

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E703I-MC
PRODUCT NAME: CXTMAIO TMI1 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

MAIN DEC CHANGE NOTICE
MAY BE REQUIRED FOR
PROGRAM TO OPERATE

1. ABSTRACT

TMA IS AN IOMODX THAT EXERCISES UP TO 8 TAPE DRIVES ON AN
TM11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES,
BACKSPACES, READS, AND IN-CORE COMPARISONS.
ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 TAPE DRIVES WITH A TM11 CONTROLLER
STORAGE: TMA REQUIRES:
1. DECIMAL WORDS: 950
2. OCTAL WORDS: 1666
3. OCTAL BYTES: 3554

3. PASS DEFINITION

ONE PASS OF THE TMA MODULE CONSISTS OF 256 CYCLES OF THE BASIC TEST
SEQUENCE (WRITES BACKSPACE, READ, DATA-CHECK). THE TEST SEQUENCE
WRITES 1024 WORDS, BACKSPACES SAME, READS THE FIRST 256 WORDS,
AND DATA-CHECKS SAME.

4. EXECUTION TIME

ONE PASS OF TMA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.
CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172520, VECTOR: 224, RPI: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

7. MODULE OPERATION

TEST SEQUENCE:

- A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
- B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
- C. GET A FRESH BLOCK OF DATA
- D. GET A DRIVE ADDRESS
- E. DO A WRITE -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
- F. DO A BACKSPACE -- IF ERRORS, REPORT
- G. DO A READ -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
- H. DO A DATA-CHECK -- IF ERRORS, REPORT AND CONTINUE
- I. IF END OF PASS, REPORT AND GO TO C
- J. IF END OF DRIVES, GO TO C ELSE GO TO D
- K. IF END OF TAPE, REWIND ALL DRIVES AND GO TO B

OPERATION OPTIONS

8.

SRI BIT 0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR
IS ASSUMED AND THE DRIVE IS DROPPED

SRI BIT 0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED, THE FUNCTION IS ABORTED AND
THE TESTING CONTINUES

9. NON-STANDARD PRINTOUTS

- A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN
THE DEC/X11 DOCUMENT
- B. ERROR MESSAGES DUMP THE CONTENTS OF THE 6 TM11 REGISTERS
AND THE CYCLE COUNT IN THE FOLLOWING ORDER:
MTS MTC MTBPC MTCMA MTD MTRD MTRD CYCLE COUNT
- C. THE CYCLE COUNT LOCATION IN THE ERROR MESSAGE CONTAINS THE
CYCLE COUNT AT THE TIME OF THE ERROR. THIS SHOULD
AID IN NOTING ANY BAD SPOTS ON A TAPE.

```

000000- IOMODY <TMAI > 172520 224 5 0 0 256 22 BUFIN, 256, 1024.
000000- MODULE 150000 TMAI 172520 224 5 0 0 256 22 BUFIN, 256, 1024.
; .TITLE TMAI DEC/111 SYSTEM EXERCISER MODULE
; DDSCOM VERSION 6 23-MAY-78
*****LIST BIN*****
000000- BEGIN:
000000- 046524 044501 040 MODNAM: .ASCII /TMAI / ;MODULE NAME.
000000- 000000 000000 XFLAG: .BYTE OPEN ;USED TO KEEP TRACK OF WBUFV USAGE
000000- 172520 ADDR: 172520+0 ;1ST DEVICE ADDR.
000000- 006224 VECTUR: 22420 ;1ST DEVICE VECTOR.
000000- 240 BR1: .BYTE PRTV5+0 ;1ST BR LEVEL.
000000- 000000 BR2: .BYTE PRTV0+0 ;2ND BR LEVEL.
000000- 000001 BYTD1: 041 ;DEVICE INDICATOR 1.
000000- 000000 SR1: OPEN ;SWITCH REGISTER 1.
000000- 000000 SR2: OPEN ;SWITCH REGISTER 2.
000000- 000000 SR3: OPEN ;SWITCH REGISTER 3.
000000- 000000 SR4: OPEN ;SWITCH REGISTER 4.
*****LIST BIN*****
000000- STAT: 150000 ;STATUS WORD.
000000- 150000 SPOINT: PART ;MODULE START ADDR.
000000- 000252 MODSP: MODSP ;MODULE STACK POINTER.
000000- 000000 PASCNT: 0 ;PASS COUNTER.
000000- 000256 ICOUNT: 256 ;# OF ITERATIONS PER PASS=256
000000- 000000 SCOUNT: 0 ;LOC TO COUNT ITERATIONS
000000- 000000 HRDCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
000000- 000000 SOFPAS: 0 ;LOC TO SAVE TOTAL HARD ERRORS
000000- 000000 SRSCNT: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
000000- 000000 RANNUM: 0 ;LOC TO SAVE HARD ERRORS PER PASS
000000- 000000 CONFIG: 0 ;# OF SYS ERRORS ACCUMULATED
000000- 000000 RES: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000000- 000000 RES1: 0 ;RESERVED FOR MONITOR USE
000000- 000000 SVR0: OPEN ;RESERVED FOR MONITOR USE
000000- 000000 SVR1: OPEN ;LOC TO SAVE R0.
000000- 000000 SVR2: OPEN ;LOC TO SAVE R1.
000000- 000000 SVR3: OPEN ;LOC TO SAVE R2.
000000- 000000 SVR4: OPEN ;LOC TO SAVE R3.
000000- 000000 SVR5: OPEN ;LOC TO SAVE R4.
000000- 000000 SVR6: OPEN ;LOC TO SAVE R5.
000000- 000000 CSRA: OPEN ;LOC TO SAVE R6.
000000- 000000 SBADR: ;ADDR OF CURRENT CSR.
000000- 000000 ACSR: OPEN ;ADDR OF GOOD DATA, OR
000000- 000000 WASADR: OPEN ;CONTENTS OF CSR.
000000- 000000 ASTAT: OPEN ;ADDR OF BAD DATA, OR
000000- 000000 ERRTP: OPEN ;STATUS REG CONTENTS.
000000- 000000 ASB: OPEN ;TYPE OF ERROR.
000000- 000000 AWAS: OPEN ;EXPECTED DATA.
000000- 000000 RSTRT: RESTART ;ACTUAL DATA.
000000- 000000 WDTC: OPEN ;RESTART ADDRESS AFTER END OF PASS
000000- 000000 WDFR: OPEN ;WORDS TO MEMORY PER ITERATION
000000- 000000 INTR: OPEN ;WORDS FROM MEMORY PER ITERATION
000000- 00022 IDNUM: 22 ;# OF INTERRUPTS PER ITERATION
;MODULE IDENTIFICATION NUMBER=22

