SEQ 0001

.REM _

IDENTIFICATION

PRODUCT CODE: AC-E896C-MC

PRODUCT NAME: CXTRAC0 TP79F MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

TRA IS AN IOMOD THAT EXERCISES A TAPE DRIVE ON AN TR79F 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 TAPE DRIVE WITH A TR79F CONTROLLER

STORAGE:: TRA REQUIRES:
1. DECIMAL WORDS: 1089
2. OCTAL WORDS: 02101
3. OCTAL BYTES: 4202

3. PASS DEFINITION

ONE PASS OF THE TRA MODULE CONSISTS OF 512 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 TRA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 164000, VECTOR: 170, BR1: 4, 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 4 TR79F REGISTERS AND THE CYCLE COUNT IN THE FOLLOWING ORDER:

TRCR TRST TRWC TRBA 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.
IF THE OPERATOR IS REASONABLEY SURE OF A BAD SPOT ON THE TAPE HE CAN ENTER THE CYCLE COUNT OF THE SUSPECTED BAD SPOT IN THE TABLE 'BADSPT' AND THE PROGRAM WILL TREAT IT AS BAD NOT AN ERROR, WITHOUT INDICATING AN ERROR. THESE BAD SPOTS SHOULD BE ENTERED IN THE TABLE, FROM THE TOP, WITH A 177777 AS AN END OF THE ENTRIES. THERE IS ONLY ROOM FOR 16 ENTRIES, DO NOT WRITE OVER THE 17TH LOCATION IN THE TABLE.

