

.REM -

IDENTIFICATION  
-----

PRODUCT CODE: AC-F425A-MC  
PRODUCT NAME: CXMNDA0 MNDA (D/A) MOD  
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 OR EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1979, 1979 DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT:  
-----

"MND" IS A BKMOD THAT EXERCISES THE MNDA DIGITAL TO ANALOG CONVERTER. A CONFIDENCE LOGIC TEST IS EXECUTED ON THE DAC0, DAC1, DAC2 AND DAC3 REGISTERS. ALL LOGIC ERRORS ARE REPORTED TO THE CONSOLE TELETYPE. "MND" WILL TEST MULTIPLE UNITS AS SELECTED BY DEVCNT/DVID1.

2. REQUIREMENTS:  
-----

HARDWARE: ONE MNDA (D/A)

STORAGE: MND REQUIRES:  
DECIMAL WORDS: 399  
OCTAL WORDS: 617  
OCTAL BYTES: 1436

3. PASS DEFINITION:  
-----

ONE PASS OF THE MND MODULE CONSISTS OF FLOATING A 1 AND A 0 ACROSS THE FOUR D TO A REGISTERS 3000(8) TIMES. THIS RESULTS IN 340,000 BUS REFERENCES TO THE MNDA OPTION

4. EXECUTION TIME:  
-----

VARIABLES WITH THE NUMBER OF OTHER DEVICES BEING RUN. THIS SHOULD TAKE AN AVERAGE OF THIRTY SECONDS TO COMPLETE ONE PASS WHEN RUNNING ALONE.

5. CONFIGURATION PARAMETERS:  
-----

DEFAULT PARAMETERS:

DVA: 171060, VCT: N/A, BR1: N/A, DEVCNT: 1, SRI: N/A

REQUIRED PARAMETERS:

NONE

6. DEVICE OPTION SETUP:  
-----

NONE.

7. MODULE OPERATION:  
-----

START/RESTART:

THIS CODE WILL USE THE VALUE CONTAINED IN LOCATION "ADDR" TO BE THE BASE ADDRESS OF THE MNCDA. THE BUS ADDRESS OF EACH DAC IS PRIMED IN THIS ROUTINE. THE INITIAL PASS COUNTER IS ALSO PRESET.

TSDAC0:

THE ABILITY OF DAC 0 REGISTER TO HOLD A FLOATING 1 PATTERN IS VERIFIED IN THIS CODE. BIT 11 OF THE REGISTER IS INITIALLY SET (4000) AND THEN ROTATED TO THE RIGHT. UPON COMPLETION, THE SAME PROCEDURE IS REPEATED EXCEPT THE INITIAL VALUE IS 3777.

TSDAC1:, TSDAC2:, TSDAC3:

SAME AS TSDAC0

DUAL:

THIS ROUTINE WILL LOAD DIFFERENT DATA INTO EACH REGISTER AND VERIFY INDEPENDANT ADDRESS SELECTION.

DONE:

IN THIS ROUTINE, THE LOCATION "PASSX" IS DECREMENTED TO DETERMINE IF THE MODULE HAS BEEN EXERCISED. IF NOT THE PROGRAM WILL LOOP TO LOCATION "LOOPA" AND REPEAT THE SEQUENCE. WHEN THE PASS COUNT HAS BEEN COMPLETED, THE "END OF PASS" IS REPORTED.

8. OPERATOR OPTIONS:  
-----

LOCATION (PASSCT) CAN BE MODIFIED TO VARY THE NUMBER OF LOOPS THRU TEST BEFORE END OF PASS IS REPORTED (TO ACCOMODATE SYSTEM CONFIGURATION).