```

```

000124- 002404 RBUFVA: BUFIN ;READ BUFFER VIRTUAL ADDRESS
000126- 000000 RBUFPA: OPEN ;READ BUFFER PHYSICAL ADDRESS
000130- 000000 RBUFEA: OPEN ;READ BUFFER EA BITS
000132- 000000 RBUSZ: 256 ;SIZE OF THE READ BUFFER
000134- 000000 WBUFVA: OPEN ;WRITE BUFFER VIRTUAL ADDRESS
000136- 000000 WBUFPA: OPEN ;WRITE BUFFER PHYSICAL ADDRESS
000140- 002000 WBUFEA: OPEN ;WRITE BUFFER EA BITS
000142- 000000 WBUSZ: 1024 ;WRITE BUFFER SIZE REQUESTED
000144- 000000 CDERC: OPEN ;WRITE BUFFER SIZE AVAILABLE
000146- 000000 CDWDC: OPEN ;CDATA/DATCK ERROR COUNT
000150- 000000 FREE: OPEN ;CDATA/DATCK WORD COUNT
;RESERVED FOR FUTURE USE
;REPT SPSIZ ;MODULE STACK STARTS HERE.
;LIST 0
;LIST
;ENDR
000252- MODSP:
*****LIST BIN*****

```

```

216
217 000252- 012767 000003 177640 START: MOV #3,INTR ;3 INTERRUPTS PER ITERATION
218 000260- 012767 002000 177630 JSR #10,WDPR ;1024 WORDS PER ITERATION FROM MEM
219 000266- 012767 000400 177620 MOV #256,WOTO ;256 WORDS TO MEM PER ITERATION
220 000300- 015967 003250 CLR FLAG ;CLEAR FLAG
221 000306- 015767 002510 MOV DVID1,DVICE ;GET DRIVE INDICATOR
222 000306- 015767 002050 MOV DVICE,DRIVE ;ALSO SAVE IT IN DRIVE
223 000314- 015067 002042 CLR DRIVE ;ZERO UNIT NUMBER
224 000320- 012767 177400 MOV #400,DRVSFT ;INITIALIZE THE SHIFTED DRIVE #
225
226
227 000326- 132737 000004 000041 %FIND IF TM IS LOAD MEDIUM, IF SO, DROP LOAD DRIVE IF SELECTED
228 000334- 001421 BITB #BIT2,@#41 ;IS TM THE LOAD MEDIUM?
229 000336- 113700 BEQ YES ;NO, CONTINUE
230 000342- 013701 MOVB @#40,R0 ;GET LOAD DRIVE #
231 000346- 135700 MOV #4,R1 ;SET UP DRIVE MASK
232 000350- 001403 TSTB R0,R1 ;HAVE LOAD DRIVE YET?
233 000352- 006301 BEQ YES ;YES, GO DROP IT
234 000354- 105300 ASL R1 ;NO, SHIFT DRIVE MASK
235 000356- 000773 BR R0 ;COUNT SHIFTS, LOAD DRIVE MASK
236 000360- 030167 001776 3S: BIT R1,DVICE ;LOAD DRIVE SELECTED?
237 000366- 004767 JSR PC,DRP ;NO, CONTINUE
238 000372- 104403 MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
239
240
241 000400- 012767 177777 001760 1S: MOV #-1,DRIVE ;INITIALIZE DRIVE COUNTER
242 000406- 004767 JSR PC,SETUP ;GENERATE REGISTER ADDRESSES
243 000412- 004767 JSR PC,SETZET ;INITIALIZE #N REGS. AND ALL DRIVES
244 000415- 005767 TST DVICE ;DROP THE MODULE?
245 000424- 004767 BEQ YES ;YES
246 000430- 005067 JSR PC,REWIND ;REWIND ALL DRIVES
247 000434- 000403 CLR RR ;ZERO TOTAL CYCLE COUNTER
248 000436- 005767 RR CONT
249 000442- 001703 RESTR: TST PASCNT ;ANY PASSES MADE YET?
250 000444- 001703 BEQ START ;NO BRANCH
251
252 000444- 104415 000000- 000124- CONT: GETPAS,REGIN,RRUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
253 000452- 016767 177454 001722 MOV RBUFVZ,WCNT2 ;SAVE READ BUFFER SIZE
254 000460- 005467 NEG WCNT2 ;GET THE 2'S COMPLEMENT
255 000484- 006367 ASL WCNT2 ;DOUBLE IT TO GET A BYTE COUNT
256
257
258 000470- 104414 000000- 001676 STRT: CHWBUFZ,BEGIN ;GET WRITE BUFFER INFORMATION
259 000474- 016767 MOV RBUFVZ,WCNT1 ;SAVE WRITE BUFFER SIZE
260 000502- 005467 NEG WCNT1 ;GET THE 2'S COMPLEMENT
261 000506- 006367 ASL WCNT1 ;DOUBLE IT TO GET A BYTE COUNT
262
263
264 000512- 004767 001060 NEXT: JSR PC,DRVADR ;GET A DRIVE ADDRESS
265 000516- 005767 TST DVICE ;ANY DRIVES LEFT?
266 000524- 001473 BEQ FINI ;NO, GO DROP THE MODULE
267 000532- 001356 BITB #BIT3,FLAG ;ALL DRIVES DONE?
268 000534- 042777 BNE YES ;YES, GO GET ANOTHER BLOCK
269 000540- 056777 JSR PC,NOTRDY ;CLEAR OUT UNIT NUMBER
270 000550- 032777 RR DRVSFT,@MTC ;LOAD SELECTED DRIVE NUMBER
271 000554- 001406 BIT #BIT2,@MTC ;WRITE PROTECTED?

