

-REM -

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 TOMOD THAT USES PROGRAMMED INTERRUPTS. IT PERFORMS A WAS-IS TEST ON CHANNELS ZERO THROUGH THREE. ONE CONVERSION FOR EACH CHANNEL IS LOADED INTO A TABLE FOLLOWED BY A SECOND CONVERSION FOR EACH CHANNEL LOADED INTO A SECOND TABLE. THE 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 UNIBUS 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.

MODIFY "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 8)

6. DEVICE/OPTION SET-UP

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

7. MODULE 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

ADAE DEC/X11 ADO1 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
; DDXCOM VERSION 8 23-MAY-78
;*****LIST BIN*****
000000* RFLIN: ;
000000* MODNAM: -ASCII /ADAE /MODULE NAME
000000* 042101 042501 040 XFLAG: -BYTE OPEN ;USED TO KEEP TRACK OF WBUF USAGE
000000* 000 ADDR: 176770+0 ;1ST DEVICE ADDR
000000* 000010 000130 VECTOR: 130+0 ;1ST DEVICE VECTOR
000000* 000012* 000 SR1: -BYTE PRTV5+0 ;1ST RR LEVEL
000000* 000013* 000 SR2: -BYTE PRTV+0 ;2ND RR LEVEL
000000* 000014* 000001 DVID1: +1 ;DEVICE INDICATOR 1
000000* 000016* 000000 SR1: OPEN ;SWITCH REGISTER 1
000000* 000017* 000000 SR2: OPEN ;SWITCH REGISTER 2
000000* 000018* 000000 SR3: OPEN ;SWITCH REGISTER 3
000000* 000021* 000000 SR4: OPEN ;SWITCH REGISTER 4
;*****
000026* 140000 ;STATUS WORD
000030* 000232* INIT: START ;MODULE START ADDR
000032* 000224* SPOINT: MODSP ;MODULE STACK POINTER
000034* 000000 PASCNT: 0 ;PASS COUNTER
000036* 000000 ICOUNT: 10000. ;# OF ITERATIONS PER PASS=10000
000040* 000000 SOFCNT: 0 ;LOC TO COUNT ITERATIONS
000042* 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000044* 000000 SDFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000046* 000000 HRDPAS: 0 ;LOC TO SAVE SORT ERRORS PER PASS
000050* 000000 SYSCNT: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000052* 000000 RANNUM: 0 ;# OF SYS ERRORS ACCUMULATED
000054* 000000 COMPTG: 0 ;HOLDS RNDOM # WHEN RAND MACRO IS CALLED
000056* 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000060* 000000 RFS2: 0 ;RESERVED FOR MONITOR USE
000062* 000000 SVR0: OPEN ;LOC TO SAVE R0
000064* 000000 SVR1: OPEN ;LOC TO SAVE R1
000066* 000000 SVR2: OPEN ;LOC TO SAVE R2
000070* 000000 SVR3: OPEN ;LOC TO SAVE R3
000072* 000000 SVR4: OPEN ;LOC TO SAVE R4
000074* 000000 SVR5: OPEN ;LOC TO SAVE R5
000076* 000000 SVR6: OPEN ;LOC TO SAVE R6
000100* 000000 CSRA: OPEN ;ADDR OF CURRENT CSR
000102* 000000 SRADR: ;ADDR OF GOOD DATA, OR
000104* 000000 WADR: ;CONTENTS OF CSR
000106* 000000 ASTAT: OPEN ;ADDR OF BAD DATA, OR
000108* 000000 ERRTYP: ;STATUS REG CONTENTS
000110* 000000 ASB: OPEN ;TYPE OF ERROR
000112* 000000 RESTR: OPEN ;EXPECTED DATA
000114* 000000 WDTO: OPEN ;ACTUAL DATA
000116* 000000 WFER: OPEN ;RESTART ADDRESS AFTER END OF PASS
000120* 000000 INTR: OPEN ;WORDS TO MEMORY PER ITERATION
; WORDS FROM MEMORY PER ITERATION
; # OF INTERRUPTS PER ITERATION
```