9. NON-STANDARD PRINTOUTS:  
-----

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

```
151 000000"
152 000000"
153
154
155
156 *****
157 000000" 047115 040504 040
158 000000" 000
159 000005" 000
160 000006" 171060
161 000010" 000000
162 000012" 000
163 000013" 000
164 000014" 000001
165 000016" 000000
166 000020" 000000
167 000022" 000000
168 000024" 000000
169
170 000026" 040020
171 000030" 000236"
172 000032" 000224"
173 000034" 000000
174 000036" 003000
175 000040" 000000
176 000042" 000000
177 000044" 000000
178 000046" 000000
179 000050" 000000
180 000052" 000000
181 000054" 000000
182 000056" 000000
183 000058" 000000
184 000060" 000000
185 000062" 000000
186 000064" 000000
187 000066" 000000
188 000070" 000000
189 000072" 000000
190 000074" 000000
191 000076" 000000
192 000100" 000000
193 000102" 000000
194 000102" 000000
195 000104" 000000
196 000104" 000000
197 000106" 000000
198 000106" 000000
199 000110" 000000
200 000112" 000236"
201 000114" 000000
202 000116" 000000
203 000120" 000000
204 000122" 000000
205 000040
206 .NLIST

BKMOD <MNDA >,171060,0,0,0,0,0,0,0,3000,0
MODULE 40020,MNDA ,171060,0,0,0,0,33000,0
.TITLE MNDA DEC/X11 SYSTEM EXERCISER MODULE
DDXCOM VERSION 6 23-MAY-78
.LIST BJJ
*****
BEGIN:
MODNAM: .ASCII /MNDA / ;MODULE NAME.
XFAC: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFF USAGE
ADDR: 171060+0 ;1ST DEVICE ADDR.
VECTOR: 0+0 ;1ST DEVICE VECTOR.
BR1: .BYTE PRTY0+0 ;1ST BR LEVEL.
BR2: .BYTE PRTY0+0 ;2ND BR LEVEL.
DVID1: 0+1 ;DEVICE INDICATOR 1.
SR1: OPEN ;SWITCH REGISTER 1
SR2: OPEN ;SWITCH REGISTER 2
SR3: OPEN ;SWITCH REGISTER 3
SR4: OPEN ;SWITCH REGISTER 4
*****
STAT: 40020 ;STATUS WORD.
INIT: START ;MODULE START ADDR.
SPOINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICONT: 3000 ;# OF ITERATIONS PER PASS=3000
ICOUNT: 0 ;LOC TO COUNT ITERATIONS
SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
RANNUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CONFIG: ;RESERVED FOR MONITOR USE
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
SVR0: OPEN ;LOC TO SAVE R0.
SVR1: OPEN ;LOC TO SAVE R1.
SVR2: OPEN ;LOC TO SAVE R2.
SVR3: OPEN ;LOC TO SAVE R3.
SVR4: OPEN ;LOC TO SAVE R4.
SVR5: OPEN ;LOC TO SAVE R5.
SVR6: OPEN ;LOC TO SAVE R6.
CSRA: OPEN ;ADDR OF CURRENT CSR.
SRADR: ;ADDR OF GOOD DATA, OR
ACSR: OPEN ;CONTENTS OF CSP.
WASADR: ;ADDR OF BAD DATA, OR
ASAT: OPEN ;STATUS REG CONTENTS.
ERRTYP: ;TYPE OF ERROR
ASB: OPEN ;ACTUAL DATA.
AWAS: OPEN ;RESTART ADDRESS AFTER END OF PASS
RSTFT: ;WORDS TO MEMORY PER ITERATION
WDTO: OPEN ;WORDS FROM MEMORY PER ITERATION
WDFR: OPEN ;# OF INTERRUPTS PER ITERATION
INTR: OPEN ;MODULE IDENTIFICATION NUMBER=0
IDNUM: 0 ;MODULE STACK STARTS HERE.
.REPT SPSIZ
.NLIST
```

```
207          .WOPD 0
208          .LIST
209          .ENDR
210 000224*   MODSP;
211          ;*****
212
213          ;DEVICE BUS ADDRESS
214
215
216 000224* 000006*   DAC0:  ADDR          ;BUS ADDRESS OF DAC 0
217 000226* 000010*   DAC1:  ADDR+2      ;BUS ADDRESS FOR DAC 1
218 000230* 000012*   DAC2:  ADDR+4      ;          2
219 000232* 000014*   DAC3:  ADDR+6      ;          3
220 000234* 000001   TEMP:  BIT0