```

```

272 000560- 004767 000754 JSR PC,DRP ;YES, DROP THE DRIVE
273 000564- 104403 000000- 003530- RR MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
274 000574- 000747 NEXT ;GO ON TO NEXT DRIVE
275 000574- 032777 BIT #BIT0,@MTC ;DRIVE READY?
276 000602- 001003 BNE YES ;YES, CONTINUE
277 000604- 004767 JSR PC,NOTRDY ;NO, WAIT FOR READY
278 000610- 006727 RR STRT ;TRY AGAIN
279 000612- 005067 CLR TRYL ;ZERO RETRY COUNTERS

```

```

280
281
282 000616- 004567 000222 RITE: JSR R5,WRITE ; WRITE SOME DATA
283 000625- 004735 ; IF ERRORS, TRY IT AGAIN
284 000624- 132767 000004 002716 BR TRV1 ; DID THE TAPE REACH EOT?
285 000633- 001416 ; NO, TO EOT ROUTINE
286 000633- 142767 000004 002706 RITEOT: BICB #RIT2,FLAG ; COUNT THE RETRYS
287 000643- 004767 000404 ; LIMIT EXCEEDED?
288 000644- 005067 001524 ; YES, GO REPORT IT
289 000652- 000706 ; NO, BACKUP TO TRY AGAIN
290 000654- 004567 000216 BACK: JSR R5,BACKSP ; REWIND ALL DRIVES
291 000663- 000240 ; ZERO TOTAL CYCLE COUNTER
292 000664- 004567 000226 REED: NOP ; START OVER AT BEGINNING OF TAPES
293 000666- 000440 ; BACKSPACE THE DRIVE
294 000670- 104412 000000- 000126- BR RETRY2 ; ERROR RETURN
295 000676- 000700 ; READ THE DATA WRITTEN
; CDATA$,BEGIN,RRUPFA ; IF ERRORS, TRY AGAIN
; +2 ; REQUEST FOR MONITOR TO CHECK DATA
; IF ERROR, CONTINUE
296
297
298 000700- 005267 001472 INC CYCKNT ; ADD 1 TO TOTAL CYCLE COUNT
299 000704- 104413 000000- ENDITS,BEGIN ; SIGNAL END OF ITERATION.
300 000710- 000700 BR NEXT ; MONITOR SHALL TEST END OF PASS
301 ; NO, CONTINUE
302
303 000712- FINI: 000000- ; ENDS,BEGIN ; DROP THE MODULE
304 000712- 104410 000000- ;
305
306
307
308 000716- 132767 000004 002624 RETRY1: BITR #RIT2,FLAG ; TAPE REACH EOT?
309 000724- 001343 ; YES - TO EOT ROUTINE
310 000732- 105267 002620 INCB TRV1 ; COUNT THE RETRYS
311 000732- 12767 000003 002612 CMPR #3,TRV1 ; LIMIT EXCEEDED?
312 000742- 004567 000130 JSR R5,BACKSP ; YES, GO REPORT IT
313 000750- 000722 ; NO, BACKUP TO TRY AGAIN
314 000752- 104403 000000- 003514- 1$: NOP RITE ; ERROR RETURN
315 000752- 104403 ; ASCII MESSAGE CALL WITH COMMON HEADER
316 000764- 000240 JSR R5,WRITE ; SKIP SOME TAPE, WRITE WITH EXTENDED IRC
317 000766- 000413 BR NEXT ; ERROR RETURN
318 ; GO ON TO NEXT DRIVE
319
320 000770- 105267 002557 RETRY2: INCB TRV2 ; COUNT RETRYS
321 000774- 12767 000003 002551 CMPB #3,TRV2 ; LIMIT EXCEEDED?
322 001004- 000723 ; YES, GO REPORT IT
323 001006- 104403 000000- 003522- 1$: BR BACK ; NO, BACKUP TO TRY AGAIN
324 001014- 000240 ; ASCII MESSAGE CALL WITH COMMON HEADER
325 001014- 000240 ; GO ON TO NEXT DRIVE
326
327
328
329
330
331

