```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E685G-MC

PRODUCT NAME: CXTCAG0 DEC/X11 TC11 MODULE

DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

TCA EXERCISES A TC11 DECTAPE CONTROL AND UP TO EIGHT (8) DECTAPE DRIVES. BASIC TEST SEQUENCE CONSISTS OF WRITING 1024 WORDS (4 BLOCKS) IN FORWARD DIRECTION, READING 1ST 256 WORDS (1 BLOCK) IN FORWARD DIRECTION, AND THEN CHECKING THE DATA. THE BASIC SEQUENCE IS REPEATED USING A DIFFERENT DRIVE EACH TIME, UNTIL A PASS IS COMPLETED. WITH EACH WRITE OR READ BEING PRECEEDED BY A SEARCH SEQUENCE. EACH SEARCH AND DATA TRANSFER OR DATA CHECK IS RETRYED UP TO A LIMIT AND THEN THE DRIVE IS EITHER,DEPENDING ON SR1, DROPPED OR THAT BLOCK IS SKIPPED. WHEN THE END OF TAPE IS REACHED, THEN THE READS AND WRITES GO IN REVERSE UNTIL THE BEGINING OF THE TAPE IS REACHED, ETC.

2. REQUIREMENTS

HARDWARE: TC11 DECTAPE CONTOL, AND ONE TU56 DUAL DECTAPE TRANSPORT.
STORAGE:: TCA REQUIRES:
1. DECIMAL WORDS: 895
2. OCTAL WORDS: 1577
3. OCTAL BYTES: 3376

3. PASS DEFINITION

ONE PASS OF TCA MODULE CONSISTS OF 40 ITERATIONS OF BASIC TEST SEQUENCE, WHICH RESULTS IN:

160 BLOCKS WRITTEN, 40 BLOCKS READ.

4. EXECUTION TIME

TCA RUNNING ALONE, WITH ONE DECTAPE DRIVE, ON PDP-11/05 TAKES APPROXIMATELY 1 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 177340, VECTOR: 214, BR1: 6, DEVCNT: 1, SR1: 0

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

EACH DECTAPE DRIVE MUST BE:

A. LOADED WITH A FORMATTED DECTAPE.

B. SET TO REMOTE.

C. WRITE ENABLED.

7. MODULE OPERATION

TEST SEQUENCE:

A. SELECT A DRIVE (ERROR AND MODULE DROPPED IF NONE AVAILABLE).

B. WRITE 4 BLOCKS FWD (1024 WORDS).

C. READ THE FIRST BLOCK WRITTEN (256 WORDS).

D. CHECK DATA (256 WORDS).

E. REPEAT A THROUGH D 40 TIMES FOR ONE PASS.

NOTES: TCA DOES NOT USE DRIVE 0 IF LOAD MEDIUM IS DECTAPE.

8. OPERATION OPTIONS

MODULE LOCATION "DVID1" MAY BE CHANGED TO TEST OTHER THAN A FULL COMPLEMENT OF DRIVES. "DVID1" BITS 0 THROUGH 7 ONLY APPLY. ONE BIT INDICATES A DRIVE. BIT0= DRIVE 0, ETC.

LOCATION "RTLMT" CONTAINS A 2 TO INDICATE 3 RETRYS. THIS MAY BE CHANGED FROM 0=NO RETRYS UP TO 377=256 RETRYS.

SR1 IS A 0 CAUSING THE MODULE TO SKIP AN OFFENDING BLOCK AFTER THE RETRY LIMIT IS EXCEEDED. IF A 1 IS PUT IN BIT 0 THE DRIVE WILL BE DROPPED WHEN THE LIMIT IS EXCEEDED.

9. NON STANDARD PRINTOUTS

NONE. ALL PRINTOUTS HAVE STANDARD MEANINGS AS REPRESENTED IN DEC/X11 DOCUMENTATION.

THERE IS AN ERROR MSG FOR EVERY ERROR AND THERE ARE EXPLAINATORY MESSAGES THAT COME WITH SOME BUT NOT ALL. THESE EXTRA MESSAGES ALSO INCLUDE THE FOLLOWING ENDING
D<X>B<YYY>
WHERE X IS THE DRIVE NUMBER AND YYY IS THE FAILING BLOCK NUMBER.