SEQ 0004

```
NOTE!!!!WHEN THE ENTRY IS MADE INTO THE BAD SPOT TABLE
EACH ONE MUST BE MADE ONE AT A TIME. BE SURE TO RUN
THE PROGRAM EACH TIME AN ENTRY IS MADE BECAUSE THE CYCLE COUNT
IS ALTERED BY TWO OR THREE EACH TIME THE  TABLE HAS A NEW ENTRY.
FOR EXAMPLE; IF THERE ARE ERRORS AT CYCLE COUNTS 47
 64 AND 75, IF 47 IS ENTERED IN THE TABLE THE NEXT
TWO CYCLE COUNT ERRORS WILL BE 65 AND 76.
THEREFORE THE BEST WAY IS TO MAKE ONE ENTRY THEN RUN THE
PROGRAM TO DETERMINE THE PROPER CYCLE COUNT  FOR THE
NEXT BAD SPOT.
DUE TO THE NATURE OF MAG TAPES, THIS BAD SPOT TABLE MAY NOT
NOT ALWAYS WORK SINCE THE DRIVE MAY WRITE THE 64TH (OR WHICH-EVER)
RECORD IN SLIGHTLY DIFFERENT PLACES EACH PASS DOWN THE TAPE.


NOTE:   DUE TO THE TAPE DRIVE BEING ODD PARITY, THE WRITE
        BUFFER IS CONTAINED IN THE MODULE WITH AN ODD PARITY
        DATA PATTERN.
```

```
000000'  -          IDMOD   <TRAC  >,164000,170,4,0,0,512.,125
000000'             MODULE  140000,TRAC  ,164000,170,4,0,0,512.,125
                    .TITLE  TRAC DEC/X11 SYSTEM EXERCISER MODULE
                ;   IDEXCOM VERSION  6      23-MAY-78
                                            .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 051124  041501     040  MODNAM: .ASCII  /TRAC / ;MODULE NAME.
000005'     040                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 164000                  ADDR:   164000+0                ;1ST DEVICE ADDR.
000010' 000170                  VECTCR: 170+0                   ;1ST DEVICE VECTOR.
000012'     200                 PR1:    .BYTE   PRTY4+0         ;1ST BR LEVEL.
000013'     004                 PR2:    .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' 140000                  STAT:   140000                  ;STATUS WORD.
000030' 000232'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 001000                  ICONT:  512.                    ;# OF ITERATIONS PER PASS=512.
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:
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'                         SRADR:                          ;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' 000224'                 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' 000125                  IDNUM:  125                     ;MODULE IDENTIFICATION NUMBER=125
```



```
                    000040                  .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                            .NLIST
                                            .WORD   0
                                            .LIST
                                            .ENDR
                    000224'                 MODSP:
                                ;*****************************************************************************
```

```
                                .GLOBL  SR
000224' 005767  177604          RESTRT: TST     PASCNT          ;ANY PASSES YET?
000230' 001076                  BNE     REST1           ;YES, THIS IS A RESTART
000232' 012767  002000  177656  START:  MOV     #1024.,WDFR     ;1024 WORDS FROM MEM/ITERATION
000240' 012767  000400  177646          MOV     #256.,WDTO      ;256 WORDS TO MEM/ITERATION
000246' 012767  000004  177644          MOV     #4,INTR         ;4 INTERRUPTS/ITERATION
000254' 105067  003530                  CLRB    FLAG            ; CLEAR FLAGS
000260' 016767  177530  002252          MOV     DVID1,DVICE     ; GET DRIVE INDICATOR
000266' 016767  002246  002246          MOV     DVICE,DRIVE     ; ALSO SAVE IT IN DRIVE
000274' 005067  002244                  CLR     DRVVE           ; ZERO UNIT NUMBER
000300' 012767  177400  002240          MOV     #-400,DRVSFT    ; INITIALIZE THE SHIFTED DRIVE #
                                ;
000306' 132737  040000  000041          BITB    #BIT14,@#41     ;IF TM IS THE LOAD MEDIUM THEN
000314' 001424                  BEQ     3$              ;BEGIN
000316' 113700  000040                  MOVB    @#40,R0         ;   GET LOAD-DEVICE NUMBER
000322' 012701  000001                  MOV     #1,R1           ;   INITIALIZE DEVICE MASK
000326' 105700                  1$:     TSTB    R0              ;   WHILE R0>0 DO
000330' 001403                  BEQ     2$              ;   BEGIN
000332' 006301                  ASL     R1              ;       SHIFT DEVICE MASK TO NEXT
000334' 105300                  DECB    R0              ;       DOWNCOUNT DEVICE NUMBER
000336' 000773                  BR      1$              ;   END
000340' 130167  002176          2$:     BITB    R1,DRIVE        ;   IF LOAD-DEVICE SELECTED THEN
000344' 001410                  BEQ     3$              ;   BEGIN
000346' 113767  000040  002170          MOVB    @#40,DRVVE      ;       MOVE LOAD-DEVICE NUMBER TO DRVVE
000354' 004767  001300                  JSR     PC,DROP         ;       DROP LOAD-DEVICE
000360' 104403  000000' 003770'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
                                ;   END
                                ;END
                                ;
000366' 012767  177777  002150  3S:     MOV     #-1,DRVVE       ; INITIALIZE DRIVE COUNTER
000374' 004767  001750                  JSR     PC,SETUP        ; GENERATE REGISTER ADDRESSES
000400' 004767  002012                  JSR     PC,REZET        ; INITIALIZE TM REGS. AND ALL DRIVES
000404' 005767  002130                  TST     DVICE           ; DROP THE MODULE ?
000410' 001002                  BNE     11$             ;NO
000412' 000167  000372                  JMP     FINI            ;YES
000416' 004767  000672          11$:    JSR     PC,REWIND       ; REWIND ALL DRIVES
000422' 005067  002126                  CLR     CYCKNT          ; ZERO TOTAL CYCLE COUNTER
                                ;
000426'                         REST1:
000426' 104415  000000' 004022'         GETPAS,BEGIN, IBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT IBUFVA
000434' 016767  003372  002116          MOV     IBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
000442' 005467  002112                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
                                ;
000446'                         STRT:
000446' 104415  000000' 004014'         GETPAS,BEGIN, OBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT OBUFVA
000454' 016767  003350  002074          MOV     OBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
000462' 005467  002070                  NEG     WCNT1           ; GET THE 2'S COMPLEMENT
                                ;
000466' 004767  001224          NEXT:   JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
000472' 005767  002042                  TST     DVICE           ; ANY DRIVES LEFT ?
000476' 001544                  BEQ     FINI            ; NO, GO DROP THE MODULE
000500' 132767  000010  003302          BITB    #BIT3,FLAG      ; ALL DRIVES DONE ?
000506' 001357                  BNE     STRT            ; YES, GO GET ANOTHER BLOCK
000510' 042777  001400  003052          BIC     #1400,@MTC      ; CLEAR OUT UNIT NUMBER
000516' 056777  002024  003044          BIS     DRVSFT,@MTC     ; LOAD SELECTED DRIVE NUMBER
```



```
000524' 032777  000004  003034          BIT     #BIT2,@MTS      ; WRITE PROTECTED ?
000532' 001406                  BEQ     1$              ; NO, CONTINUE