```

```

331
332
333 001016- 032767 000001 176772 NEXTA: BIT #BIT0,SRI ; DROP THE DRIVE?
334 001024- 001405 ; NO, SKIP TO NEXT DRIVE
335 001025- 004767 000506 JSR PC,DROP ; YES, DROP OFFENDING DRIVE
336 001040- 000167 177448 ; ASCII MESSAGE CALL WITH COMMON HEADER
337 ; GO ON TO NEXT DRIVE
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377

```

----- TMI1 TAPE DRIVERS -----

```

001044- 012767 060105 001304 WRITE: MOV #60105,FUNC ; LOAD WRITE FUNCTION
001052- 016777 001322 002334 MOV MCNT1,&MTBRC ; LOAD BYTE COUNT
001060- 016777 177050 002330 MOV WRUPFA,&MTCMA ; LOAD BUFFER ADDRESS
001065- 216767 177044 001264 MOV WRUPEA,XMEM ; LOAD EXTENDED MEMORY BITS
001074- 000440 BR ; CONTINUE
001076- 000440 GOGO ; LOAD BACKSPACE FUNCTION
001104- 012767 060113 001252 MOV #60113,FUNC ; LOAD BYTE COUNT
001112- 000431 MOV #1,&MTBRC ; CONTINUE
001114- 000431 BR ; CONTINUE
001117- 012767 060103 001234 READ: MOV #60103,FUNC ; LOAD READ FUNCTION
001117- 012767 001254 002264 MOV MCNT1,&MTBRC ; LOAD BYTE COUNT
001130- 016777 176772 002260 MOV RUPFA,&MTCMA ; LOAD BUFFER ADDRESS
001136- 016767 176766 001214 MOV RRUPEA,XMEM ; LOAD EXTENDED MEMORY BITS
001144- 000440 BR ; CONTINUE
001148- 000440 GOGO ; LOAD WRITE EXTENDED FUNCTION
001154- 016777 060115 001202 MOV #60115,FUNC ; LOAD BYTE COUNT
001162- 016777 176746 002226 MOV WRUPFA,&MTCMA ; LOAD BUFFER ADDRESS
001170- 016767 176742 001162 MOV WRUPEA,XMEM ; LOAD EXTENDED MEMORY BITS
361
362 001176- 012777 001232- 176604 GOGO: MOV #NTRUPT,&VECTOR ; SET INTERRUPT ENTRY POINTER
363 001203- 026767 001160 001144 BTS DRVSFT,FUNC ; LOAD DRIVE UNIT NUMBER
364 001220- 016777 001132 001124 BTS XMEM,FUNC ; LOAD EXTENDED MEMORY BITS
365 001226- 104400 000000- EXECUTE THE FUNCTION
366 ; EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
367
368 001232- NTRUPT:
369
370 001232- 000004 000000- 001240- ; TRQS,BEGIN,1$ ; QUEUE UP TO CONTINUE AT 1$ AND RTI
371 ;
372
373 001240- 004567 060576 1$: JSR R5,ERRORS ; GO CHECK FOR ERRORS
374 001244- 000205 RTS R5 ; ERRORS DETECTED, RETURN
375 001250- 002728 TST (R5)+ ; NO ERRORS, SKIP RETRY
376 001250- 002728 RTS R5 ; RETURN OR
377 ;

```

```

378
379
380 001252- 012767 001104 001104 REWIND: MOV DVICE,DRIVE ; GET ACTIVE DRIVES
381 001266- 012767 001100 001100 MOV #8,DVCNUM ; LOAD MAXIMUM NUMBER OF DRIVES
382 001266- 012767 001100 001100 MOV DVCNUM,R1 ; PUT IT INTO A COUNTER
383
384 001272- 002241
385 001272- 002241
386 001300- 005367 001064 1$: CLC ; MAKE SURE C-BIT IS CLEAR
387 001302- 005367 001064 BCS ZS ; CHECK FOR ACTIVE DRIVE
388 001306- 005301 001064 2$: DEC R1 ; IT'S ACTIVE --- BRANCH
389 001310- 005370 BGT 1$ ; NOT ACTIVE SUBTRACT FROM TOTAL
390 ; NO, CONTINUE
391
392 001312- 012767 000100 002066 MOV #100,CLK1 ; LOAD THE 2ND TIMER
393 001320- 012767 177777 001040 MOV #1,DRIVE ; INITIALIZE THE DRIVE COUNTER
394 001330- 005967 001024 001024 CLR DVCNT ; CLEAR DRIVE COUNTER FOR ISR.
395 001340- 012767 177400 001022 MOV DVICE,DRIVE ; RESTORE DRIVE INDICATOR
396 001340- 004767 000224 3$: MOV #400,DRVSFT ; INITIALIZE SHIFTED DRIVE NUMBER
397 001350- 132767 000610 002170 PC,DRVADR ; GO GET A DRIVE NUMBER
398 001360- 001017 BITB #BITS,FLAG ; ALL DRIVES DONE ?
399 001360- 004767 BNE 4$ ; YES, GO WAIT FOR COMPLETION
400 001366- 042777 003400 002016 BIC #3400,&MTC ; CONTROLLER READY ?
401 001370- 012767 000770 000754 MOV DRVSFT,FUNC ; YES, CLEAR OUT OLD UNIT NUMBER
402 001374- 012767 000770 000754 MOV #17,FUNC ; LOAD NEW UNIT NUMBER
403 001410- 012767 000774 001774 BJS #17,FUNC ; LOAD REWIND FUNCTION
404 001416- 000753 BR 3$ ; EXECUTE THE REWIND
405 ; GO ON FOR THE NEXT DRIVE
406
407 001420- 012767 077777 001756 4$: MOV #77777,CLK ; SET THE TIMER
408 001426- 004767 JSR PC,DRVADR ; GO GET A DRIVE NUMBER
409 001430- 132767 000010 002110 BITB #BITS,FLAG ; ALL DRIVES DONE ?
410 001440- 001036 BNE 4$ ; YES, GET OUT
411 001450- 056777 003400 001744 BIC #3400,&MTC ; CLEAR OUT OLD UNIT NUMBER
412 001456- 000010 001744 6$: BJS DRVSFT,&MTC ; LOAD NEW UNIT NUMBER
413
414 001456- 104407 000000- BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR.
415 001466- 032777 000041 001714 BREAKS,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
416 001474- 001354 BNE #1,&MTC ; DRIVE READY AND AT BOT ?
417 001476- 005367 001702 DEC CLK ; YES, GO CHECK THE NEXT DRIVE
418 001478- 005367 001676 BGT CLK ; NO, OUT OF TIME
419 001504- 005367 001676 BGT CLK ; NO, WAIT SOME MORE
420 001510- 001407 BEQ #40,CLK ; YES, WAIT ANOTHER 40 SECONDS ?
421 001512- 012767 077777 001664 MOV #77777,CLK ; RESET THE TIMER
422 001522- 006767 BR 7$ ; TIME-OUT
423 001522- 012767 000013 176356 MOV #13,ERRTYP ; REWIND ERROR
424 001530-
425 001530- 104405 000000- 000000 ;*****
426 001536- 000207 8$: RTS PC ; RETURN
427 ;*****
428 ;*****
429 ;*****
430 ;*****
431 ;*****
432 ;*****
433 ;*****
  
