.REM _

IDENTIFICATION

PRODUCT CODE: AC-E703I-MC

PRODUCT NAME: CXTMAIC TM11 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

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 (WRITE, 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.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172520, VECTOR: 224, BR1: 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

8. OPERATION OPTIONS

SR1 BIT 0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR IS ASSUMED AND THE DRIVE IS DROPPED

SR1 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 MTBRC MTCMA MTD 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.

```
                        IOMODX  <TMAI >,172520,224,5,0,0,256,22,BUFIN,256.,1024.
                        MODULE  150000,TMAI ,172520,224,5,0,0,256,22,BUFIN,256.,1024.
                        .TITLE  TMAI DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                               .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 046524  044501     040  MODNAM: .ASCII  /TMAI / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   0       ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 172520                  ADDR:   172520+0        ;1ST DEVICE ADDR.
000010' 000224                  VECTOR: 224+0           ;1ST DEVICE VECTOR.
000012'     240                 BR1:    .BYTE   PRTY5+0 ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY0+0 ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1             ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN            ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN            ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN            ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN            ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 150000                  STAT:   150000          ;STATUS WORD.
000030' 000252'                 INIT:   START           ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP           ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0               ;PASS COUNTER.
000036' 000256                  ICONT:  256             ;# OF ITERATIONS PER PASS=256
000040' 000000                  ICOUNT: 0               ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0               ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0               ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0               ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0               ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0               ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0               ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                 ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0               ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   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'                         SBADR:                  ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN            ;CONTENTS OF CSR.
000104'                         WASADR:                 ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN            ;STATUS REG CONTENTS.
000106'                         ERRTYP:                 ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN            ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN            ;ACTUAL DATA.
000112' 000436'                 RSTRT:  RESTRT          ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN            ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN            ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN            ;# OF INTERRUPTS PER ITERATION
000122' 000022                  IDNUM:  22              ;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' 000400                  RBUFSZ: 256.            ;SIZE OF THE READ BUFFER
000134' 000000                  WBUFPA: OPEN            ;WRITE BUFFER PHYSICAL ADDRESS
000136' 000000                  WBUFEA: OPEN            ;WRITE BUFFER EA BITS
000140' 002000                  WBUFRQ: 1024.           ;WRITE BUFFER SIZE REQUESTED
000142' 000000                  WBUFSZ: OPEN            ;WRITE BUFFER SIZE AVAILABLE
000144' 000000                  CDERCT: OPEN            ;CDATA/DATCK ERROR COUNT
000146' 000000                  CDWDCT: OPEN            ;CDATA/DATCK WORD COUNT
000150' 000000                  FREE:   OPEN            ;RESERVED FOR FUTURE USE
        000940                          .REPT   SPSIZ   ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000252'                         MODSP:
                                ;*****************************************************************************
```

```
                                
000252' 012767  000003  177640  START:  MOV     #3,INTR         ;3 INTERRRUPTS PER ITERATION
000260' 012767  002000  177630          MOV     #1024.,WDFR     ;1024 WORDS PER ITERATION FROM MEM
000266' 012767  000400  177620          MOV     #256.,WDTO      ;256 WORDS TO MEM PER ITERATION
000274' 105067  003250                  CLRB    FLAG            ; CLEAR FLAGS
000300' 016767  177510  002054          MOV     DVID1,DVICE     ; GET DRIVE INDICATOR
000306' 016767  002050  002050          MOV     DVICE,DRIVE     ; ALSO SAVE IT IN DRIVE
000314' 005067  002046                  CLR     DRVVE           ; ZERO UNIT NUMBER
000320' 012767  177400  002042          MOV     #-400,DRVSFT    ; INITIALIZE THE SHIFTED DRIVE #
                                
                                ;FIND IF TM IS LOAD MEDIUM, IF SO, DROP LOAD DRIVE IF SELECTED
000326' 132737  000004  000041          BITB    #BIT2,@#41      ; IS TM THE LOAD MEDIUM ?
000334' 001421                          BEQ     1$              ; NO, CONTINUE
000336' 113700  000040                  MOVB    @#40,R0         ; GET LOAD DRIVE #
000342' 012701  000001                  MOV     #1,R1           ; SET UP DRIVE MASK
