF TABLES ??
290 000672 001335 BNE ADLOOP ;LOOP UNTIL DONE
291 000674 104413 000000 SNOITS,BEGIN ;SIGNAL END OF ITERATION.
292 ;MONITOR SHALL TEST END OF PASS
293
294 000700 005067 000052 ADCONT: CLR ADTEMP ;INITIALIZE ADTEMP
295 000704 005067 000050 CLR CHOUT ;
296 000710 105077 177312 CLR @ADCSRO ;START CONVERSION
297 000714 104400 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
298
299 000720 016767 000036 177156 DATERR: MOV POINTA,WASADR ;LOAD WAS ADDR AS LOC IN DATER TAB.#1
300 000726 016767 000032 177146 MOV POINTB,SBADR ;LOAD S/B ADDR AS LOC IN DATER TAB.#2
301 000734 017767 000022 177146 MOV @POINTA,AWAS ;DIFFERENCE BETWEEN AWAS AND ASB IS
302 000742 017767 000016 177136 MOV @POINTB,ASB ;THE ERROR SHOULD BE 1
303 000750 000207 RTS ;RETURN TO PRINT ERROR
```

```
304
305
306 000752 000001 SPREAD: 1
307
308 000754 000000 TRPTR: OPEN
309 000756 000000 ADTEMP: OPEN
310 000760 000000 CHOUT: OPEN
311 000762 000000 POINTA: OPEN
312 000764 000000 POINTB: OPEN
313 000766 000000 ADTBLA: OPEN
314 000776 000776 ADTBLB: OPEN
315 000776 000000 .=-*6
316 001006 .=-*6
317
318 000001 .END
```


SVR1	000064R	174#								
SVR2	000066R	175#								
SVR3	000070R	176#								
SVR4	000072R	177#								
SVR5	000074R	178#								
SVR6	000076R	179#								
SYSCNT	000052R	168#								
TRPTR	000754R	229#	252*	253	255*	257	259*	261*	308#	
TRPDFD=	000022	200#								
VECTDR	000010R	149#								218
WKSADR	000104R	183#								299*
WDR	000116R	190#								
WDT0	000114R	189#								209*
XFLAG	000005R	147#								
.	= 001006R	314#								316#

. AFS. 000000 000
001006 001

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

XADAE0, XADAE0/SOL/CRF:SYM=DDXCOM, XADAE0
RUN-TIME: 1 1 .2 SECONDS
RUN-TIME RATIO: 11/2=4.0
CORE USED: 7K (13 PAGES)