```

```

434 001540- 012701 000001 DROP: MOV #1,R1 ; INITIALIZE DROP PICKER
435 001550- 001103 000516 MOV #1,DRIVE,RO ; GET THE DRIVE NUMBER
436 001552- 006301 1$: BEQ ZS ; IF DRIVE 0 GO DROP IT
437 001554- 005300 DEC RO ; IS THIS THE NEXT DRIVE
438 001554- 005300 BNE 1$ ; NO, TAKE THE ONE ?
439 001560- 040172 000576 2$: BNE 1$ ; NO, LOOK AGAIN
440 001560- 040172 BIC #1,DRVICE ; DROP THE DRIVE
441 ;*****
442 ;*****
443 ;*****
444 001564- 104420 000000- 002366- OTOAS,BEGIN,DRIVE,ADRI ; CONVERT DRIVE TO ASCII AND
445 001572- 003540- ; STORE AT ADRI
446 ;*****
447 ;*****
448 ;*****
449 ;*****
450 ;*****
451 001576- 005267 000564 DRVADR: INC DRIVE ; COUNT A DRIVE
452 001582- 062767 000400 000560 ADD #BITS,DRVSFT ; DRIVE COUNT LINED UP WITH MTC
453 001610- 142767 000010 000732 BITB #BITS,FLAG ; CLEAR END OF DRIVES FLAG
454 001616- 022767 000010 000542 CMP #8,DRIVE ; ALL DRIVES CHECKED ?
455 001624- 001404 BEQ 1$ ; YES, GO FLAG END OF DRIVES
456 001626- 006264 BCS ZS ; NO, IS NEXT DRIVE CHOSEN ?
457 001632- 103361 BPC DRVADR ; NO, GO TRY ANOTHER DRIVE
458 001634- 000207 RTS PC ; RETURN
459
460 001636- 152767 000010 001704 1$: BITB #BITS,FLAG ; SET END OF DRIVES FLAG
461 001644- 012767 177777 000514 MOV #1,DRIVE ; RESET DRIVE COUNTER
462 001652- 012767 177400 000510 MOV #400,DRVSFT ; ZERO THE SHIFTED DRIVE #
463 001656- 000476 000476 000476 MOV DVICE,DRIVE ; RESTORE CHOSEN DRIVES
464 ; RETURN
465
466
467
468 001670- 012767 177777 000470 NOTRDY: MOV #1,DRIVE ; START WITH FIRST DRIVE
469 001676- 012767 177400 000464 MOV #400,DRVSFT ;
470 001704- 016767 000452 MOV DVICE,DRIVE ; RESET DRIVE SELECT
471 001716- 004767 177460 000452 JSR PC,DRVADR ; GET A DRIVE ADDRESS
472 001724- 001012 000010 001624 1$: BITB #BITS,FLAG ; ALL DRIVES CHECKED ?
473 001726- 016777 BNE 1$ ; YES, RETURN
474 001730- 000436 001456 MOV DRVSFT,&MTC ; NO, LOAD NEXT DRIVE ADDRESS
475 001740- 002767 000001 001446 BIT #BIT0,&MTC ; IS THIS DRIVE READY ?
476 001744- 004767 BNE 1$ ; YES, CONTINUE
477 001750- 000760 BR 1$ ; NO, WAIT FOR IT
478 001752- 000207 2$: RTS PC ; GO CHECK REST OF DRIVES
479 ; RETURN
480
481
482
483
484 001754- 016767 001430 176116 ERSUB1: MOV #MTC,CSRA ; LOAD ADDR OF CURRENT CSR
485 001762- 017767 001422 176116 MOV #MTC,ACSR ; LOAD CONTENTS OF CURRENT CSR
486 001770- 000207 RTS PC ; RETURN
487
  