```
000122* 000044 IDNUM: 44 ;MODULE IDENTIFICATION NUMBER=44
000040 ;MODULE STACK STARTS HERE.
;REPT SPSTZ
;NLIST
;WORD 0
;LIST
;ENDR
000224* MODSP: ;*****
;*****
;SOME MODULE VARIABLES
ADCSR: OPEN
ADCSR0: OPEN
ADDRR: OPEN
;MODULE INITIALIZATION
200
201
202
203
204 000224* 000000
205 000226* 000000
206 000230* 000000
207
208
209 000232* 012767 000010 177654 START: MOV #8,WDTO ;8. WORDS TO MEM PER ITERATION
210 000240* 012767 000010 177652 MOV #8,INTR ;8. INTERRUPTS
211
212 000246* 016705 177534 RESTR: MOV ADDR,R5 ;GET THE FIRST ADDRESS
213 000252* 010567 177746 MOV R5,ADCSR ;SET UP THE CSR ADDRESS
214 000256* 105725 177742 TSTR (5)* ;CALCULATE ADDR ADDRESS
215 000260* 010567 177742 MOV R5,ADCSR0 ;SET UP HI BYTE OF CSR ADDRESSING
216 000264* 105725 177736 TSTR ;ADD *1 FOR DATA BUFFER ADDRESS
217 000266* 010567 177736 MOV R5,ADDRR ;SET UP THE ADDR ADDRESS
218 000272* 016700 177512 MOV VFCR,P0 ;GET THE VECTOR ADDRESS
219 000276* 016700 000416* MOV #AD010(0)+ ;POINT INTERRUPTS TO AD010
220 000300* 016700 177504 MOV R5,R1 ;SET UP THE PRIORITY LEVEL
221 000306* 016767 177712 177564 MOV ADCSR,CSRA ;MOV CSR ADDRESS TO CSRA
222 000314* 012767 000766* 000432 ST: MOV #ADTBLA,TRPTR ;LOAD POINTER TO TABLE A
223 000322* 005297 000430 INC ADTEMP ;SET FLAG TO INDICATE USING A TABLE
224 000326* 112777 000134 177670 MOVR #13,ADCSR ;TABLE INTERRUPTS, SET GAIN (130=X8,
;120=X4,110=X2,100=X1)
225
226 000334* 032767 000001 177454 BIT #1,SR1 ;TEST SR1 SET FOR 11OR 12 BITS
227 000342* 001404 000002 000400 BRG LS ;NO CHECK FOR 14 BITS
228 000344* 017767 000002 000400 MOV #2,SPREAD ;YES, LOAD PROPER SPREAD VAL'VE
229 000352* 000413 000000 CONTINUE
230 000354* 032767 000002 177434 1S: BIT #2,SR1 ;14 BIT DEVICE?
231 000362* 001404 000010 000360 BRG ZS ;NO GO LOAD DEFAULT
232 000364* 012767 000010 000360 MOV #10,SPREAD ;YES LOAD PROPER SPREAD VAL'VE
233 000372* 000403 000000 CONTINUE
234 000374* 012767 000001 000350 2S: MOV #1,SPREAD ;MAINTAIN COUNT SPREAD OF 1
235 000402* 105077 177620 CLR ADCSR0 ;START CONVERSION VIA DATOB TO HIGH BYTE
236 000406* 005057 000346 CLR CHOUT ;
237 000412* 104400 000000* EXIT$BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
;INTERRUPT SERVICE ROUTINES
238
239
240
241 000416* ADO1:
242
243 000416* 000004 000000* 000424* ;-----
244 ;TRQS,BEGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTY
245 ;ST ADCSR ; CHECK FOR ERROR
246 1S: BPL ZS ;BRANCH IF NONE
247 000432* 017767 177566 177442 MOV #ADCSR,ACSR ;MOV CONTENTS OF CSP TO ACSR
```