000346' 105700                  2$:     TSTB    R0              ; HAVE LOAD DRIVE YET?
000350' 001403                          BEQ     3$              ; YES, GO DROP IT
000352' 006301                          ASL     R1              ; NO, SHIFT DRIVE MASK
000354' 105300                          DECB    R0              ; COUNT SHIFTS,
000356' 000773                          BR      2$              ;  UNTIL HAVE LOAD DRIVE MASK
000360' 030167  001776          3$:     BIT     R1,DVICE        ; LOAD DRIVE SELECTED?
000364' 001405                          BEQ     1$              ; NO, CONTINUE
000366' 004767  001146                  JSR     PC,DROP         ; YES, GO DROP IT
000372' 104403  000000' 003530'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
                                
000400' 012767  177777  001760  1$:     MOV     #-1,DRVVE       ; INITIALIZE DRIVE COUNTER
000406' 004767  001564                  JSR     PC,SETUP        ; GENERATE REGISTER ADDRESSES
000412' 004767  001642                  JSR     PC,REZET        ; INITIALIZE TM REGS. AND ALL DRIVES
000416' 005767  001740                  TST     DVICE           ; DROP THE MODULE ?
000422' 001533                          BEQ     FINI            ; YES
000424' 004767  000622                  JSR     PC,REWIND       ; REWIND ALL DRIVES
000430' 005067  001742                  CLR     CYCKNT          ; ZERO TOTAL CYCLE COUNTER
000434' 000403                          BR      CONT
000436' 005767  177372          RESTRT: TST     PASCNT          ;ANY PASSES MADE YET ?
000442' 001703                          BEQ     START           ;NO BRANCH
000444'                         CONT:
000444' 104415  000000' 000124'         GETPAS,BEGIN,,RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
000452' 016767  177454  001722          MOV     RBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
000460' 005467  001716                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
000464' 006367  001712                  ASL     WCNT2           ; DOUBLE IT TO GET A BYTE COUNT
                                
000470'                         STRT:
000470' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
000474' 016767  177442  001676          MOV     WBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
000502' 005467  001672                  NEG     WCNT1           ; GET THE 2'S COMPLEMENT
000506' 006367  001666                  ASL     WCNT1           ; DOUBLE IT TO GET A BYTE COUNT
                                
000512' 004767  001060          NEXT:   JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
000516' 005767  001640                  TST     DVICE           ; ANY DRIVES LEFT ?
000522' 001473                          BEQ     FINI            ; NO, GO DROP THE MODULE
000524' 132767  000010  003016          BITB    #BIT3,FLAG      ; ALL DRIVES DONE ?
000532' 001356                          BNE     STRT            ; YES, GO GET ANOTHER BLOCK
000534' 042777  003400  002650          BIC     #3400,@MTC      ; CLEAR OUT UNIT NUMBER
000542' 056777  001622  002642          BIS     DRVSFT,@MTC     ; LOAD SELECTED DRIVE NUMBER
000550' 032777  000004  002632          BIT     #BIT2,@MTS      ; WRITE PROTECTED ?
000556' 001406                          BEQ     1$              ; NO, CONTINUE
```



```
000560' 004767  000754                  JSR     PC,DROP         ; YES, DROP THE DRIVE
000564' 104403  000000' 003530'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
000572' 000747                          BR      NEXT            ; GO ON TO NEXT DRIVE
000574' 032777  000001  002606  1$:     BIT     #BIT0,@MTS      ; DRIVE READY ?
000602' 001003                          BNE     2$              ; YES, CONTINUE
000604' 004767  001060                  JSR     PC,NOTRDY       ; NO, WAIT FOR READY
000610' 000727                          BR      STRT            ; TRY AGAIN
000612' 005067  002734          2$:     CLR     TRYI            ; ZERO RETRY COUNTERS
```

```
                                ;
000616' 004567  000222          RITE:   JSR     R5,WRITE        ; WRITE SOME DATA
000622' 000435                  BR      RETRY1          ; IF ERRORS, TRY IT AGAIN
000624' 132767  000004  002716          BITB    #BIT2,FLAG      ; DID THE TAPE REACH EOT ?
000632' 001410                  BEQ     BACK            ; NO, CONTINUE
000634' 142767  000004  002706  RITEOT: BICB    #BIT2,FLAG      ; YES, CLEAR THE EOT FLAG
000642' 004767  000404          JSR     PC,REWIND       ; REWIND ALL DRIVES
000646' 005067  001524          CLR     CYCKNT          ; ZERO TOTAL CYCLE COUNTER
000652' 000706                  BR      START           ; START OVER AT BEGINNING OF TAPES
000654' 004567  000216          BACK:   JSR     R5,BACKSP       ; BACKSPACE THE DRIVE
000660' 000240                  NOP                     ; ERROR RETURN
000662' 004567  000226          REED:   JSR     R5,READ         ; READ THE DATA WRITTEN
000666' 000440                  BR      RETRY2          ; IF ERRORS, TRY AGAIN
000670' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
000676' 000700'                 .+2                     ; IF ERROR, CONTINUE
                                ;
000700' 005267  001472          INC     CYCKNT          ; ADD 1 TO TOTAL CYCLE COUNT
000704' 104413  000000'         ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                        ;MONITOR SHALL TEST END OF PASS
000710' 000700                  BR      NEXT            ; NO, CONTINUE
                                ;
000712'                 FINI:
000712' 104410  000000'         END$,BEGIN              ; DROP THE MODULE
                                ;----------------------
                                ;
                                ;
                                ;