```

```

488
489
490
491
492 001777 012767 077777 001404 WAIT: MOV #77777,CLK ; SET THE TIMER
493 002000 ; YES, RETURN
494 002000 104407 000000 ; TEMPORARY RETURN TO MONITOR...
495 002004 104407 000000 ; THEN CONTINUE AT NEXT INSTRUCTION.
496
497 002010 032777 000001 001372 BIT #BIT0,@MTS ; DRIVE READY ?
498 002016 001010 ; YES, RETURN
499 002020 003367 001360 BNE CLK ; NO, WAIT SOME MORE ?
500 002024 001365 ; YES, WAIT
501 002026 004767 177506 ; TIME-OUT, DROP THE DRIVE
502 002032 104403 000000 003530 JSR PC,DRP ; ASCII MESSAGE CALL WITH COMMON HEADER
503 002040 006207 ; RETURN
504
505
506
507
508
509 002042 004767 177706 ERRORS: JSR PC,ERSUB1 ; LOAD ERROR INFORMATION
510 002046 005777 001340 TST @MTC ; ANY ERRORS ?
511 002050 009047 ; NO, RETURN OK
512 002054 001414 RPL #BIT9,@MTS ; RECORD LENGTH ERROR ?
513 002062 001414 ; NO, CONTINUE
514 002064 032777 000002 001320 BIT #BIT1,@MTC ; WAS THE COMMAND A READ ?
515 002074 001414 ; NO, REPORT THE ERROR
516 002102 001430 ; YES, RLE ERR EXPECTED, CHECK OTHER ERRORS
517 002104 042767 001000 175770 BIC #BIT9,ACSR ; RETURN OK
518 002114 000435 ; ELIMINATE RLE BIT FROM
519 002122 001411 002000 001266 1$: RR ; ERROR REPORT & GO REPORT IT
520 002124 152767 000004 001416 BIT #BIT10,@MTS ; AT END OF TAPE ?
521 002130 001417 174600 001250 BIC #BIT2,FLAG ; YES, SET EOT FLAG
522 002142 005067 175740 ; NO, TEST OTHER ERRORS
523 002146 ; NONE RETURN OK
524 ; UNKNOWN ERROR
525
526
527
528 002146 104405 000000 003410 *****
529 002154 012777 010000 001230 HDRS,BEGIN,TABLE ;
530 002162 006404 BR 55 ; ISSUE A POWER CLEAR
531 ; RETURN TO DO A RETRY
532
533 002164 012777 010000 001220 3$: MOV #BIT12,@MTC ; ISSUE A POWER CLEAR
534 002172 005725 ; SKIP THE RETRY
535 002174 006205 ; RETURN
536

```

```

537
538
539 002176 016700 175604 SETUP: MOV ADDR,R0 ; GET DEVICE ADDRESS
540 002202 010067 001292 R0,MTC ; GENERATE CONTROLLER REGS. ADDRESSES
541 002206 005720 TST (R0)+
542 002210 010067 001176 MOV R0,MTC
543 002214 005720 TST (R0)+
544 002216 010067 001172 MOV R0,@BRC
545 002222 005720 TST (R0)+
546 002230 010067 001166 MOV R0,@MTCMA
547 002234 005720 TST (R0)+
548 002236 010067 001162 MOV R0,@MTD
549 002240 010067 001156 TST (R0)+
550 ; GO DROP THE MODULE
551
552 002244 016700 175540 MOV VECTOR,R0 ; GET THE VECTOR ADDRESS
553 002250 116760 175536 MOV#R,B1,2(R0) ; SET PRIORITY
554
555 002256 000207 2$: RTS PC ; RETURN
556
557
558
559
560
561 002260 012777 010000 001124 REZET: MOV #BIT12,@MTC ; EXECUTE POWER CLEAR
562 002266 004767 000000 JSR PC,WAIT1 ; GO WAIT FOR CONTROLLER READY
563 002270 004767 177372 JSR PC,NOTRDV ; MAKE SURE ALL CHOSEN DRIVES ARE READY
564 002276 006207 ; RETURN
565
566
567
568
569
570
571 002300 012767 077777 001076 WAIT1: MOV #77777,CLK ; SET THE TIMER
572 002312 100417 ; CONTROLLER READY ?
573 002314 104407 000000 ; YES, CONTINUE
574 002320 104407 000000 BREAKS,BEGIN ; TEMPORARY RETURN TO MONITOR...
575 002324 003367 001054 BREAKS,BEGIN ; THEN CONTINUE AT NEXT INSTRUCTION.
576 002330 001366 ; WAIT SOME MORE ?
577 002332 012767 000003 175546 BNE #3,ERRTYP ; YES
578 ; CONTROLLER NOT READY
579 *****
580 HDRS,BEGIN,TABLE ; CONTROLLER NOT READY
581 *****
582 002346 000167 176340 JMP FNI ; GO DROP THE MODULE
583 002352 006207 2$: RTS PC ; READY, RETURN
584