221          ;INITIALIZATION CODE
222 000236*
223 000236* 012767 000001 177770   RESTRT: MOV  #BIT0,TEMP ;LOAD UNIT POINTER
224 000244* 016767 177536 177752   LOOPA:  MOV  ADDR,DAC0 ;LOAD BUS ADDRESS
225 000252* 016767 177746 177746   MOV  DAC0,DAC1 ; FOR
226 000260* 062767 000002 177740   ADD  #2,DAC1
227 000266* 016767 177734 177734   MOV  DAC1,DAC2 ; DIFFERENT
228 000274* 062767 000002 177726   ADD  #2,DAC2
229 000302* 016767 177722 177722   MOV  DAC2,DAC3 ; DAC
230 000310* 062767 000002 177714   ADD  #2,DAC3 ;BUS ADDRESSES
231 000316* 012767 010000 177556   TSDAC0: MOV #BIT12,ACSR ;LOAD EXPECTED
232 000324* 012777 004000 177672   MOV  #BIT11,0DAC0 ;LOAD DAC0 REGISTER
233 000332* 016767 177666 177540   MOV  DAC0,CSRA ;LOAD BUS ADDRESS
234 000340* 006267 177536          1S:  ASR  ACSR ;SHIFT THE EXPECTED
235 000344* 001043          BNE  4S ;R IF DONE
236 000346* 017767 177652 177530   MOV  0DAC0,ASTAT ;READ THE REGISTER
237 000354* 026767 177522 177522   CMP  ACSR,ASTAT ;COMPARE
238 000362* 001403          BEQ  2S ;R IF SAME
239
240 000364* 104405 000000* 000000   ;*****
241          HDRS,REGIN,NULL ;DAC0 FAILED TO HOLD THE FLOATING 1 PATTERN
242          ;*****
242 000372* 005167 177504          2S:  COM  ACSR ;COMPLEMENT DATA
243 000376* 005177 177622          COM  0DAC0 ;COMPLEMENT DATA IN DAC0
244 000402* 042767 170000 177472   BIC  #170000,ACSR ;MASK OFF UNUSED BITS
245 000410* 017767 177610 177466   MOV  0DAC0,ASTAT ;READ DAC0
246 000416* 026767 177460 177460   CMP  ACSR,ASTAT ;COMPARE
247 000424* 001403          BEQ  3S ;R IF SAME
248
249 000426* 104405 000000* 000000   ;*****
250          HDRS,REGIN,NULL ;DAC0 FAILED TO HOLD THE FLOATING 0 PATTERN
251          ;*****
251 000434* 005167 177442          3S:  COM  ACSR ;COMPLEMENT EXPECTED
252 000440* 005177 177560          COM  0DAC0 ;COMPLEMENT DATA
253 000444* 042767 170000 177430   BIC  #170000,ACSR ;MASK EXPECTED DATA
254 000452* 000732          BR   1S ;TEST MORE BITS
255
256 000454*          4S:
```

```
257 000454* 012767 010000 177420   TSDAC1: MOV #BIT12,ACSR ;LOAD EXPECTED
258 000462* 012777 004000 177536   MOV  #BIT11,0DAC1 ;LOAD DAC1 REGISTER
259 000470* 016767 177532 177402   MOV  DAC1,CSRA ;LOAD BUS ADDRESS
260 000476* 006267 177400          1S:  ASR  ACSR ;SHIFT THE EXPECTED
261 000502* 001043          BNE  4S ;R IF DONE
262 000504* 017767 177516 177372   MOV  0DAC1,ASTAT ;READ THE REGISTER
263 000512* 026767 177364 177364   CMP  ACSR,ASTAT ;COMPARE
264 000520* 001403          BEQ  2S ;R IF SAME
265
266 000522* 104405 000000* 000000   ;*****
267          HDRS,REGIN,NULL ;DAC1 FAILED TO HOLD THE FLOATING 1 PATTERN
268          ;*****
268 000530* 005167 177346          2S:  COM  ACSR ;COMPLEMENT DATA
269 000534* 005177 177466          COM  0DAC1 ;COMPLEMENT DATA IN DAC1
270 000540* 042767 170000 177334   BIC  #170000,ACSR ;MASK OFF UNUSED BITS
271 000546* 017767 177454 177330   MOV  0DAC1,ASTAT ;READ DAC1
272 000554* 026767 177322 177322   CMP  ACSR,ASTAT ;COMPARE
273 000562* 001403          BEQ  3S ;R IF SAME
274
275 000564* 104405 000000* 000000   ;*****
276          HDRS,REGIN,NULL ;DAC1 FAILED TO HOLD THE FLOATING 0 PATTERN