000534' 004767  001120                  JSR     PC,DROP         ; YES, DROP THE DRIVE
000540' 104403  000000' 003770'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
000546' 000747                  BR      NEXT            ; GO ON TO NEXT DRIVE
000550' 032777  000200  003012  1$:     BIT     #BIT7,@MTC      ; DRIVE READY ?
000556' 001003                  BNE     2$              ; YES, CONTINUE
000560' 004767  001224                  JSR     PC,NOTRDY       ; NO, WAIT FOR READY
000564' 000730                  BP      STRT            ; TRY AGAIN
000566' 005067  003220          2$:     CLR     TRY1            ; ZERO RETRY COUNTERS
000572' 005267  001756                  INC     CYCKNT
```

```
                                ;
                                ;
000576' 004567 000322   RITE:   JSR     R5,WRITE        ; WRITE SOME DATA
000602' 005004          RETRY1: BR      RETRY1          ; IF ERRORS, TRY IT AGAIN
000604' 132767 000004 003176    BITB    #BIT2,FLAG      ; DID THE TAPE REACH EOT ?
000612' 001416          BEQ     BACK                    ; NO, CONTINUE
000614' 142767 000004 003166    BICB    #BIT2,FLAG      ; YES, CLEAR THE EOT FLAG
000622' 004767 000466           JSR     PC,REWIND       ; REWIND ALL DRIVES
000626' 005067 001722           CLR     CYCKNT          ; ZERO TOTAL CYCLE COUNTER
000632' 000705                  BR      STRT            ; START OVER AT BEGINNING OF TAPES
000634' 004567 000316   BACK:   JSR     R5,BACKSP       ; BACKSPACE THE DRIVE
000640' 000240          REED:   NOP                     ; ERROR RETURN
000642' 004567 000332           JSR     R5,READ         ; READ THE DATA WRITTEN
000646' 000500          DATCK:  BR      RETRY2          ; IF ERRORS, TRY AGAIN
000650' 010067 177206           MOV     R0,SVR0         ;SAVE R0
000654' 010167 177204           MOV     R1,SVR1         ;SAVE R1
000660' 005067 001644           CLR     COUNT
000664' 012700 004034'          MOV     #CBUF,R0        ;LOAD GOOD POINTER
000670' 012701 004120'          MOV     #IBUF,R1        ;LOAD BAD POINTER
000674' 021777 002000 000000G DAT1:     CMP     #RIT10,@SR
000702' 001004                  BNE     1$
000704' 022767 000003 001616            CMP     #3,COUNT
000712' 003425                  BLE     OKX
000714' 021011          1$:     CMP     @R0,@R1         ;DATA GOOD?
000716' 001417                  BEQ     OK              ;BR=YES
000720' 005267 001604           INC     COUNT
000724' 011067 177156           MOV     @R0,ASB         ;LOAD ERROR INFO
000730' 011167 177154           MOV     @R1,AWAS
000734' 010067 177142           MOV     R0,BBADR
000740' 013167 177140           MOV     R1,WASADR
000744' 016767 002620 177126            MOV     MTC,CSRA
                                ;*****************************************************************
000752' 104404 000000'          DATERS,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************
000756' 022021          OK:     CMP     (R0)+,(R1)+     ;STEP POINTERS
000760' 022700 004116'          CMP     #CBUFEN,R0      ;DONE?
000764' 001343                  BNE     DAT1            ;BR=NO
000766'                 OKX:
000766' 012700 004120'  BUF:    MOV     #IBUF,R0                ;CLEAR OUT THE INPUT BUFFER
000772' 005020          1$:     CLR     (R0)+
000774' 022700 004152'          CMP     #IBUF+26.,R0    ;DONE?
001000' 001374                  BNE     1$              ;BR=NO
001002'                 PASS:
001002' 104413 000000'          ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                        ;MONITOR SHALL TEST END OF PASS
001006' 000627                  BR      NEXT
001010'                 FINI:
001010' 104410 000000'          ENDS,BEGIN              ; DROP THE MODULE
                                ;       -------------------------
                                ;
                                ;
001014' 105267 002772   RETRY1: INCB    TRY1            ; COUNT THE RETRYS
001020' 122767 000003 002764            CMPB    #3,TRY1         ; LIMIT EXCEEDED ?
001026' 001404                  BEQ     1$              ; YES, GO REPORT IT
001030' 004567 000122           JSR     R5,BACKSP       ; NO, BACKUP TO TRY AGAIN
001034' 000240                  NOP                     ;  ERROR RETURN
```



```
001036' 000657                  BR      RITE            ; GO TRY AGAIN
001040'                 1$:
001040' 104403 000000' 003754'  MSGNS,BEGIN,EXCED1      ;ASCII MESSAGE CALL WITH COMMON HEADER
001046' 000413                  BR      NEXTA           ; GO ON TO NEXT DRIVE
                                ;       -------------------------
001050' 105267 002737   RETRY2: INCB    TRY2            ; COUNT RETRYS
001054' 122767 000003 002731            CMPB    #3,TRY2         ; LIMIT EXCEEDED ?
001062' 001401                  BEQ     1$              ; YES, GO REPORT IT
001064' 000663                  BR      BACK            ; NO, BACKUP TO TRY AGAIN
001066'                 1$:
001066' 104403 000000' 003762'  MSGNS,BEGIN,EXCED2      ;ASCII MESSAGE CALL WITH COMMON HEADER
001074' 000240                  NOP                     ; GO ON TO NEXT DRIVE
                                ;       -------------------------
```

```
                                ;
                                ;       -----  TM11 TAPE DRIVERS  -----
                                ;
001124' 012767  000103  001402  WRITE:  MOV     #103,FUNC       ; LOAD WRITE FUNCTION
001132' 016777  001420  002432          MOV     WCNT1,@MTBRC    ; LOAD BYTE COUNT
001140' 016777  002652  002426          MOV     CRUFPA,@MTCMA   ; LOAD BUFFER ADDRESS
001146' 016767  002646  001362          MOV     CRUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS
001154' 000425                          BR      GOGO            ; CONTINUE
001156' 012767  000111  001350  BACKSP: MOV     #111,FUNC       ; LOAD BACKSPACE FUNCTION
001164' 012777  177777  002400          MOV     #-1,@MTBRC      ; LOAD BYTE COUNT
001172' 005067  001340                  CLR     XMEM            ;CLEAR EXTENDED ADDRESS BITS
001176' 000414                          BR      GOGO            ; CONTINUE
001200' 012767  000105  001326  READ:   MOV     #105,FUNC       ; LOAD READ FUNCTION
001206' 016777  001346  002356          MOV     WCNT2,@MTBRC    ; LOAD BYTE COUNT
001214' 016777  002604  002352          MOV     IRUFPA,@MTCMA   ; LOAD BUFFER ADDRESS
001222' 016767  002600  001306          MOV     IRUFEA,XMEM     ; LOAD EXTENDED MEMORY BITS

001230' 012777  001274' 176552  GOGO:   MOV     #NTRUPT,@VECTOR ; SET INTERUPT ENTRY POINTER
001236' 056767  001304  001270          BIS     DRVSFT,FUNC     ; LOAD DRIVE UNIT NUMBER
001244' 000367  001266                  SWAB    XMEM            ;ADJUST FOR TM79
001250' 056767  001262  001256          BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
001256' 005077  002304                  CLR     @MTS            ;CLEAR INHIBIT
001262' 016777  001246  002300          MOV     FUNC,@MTC       ; EXECUTE THE FUNCTION
001270' 104400  000000'                 EXITS,REGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

001274'                         NTRUPT:
                                ;-------------------------------------------------------------------