```
                ;TC11 DEC/X11 EXERCISER MODULE.
000000'                 IOMODX  <TCAG  >,<177340,214,6,0,0,40.,10,RBUF,256.,1024.
000000'                 MODULE  150000,TCAG  ,<177340,214,6,0,0,40.,10,RBUF,256.,1024.
                        .TITLE  TCAG DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 041524  043501      040 MODNAM: .ASCII  /TCAG / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN                    ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177340                  ADDR:   177340+0                        ;1ST DEVICE ADDR.
000010' 000214                  VECTOR: 214+0                           ;1ST DEVICE VECTOR.
000012'     300                 BR1:    .BYTE   PRTY6+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' 000326'                 INIT:   START                           ;MODULE START ADDR.
000032' 000252'                 SPOINT: MODSP                           ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                               ;PASS COUNTER.
000036' 000050                  ICONT:  40.                             ;# OF ITERATIONS PER PASS=40.
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, CR
000102' 000000                  ACSR:   OPEN                            ;CONTENTS OF CSR.
000104'                         WASADR:                                 ;ADDR OF BAD DATA, CR
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' 000504'                 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' 000010                  IDNUM:  10                              ;MODULE IDENTIFICATION NUMBER=10
```



```
000124' 002376'                 RBUFVA: RBUF                            ;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
        000040                          .REPT   SPSIZ                   ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR

000252'                         MODSP:  ;*****************************************************************************
```

```
                    010000                      ILO=BIT12               ;ILLEGAL OPERATION
                    004000                      SELE=BIT11              ;SELECTION ERROR
                    000100                      IE=BIT6                 ;INTERRUPT ENABLE
                    002000                      BM=BIT10                ;BLOCK MISSED
                    001000                      DATM=BIT9               ;DATA MISSED
                    004000                      REV=BIT11               ;ON FOR REVERSE DIRECTION
000252' 000                 INTSW:  .BYTE   OPEN            ;INTERRUPT SWITCH, OFF=SEARCH, ON=DATA
000253' 000                 REVCNT: .BYTE   OPEN            ;# OF TIMES REVERSED DIRECTION WHILE SEARCHING
000254' 000                 DIRIND: .BYTE   OPEN            ;DIRECTION INDICATOR TO SHOW WHICH DIR. FOR DATA XFER 0=
000255' 000                 FLAG:   .BYTE   OPEN            ;SET TO SHOW TAPE OVERFLOWED INTO END REGION
000256' 000                 DRTRYC: .BYTE   OPEN            ;**DATA RETRY COUNTER
000257' 000                 SRTRYC: .BYTE   OPEN            ;**SEARCH RETRY COUNTER
                                    .EVEN
000260' 000002              RTLMT:  2                       ;**RETRY LIMIT
000262' 000000              UNIT:   OPEN
000264' 000000              CMND:   OPEN                    ;TEMP LOCATION FOR NEXT COMMAND DURING SEARCH
000266' 000000              BATMP:  OPEN                    ;TEMP LOCATION FOR TCBA VALUE
000270' 000000              BLOCK:  OPEN                    ;CURRENT BLOCK #
000272' 000000              TCST:   OPEN                    ;ADDR OF CONTROL AND STATUS REG
000274' 000000              TCCM:   OPEN                    ;"    "  COMMAND REGISTER
000276' 000000              TCWC:   OPEN                    ;"    "  WORD COUNT REG
000300' 000000              TCBA:   OPEN                    ;"    "  BUS ADDRESS REG
000302' 000000              TCDT:   OPEN                    ;"    "  DATA REGISTER
000304' 000000              TCWCS:  OPEN                    ;LOC TO SAVE CONTENTS OF TCWC
000306' 000000              TCBAS:  OPEN                    ;"   "   "   "   "  TCBA
000310' 000000              USELCT: OPEN                    ;A BIT /DEVICE LIST OF DRIVES TO BE EXERSIZED
000312' 000000              EABITS: OPEN                    ;HOLDS EXTENDED ADDRESS BITS
000314' 000000              WCNT1:  OPEN                    ;WRITE BUFFER COUNT (2'S COMP)
000316' 000000              WCNT2:  OPEN                    ;READ DITO
000320' 000000              BLK1:   OPEN
000322' 000000              BLK2:   OPEN
000324' 000000              RISV:   OPEN                    ;R1 SAVE LOC.
```