000716' 132767  000004  002624  RETRY1: BITB    #BIT2,FLAG      ;TAPE REACH EOT?
000724' 001343                  BNE     RITEOT          ;YES - TO EOT ROUTINE
000726' 105267  002620          INCB    TRY1            ; COUNT THE RETRYS
000732' 122767  000003  002612          CMPB    #3,TRY1         ; LIMIT EXCEEDED ?
000740' 001404                  BEQ     1$              ; YES, GO REPORT IT
000742' 004567  000130                  JSR     R5,BACKSP       ; NO, BACKUP TO TRY AGAIN
000746' 000240                  NOP                     ; ERROR RETURN
000750' 000722                  BR      RITE            ; GO TRY AGAIN
000752'                 1$:
000752' 104403  000000' 003514'         MSG$S,BEGIN,EXCED1      ;ASCII MESSAGE CALL WITH COMMON HEADER
000760' 004567  000162          JSR     R5,WRITEX       ; SKIP SOME TAPE, WRITE WITH EXTENDED IRG
000764' 000240                  NOP                     ; ERROR RETURN
000766' 000413                  BR      NEXTA           ; GO ON TO NEXT DRIVE
                                ;
000770' 105267  002557  RETRY2: INCB    TRY2            ; COUNT RETRYS
000774' 122767  000003  002551          CMPB    #3,TRY2         ; LIMIT EXCEEDED ?
001002' 001401                  BEQ     1$              ; YES, GO REPORT IT
001004' 000723                  BR      BACK            ; NO, BACKUP TO TRY AGAIN
001006'
001006' 104403  000000' 003522'         MSG$S,BEGIN,EXCED2      ;ASCII MESSAGE CALL WITH COMMON HEADER
001014' 000240                  NOP                     ; GO ON TO NEXT DRIVE
                                ;                       ;----------------------
```



```
001016' 032767  000001  176772  NEXTA:  BIT     #BIT0,SR1       ; DROP THE DRIVE ?
001024' 001405                  BEQ     1$              ; NO, SKIP TO NEXT DRIVE
001026' 004767  000506          JSR     PC,DROP         ; YES, DROP OFFENDING DRIVE
001032' 104403  000000' 003530'         MSG$S,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
001040' 000167  177446          1$:     JMP     NEXT            ; GO ON TO NEXT DRIVE
                                ;                       ;----------------------
                                ;
                                ;               -----  TM11 TAPE DRIVERS  -----
001044' 012767  060105  001304  WRITE:  MOV     #60105,FUNC     ; LOAD WRITE FUNCTION
001052' 016777  001322  002334          MOV     WCNT1,@MTBRC    ; LOAD BYTE COUNT
001060' 016777  177050  002330          MOV     WBUFPA,@MTCMA   ; LOAD BUFFER ADDRESS
001066' 016767  177044  001264          MOV     WBUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS
001074' 000440                  BR      GOGO            ; CONTINUE
001076' 012767  060113  001252  BACKSP: MOV     #60113,FUNC     ; LOAD BACKSPACE FUNCTION
001104' 012777  177777  002302          MOV     #-1,@MTBRC      ; LOAD BYTE COUNT
001112' 000431                  BR      GOGO            ; CONTINUE
001114' 012767  060103  001234  READ:   MOV     #60103,FUNC     ; LOAD READ FUNCTION
001122' 016777  001254  002264          MOV     WCNT2,@MTBRC    ; LOAD BYTE COUNT
001130' 016777  176772  002260          MOV     RBUFPA,@MTCMA   ; LOAD BUFFER ADDRESS
001136' 016767  176766  001214          MOV     RBUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS
001144' 000414                  BR      GOGO            ; CONTINUE
001146' 012767  060115  001202  WRITEX: MOV     #60115,FUNC     ; LOAD WRITE EXTENDED FUNCTION
001154' 016777  001220  002232          MOV     WCNT1,@MTBRC    ; LOAD BYTE COUNT
001162' 016777  176746  002226          MOV     WBUFPA,@MTCMA   ; LOAD BUFFER ADDRESS
001170' 016767  176742  001162          MOV     WBUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS
                                ;
                                ;
001176' 012777  001232' 176604  GOGO:   MOV     #NTRUPT,@VECTOR ; SET INTERUPT ENTRY POINTER
001204' 016767  001160  001144          BIS     DRVSFT,FUNC     ; LOAD DRIVE UNIT NUMBER
001212' 056767  001142  001136          BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
001220' 016777  001132  002164          MOV     FUNC,@MTC       ; EXECUTE THE FUNCTION
001226' 104400  000000'         EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;
001232'                 NTRUPT:
                                ;-------------------------------------------------------------------
001232' 000004  000000' 001240'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                ;-------------------------------------------------------------------
                                ;
001240' 004567  000576          1$:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
001244' 000205                  RTS     R5              ; ERRORS DETECTED, RETURN
001246' 005725                  TST     (R5)+           ; NO ERRORS, SKIP RETRY
001250' 000205                  RTS     R5              ; RETURN OK
                                ;                       ;----------------------
```

```

001252' 016767 001104 001104 REWIND: MOV     DVICE,DRIVE         ; GET ACTIVE DRIVES
001260' 012767 000010 001104         MOV     #8.,DVCNUM          ; LOAD MAXIMUM NUMBER OF DRIVES
001266' 016701 001100                MOV     DVCNUM,R1           ; PUT IT INTO A COUNTER

001272' 000241                1$:    CLC                         ; MAKE SURE C-BIT IS CLEAR
001274' 006267 001064                ASR     DRIVE               ; CHECK FOR ACTIVE DRIVE
001300' 103402                       BCS     2$                  ; IT'S ACTIVE --- BRANCH
001302' 005367 001064                DEC     DVCNUM              ; NOT ACTIVE, SUBTRACT FROM TOTAL
001306' 005301                2$:    DEC     R1                  ; ALL 8 CHECKED ?
001310' 003370                       BGT     1$                  ; NO, CONTINUE

001312' 012767 000100 002066         MOV     #100,CLK1           ; LOAD THE 2ND TIMER