277          ;*****
277 000572* 005167 177304          3S:  COM  ACSR ;COMPLEMENT EXPECTED
278 000576* 005177 177424          COM  0DAC1 ;COMPLEMENT DATA
279 000602* 042767 170000 177272   BIC  #170000,ACSR ;MASK EXPECTED DATA
280 000610* 000732          BR   1S ;TEST MORE BITS
281
282 000612*
283 000612* 012767 010000 177262   TSDAC2: MOV #BIT12,ACSR ;LOAD EXPECTED
284 000620* 012777 004000 177402   MOV  #BIT11,0DAC2 ;LOAD DAC2 REGISTER
285 000626* 016767 177376 177244   MOV  DAC2,CSRA ;LOAD BUS ADDRESS
286 000634* 006267 177242          1S:  ASR  ACSR ;SHIFT THE EXPECTED
287 000640* 001043          BNE  4S ;R IF DONE
288 000642* 017767 177362 177234   MOV  0DAC2,ASTAT ;READ THE REGISTER
289 000650* 026767 177226 177226   CMP  ACSR,ASTAT ;COMPARE
290 000656* 001403          BEQ  2S ;R IF SAME
291
292 000660* 104405 000000* 000000   ;*****
293          HDRS,REGIN,NULL ;DAC2 FAILED TO HOLD THE FLOATING 1 PATTERN
294          ;*****
294 000666* 005167 177210          2S:  COM  ACSR ;COMPLEMENT DATA
295 000672* 005177 177332          COM  0DAC2 ;COMPLEMENT DATA IN DAC2
296 000676* 042767 170000 177176   BIC  #170000,ACSR ;MASK OFF UNUSED BITS
297 000704* 017767 177320 177172   MOV  0DAC2,ASTAT ;READ DAC2
298 000712* 026767 177164 177164   CMP  ACSR,ASTAT ;COMPARE
299 000720* 001403          BEQ  3S ;R IF SAME
300
301 000722* 104405 000000* 000000   ;*****
302          HDRS,REGIN,NULL ;DAC2 FAILED TO HOLD THE FLOATING 0 PATTERN
303          ;*****
303 000730* 005167 177146          3S:  COM  ACSR ;COMPLEMENT EXPECTED
304 000734* 005177 177270          COM  0DAC2 ;COMPLEMENT DATA
305 000740* 042767 170000 177134   BIC  #170000,ACSR ;MASK EXPECTED DATA
306 000746* 000732          BR   1S ;TEST MORE BITS
307
308 000750*          4S:
```

```
309 000750 012767 010000 177124 TSDAC3: MOV #RIT12,ACSR ;LOAD EXPECTED
310 000756 012777 004000 177246 MOV #RIT11,0DAC3 ;LOAD DAC3 REGISTER
311 000764 016767 177242 177106 MOV DAC3,CSRA ;LOAD BUS ADDRESS
312 000772 006267 177104 15: ASR ACSR ;SHIFT THE EXPECTED
313 000776 001043 48 BNE ;BR IF DONE
314 001000 017767 177226 177076 MOV #0AC3,ASTAT ;READ THE REGISTER
315 001006 026767 177070 177070 CMP ACSR,ASTAT ;COMPARE
316 001014 001403 28 REQ ;BR IF SAME
317
318 001016 104405 000000 000000 ;*****
HRDERS,BEGIN,NULL ;DAC3 FAILED TO HOLD THE FLOATING 1 PATTERN
319 ;*****
320 001024 005167 177052 28: COM ACSR ;COMPLEMENT DATA
321 001030 005177 177176 COM 0DAC3 ;COMPLEMENT DATA IN DAC3
322 001034 042767 170000 177040 BIC #170000,ACSP ;MASK OFF UNUSED BITS
323 001042 017767 177164 177034 MOV #0AC3,ASTAT ;READ DAC3
324 001050 026767 177026 177026 CMP ACSR,ASTAT ;COMPARE
325 001056 001403 38 REQ ;BR IF SAME
326 ;*****
327 001060 104405 000000 000000 HRDERS,BEGIN,NULL ;DAC3 FAILED TO HOLD THE FLOATING 0 PATTERN
328 ;*****
329 001066 005167 177010 38: COM ACSR ;COMPLEMENT EXPECTED
330 001072 005177 177134 COM 0DAC3 ;COMPLEMENT DATA
331 001076 042767 170000 176776 BIC #170000,ACSR ;MASK EXPECTED DATA
332 001104 000732 15 BR ;TEST MORE BITS
333
334 001106 48:
335
336 ;TEST FOR DUAL ADDRESSING
337
338 001106 012777 000000 177110 DUAL: MOV #0,0DAC0 ;LOAD DAC 0