001274' 000004  000000' 001302'         $IRQS,REGIN,1S          ; QUEUE UP TO CONTINUE AT 1S AND RTI
                                ;-------------------------------------------------------------------

001302' 004567  000654          1S:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
001306' 000205                          RTS     R5              ; ERRORS DETECTED, RETURN
001310' 005725                          TST     (R5)+           ; NO ERRORS, SKIP RETRY
001312' 000205                          RTS     R5              ; RETURN OK
                                ;       -------------------------
```



```
001314' 016767  001220  001220  REWIND: MOV     DVICE,DRIVE     ; GET ACTIVE DRIVES
001322' 012767  000004  001220          MOV     #4.,DVCNUM      ; LOAD MAXIMUM NUMBER OF DRIVES
001330' 016701  001214                  MOV     DVCNUM,R1       ; PUT IT INTO A COUNTER

001334' 000241                  1$:     CLC                     ; MAKE SURE C-BIT IS CLEAR
001336' 006067  001200                  ASR     DRIVE           ; CHECK FOR ACTIVE DRIVE
001342' 103402                          BCS     2$              ; IT'S ACTIVE --- BRANCH
001344' 005367  001200                  DEC     DVCNUM          ; NOT ACTIVE, SUBTRACT FROM TOTAL
001350' 005301                  2$:     DEC     R1              ; ALL 4 CHECKED ?
001352' 003370                          BGT     1$              ; NO, CONTINUE

001354' 012767  000062  002202          MOV     #50.,CLK1       ; LOAD THE 2ND TIMER
001362' 012767  177777  001154          MOV     #-1,DRVVE       ; INITIALIZE THE DRIVE COUNTER
001370' 005067  001156                  CLR     DVCNT           ; CLEAR DEVICE COUNTER FOR ISR
001374' 016767  001140  001140          MOV     DVICE,DRIVE     ; RESTORE DRIVE INDICATOR
001402' 012767  177400  001136          MOV     #-400,DRVSFT    ; INITIALIZE SHIFTED DRIVE NUMBER
001410' 004767  000322          3$:     JSR     PC,DRVADR       ; GO GET A DRIVE NUMBER
001414' 132767  000010  002366          BITB    #BIT3,FLAG      ; ALL DRIVES DONE ?
001422' 001025                          BNE     4$              ; YES, GO WAIT FOR COMPLETION
001424' 004767  001024                  JSR     PC,WAIT1        ; CONTROLLER READY ?
001430' 042777  001400  002132          BIC     #1400,@MTC      ; YES, CLEAR OUT OLD UNIT NUMBER
001436' 016767  001104  001070          MOV     DRVSFT,FUNC     ; LOAD NEW UNIT NUMBER
001444' 005077  002116                  CLR     @MTS            ;CLEAR INHIBIT
001450' 032777  000040  002110          BIT     #40,@MTS        ;AT LOAD PT ALLREADY?
001456' 001006                          BNE     33$             ;YES DON'T REWIND
001460' 052767  000021  001046          BIS     #21,FUNC        ; LOAD REWIND FUNCTION
001466' 016777  001042  002074          MOV     FUNC,@MTC       ; EXECUTE THE REWIND
001474' 000745                  33$:    BR      3$              ; GO ON FOR THE NEXT DRIVE

001476' 012767  077777  002056  4$:     MOV     #77777,CLK      ; SET THE TIMER
001504' 004767  000206  5$:     JSR     PC,DRVADR       ; GO GET A DRIVE NUMBER
001510' 132767  000010  002272          BITB    #BIT3,FLAG      ; ALL DRIVES DONE ?
001516' 001067                          BNE     8$              ; YES, GET OUT
001520' 042777  001400  002042          BIC     #1400,@MTC      ; CLEAR OUT OLD UNIT NUMBER
001526' 056777  001014  002034          BIS     DRVSFT,@MTC     ; LOAD NEW UNIT NUMBER
001534'                         6$:
001534' 104407  000000'                 BREAK$,REGIN            ;TEMPORARY RETURN TO MONITOR....
001540' 104407  000000'                 BREAK$,REGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001544' 032777  000040  002014          BIT     #40,@MTS        ; DRIVE  AT BOT ?
001552' 001416                          BEQ     66$             ;NO GO WAIT
001554' 032777  000200  002006          BIT     #BIT7,@MTC      ;IS DRIVE READY?
001562' 001412                          BEQ     66$             ;NO GO WAIT!
001564' 005077  001776                  CLR     @MTS            ;CLEAR INHIBIT
001570' 052777  000033  001772          BIS     #33,@MTC        ;WRITE IDB
001576' 032777  000200  001764  61$:    BIT     #BIT7,@MTC      ;WAIT FOR READY
001604' 001774                          BEQ     61$
001606' 000736                          BR      5$              ;  GO CHECK THE NEXT DRIVE
001610' 005367  001746          66$:    DEC     CLK             ;  OUT OF TIME ?
001614' 003347                          BGT     6$              ; NO, WAIT SOME MORE
001616' 005367  001742                  DEC     CLK1            ; YES, WANT ANOTHER 40 SECONDS ?
001622' 001407                          BEQ     7$              ; NO, TIME-OUT
001624' 012767  077777  001730          MOV     #77777,CLK      ; RESET THE TIMER
001632' 000740                          BR      6$              ; WAIT SOME MORE
001634' 104403  000000' 003750'         MSGNS,REGIN,RWDERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
```

```
001642' 012767  000013  176236  7$:     MOV     #13,ERRTYP      ;REWIND ERROR
                                        ;*****************************************************************
001650' 104405  000000' 000000          ERDES,BEGIN,NULL        ; REWIND ERROR, REWIND NOT COMPLETE
                                        ;*****************************************************************

001656' 000207                  8$:     RTS     PC              ; RETURN
                                        ;       -------------------------


001660' 012701  000001          DROP:   MOV     #1,R1           ; INITIALIZE DROP PICKER
001664' 116700  000654          1$:     MOVB    DRVVE,R0        ; GET THE DRIVE NUMBER
001670' 001403                  2$:     BEQ     2$              ; IF DRIVE 0 GO DROP IT
001672' 006301                          ASL     R1              ; NO, AIM AT THE NEXT DRIVE
001674' 005300                          DEC     R0              ; IS THIS THE ONE ?
001676' 001375                          BNE     1$              ; NO, LOOK AGAIN
001700' 040167  000634          2$:     BIC     R1,DVICE        ; DROP THE DRIVE
                                        ;*****************************************************************
                                        ;CONVERT DRVVE TO ASCII AND
001704' 104420  000000' 002544'         OTOA$,BEGIN,DRVVE,ADR1  ;STORE AT ADR1
001712' 004000'                         ;*****************************************************************
001714' 000207                          RTS     PC              ; RETURN
                                        ;       -------------------------


001716' 005267  000622          DRVADR: INC     DRVVE           ; COUNT A DRIVE
001722' 062767  000400  000616          ADD     #BIT8,DRVSFT    ; DRIVE COUNT LINED UP WITH MTC
001730' 142767  000010  002052          BICB    #BIT3,FLAG      ; CLEAR END OF DRIVES FLAG
001736' 022767  000004  000600          CMP     #4.,DRVVE       ; ALL DRIVES CHECKED ?
001744' 001404                          BEQ     1$              ; YES, GO FLAG END OF DRIVES
001746' 006367  000570                  ASR     DRIVE           ; NO, IS NEXT DRIVE CHOSEN ?
001752' 103361                          BCC     DRVADR          ; NO, GO TRY ANOTHER DRIVE