```

248 000440 005067 177442 CLR ERRTP ;UNKNOWN ERROR TYPE
249 *****
250 000444 104405 000000 000000 HDRERR,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 000776 000256 JNE AD1 ;BRANCH IF NOT FILLED
255 000470 017767 000776 000256 MOV #ADTBLB,@TRPTR ;LOAD POINTER TO TABLE B
256 000476 000422 000776 000256 JRV ADCK
257 000500 026727 000250 001004 AD1: CMP TRPTR,#ADTBLB+6 ;CHECK FOR TABLE B BEING FULL
258 000506 001004 000250 001004 JNE AD2 ;BRANCH IF NOT FULL
259 000510 017767 000766 000236 MOV @TBLA,@TRPTR ;LOAD POINTER TO POINT RACK AT TABLE A
260 000516 000412 000766 000236 JRV ADCK
261 000520 062767 000002 000226 AD2: ADD #2,@TRPTR ;MOVE THE TABLE POINTER TO NEXT WORD
262 000526 005267 000226 000226 INC CHOUT ;ERROR MAY BE SET SO USE A MOVE TO CLEAR IT AND START CONVERSION
263 *****
264 000532 116777 000222 177466 MOV#R,CHOUT,@ADCSRO ;THE MOVE THAT DOES IT I THINK
265 000540 104400 000000 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
266 000544 005767 000206 000206 ANCK: TST ADTEMP ;READY TO START TABLE B ??
267 000550 001053 000206 000206 JNE ADCONT ;BR IF YES
268 *****
269 ;ROUTINE TO CHECK DATA FOR ACCURACY OF CONVERSION
270 *****
271 000552 012767 000766 000292 MOV #ADTBLA,@POINTA ;LOAD POINTER TO TABLE A
272 000560 012767 000776 000176 MOV #ADTBLB,@POINTB ;LOAD POINTER TO TABLE B
273 000566 017767 000170 000162 ADLOOP: MOV @POINTA,@DTEMP ;PUT A TABLE DATA IN ADTEMP
274 000574 167767 000164 000154 SUB @POINTB,@DTEMP ;SUBTRACT SECOND READING ON THE SAME CHANNEL
275 000580 004676 000144 000146 ADD SPREAD,@DTEMP ;TEST FOR DIFFERENCE TO EXCEED C.S.
276 000586 100004 000144 000146 JPL IS ;BR IF OK
277 000612 004767 000102 JSR PC,DATERR ;GO LOAD ASB,AWAS, AND ACSR
278 *****
279 000616 104404 000000 000000 DATERR,BEGIN ;DATA ERROR!!!
280 *****
281 000622 166767 000124 000126 1S: SUB SPREAD,@DTEMP ;FIND ORIGINAL DIFFERENCE
282 000630 166767 000116 000120 SUB SPREAD,@DTEMP ;TEST FOR DIFFERENCE TO EXCEED C.S..
283 000636 066767 000054 000054 JMI 2S ;BRANCH IF OK
284 000640 004767 000054 000054 JSR PC,DATERR ;GO LOAD ASB,AWAS, AND ACSR
285 *****
286 000644 104404 000000 000000 DATERR,BEGIN ;DATA ERROR!!!
287 *****
288 000650 062767 000002 000104 2S: ADD #2,@POINTA ;MOVE POINTER A
289 000656 062767 000002 000100 ADD #2,@POINTB ;MOVE POINTER B
290 000664 026727 000072 000776 CMP POINTA,#ADTBLA+10 ;END OF TABLES ??
291 000672 001335 000072 000776 JNE ADLOOP ;LOOP UNTIL DONE
292 000674 104413 000000 000000 ENDTLS,BEGIN ;SIGNAL END OF ITERATION.
293 *****
294 000700 005067 000052 000052 ADCONT: CLR ADTEMP ;MONITOR SHALL TEST END OF PASS
295 000704 005067 000050 000050 CLR CHOUT ;INITIALIZE ADTEMP
296 000710 105077 177312 000000 CLR @ADCSRO ;START CONVERSION
297 000714 104400 000000 000000 EXITS,BEGIN ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
298 *****
299 000720 016767 000036 177156 DATERR: MOV POINTA,#ASADR ;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,@WAS ;DIFFERENCE BETWEEN AWAS AND ASB IS
302 000742 017767 000016 177136 MOV @POINTB,@ASB ;THE ERROR SHOULD BE 1
303 000750 000207 000016 177136 RTS PC ;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 000000 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#							
SVSCMT	000022R	158#							
TBPTR	000754R	222*	253	255*	257	259*	261*	308#	
TRPOFD=	000022	200#							
VECTOR	000010R	149#							218
WASADR	000104R	183#							299*
WDR	000116R	190#							
WDT0	000114R	189#							209*
XPLAG	000005R	147#							
.	= 001006R	314#							316#

. A9S. 000000 000  
001006 001

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

KADAE0,KADAE0/SOL/CRF:SYM=DDXCOM,KADAE0  
RUN-TIME: 1 1 .2 SECONDS  
RUN-TIME RATIO: 11/2=4.0  
CORE USED: 7K (13 PAGES)