```



```

585 002354* 000000
586 002355* 000000
587 002356* 000000
588 002357* 000000
589 002358* 000000
590 002359* 000000
591 002360* 000000
592 002361* 000000
593 002362* 000000
594 002363* 000000
595 002364* 000000
596 002365* 000000
597 002366* 000000
598 002367* 000000
599 002368* 000000
600 002369* 000000
601 002370* 000000
602 002371* 000000
603 002372* 000000
604 002373* 000000
605 002374* 000000
606 002375* 000000
607 002376* 000000
608 002377* 000000
609 002378* 000000
610 002379* 000000
611 002380* 000000
612 002381* 000000
613 002382* 000000
614 002383* 000000
615 002384* 000000
616 002385* 000000
617 002386* 000000
618 002387* 000000
619 002388* 000000
620 002389* 000000
621 002390* 000000
622 002391* 000000
623 002392* 000000
624 003514* 003476*
625 003515* 003476*
626 003516* 003476*
627 003517* 003476*
628 003518* 003476*
629 003519* 003476*
630 003520* 003476*
631 003521* 003476*
632 003522* 003476*
633 003523* 003476*
634 003524* 003476*
635 003525* 003476*
636 003526* 003476*
637 003527* 003476*
638 003528* 003476*
639 003529* 003476*
640 003530* 003476*
    
```

```

CNT: 0
FUNC: 0
WFM: 0
DVICE: 0
DRIVE: 0
DRVSFT: 0
DVCNUM: 0
DVCNMT: 0
WCNT1: 0
WCNT2: 0
WUPIN: 0
CLK: 0
TABLE: 0
MTC: 0
MTRPC: 0
MTCMA: 0
MTRD: 0
MTRD: 0
CYCKNT 177777
CYC: 177777
MES2: .ASCIZ " DRIVE "
MES3: .ASCIZ " DROPPED%"
MES4: .ASCIZ " RETPV EXCEEDED%"
MES5: .ASCIZ " WRITE"
MES6: .ASCIZ " READ"
.EVEN
EXCED1: MES5
MES4
EXCED2: MES6
MES4
DRP: MES2
NUMR 177777
ADRI: .PLKB 5
NUMR: .BYTE 0
FLAG: .WORD 0
TRY1: .EVEN 0
TRY2: .BYTE 0
    
```

```

641 .EVEN
642
643 00C001 .END
    
```


PRTV0	=	000000	156#	216#				
PRTV1	=	000040	216#					
PRTV2	=	000100	216#					
PRTV3	=	000140	216#					
PRTV4	=	000200	156#					
PRTV5	=	000240	216#	216#				
PRTV6	=	000300	216#					
PRTV7	=	000340	216#					
PS	=	177776	216#					
PSM	=	177776	216#					
PUSH	=	005746	216#					
PUSH2	=	024646	216#					
RANDS	=	104417	216#					
RANNU	=	00054R	177#					
RBUFEA	=	000130R	200#	354				
RBUFPA	=	000125R	189#	294	353			
RBUFSZ	=	000125R	181#	253				
RBUFVA	=	000124R	188#	252				
READ	=	001114R	292#	351#				
READ	=	000652R	292#					
RES1	=	000346R	193#					
RES2	=	000056R	178#	249#				
RES2	=	000060R	177#					
RETRY1	=	000746R	283#	309#				
RETRY2	=	000746R	283#	323#				
REWIND	=	001252R	446#	387#				
REZET	=	002260R	243#	561#	390#			
RFILE	=	000616R	282#	316				
RSTART	=	000112R	193#	310				
SBADR	=	000102R	186#					
SDFUP	=	000049R	174#	539#				
SDFCNT	=	000049R	174#					
SOPERS	=	104406	216#					
SUPPAS	=	000046R	171#					
SPLIT	=	000037R	165#					
SP1Z	=	000040	165#					
SRI	=	000016R	158#	209				
SRI	=	000020R	155#	333				
SRI	=	000022R	160#					
SRI	=	000024R	161#					
START	=	000252R	184#	217#	250			
START	=	000026R	163#					
START	=	000070R	257#	267	278	289		
SVRO	=	000052R	178#					
SVR1	=	000064R	178#					
SVR2	=	000066R	180#					
SVR3	=	000070R	181#					
SVR4	=	000072R	183#					
SVR5	=	000074R	183#					
SVR6	=	000076R	184#					
SYSCNT	=	000052R	173#					
TABLE	=	013452R	578#	579	601#			
TRPDFD	=	000022	278#					
TRV1	=	003552R	299#	311*	312	639#		
TRY2	=	003553R	323*	324	640#			

VECTOR	=	000010R	154#	362*	552			
WAIT	=	001772R	477#	492#				
WAIT1	=	002300R	399#	562	570#			
WASADR	=	000104R	188#					
WBUFEA	=	000136R	203#	346	359			
WBUFPA	=	000134R	202#	345	358			
WBUFRQ	=	000140R	204#					
WBUFSZ	=	000142R	205#					
WCNT1	=	022406R	259#	260*	344	357	596#	
WCNT2	=	002402R	253#	254*	352	597#		
WDFR	=	000116R	185#	218*				
WDTO	=	000114R	194#	219*				
WRITE	=	001044R	242#	343#				
WRITEX	=	001045R	243#	356#				
XFLAG	=	000005R	159#					
XNEM	=	022360R	346*	354*	364	588#		
.	=	003554R	295	598#	621#	634#	638#	

. ABS. 000000 000
 003554 001

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0
 XTMAIO,XTMAIO/SOL/CRF:SYN=DDXCOM,XTMAIO
 RUN-TIME: 1 2 .3 SECONDS
 RUN-TIME RATIO: 18/4=4.6
 CORE USED: 7K (13 PAGES)