001320' 012767 177777 001040         MOV     #-1,DRYVE           ; INITIALIZE THE DRIVE COUNTER
001326' 005067 001042                CLR     DVCFNT              ; CLEAR DEVICE COUNTER FOR ISR.
001332' 016767 001024 001024         MOV     DVICE,DRIVE         ; RESTORE DRIVE INDICATOR
001340' 012767 177400 001022         MOV     #-400,DRVSFT        ; INITIALIZE SHIFTED DRIVE NUMBER
001346' 004767 000224         3$:    JSR     PC,DRVADR           ; GO GET A DRIVE NUMBER
001352' 132767 000010 002170         BITB    #BIT3,FLAG          ; ALL DRIVES DONE ?
001360' 001017                       BNE     4$                  ; YES, GO WAIT FOR COMPLETION
001362' 004767 000712                JSR     PC,WAIT1            ; CONTROLLER READY ?
001366' 042777 003400 002016         BIC     #3400,@MTC          ; YES, CLEAR OUT OLD UNIT NUMBER
001374' 016767 000770 000754         MOV     DRVSFT,FUNC         ; LOAD NEW UNIT NUMBER
001402' 052767 000017 000746         BIS     #17,FUNC            ; LOAD REWIND FUNCTION
001410' 016777 000742 001774         MOV     FUNC,@MTC           ; EXECUTE THE REWIND
001416' 000753                       BR      3$                  ; GO ON FOR THE NEXT DRIVE

001420' 012767 077777 001756  4$:    MOV     #77777,CLK          ; SET THE TIMER
001426' 004767 000144         5$:    JSR     PC,DRVADR           ; GO GET A DRIVE NUMBER
001432' 132767 000010 002110         BITB    #BIT3,FLAG          ; ALL DRIVES DONE ?
001440' 001036                       BNE     8$                  ; YES, GET OUT
001442' 042777 003400 001742         BIC     #3400,@MTC          ; CLEAR OUT OLD UNIT NUMBER
001450' 056777 000714 001734         BIS     DRVSFT,@MTC         ; LOAD NEW UNIT NUMBER
001456'                       6$:
001456' 104407 000000'               BREAKS,BEGIN                ;TEMPORARY RETURN TO MONITOR....
001462' 104407 000000'               BREAKS,BEGIN                ;THEN CONTINUE AT NEXT INSTRUCTION.
001466' 032777 000041 001714         BIT     #41,@MTS            ; DRIVE READY AND AT BOT ?
001474' 001354                       BNE     5$                  ; YES, GO CHECK THE NEXT DRIVE
001476' 005367 001702                DEC     CLK                 ; NO, OUT OF TIME ?
001502' 003365                       BGT     6$                  ; NO, WAIT SOME MORE
001504' 005367 001676                DEC     CLK1                ; YES, WANT ANOTHER 40 SECONDS ?
001510' 001407                       BEQ     7$                  ; NO, TIME-OUT
001512' 012767 077777 001664         MOV     #77777,CLK          ; RESET THE TIMER
001520' 000756                       BR      6$                  ; WAIT SOME MORE
001522' 012767 000013 176356         MOV     #13,ERRTYP          ;REWIND ERROR
001530'                       7$:
                                     ;*************************************************************
001530' 104405 000000' 000000        HRDERS,BEGIN,NULL           ; REWIND ERROR, REWIND NOT COMPLETE
                                     ;*************************************************************

001536' 000207                8$:    RTS     PC                  ; RETURN
                              ;      ------------------------
```



```
001540' 012701 000001         DROP:  MOV     #1,R1               ; INITIALIZE DROP PICKER
001544' 016700 000616                MOV     DRYVE,R0            ; GET THE DRIVE NUMBER
001550' 001403                       BEQ     2$                  ; IF DRIVE 0 GO DROP IT
001552' 006301                1$:    ASL     R1                  ; NO, AIM AT THE NEXT DRIVE
001554' 005300                       DEC     R0                  ; IS THIS THE ONE ?
001556' 001375                       BNE     1$                  ; NO, LOOK AGAIN
001560' 040167 000576         2$:    BIC     R1,DVICE            ; DROP THE DRIVE
                                     ;*************************************************************
                                                                 ;CONVERT DRYVE TO ASCII AND
                                                                 ;STORE AT ADR1
001564' 104420 000000' 002366'       OTOAS,BEGIN,DRYVE,ADR1
001572' 003540'
                                     ;*************************************************************
001574' 000207                       RTS     PC                  ; RETURN
                              ;      ------------------------


001576' 005267 000564         DRVADR: INC    DRYVE               ; COUNT A DRIVE
001602' 062767 000400 000560         ADD     #BIT8,DRVSFT        ; DRIVE COUNT LINED UP WITH MTC
001610' 142767 000010 001732         BICB    #BIT3,FLAG          ; CLEAR END OF DRIVES FLAG
001616' 022767 000010 000542         CMP     #8.,DRYVE           ; ALL DRIVES CHECKED ?
001624' 001404                       BEQ     1$                  ; YES, GO FLAG END OF DRIVES
001626' 006267 000532                ASR     DRIVE               ; NO, IS NEXT DRIVE CHOSEN ?
001632' 103361                       BCC     DRVADR              ; NO, GO TRY ANOTHER DRIVE
001634' 000207                       RTS     PC                  ; RETURN

001636' 152767 000010 001704  1$:    BISB    #BIT3,FLAG          ; SET END OF DRIVES FLAG
001644' 012767 177777 000514         MOV     #-1,DRYVE           ; RESET DRIVE COUNTER
001652' 012767 177400 000510         MOV     #-400,DRVSFT        ; ZERO THE SHIFTED DRIVE #
001660' 016767 000476 000476         MOV     DVICE,DRIVE         ; RESTORE CHOSEN DRIVES
001666' 000207                       RTS     PC                  ; RETURN
                              ;      ------------------------


001670' 012767 177777 000470  NOTRDY: MOV    #-1,DRYVE           ; START WITH FIRST DRIVE
001676' 012767 177400 000464         MOV     #-400,DRVSFT        ;
001704' 016767 000452 000452         MOV     DVICE,DRIVE         ; RESET DRIVE SELECT