001754' 000207                  1$:     RTS     PC              ; RETURN


001756' 152767  000010  002024  1$:     BISB    #BIT3,FLAG      ; SET END OF DRIVES FLAG
001764' 012767  177777  000552          MOV     #-1,DRVVE       ; RESET DRIVE COUNTER
001772' 012767  177400  000546          MOV     #-400,DRVSFT    ; ZERO THE SHIFTED DRIVE #
002000' 016767  000534  000534          MOV     DVICE,DRIVE     ; RESTORE CHOSEN DRIVES
002006' 000207                          RTS     PC              ; RETURN
                                        ;       -------------------------


002010' 012767  177777  000526  NOTRDY: MOV     #-1,DRVVE       ; START WITH FIRST DRIVE
002016' 012767  177400  000522          MOV     #-400,DRVSFT    ;
002024' 016767  000510  000510          MOV     DVICE,DRIVE     ; RESET DRIVE SELECT
002032' 004767  177660          1$:     JSR     PC,DRVADR       ; GET A DRIVE ADDRESS
002036' 132767  000010  001744          BITB    #BIT3,FLAG      ; ALL DRIVES CHECKED ?
002044' 001312                          BNE     2$              ; YES, RETURN
002046' 016777  000474  001514          MOV     DRVSFT,@MTC     ; NO, LOAD NEXT DRIVE ADDRESS
002054' 032777  000200  001506          BIT     #BIT7,@MTC      ; IS THIS DRIVE READY ?
002062' 001363                          BNE     1$              ; YES, CONTINUE
002064' 004767  000022                  JSR     PC,WAIT         ; NO, WAIT FOR IT
002070' 000760                          BR      1$              ; GO CHECK REST OF DRIVES
002072' 000207                  2$:     RTS     PC              ; RETURN
```



```
                                        ;       -------------------------


002074' 016767  001466  175776  ERSUP1: MOV     MTS,CSRA        ; LOAD ADR. OF CURRENT CSR
002102' 117767  001460  175772          MOVB    @MTS,ACSR       ; LOAD CONTENTS OF CURRENT CSR
002110' 000207                          RTS     PC              ; RETURN
                                        ;       -------------------------
```

```
                                ;
                                ;
                                ;
                                ;
002112' 012767  077777  001442  WAIT:   MOV     #77777,CLK      ; SET THE TIMER
002120'                         1$:
002120' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002124' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
                                ;
002130' 032777  000200  001432          BIT     #BIT7,@MTC      ; DRIVE READY ?
002136' 001010                          BNE     2$              ; YES, RETURN
002140' 005367  001416                  DEC     CLK             ; NO, WAIT SOME MORE ?
002144' 001365                          BNE     1$              ; YES, WAIT
002146' 004767  177506                  JSR     PC,DROP         ; TIME-OUT, DROP THE DRIVE
002152' 104403  000000' 003770'         MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
002160' 000207                  2$:     RTS     PC              ; RETURN
                                ;       -------------------------
                                ;
                                ;
                                ;
                                ;
002162' 004767  177706          ERRORS: JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
002166' 032777  000200  001372          BIT     #BIT7,@MTS      ; AT END OF TAPE ?
002174' 001403                          BEQ     1$              ; NO, CHECK FOR ERROR
002176' 152767  000004  001604          BISB    #BIT2,FLAG      ; YES, SET EOT FLAG
002204' 005777  001360          1$:     TST     @MTC            ;ANY ERRORS?
002210' 100055                          BPL     4$              ;BR = NO
                                ;
002212' 012701  003706'                 MOV     #BADSPT,R1              ;SET BAD POINTER
002216' 021167  000332          22$:    CMP     (R1),CYCKNT             ;MATCH?
002222' 001403                          BEQ     33$                     ;YES
002224' 005721                          TST     (R1)+                   ;NO TRY AGAIN
002226' 100373                          BPL     22$
002230' 000414                          BR      2$                      ;NEW ERROR
002232' 012777  004000  001330  33$:    MOV     #BIT11,@MTC             ;POWER CLEAR
002240' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002244' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002250' 005077  001312                  CLR     @MTS            ;CLEAR INHIBIT
002254' 022626                          CMP     (SP)+,(SP)+
002256' 000167  176204                  JMP     NEXT
                                ;
002262' 012767  000001  175616  2$:     MOV     #1,ERPTYP       ;DATA ERROR
                                ;*****************************************************************
002270' 104406  000000' 003566'         SCOPE$,BEGIN,TABLE      ;
                                ;*****************************************************************
002276' 012777  004000  001264          MOV     #BIT11,@MTC     ; ISSUE A POWER CLEAR
002304' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002310' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002314' 005077  001246                  CLR     @MTS            ;CLEAR INHIBIT
002320' 000412                          BR      5$              ; RETURN TO DO A(RETRY
                                ;
002322' 012777  004000  001240  3$:     MOV     #BIT11,@MTC     ; ISSUE A POWER CLEAR
002330' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002334' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002340' 005077  001222                  CLR     @MTS            ;CLEAR INHIBIT
```



```
002344' 005725                  4$:     TST     (R5)+           ; SKIP THE RETRY
002346' 000205                  5$:     RTS     R5              ; RETURN
                                ;       -------------------------
                                ;
```

```
                        ;
                        ;
002350' 016706  175432          SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
002354' 010067  001210          MOV     R0,MTC          ; GENERATE CONTROLLER REGS. ADDRESSES
002360' 005720                  TST     (R0)+
002362' 010067  001200          MOV     R0,MTS
002366' 005720                  TST     (R0)+
002370' 010067  001176          MOV     R0,MTBRC
002374' 005720                  TST     (R0)+
002376' 010067  001172          MOV     R0,MTCMA
                        ;
002402' 016700  175402          MOV     VECTOR,R0       ; GET THE VECTOR ADDRESS
002406' 116760  175403  000002  MOVB    BP1,2(R0)       ; SET PRICRITY
                        ;
002414' 000207          2$:     RTS     PC              ; RETURN
                        ;
                        ;
                        ;
                        ;
002416' 012777  004000  001144  REZET:  MOV     #BIT11,@MTC     ; EXECUTE POWER CLEAR
002424' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002430' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002434' 012777  000001  001124  MOV     #BIT0,@MTS      ;CLEAR INHIBIT
002442' 004767  000006          JSR     PC,WAIT1        ; GO WAIT FOR CONTROLLER READY