DIAGNOSTIC ENGINEERING

33

digital

DECO DEPO SUBMISSION

NEW CHANGE DELETE FOR RELEASE ENG. USE

PRODUCT IDENTIFICATION																
MD	LIBRARY ZZ	PRODUCT NUMBER CXTMA	REV 1	PATCH 1	ECO TALLY DI	PRODUCT DATE DD MMM YY Nov 17 78	STATUS OBSOLETE	DISTRIBUTION X G R	1ST COPY - RIGHT YEAR 1973	LAST COPY - RIGHT YEAR 1978						
TITLE CXTMAIL TM11 MOUDLE																
AUTHOR D. BUTENHOF			MAINTAINING GROUP DEC/X11 SPT GRP			MAINTAINED BY D. BUTENHOF		SUBMITTING ENGINEER D. BUTENHOF								
PRODUCT COMPONENTS																
CK	DESCRIPTION	PRODUCT NO	REV	CK	DESCRIPTION	PRODUCT NO	REV									
	DOCUMENT				INDEX											
	LISTING				SOURCE MEDIA											
	OBJECT MEDIA				TEST MEDIA											
X	DEPO	AF-E7031-M1														
PRODUCTS OBSOLETE (other than previous version)																
LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV								
MD			MD			MD										
PRODUCT CHARACTERISTICS																
PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2-digit codes representing the Processor the product operates with. See separate instructions.)																
03	04	05	10	20	21	34	35	40	45	50	55	60	70			
OPERATIONAL CODES (Enter all applicable 2-digit codes that describe the product. See separate instructions.)																
02	03	04	06	50												
ACT/APT/XXDP	EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME										
INFORMATION FIELD		1132	Y N	Y N	SECONDS	SECONDS										
DECO/DEPO INFORMATION																
PROBLEM REPORTS CLOSED:																
DEVICE AFFECTED DEC/X11				MULTIMEDIA AFFECTED? <input type="checkbox"/> YES <input checked="" type="checkbox"/> NO												
IT NUMBERS		ZJ129-RZ,FR	ZJ239-RB,RY	ZJ240-RB,RE	ZJ240-FR	ZJ215-RY,RZ	ZJ239-RZ,PB,FR									
		ZJ240-RZ,PB	ZJ239-FR	ZJ130-RB												
PROBLEM: DOES BIT TEST FOR LOAD MEDIA CHECK INSTEAD OF COMPARE; WILL DROP SELF IF ACTUAL LOAD MEDIA SETS IT'S DESIGNATED BIT. (FOR EXAMPLE, IF LOAD MEDIA IS RL DISK)																
UNCONDITIONAL PATCH																
SOLUTION: PATCH BIT TEST TO COMPARE BYTE																
DEPO PATCH AREA																
CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO											
326	132737	122737														
SUBMITTING ENGINEER		MANUFACTURING ENGINEER		SUPPORT ENGINEER		CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER Q98-05314										
DATE: 8-nov-78		DATE: 29-nov-78		DATE:												
MAINTAINER		FIELD SERVICE		WAIVERING MANAGER		COORDINATION NO. MC#2503										
DATE: 8-nov-78		DATE:		DATE:												

DIAGNOSTIC ENGINEERING



DECO DEPO SUBMISSION

FOR RELEASE ENG. USE
 NEW CHANGE DELETE

PRODUCT IDENTIFICATION

D	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE			STATUS	DISTRIBUTION		1ST COPY - RIGHT YEAR	LAST COPY - RIGHT YEAR
	ZZ	CXTMA	I	2	02	DD	MMM	YY	OBSOLETE	G	R	1973	1978

TITLE: CXTMAI2 TML1 MODULE
 AUTHORD. BUTENHOF MAINTAINING GROUDEC/X11 SPT MAINTAINER D. BUTENHOF SUBMITTING ENGINEER D. BUTENHOF

PRODUCT COMPONENTS

CK	DESCRIPTION	PRODUCT NO	REV	CK	DESCRIPTION	PRODUCT NO	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
X		AF-E703 I-M2					

PRODUCTS OBSOLETE (other than previous version)

LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV	LIBRARY	PRODUCT NUMBER	REV
MD			MD			MD		

PRODUCT CHARACTERISTICS

PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2 digit codes representing the Processor the product operates with. See separate instructions.)
 03 04 05 10 20 21 34 35 40 45 50 55 60 70

OPERATIONAL CODES (Enter all applicable 2 digit codes that describe the product. See separate instructions.)
 02 03 04 06 50

ACT/APT/XXDP	EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
INFORMATION FIELD			X Y <input type="checkbox"/> N	X Y <input type="checkbox"/> N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED: _____

ICE AFFECTED DEC/X11 _____ MULTIMEDIA AFFECTED? YES NO

KIT NUMBERS	ZJ129-RZ,FR	ZJ239-RB,RV	ZJ240-RB,RE	ZJ240-FR	ZJ215-RV,RZ	ZJ239-RZ,PB,FR
	ZJ240-RZ,PB	ZJ239-FR				

PROBLEM:
 WHEN PREVIOUS PATCH CHANGED BIT TEST TO COMPARE, BRANCH CONDITIONS WERE LEFT UNCHANGED, RESULTING IN DROP UNLESS MEDIA IS LOAD MEDIA.

SOLUTION:
 PATCH BEQ TO BNE

DEPO PATCH AREA

FROM	TO	CHANGE I/C	FROM	TO
334	1421	1021		

SUBMITTING ENGINEER DATE: <i>[Signature]</i>	MANUFACTURING ENGINEER DATE: <i>[Signature]</i>	SUPPORT ENGINEER DATE:	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER 098-05314
PROJECT MANAGER DATE:	PROJECT ENGINEER DATE:	COORDINATOR DATE:	<i>mc 2786</i>