```
                                    ;MODULE CODE STARTS HERE.
000326' 012767  000400  177560  START:  MOV     #256.,WDTC      ;256 WORDS TO MEM PER ITERATION
000334' 012767  002000  177554          MOV     #1024.,WDFR     ;1024 WORDS FROM MEM PER ITERATION
000342' 012767  000021  177550          MOV     #17.,INTR       ;17 INTERRUPTS PER ITERATION
000350' 016705  177432                  MOV     ADDR,R5         ;GET DEVICE ADDR.
000354' 010567  177712                  MOV     R5,TCST         ; SAVE IT
000360' 005725                          TST     (R5)+           ; GET COMMAND REGISTER ADDRESS
000362' 010567  177512                  MOV     R5,CSRA         ; LOAD ADDRESS OF COMMAND REG.
000366' 010567  177702                  MOV     R5,TCCM         ;PUT THIS ADDRESS IN MY POINTER FOR TC COMMANDS
000372' 005725                          TST     (R5)+           ;MAKE ADDRESS OF WORD COUNT REG
000374' 010567  177676                  MOV     R5,TCWC         ;PUT IT THERE
000400' 005725                          TST     (R5)+           ;MAKE ADDRESS OF BUS ADDRESS REG
000402' 010567  177672                  MOV     R5,TCBA         ;PUT IT THERE
000406' 005725                          TST     (R5)+           ;MAKE ADDRESS OF DATA REG
000410' 010567  177666                  MOV     R5,TCDT         ;PUT IT THERE
000414' 016700  177370                  MOV     VECTOR,R0       ;LOAD TC11 VECTOR.
000420' 012720  011122'                 MOV     #DTINT,(0)+
000424' 016720  177362                  MOV     BR1,(0)+
000430' 016767  177360  177652          MOV     DVIDI,USELCT    ;GET AVAILABLE UNITS.
000436' 105067  177613                  CLRB    FLAG            ; ZERO FLAGS
000442' 005067  177622                  CLR     BLOCK           ;START AT BLOCK 0 AND INDICATE THAT THIS IS A NEW START
000446' 122767  000001  000041          CMPB    #1,41           ;LOAD MEDIUM DECTAPE?
000454' 001013                          BNE     RESTRT          ;BR IF NOT.
000456' 113700  000040                  MOVB    @#40,R0         ;GET DRIVE NUMBER
000462' 012701  000001                  MOV     #1,R1           ;DRIVE MASK: INITIALIZE TO DRIVE 0
000466' 105700          1$:     TSTB    R0              ;IF DRIVE FOUND THEN
000470' 001403                  BEQ     2$              ;GO TO 2$
000472' 006301                  ASL     R1              ;ELSE SHIFT MASK TO NEXT DRIVE
000474' 105300                  DECB    R0              ;DECNCOUNT DRIVE #
000476' 000773                  BR      1$              ;CHECK AGAIN
000500' 040167  177604  2$:     BIC     R1,USELCT       ;DISABLE DRIVE .
                        RESTRT:
000504' 104415  000000' 000124'         GETPAS,BEGIN,,RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
000512' 016767  177414  177576          MOV     RBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
000520' 005467  177572                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
000524' 005767  177540                  TST     BLOCK           ;IS THIS A RESTART OR START?
000530' 001004                          BNE     SEGB            ;BRANCH IF RESTART
000532' 112767  000010  177514          MOVB    #BIT3,DIRIND    ;START IN REVERSE PUT
000540' 000453                          BR      GOBK            ;THIS JMP PUTS IT INTO FORWARD AND INSURES ALL CODE IS
                                                                ;RIGHT FOR THIS DIRECTION
000542'                 SEGB:
000542' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
000546' 016767  177370  177540          MOV     WBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
000554' 005467  177534                  NEG     WCNT1           ; GET THE 2'S COMPLEMENT
000560' 062767  000004  177502  STEP:   ADD     #4,BLOCK        ; STEP 4 BLOCKS (1024 WORDS)
000566' 022767  001101  177474          CMP     #577.,BLOCK     ; LEGAL BLOCK NUMBER ?
000574' 100435                          BMI     GOBK            ;BRANCH IF BLOCK # IS TOO BIG
000576' 026727  177466  000000          CMP     BLOCK,#0        ;BY CHANGING THIS #0, WHICH IS THE DEFAULT
                        ;LOWER BLOCK #, AND THE #577 TWO INSTRUCTIONS UP, WHICH IS THE DEFAULT UPPER
                        ;BLOCK LIMIT, YOU CAN FORCE THE DECTAPES TO STAY BETWEEN ANY TWO BLOCK NUMBERS YOU DESIR
000604' 100431                          BMI     GOBK            ;BRANCH IF BLOCK # IS TOO SMALL