001712' 004767 177660         1$:    JSR     PC,DRVADR           ; GET A DRIVE ADDRESS
001716' 132767 000010 001624         BITB    #BIT3,FLAG          ; ALL DRIVES CHECKED ?
001724' 001012                       BNE     2$                  ; YES, RETURN
001726' 016777 000436 001456         MOV     DRVSFT,@MTC         ; NO, LOAD NEXT DRIVE ADDRESS
001734' 032777 000001 001446         BIT     #BIT0,@MTS          ; IS THIS DRIVE READY ?
001742' 001363                       BNE     1$                  ; YES, CONTINUE
001744' 004767 000022                JSR     PC,WAIT             ; NO, WAIT FOR IT
001750' 000760                       BR      1$                  ; GO CHECK REST OF DRIVES
001752' 000207                2$:    RTS     PC                  ; RETURN
                              ;      ------------------------



001754' 016767 001430 176116  ERSUB1: MOV    MTS,CSRA            ; LOAD ADR. OF CURRENT CSR
001762' 017767 001422 176112         MOV     @MTS,ACSR           ; LOAD CONTENTS OF CURRENT CSR
001770' 000207                       RTS     PC                  ; RETURN
                              ;      ------------------------
```

```
001772' 012767  077777  001404  WAIT:   MOV     #77777,CLK              ; SET THE TIMER
002000'                         1$:
002000' 104407  000000'                 BREAK$,BEGIN                    ;TEMPORARY RETURN TO MONITOR....
002004' 104407  000000'                 BREAK$,BEGIN                    ;THEN CONTINUE AT NEXT INSTRUCTION.

002010' 032777  000001  001372          BIT     #BIT0,@MTS              ; DRIVE READY ?
002016' 001010                          BNE     2$                      ; YES, RETURN
002020' 005367  001360                  DEC     CLK                     ; NO, WAIT SOME MORE ?
002024' 001365                          BNE     1$                      ; YES, WAIT
002026' 004767  177506                  JSR     PC,DROP                 ; TIME-OUT, DROP THE DRIVE
002032' 104403  000000' 003530'         MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
002040' 000207                  2$:     RTS     PC                      ; RETURN
                                ;       ------------------------




002042' 004767  177706          ERRORS: JSR     PC,ERSUB1               ; LOAD ERROR INFORMATION
002046' 005777  001340                  TST     @MTC                    ; ANY ERRORS ?
002052' 100047                          BPL     4$                      ; NO, RETURN OK
002054' 032777  001000  001326          BIT     #BIT9,@MTS              ; RECORD LENGTH ERROR ?
002062' 001414                          BEQ     1$                      ; NO, CONTINUE
002064' 032777  000002  001320          BIT     #BIT1,@MTC              ; WAS THE COMMAND A READ ?
002072' 001425                          BEQ     2$                      ; NO, REPORT THE ERROR
002074' 032777  174600  001306          BIT     #174600,@MTS            ;YES,RLE ERR EXPECTED,CHECK OTHER ERRORS
002102' 001430                          BEQ     3$                      ;RETURN OK
002104' 042767  001000  175770          BIC     #BIT9,ACSR              ;ELIMINATE RLE BIT FROM
002112' 000415                          BR      2$                      ;ERROR REPORT & GO REPORT IT
002114' 032777  002000  001266  1$:     BIT     #BIT10,@MTS             ; AT END OF TAPE ?
002122' 001411                          BEQ     2$                      ; NO, REPORT THE ERROR
002124' 152767  000004  001416          BISB    #BIT2,FLAG              ; YES, SET EOT FLAG
002132' 032777  174600  001250          BIT     #174600,@MTS            ;NO / TEST OTHER ERRORS
002140' 001411                          BEQ     3$                      ;NONE RETURN OK
002142' 005067  175740                  CLR     ERRTYP                  ;UNKNOWN ERROR
002146'                         2$:
                                        ;*****************************************************************
002146' 104405  000000' 003410'         HRDER$,BEGIN,TABLE              ;
                                        ;*****************************************************************
002154' 012777  010000  001230          MOV     #BIT12,@MTC             ; ISSUE A POWER CLEAR
002162' 000404                          BR      5$                      ; RETURN TO DO A RETRY

002164' 012777  010000  001220  3$:     MOV     #BIT12,@MTC             ; ISSUE A POWER CLEAR
002172' 005725                  4$:     TST     (R5)+                   ; SKIP THE RETRY
002174' 000205                  5$:     RTS     R5                      ; RETURN
                                ;       ------------------------
```



```
002176' 016700  175604          SETUP:  MOV     ADDR,R0                 ; GET DEVICE ADDRESS
002202' 010067  001202                  MOV     R0,MTS                  ; GENERATE CONTROLLER REGS. ADDRESSES
002206' 005720                          TST     (R0)+
002210' 010067  001176                  MOV     R0,MTC
002214' 005720                          TST     (R0)+
002216' 010067  001172                  MOV     R0,MTBRC
002222' 005720                          TST     (R0)+
002224' 010067  001166                  MOV     R0,MTCMA
002230' 005720                          TST     (R0)+
002232' 010067  001162                  MOV     R0,MTD
002236' 005720                          TST     (R0)+
002240' 010067  001156                  MOV     R0,MTRD

002244' 016700  175540                  MOV     VECTOR,R0               ; GET THE VECTOR ADDRESS
002250' 116760  175536  000002          MOVB    BPI,2(R0)               ; SET PRIORITY

002256' 000207                  2$:     RTS     PC                      ; RETURN
                                ;       ------------------------




002260' 012777  010000  001124  REZET:  MOV     #BIT12,@MTC             ; EXECUTE POWER CLEAR