002446' 004767  177336          JSR     PC,NOTRDY       ; MAKE SURE ALL CHOSEN DRIVES ARE READY
002452' 000207          2$:     RTS     PC              ; RETURN
                        ;
                        ;
                        ;
                        ;
002454' 012767  077777  001100  WAIT1:  MOV     #77777,CLK      ; SET THE TIMER
002462' 105777  001102          1$:     TSTB    @MTC            ; CONTROLLER READY ?
002466' 100417                  BMI     2$              ; YES, CONTINUE
002470' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002474' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002500' 005367  001056          DEC     CLK             ; WAIT SOME MORE ?
002504' 001366                  BNE     1$              ; YES
002506' 012767  000003  175372  MOV     #3,ERRTYP       ;CONTROLLER NOT READY
                        ;*****************************************************************
002514' 104405  000000' 003566' ERROR$,BEGIN,TABLE      ; CONTROLLER NOT READY
                        ;*****************************************************************
002522' 000167  176262          JMP     FINI            ; GO  DROP THE MODULE
002526' 000207          2$:     RTS     PC              ; READY, RETURN
                        ;
                        ;
```



```
                        ;
002532' 000000          COUNT:  0
002532' 000000          CNT:    0
002534' 000000          FUNC:   0
002536' 000000          XMEM:   0
002540' 000000          DVICE:  0
002542' 000000          DRIVE:  0
002544' 000000          DRVVEC: 0
002546' 000000          DRVSFT: 0
002550' 000000          DVCNUM: 0
002552' 000000          DVCKNT: 0
002554' 000000          CYCKNT: 0
002556' 000000          WCNT1:  0
002560' 000000          WCNT2:  0
002562' 000400          BUFIN:  .BLKW   256.
003562' 000000          CLK:    0
003564' 000000          CLK1:   0
003566'                 TABLE:
003566' 000000          MTS:    0
003570' 000000          MTC:    0
003572' 000000          MTBRC:  0
003574' 000000          MTCMA:  0
003576' 002554'         CYC:    CYCKNT
003600' 177777                  177777
003602' 020040  051040  053505  MES1:   .ASCIZ  "  REWIND ERROR"
003610' 047111  020104  051105
003616' 047522  000122
003623' 020040  051104  053111  MES2:   .ASCIZ  "  DRIVE  "
003634' 020105  000040
003643' 020040  051104  050117  MES3:   .ASCIZ  "  DROPPED%"
003642' 042526  022504  000
003647'     040 042522  051124  MES4:   .ASCIZ  " RETRY EXCEEDED%"
003654' 020131  054105  042503
003662' 042105  000045
003670' 053440  044522  042524  MES5:   .ASCIZ  " WRITE"
003676'     000
003677'     040 042522  042101  MES6:   .ASCIZ  " READ"
003704'     000
                003706'                 .EVEN
                        ;
                        ;
003706' 177777          BADSPT: -1
003710' 177777                  -1
003712' 177777                  -1
003714' 177777                  -1
003716' 177777                  -1
003720' 177777                  -1
003722' 177777                  -1
003724' 177777                  -1
003726' 177777                  -1
003730' 177777                  -1
003732' 177777                  -1
003734' 177777                  -1
003736' 177777                  -1
003740' 177777                  -1
003742' 177777                  -1
```

```
003744' 177777                  -1
003746' 177777                  -1

003750' 003602'         RWDERR: MES1
003752' 177777                  177777
003754' 003670'         EXCED1: MES5
003756' 003647'                 MES4
003760' 177777                  177777
003762' 003677'         EXCED2: MES6
003764' 003647'                 MES4
003766' 177777                  177777
003770' 003622'         DRP:    MES2
003772' 004065'                 NUMR
003774' 003634'                 MES3
003776' 177777                  177777
004000' 000005          ADR1:   .BLKB   5
004005'     000         NUMR:   .BYTE   0
004006' 000000                  .WORD   0
004010'     000         FLAG:   .BYTE   0
004012'                         .EVEN
004012'     000         TRY1:   .BYTE   0
004013'     000         TRY2:   .BYTE   0
                                .EVEN

004014' 004034'         OBUFVA: OBUF
004016' 000000          OBUFFA: 000
004020' 000000          OBUFBA: 000
004022' 004120'         IBUFVA: IBUF
004024' 000000          IBUFFA: 000
004026' 000000          IBUFBA: 000

004030' 000031          OBUFSZ: 25.
004032' 000031          IBUFSZ: 25.

004034' 000001          OBUF:   1                       ;OUTPUT BUFFER
004036' 000002                  2
004040' 000004                  4
004042' 000010                  10
004044' 000020                  20
004046' 000040                  40
004050' 000100                  100
004052' 000200                  200
004054' 000652                  652
004056' 000525                  525
004060' 000652                  652
004062' 000525                  525
004064' 000625                  625
004066' 000525                  525
004070' 000625                  625
004072' 000525                  525
004074' 000400                  400
004076' 000400                  400
004100' 000400                  400
004102' 000400                  400
004104' 000777                  777
004106' 000777                  777
```



```
004110' 000777                  777
004112' 000777                  777
004114' 000523                  523
004116' 000000          OBUFEN: 000                     ;OUTPUT BUFFER END

004120' 000000          IBUF:   000                     ;INPUT BUFFER
        004202'                 .=.+48.

        000001                  .END
```

```
ACSR    000102R          203#    519*
ADDR    000166R          169#    583
ADDR22= 001000           121#
ADR1    004000R          478     699#
ASR     000100R          207#    314*
ASTAT   000104R          205#
AWAS    000110R          208#    315*
BACK    000634R          293     298#    352
BACKSP  001166R          298     342     374#
BADSPT  003706R          550     566#
BEGIN   000000R          166#    247     261     266     280     320     331     335     346     354     362     390     394
                         440     441     457     460     478     528     529     536     557     558     566     569     570
                         575     576     602     603     616     617     622
BIT0    = 000001         221#    359     604
BIT1    = 000002         221#
BIT10   = 002000         221#    307
BIT11   = 004000         221#    556     568     574     601
BIT12   = 010000         221#
BIT13   = 020000         221#
BIT14   = 040000         221#    234
BIT15   = 100000         221#
BIT2    = 000004         221#    277     292     294     546
BIT3    = 000010         221#    273     421     435     487     494     506
BIT4    = 000020         221#
BIT5    = 000040         221#
BIT6    = 000100         221#
BIT7    = 000200         221#    282     444     448     509     531     544
BIT8    = 000400         221#    486
BIT9    = 001000         221#
BREAKS= 104407           221#    440     441     528     529     557     558     569     570     575     576     602     603
                         616     617
BR1     000012R          171#    583
BR2     000013R          172#