000606' 004767  001302  X1:     JSR     PC,SEQDRV       ;SELECT SEQUENTIAL DRIVE.
```

```
000612' 105067  177441          CLRB    SRTRYC          ;**CLEAR SEARCH RETRY COUNT
000616' 105067  177434          CLRB    DRTRYC          ;**CLEAR DATA RETRY COUNT
000622' 004767  000130  WRITE:   JSR     PC,WDATAF       ;WRITE BLOCK FORWARD.
000626' 000407                  BR      SEGC            ;ERROR RETURN.
```



```
000630' 004767  000154          READ:   JSR     PC,RDATAF       ; YES, READ BLOCK FORWARD
000634' 000404                  BR      SEGC            ;ERROR RETURN.
000636' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
000644' 000646'                 .+2                     ; IF ERROR, CONTINUE
000646'                         SEGC:
000646' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                        ;MONITOR SHALL TEST END OF PASS
000652' 132767  000001  177375          BITB    #BIT0,FLAG      ; NO, END OF TAPE ?
000660' 001730                  BEQ     SEGB            ; NO, CONTINUE
000662' 142767  000001  177365          BICB    #BIT0,FLAG      ; YES, CLEAR FLAG AND DROP INTO GOBK


                                ;SUB TO REVERSE DIRECTION OF TAPE MOTION. THIS IS DONE BY CHANGING THE VALUE OF
                                ;MANY COUNTING AND LIMIT CONSTANTS AND 1 INSTRUCTION WHICH IS EITHER A NOP
                                ;OR A NEGATE.

000670' 105767  177360          GOBK:   TSTB    DIRIND          ;WHICH WAY WE GOING NOW?
000674' 001014                  BNE     BACK            ;BR IF BACKWARD
000676' 112767  000010  177350          MOVB    #BIT3,DIRIND    ;MUST GO INTO REVERSE
000704' 012767  177774  177650          MOV     #-4,STEP+2      ;DEC BY 4 CURRENT BLOCK #
000712' 005067  001034                  CLR     ROCM+8.         ;CHANGE MAX BLOCK# TO 0
000716' 012767  000240  001022          MOV     #240,MAYBE      ;CHANGE SENSE OF SUBTRACTION
000724' 000706                  BR      SEGB
000726' 105067  177322          BACK:   CLRB    DIRIND          ;MUST GO FORWARD
000732' 012767  000004  177622          MOV     #4,STEP+2       ;INC BLOCK # BY 4
000740' 012767  001101  001004          MOV     #577.,ROCM+8.   ;CHANGE MAX BLOCK # TO 577
000746' 012767  005400  000772          MOV     #5400,MAYBE     ;PUTS A NEGATE INST. IN PLACE OF NOP
000754' 000672                  BR      SEGB