339 001114 012777 002525 177104 MOV #2525,0DAC1 ;LOAD DAC 1
340 001122 012777 005252 177100 MOV #5252,0DAC2 ;LOAD DAC 2
341 001130 012777 007777 177074 MOV #7777,0DAC3 ;LOAD DAC 3
342
343 001136 016767 177062 176734 MOV DAC0,CSRA ;LOAD DAC 0 BUS ADDRESS
344 001144 017767 177054 176732 MOV #0AC0,ASTAT ;READ DAC 0
345 001152 012767 000000 176722 MOV #0,ACSR ;LOAD EXPECTED
346 001160 026767 176716 176716 CMP ACSR,ASTAT ;COMPARE
347 001166 001403 15 REQ ;BR IF SAME
348 ;*****
349 001170 104405 000000 000000 HRDERS,BEGIN,NULL ;DUAL ADDRESS ERROR ON DAC 0
350 ;*****
351
352 001176 016767 177024 176674 15: MOV DAC1,CSRA ;LOAD DAC 1 BUS ADDRESS
353 001204 017767 177016 176672 MOV #0AC1,ASTAT ;READ DAC 1
354 001212 012767 002525 176662 MOV #2525,ACSP ;LOAD EXPECTED
355 001220 026767 176656 176656 CMP ACSR,ASTAT ;COMPARE
356 001226 001403 28 REQ ;BR IF SAME
357 ;*****
358 001230 104405 000000 000000 HRDERS,BEGIN,NULL ;DUAL ADDRESS ERROR ON DAC 1
359 ;*****
360
```

```
361 001236 016767 176766 176634 28: MOV DAC2,CSRA ;LOAD DAC 2 BUS ADDRESS
362 001244 017767 176760 176632 MOV #0AC2,ASTAT ;READ DAC 2
363 001252 012767 005252 176622 MOV #5252,ACSR ;LOAD EXPECTED
364 001260 026767 176616 176616 CMP ACSR,ASTAT ;COMPARE
365 001266 001403 38 REQ ;BR IF SAME
366 ;*****
367 001270 104405 000000 000000 HRDERS,BEGIN,NULL ;DUAL ADDRESS ERROR ON DAC 2
368 ;*****
369
370 001276 016767 176730 176574 38: MOV DAC3,CSRA ;LOAD DAC 3 BUS ADDRESS
371 001304 017767 176722 176572 MOV #0AC3,ASTAT ;READ DAC 3
372 001312 012767 007777 176562 MOV #7777,ACSR ;LOAD EXPECTED
373 001320 026767 176556 176556 CMP ACSR,ASTAT ;COMPARE
374 001326 001403 48 REQ ;BR IF SAME
375 ;*****
376 001330 104405 000000 000000 HRDERS,BEGIN,NULL ;DUAL ADDRESS ERROR ON DAC 3
377 ;*****
378
379 001336 48:
380
381 001336 006367 176672 DONE: ASL TEMP ;CHANGE UNIT POINTER
382 001342 022767 000400 176664 CMP #RIT0,TEMP ;TEST IF LAST UNIT
383 001350 001404 38 REQ ;BR IF YES
384 001352 036767 176656 176434 BIT #RIT0,DVID1 ;TEST IF UNIT IS SELECTED
385 001360 001006 15 BNE ;BR IF UNIT IS TO BE TESTED
386 001362 012767 000001 176644 38: MOV #RIT0,TEMP ;RELOAD UNIT POINTER
387 001370 104413 000000 ENDIRS,BEGIN ;SIGNAL END OF ITERATION,
388 ;MONITOR SHALL TEST END OF PASS
389 28 BR ;BR IF NOT DONE
390 001376 062767 000010 176620 15: ADD #10,0DAC0 ;UPDATE ADDRESS
391 001404 062767 000010 176614 ADD #10,0DAC1
392 001412 062767 000010 176610 ADD #10,0DAC2
393 001420 062767 000010 176604 ADD #10,0DAC3
394 001426 000167 176664 JMP TSDAC0
395 001432 000167 176606 28: JMP LOOPA
396
397 000001 .END
```



SVR6	000076R	191#					
SYSCNT	000057R	100#					
TEMP	000234R	220#	223*	301*	302	304	306*
TRPDFD	000022	212#					
TSDAC0	000316R	231#	394				
TSDAC1	000454R	257#					
TSDAC2	000612R	283#					
TSDAC3	000750R	309#					
VECTOR	000010R	161#					
WASADR	000104R	195#					
WDFR	000116R	202#					
WDT0	000114R	201#					
XFLAG	000005R	159#					

. ABS. 000000 000  
001436 001

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

XMNDA0, XMNDA0/SOL/CRF; SYN=DDXCOM, XMNDA0  
RUN-TIME: 1 1 .2 SECONDS  
RUN-TIME RATIO: 00/3=23.0  
CORE USED: 7K (13 PAGES)