BTODS = 104421           221#
BUF     000766R          326#
BUFIN   002552R          642#
CDATAS= 104412           221#
CLK     003562R          433*    451*    455*    526*    533*    613*    618*    643#
CLK1    003564R          415*    453*    644#
CNT     002532R          636#
CONFIG  000656R          191#
COUNT   002530R          304*    309     313*    629#
CSRA    000156R          201#    318*    518*
CVC     003576R          650#
CYCNT   002554R          258*    287*    296*    551     639#    650
DATCK   000660R          302#
DATCKS= 104411           221#
DATERS= 104404           221#    320
DATI    000674R          307#    324
DRIVE   002545R          230*    233     404*    409*    418*    490*    497*    504*    634#
DROP    001166R          246     279     361     468#    535
DRP     003772R          247     280     362     536     695#
DRVADR  001716R          270     420     434     485#    491     505
DRVSET  002546R          232*    276     385     419*    425     438     486*    496*    503*    508     636#
DRYVE   002544R          231*    245*    251*    416*    469     478     485*    488     495*    502*    635#
DVCKNT  002552R          417*    638#
```



```
DVCNUM  002550R          405*    406     411*    637#
DVICE   002540R          229*    230     254     271     404     418     474*    497     504     633#
DVID1   002514R          173#    539
ENDITS= 104433           221#    331
ENDS  = 104410           221#    335
ERRORS  002562R          597     643#
ERRTYP  002566R          206#    458*    564*    620*
ERSUB1  002574R          516#    543
EXCEP1  003754R          346     689#
EXCEP2  003762R          354     692#
EXITS = 104400           221#    360
FINI    001010R          256     272     334#    624
FLAG    004310R          228*    273     292     294*    421     435     487*    494*    506     546*    702#
FUNC    002534R          369*    374*    378*    385*    387*    389     425*    429*    430     631#
GETPAS= 104416           221#    261     266
GOGO    001230R          373     377     384#
GWBUFS= 104414           221#
HRDCNT  003044R          186#
HRDERS= 104405           221#    460     622
HRDPAS  000650R          188#
IBUF    004120R          326     328     711     745#
IBUFEA  004026R          381     713#
IBUFPA  004024R          380     712#
IBUFSZ  004032R          262     716#
IBUFVA  004022R          261     711#
ICONT   003036R          183#
ICOUNT  000640R          194#
IDNUM   000122R          213#
INIT    000030R          180#
INTR    000132R          212#    227*
MAP22S= 104416           221#
MES1    003622R          653#    687
MES2    003626R          655#    695
MES3    003634R          657#    697
MES4    003647R          659#    690     693
MES5    003670R          662#    684
MES6    003677R          664#    692
MODNAM  000666R          187#
MODSP   000224R          191     219#
MSGNS = 104403           221#    247     280     346     354     362     457     536
MSGSS = 104402           221#
MSGS  = 104401           221#
MTBRC   003572R          376*    375*    379*    588*    648#
MTC     003570R          275*    276*    282     318     389*    424*    430*    437*    438*    444     447*    448     508*
                         520     531     547     556*    568*    574*    584*    601*    614     647#
MTCMA   003574R          377*    380*    590*    649#
MTS     003566R          277     388*    426*    427     442     446*    518     519     544     559*    571*    577*    586*
                         604*    646#
NEXT    000466R          270#    281     333     363     561
NEXTA   001076R          347     359#
NOTRDY  002016R          284     502#    606
NTRUPT  001274R          394     395#
NULL  = 000000           221#    460
NUMB    004006R          696     700#
OBUF    004034R          305     718     718#
OBUFEA  004020R          372     710#
```

```
OBUFEN  004116R          323     743#
OBUFPA  004016R          311     709#
OBUFSZ  004030R          267     715#
OBUFVA  004014R          266     708#
OK      000756R          312     322#
OKX     000766R          316     325#
OPEN    = 000000         168     174     175     176     177     194     195     196     197     198     199     200     201
                         203     205     207     208     210     211     212     221#
OTOA$ = 104420           221#    478
PASCNT  000034R          182#    223
PASS    001002R          330#
PIRQ$ = 000004           221#    394
POPSP = 005726           221#
POPSP2= 222626           221#
PRTV  = 000000           221#
PRTV0 = 000000           172     221#
PRTV1 = 000040           221#
PRTV2 = 000100           221#
PRTV3 = 000140           221#
PRTV4 = 000200           171     221#
PRTV5 = 000240           221#
PRTV6 = 000300           221#
PRTV7 = 000340           221#
PS    = 177776           221#
PSW   = 177776           221#
PUSH  = 005746           221#
PUSH2 = 024646           221#
RAND$ = 104417           221#
RANNUM  000354R          190#
READ    001200R          300     378#
REED    000642R          300#
RESTRT  000224R          209     223#
REST1   000426R          224     260#
RES1    000066R          192#
RES2    000060R          193#
RETRY1  001014R          291     339#
RETRY2  001050R          301     349#
REWIND  001314R          257     295     404#
REZET   002416R          253     601#
RITE    000576R          290#    344
RSTRT   000112R          208#
RWDERR  003760R          457     687#
SBADR   000162R          202#    316*
SETUP   002386R          255     583#
SOFCNT  000042R          185#
SOFERS= 104416           221#    566
SOFPAS  000046R          187#
SPOINT  000332R          181#
SPSIZ = 000040             1#    214
SR    = ****** G         222#    307
SR1     000176R          174#    359
SR2     000200R          175#
SR3     000222R          176#
SR4     000244R          177#
START   002232R          186     225#
STAT    000226R          179#
```



```
STRT    002446R          265#    274     285     297
SVRC    000062R          194#    302*
SVR1    000064R          195#    303*
SVR2    000066R          196#
SVR3    000070R          197#
SVR4    000072R          198#
SVR5    000074R          199#
SVR6    000076R          200#
SVSCNT  000052R          189#
TABLE   003566R          566     622     645#
TRPDPD= 003422           221#
TRY1    004512R          336*    339*    340     704#
TRY2    004618R          340*    350     705#
VECTOR  000100R          170#    384*    592
WAIT    002112R          511     526#
WAIT1   002454R          423     605     613#
WASADR  000164R          204#    317*
WCNT1   002566R          267*    368*    370     640#
WCNT2   002560R          262*    363*    379     641#
WDFR    000116R          211#    225*
WDTO    000114R          210#    226*
WRITE   001124R          290     369#
XFLAG   000065R          168#
XMEM    002536R          372*    376*    381*    386*    387     632#
.       = 004202R        642#    666#    698#    703#    746#
```

```
. ABS.  000000      000
        004202      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XTRACO,XTRACO/SOL/CRF:SYM=DDXCOM,XTRACO
 RUN-TIME:  1  2  .3  SECONDS
 RUN-TIME RATIO: 22/4=5.1
 CORE USED:  7K  (13 PAGES)
```