                                ;WRITE DATA SUB ENTRY.
000756' 112767  000115  177300  WDATAF: MOVB    #115,CMND       ;PRESET COMMAND.
000764' 016767  177144  177274          MOV     WBUFFA,BATMP    ;LOAD TEMPORY STORAGE.
000772' 016777  177316  177276          MOV     WCNT1,GTCWC     ;LOAD WORD COUNT.
001000' 016767  177132  177304          MOV     WBUFEA,EAPITS   ; SAVE EXTENDED MEMORY BITS
001006' 000414                  BR      COMMON


                                ;READ DATA SUB ENTRY.
001010' 112767  000105  177246  RDATAF: MOVB    #105,CMND       ;PRESET COMMAND.
001016' 016767  177104  177242          MOV     RBUFFA,BATMP    ;LOAD TEMPORARY STORAGE
001024' 016777  177266  177244          MOV     WCNT2,GTCWC     ;LOAD WORD COUNT.
001032' 016767  177072  177252          MOV     RBUFEA,EAPITS   ; SAVE EXTENDED MEMORY BITS
```

```
001040' 016777  177222  177232  COMMCN: MOV     BATMP,@TCBA             ;LOAD TCBA
001046' 042767  000060  177210          BIC     #60,CMND        ;SET UP EXT. MEM. BITS
001054' 056767  177232  177202          BIS     EABITS,CMND
001062' 112767  000005  177163          MOVB    #5,REVCNT       ;SET MAX REVERSE CCUNT.
001070' 105067  177156                  CLRB    INTS%           ;SET INT SWITCH TC SEARCH.
001074' 112767  000103  177160          MOVB    #103,UNIT       ;SET UP SEARCH CCMMAND.
001102' 156767  177146  177153          BISB    DIRIND,UNIT+1   ;START SEARCH SAME DIR AS XFR.
001110' 016777  177146  177156          MOV     UNIT,@TCCM      ;ISSUE IT.
001116' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                        ;TC11 INTERRUPTS HERE
001122' 105767  177124  DTINT:  TSTB    INTS%           ;DATA XFR INTERRUPT?
001126' 001146                  BNE     XFRINT          ;BR IF YES.

001130' 005777  177140          TST     @TCCM           ;NC. IN SEARCH MODE. ERROR?
001134' 100556                  BMI     DTER            ;BR IF YES.
001136' 027767  177140  177124  CMP     @TCDT,BLOCK     ;BLOCK FOUND?
001144' 001516                  BEQ     SAME            ;BR IF YES.
001146' 003017                  BGT     LARGER          ;BR IF BLOCK FOUND IS LARGER.

001150' 062777  000003  177124  LOWER:  ADD     #3,@TCDT        ;LCWER. BY 3 OR MORE?
001156' 027767  177120  177104          CMP     @TCDT,BLOCK     ;FIND OUT.
001164' 003004                  BGT     CONT            ;BR IF NOT.
001166' 032777  004000  177100          BIT     #REV,@TCCM;YES. REV BIT SET?
001174' 001017                  BNE     DTREV           ;BR IF YES TC TURN AROUND.

001176' 016777  177060  177070  CONT:   MOV     UNIT,@TCCM      ;ISSUE COMMAND.
001204' 000002                  RTI                     ;EXIT INTERRUPT.

001206' 162777  000003  177066  LARGER: SUB     #3,@TCDT        ;LARGER. BY 3 CR MCRE?
001214' 026777  177050  177060          CMP     BLOCK,@TCDT     ;FIND OUT.
001222' 003365                  BGT     CONT            ;BR IF NCT.
001224' 032777  004000  177042          BIT     #REV,@TCCM      ;GCING FWD?
001232' 001361                  BNE     CONT            ;BR IF NCT.