002266' 004767  000006                  JSR     PC,WAIT1                ; GO WAIT FOR CONTROLLER READY
002272' 004767  177372                  JSR     PC,NOTRDY               ; MAKE SURE ALL CHOSEN DRIVES ARE READY
002276' 000207                          RTS     PC                      ; RETURN
                                ;       ------------------------




002300' 012767  077777  001076  WAIT1:  MOV     #77777,CLK              ; SET THE TIMER
002306' 105777  001100          1$:     TSTB    @MTC                    ; CONTROLLER READY ?
002312' 100417                          BMI     2$                      ; YES, CONTINUE
002314' 104407  000000'                 BREAK$,BEGIN                    ;TEMPORARY RETURN TO MONITOR....
002320' 104407  000000'                 BREAK$,BEGIN                    ;THEN CONTINUE AT NEXT INSTRUCTION.
002324' 005367  001054                  DEC     CLK                     ; WAIT SOME MORE ?
002330' 001366                          BNE     1$                      ; YES
002332' 012767  000003  175546          MOV     #3,ERRTYP               ;CONTROLLER NOT READY
                                        ;*****************************************************************
002340' 104405  000000' 003410'         HRDER$,BEGIN,TABLE              ; CONTROLLER NOT READY
                                        ;*****************************************************************
002346' 000167  176340                  JMP     FINI                    ; GO DROP THE MODULE
002352' 000207                  2$:     RTS     PC                      ; READY, RETURN
                                ;       ------------------------
```

```
     002354' 000000                      CNT:     0
     002356' 000000                      FUNC:    0
     002360' 000000                      XMEM:    0
     002362' 000000                      DVICE:   0
     002364' 000000                      DRIVE:   0
     002366' 000000                      DRVVE:   0
     002370' 000000                      DRVSFT:  0
     002372' 000000                      DVCNUM:  0
     002374' 000000                      DVCKNT:  0
     002376' 000000                      CYCKNT:  0
     002400' 000000                      WCNT1:   0
     002402' 000000                      WCNT2:   0
     002404' 000400                      BUFIN:   .BLKW   256.
     003404' 000000                      CLK:     0
     003406' 000000                      CLK1:    0
     003410'                             TABLE:
     003410' 000000                      MTS:     0
     003412' 000000                      MTC:     0
     003414' 000000                      MTRPC:   0
     003416' 000000                      MTCMA:   0
     003420' 000000                      MTD:     0
     003422' 000000                      MTRD:    0
     003424' 002376'                     CYC:     CYCKNT
     003426' 177777                               177777
     003430' 020040  051104  053111      MES2:    .ASCIZ  '  DRIVE  '
     003436' 020105  000040
     003442' 020040  051104  050117      MES3:    .ASCIZ  '  DROPPED%'
     003450' 042520  022504     000
     003455'    040  042522  051124      MES4:    .ASCIZ  ' RETRY EXCEEDED%'
     003462' 020131  054105  042503
     003470' 042105  042105  000045
     003476' 053440  044522  042524      MES5:    .ASCIZ  ' WRITE'
     003504'    000
     003505'    040  042522  042101      MES6:    .ASCIZ  ' READ'
     003512'    000
             003514'                              .EVEN


     003514' 003476'                     EXCED1:  MES5
     003516' 003455'                              MES4
     003520' 177777                               177777
     003522' 003505'                     EXCED2:  MES6
     003524' 003455'                              MES4
     003526' 177777                               177777
     003530' 003430'                     DRP:     MES2
     003532' 003545'                              NUMB
     003534' 003442'                              MES3
     003536' 177777                               177777
     003540' 000005                      ADR1:    .BLKB   5
     003545'    000                      NUMB:    .BYTE   0
     003546' 000000                               .WORD   0
     003550'    000                      FLAG:    .BYTE   0
             003552'                              .EVEN
     003552'    000                      TRY1:    .BYTE   0
     003553'    000                      TRY2:    .BYTE   0
```



```
                                                  .EVEN

             000001                               .END
```

```
ACSR    000102R  187#  485*  518*
ADDR    000006R  153#  539
ADDR22= 001000   216#
ADR1    003540R  444   634#
ASB     000106R  191#
ASTAT   000104R  189#
AWAS    000110R  192#
BACK    002654R  285   290#  326
BACKSP  001076R  290   314   348#
BEGIN   000000R  150#  239   252   258   273   294   299   304   318   328   336   366   370
                 413   414   426   444   494   495   502   528   573   574   579
BIT0  = 000001   216#  275   333   475   497
BIT1  = 000002   216#  514
BIT10 = 002000   216#  520
BIT11 = 004000   216#
BIT12 = 010000   216#  530   533   561
BIT13 = 020000   216#
BIT14 = 040000   216#
BIT15 = 100000   216#
BIT2  = 000004   216#  227   270   284   286   309   522
BIT3  = 000010   216#  266   397   408   453   460   472
BIT4  = 000020   216#
BIT5  = 000040   216#
BIT6  = 000100   216#
BIT7  = 000200   216#
BIT8  = 000400   216#  452
BIT9  = 001000   216#  512   518
BREAK$= 104407   216#  413   414   494   495   573   574
BR1     000012R  155#  553
BR2     000013R  156#
BTOD$ = 104421   216#
BUFIN   002404R  198   598#
CDATA$= 104412   216#  294
CDERCT  000144R  206#
CDWDCT  000146R  207#
CLK     003404R  405*  417*  421*  492*  499*  570*  575*  599#