DIAGNOSTIC ENGINEERING

digital **DECO☐ DEPO☒ SUBMISSION☐** ☐ NEW FOR RELEASE ENG. USE ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

MD	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE DD MMM YY	STATUS	DISTRIBUTION	1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
	ZZ	CXTRA	C	1	01	01 NOV 78	☐ OBSOLETE	X G ☐ R	1976	1978

TITLE CXTRAC1 TR79F MODULE

AUTHOR D.BUTENHOF | 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-E896C-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 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
			133	X Y ☐ N	X Y ☐ N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

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

.T NUMBERS							
	ZJ130-RB	ZJ129-RZ,FR					

PROBLEM:

DOES BIT TEST FOR LOAD MEDIA CHECK INSTEAD OF COMPARE; WILL DROP SELF IF ACTUAL LOAD MEDIA SETS IT'S DESIGNATED BIT.

UNCONDITIONAL PATCH

SOLUTION:

PATCH BIT TEST TO COMPARE BYTE

DEPO PATCH AREA

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO
306	132737	122737			
308	40000	100			

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#25Ø4
8-NOV-78	DATE:	DATE:	

DIAGNOSTIC ENGINEERING

digital **DECO☐ DEPO☒ SUBMISSION☐**

FOR RELEASE ENG. USE ☐ NEW ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

MD	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE DD MMM YY	STATUS	DISTRIBUTION	1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
	ZZ	CXTRA	C	2	62	12 DEC 78	☐ OBSOLETE	X G ☐ R	1976	1978

TITLE: CXTRAC2 TR79F MODULE

AUTHOR: D.BUTENHOF | MAINTAINING GROUP: DEC/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-E896C-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
			X Y ☐ N	X Y ☐ N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

SERVICE AFFECTED: DEC/X11 | MULTIMEDIA AFFECTED? ☐ YES ☒ NO

KIT NUMBERS	ZJ130-RB	ZJ129-RZ,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
314	1424	1024			

SOFTWARE ENGINEER	MANUFACTURING ENGINEER	SUPPORT ENGINEER	CHARGE DECO DEPO TO DISCRETE PROJECT NUMBER Q98-05314
DATE: 13 DEC 78	DATE:	DATE:	
MAINTAINER	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. MC2787
13-DEC-78	DATE:	DATE	

DIAGNOSTIC ENGINEERING

digital **DECO☐ DEPO☒ SUBMISSION☐**

☐ NEW FOR RELEASE ENG. USE ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

	LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE	STATUS	DISTRIBUTION	1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
...D	ZZ	CXTRA	C	3	13	18 DEC 78	☐ OBSOLETE	K G ☐ R	1976	1978

TITLE: CXTRAC3 TR79F MODULE

AUTHOR: J.BUTENHOF | MAINTAINED BY: DEC SPT GRP | MAINTAINER: D.BUTENHOF | SUBMITTED BY: ENGINEER

PRODUCT COMPONENTS

CK	DESCRIPTION	PRODUCT NO	REV	CK	DESCRIPTION	PRODUCT NO.	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
X		AF-E896C-M3					

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 3435 40 4550 55 6070

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
				XX Y ☐ N	XX Y ☐ N	SECONDS	SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

SERVICE AFFECTED: DPC/XXDP | MULTIMEDIA AFFECTED? ☒ YES ☐ NO

KIT NUMBERS	ZJ130-RB	ZJ129-RZ,FR				

PROBLEM:

CXTRAC does an illegal global reference to the old monitor switch register. It also hangs on a tape badspot.

SOLUTION:

Install the following patch

DEPO PATCH AREA

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO
674	22777	240			
676	2000	240			
700	(undefined)	240			
702	1004	240			
1022	3	2			
1056	3	2			
2276	12777	205			

PATCH ENGINEER	MANUFACTURING ENGINEER	SUPPORT ENGINEER	CHARGE DECO DEPT TO DISCRETE PROJECT NUMBER
DATE:	DATE:	DATE:	Q9805460
	FIELD SERVICE	WAIVER MANAGER	COORDINATION NO: mc#2805
18 dec-c	DATE:	DATE:	