001234' 062767  004000  177020  DTREV:  ADD     #REV,UNIT       ;CCMPLIMENT DIRECTICN.
001242' 042767  010000  177012          BIC     #BIT12,UNIT     ;CLEAR PCSSIBLE CARRY INTO BIT 12
001250' 105367  176777                  DECB    REVCNT          ;EXHAUSTED REV ALLCWANCE?
001254' 001350                  BNE     CONT            ;BR IF NOT.
001256' 004767  000420          JSR     PC,STOP         ;YES. STCP DECTAPE.
                        ;-------------------------------------------------------------------------
001262' 000004  000000' 001270' ;       FIRQ$,BEGIN,1$          ; QUEUE UP TC CONTINUE AT 1$ ARE RTI
                        ;-------------------------------------------------------------------------
001270' 004767  000546  1$:     JSR     PC,NCW          ;GC INSERT  DRIVE AND BLOCK #
001274' 104403  000000' 002164' MSGN$,BEGIN,MP1 ;ASCII MESSAGE CALL WITH CCMMON HEADER
001302' 012767  000004  176576  MOV     #4,ERRTYP       ;BLOCK NOT FOUND
                        ;*************************************************************************
001310' 104406  000000' 000000  ;       SOFER$,BEGIN,NULL               ;BLOCK NCT FCUND
                        ;*************************************************************************
001316' 126767  176736  176733          CMPB    RTLMT,SRTRYC    ;EXCEEDED RETRY LIMIT ?
001324' 001023                  BNE     A1              ;BRANCH IF NC
```



```
001326'                 DROP:
001326' 104403  000000' 002172' MSGN$,BEGIN,MP2 ;ASCII MESSAGE CALL WITH CCMMON HEADER
001334' 005767  176456          TST     SR1             ;TC DROP OR NCT TC DROP?
001340' 001001                  BNE     A0              ;BRANCH IF YES
001342' 000207                  RTS     PC              ;NC
001344' 116701  176713  A0:     MOVB    UNIT+1,R1       ;YES
001350' 042701  177770          BIC     #177770,R1      ;LEAVE ONLY DEVICE # BITS
001354' 116101  002366'         MOVB    UNTAB(1),R1     ;GET PROPER BIT IN R0
001360' 040167  176724          BIC     R1,USELCT       ;CLEAR BIT IN USELCT TO DROP DRIVE
001364' 104403  000000' 002200' MSGN$,BEGIN,MP3 ;ASCII MESSAGE CALL WITH CCMMON HEADER
001372' 000207                  RTS     PC              ;CAN KEEP GOING
001374' 105267  176657  A1:     INCB    SRTRYC          ;BUMP RETRY COUNT UP
001400' 000617                  BR      COMMON          ;GC BACK AND TRY AGAIN


001402' 032777  004000  176664  SAME:   BIT     #REV,@TCCM      ;SAME. CHECK DIRECTION.
001410' 001011                  BNE     SRCHG           ;BR IF IN REV.
001412' 105767  176636          TSTB    DIRIND          ;FWD. WANT FWD XFR?
001416' 001267                  BNE     CONT            ;BR IF NOT.

001420' 116767  176640  176634  XFR:    MOVB    CMND,UNIT       ;PRESET COMMAND.
001426' 105167  176620          COMB    INTS%           ;SET INT SWITCH TO DATA XFR.
001432' 000661                  BR      CONT            ;GC ISSUE XFR COMMAND.
001434' 105767  176614  SRCHG:  TSTB    DIRIND          ;IN REV. REV XFR WANTED?
001440' 001367                  BNE     XFR             ;BR IF YES.
001442' 000655                  BR      CONT            ;NC. CONTINUE SAME DIR.
```

```
                                ;DATA XFR INTERRUPT SERVICED HERE.
001444' 005777  176624          XFRINT: TST     @TCCM           ;ERROR?
001450' 100410                          BMI     DTER            ;BR IF YES.
001452' 004767  000254                  JSR     PC,STOP1        ;STOP DECTAPE.
                                ;-------------------------------------------------------------------
001456' 000004  000000' 001464'         PIRQ$,BEGIN,XFRA                ; QUEUE UP TC CONTINUE AT XFRA AND RTI
                                ;-------------------------------------------------------------------
001464' 062716  000002          XFRA:   ADD     #2,(6)          ;SET UP CK EXIT.
001470' 000207                          RTS     PC              ;EXIT.