CLK1    003406R  391*  419*  600#
CNT     002354R  586#
CONFIG  000056R  175#
CONT    000444R  248   251#
CSRA    000100R  185#  484*
CYC     003424R  608#
CYCCNT  002376R  247*  288*  298*  595#  608
DATCK$= 104411   216#
DATER$= 104404   216#
DRIVE   002364R  222*  380*  385*  394*  456*  463*  470*  590#
DROP    001540R  238   272   335   434#  501
DRP     003530R  239   273   336   502   630#
DRVADR  001576R  263   396   407   451#  457   471
DRVSFT  002370R  224*  269   363   395*  401   411   452*  462*  469*  474   592#
DRVVE   002366R  223*  241*  392*  435   444   451*  454   461*  468*  591#
DVCCNT  002374R  393*  594#
DVCNUM  002372R  381*  382   387*  593#
DVICE   002362R  221*  222   236   244   264   380   394   440*  463   470   589#
DVID1   000014R  157#  221
ENDIT$= 104413   216#  299
```



```
ENDS  = 104410   216#  304
ERRORS  002042R  373   509#
ERRTYP  000106R  190#  423*  525*  577*
ERSUB1  001754R  484#  509
EXCED1  003514R  318   624#
EXCED2  003522R  328   627#
EXITS = 104400   216#  366
FINI    000712R  245   265   303#  581
FLAG    003550R  220*  266   284   286*  309   397   408   453*  460*  472   522*  637#
FREE    000150R  208#
FUNC    002356R  343*  348*  351*  356*  363*  364*  365   401*  402*  403   587#
GETPA$= 104415   216#  262
GOGO    001176R  347   350   355   362#
GWBUF$= 104414   216#  258
HRDCNT  000044R  170#
HRDERS= 104405   215#  426   528   579
HRDPAS  000050R  172#
ICONT   000036R  167#
ICOUNT  000040R  168#
IDNUM   000122R  197#
IMODX.= 000000   209#  259
INIT    000030R  164#
INTR    000120R  196#  217*
MAP22$= 104416   216#
MES2    003430R  610#  630
MES3    003442R  612#  632
MES4    003455R  614#  625   628
MES5    003476R  617#  624
MES6    003505R  619#  627
MODNAM  000000R  151#
MODSP   000252R  165   214#
MSGN$ = 104403   216#  239   273   318   328   336   502
MSGS$ = 104402   216#
MSG5  = 104401   216#
MTBRC   003414R  344*  349*  352*  357*  544*  604#
MTC     003412R  268*  269*  365*  400*  403*  410*  411*  474*  510   514   530*  533*  542*
                 561*  571   603#
MTCMA   003416R  345*  353*  358*  546*  605#
MTD     003420R  548*  606#
MTRD    003422R  550*  607#
MTS     003410R  270   275   415   475   484   485   497   512   516   520   523   540*  602#
NEXT    000512R  263#  274   301   337
NEXTA   001016R  321   333#
NOTRDY  001670R  277   468#  563
NTRUPT  002322R  362   368#
NULL  = 000000   216#  426
NUMB    003545R  631   635#
OPEN  = 000000   152   158   159   160   161   178   179   180   181   182   183   184   185
                 187   189   191   192   194   195   196   199   200   202   203   205   206
                 207   208   216#
OTOA$ = 104420   216#  444
PASCNT  000034R  166#  249
PIRQ$ = 000004   216#  370
POPSP = 005726   216#
POPSP2= 022626   216#
PRTY  = 000000   216#
```

```
PRTV0 = 000000            156     216#
PRTV1 = 000040            216#
PRTV2 = 000100            216#
PRTV3 = 000140            216#
PRTV4 = 000200            216#
PRTV5 = 000240            155     216#
PRTV6 = 000300            216#
PRTV7 = 000340            216#
PS    = 177776            216#
PSW   = 177776            216#
PUSH  = 005746            216#
PUSH2 = 024646            216#
RANDS = 104417            216#
RANNUM  000054R           174#
RBUFBA  000130R           200#    354
RBUFPA  000126R           199#    294     353
RBUFSZ  000132R           201#    253
RBUFVA  000124R           198#    252
READ    001114R           292     351#
REED    000662R           292#
RESTRT  002436R           193     249#
RES1    000056R           176#
RES2    000060R           177#
RETRY1  000716R           283     309#
RETRY2  000770R           293     323#
REWIND  001252R           246     287     380#
REZET   002260R           243     561#
RITE    000616R           282#    316
RITEOT  000634R           286#    310
RSTRT   000112R           193#
SBADR   000102R           186#
SETUP   002176R           242     539#
SOFCNT  000042R           169#
SOFERS= 104406            216#
SOFPAS  000046R           171#
SPOINT  000032R           165#
SPSIZ = 000040              1#    209
SR1     000016R           158#    333
SR2     000020R           159#
SR3     000022R           160#
SR4     000024R           161#
START   000252R           164     217#    250
STAT    000026R           163#
STRT    002470R           257#    267     278     289
SVR0    000062R           176#
SVR1    000064R           179#
SVR2    000066R           180#
SVR3    000070R           181#
SVR4    000072R           182#
SVR5    000074R           183#
SVR6    000076R           184#
SYSCNT  000052R           173#
TABLE   003410R           528     579     601#
TRPDFD= 000022            216#
TRY1    003552R           279*    311*    312     639#
TRY2    003553R           323*    324     640#
```



```
VECTOR  000010R           154#    362*    552
WAIT    001772R           477     482#
WAIT1   002300R           399     562     570#
WASADR  000104R           188#
WBUFBA  000136R           203#    346     359
WBUFPA  000134R           202#    345     358
WBUFRQ  000140R           204#
WBUFSZ  000142R           205#    259
WCNT1   002400R           259*    260*    261*    344     357     596#
WCNT2   002402R           253*    254*    255*    352     597#
WDFR    000116R           195#    218*
WDTO    000114R           194#    319*
WRITE   001044R           282     343#
WRITEX  001146R           315     356#
XFLAG   000005R           152#
XMEM    002360R           346*    354*    359*    364     588#
.       = 003554R         295     598#    621#    634#    638#

. ABS.  030000      000
        003554      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XTMAIO,XTMAIO/SOL/CRF:SYM=DDXCOM,XTMAIO
 RUN-TIME:  1  2  .3 SECONDS
 RUN-TIME RATIO: 18/4=4.6
 CORE USED:  7K  (13 PAGES)
```

33

DIAGNOSTIC ENGINEERING

digital DECO ☐ DEPO ☒ SUBMISSION ☐

FOR RELEASE ENG. USE ☐ NEW ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE DD MMM YY	STATUS	DISTRIBUTION	1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
MD	ZZ	CXTMA	1	1	Di	1 NOV 78	☐ OBSOLETE	X G ☐ R	1973	1978

TITLE CXTMA11 TM11 MOUDLE

AUTHOR D. BUTENHOF | MAINTAINING GROUP DEC/X11 SPT GRP | 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	DEPO	AF-E703I-M1					

PRODUCTS OBSOLETED (other than previous version)

	LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV
MD				M D				M D			

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 INFORMATION FIELD	EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
		1132	XX Y ☐ N	XX Y ☐ N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

DEVICE AFFECTED DEC/X11 | MULTIMEDIA AFFECTED? ☐ YES ☒ 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
8-NOV-78	DATE:	DATE:	

DIAGNOSTIC ENGINEERING

digital DECO ☐ DEPO ☒ SUBMISSION ☐

FOR RELEASE ENG. USE
☐ NEW ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

D	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE DD MMM YY	STATUS	DISTRIBUTION	1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
	ZZ	CXTMA	I	2	02	12 DEC 78	☐ OBSOLETE	☐ G ☐ R	1973	1978

TITLE CXTMAI2 TM11 MODULE

AUTHOR	MAINTAINING GROUP	MAINTAINER	SUBMITTING ENGINEER
D. BUTENHOF	DEC/X11 SPT	D. BUTENHOF	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 OBSOLETED (other than previous version)

	LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV
MD				M D				M D			

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 INFORMATION FIELD		EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
				☒ Y ☐ N	☒ Y ☐ N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED

DEVICE AFFECTED DEC/X11 — MULTIMEDIA AFFECTED? ☐ YES ☒ NO

KIT NUMBERS: ZJ129-RZ,FR ZJ239-RB,RY ZJ240-RB,RE ZJ240-FR ZJ215-RY,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

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO
334	1421	1021			

SUBMITTING ENGINEER	MANUFACTURING ENGINEER	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER 098-05314
DATE	DATE	DATE	
RELEASE ENGINEER		MAINTENANCE MANAGER	COORDINATOR MC 2786
DATE	DATE	DATE	