001472' 005777  176574          DTER:   TST     @TCST           ;END ZONE?
001476' 100003                          BPL     1$              ;BR IF NOT. TROUBLE!
001500' 105767  176546                  TSTB    INTS%           ;IN SEARCH MODE?
001504' 001653                          BEQ     DTREV           ;BR IF YES TC REVERSE.
001506' 004767  000170          1$:     JSR     PC,STOP         ;NO!. STOP DECTAPE.
                                ;-------------------------------------------------------------------
001512' 000004  000000' 001520'         PIRQ$,BEGIN,2$          ; QUEUE UP TC CONTINUE AT 2$ AND RTI
                                ;-------------------------------------------------------------------
001520' 004567  000216          2$:     JSR     R5,ROOM         ; WAS THERE ROOM ON TAPE FOR TRANSFER ?
001524' 000207                          RTS     PC              ; NO, THEREFOR NOT REALLY AN ERROR
001526' 004767  000310                  JSR     PC,NOW          ;TYPE DRIVE AND BLOCK #
001532' 032767  002000  176344          BIT     #BM,ASTAT               ;BLOCK MISSED?
001540' 001407                          BEQ     3$
001542' 012767  000005  176336          MOV     #5,ERRTYP       ;BLOCK MISSED CODE
001550' 104403  000000' 002206'         MSGN$,BEGIN,MP4 ;ASCII MESSAGE CALL WITH COMMON HEADER
001556' 000412                          BR      4$
001560' 032767  001000  176316  3$:     BIT     #DATM,ASTAT     ;DATA LATE?
001566' 001404                          BEQ     8$
001570' 012767  000002  176310          MOV     #2,ERRTYP       ;DATA LATE CODE
001576' 000402                          BR      4$              ;CC REPORT
001600' 005067  176302          8$:     CLR     ERRTYP          ;UNKNOWN ERROR
001604' 016767  176274  000546  4$:     MOV     ASTAT,CNUM      ;SAVE ASTAT
                                ;*******************************************************************
001612' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;DECTAPE ERROR
                                ;*******************************************************************
001620' 032767  004000  000532          BIT     #SELE,DNUM      ;SELECTION ERROR?
001626' 001413                          BEQ     5$              ;BR IF NO
001630' 104403  000000' 002214'         MSGN$,BEGIN,MP7 ;ASCII MESSAGE CALL WITH COMMON HEADER
001636' 012767  000007  176242          MOV     #7,ERRTYP       ;SELECTION CODE
                                ;*******************************************************************
001644' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FATAL ERROR
                                ;*******************************************************************
001652' 000167  177466                  JMP     A0              ;GIVE UP AND DROP THAT DRIVE
001656' 126767  176376  176372  5$:     CMPB    RTLMT,DRTRYC    ;EXCEEDED RETRY LIMIT?
001664' 001002                          BNE     6$              ;BRANCH IF NO
001666' 000167  177434                  JMP     DROP            ;YES
001672' 105267  176360          6$:     INCB    DRTRYC          ;BUMP RETRY COUNT
001676' 000167  177136                  JMP     COMMON

001702' 017767  176366  176172  STOP:   MOV     @TCCM,ACSR      ;TCCM CONTENTS TO ACSR.
001710' 017767  176356  176166          MOV     @TCST,ASTAT     ;TCST CONTENTS TO ASTAT.
001716' 017767  176354  176360          MOV     @TCWC,TCWCS     ;SAVE TCWC
001724' 017767  176350  176354          MOV     @TCBA,TCBAS     ;SAVE TCBA.
001732' 042777  000116  176334  STOP1:  BIC     #116,@TCCM      ;STOP DECTAPE.
001740' 000207                          RTS     PC              ;EXIT.
```



```
001742' 016700  176322          ROOM:   MOV     BLOCK,R0        ; SAVE CURRENT BLOCK #
                                ;**********************CAREFULL**************************
001746' 005400                  MAYBE:  NEG     R0              ;THIS INST IS CHANGED BY SUBROUTINE GOBK TC EITHER BE
                                                                ;A NEGATE R0 OR A NOP, DEPENDING ON
                                                                ;THE DIRECTION THE TAPE IS MOVING
                                ;**********************CAREFULL**************************
001750' 012701  001101                  MOV     #577.,R1        ; LOAD MAX. NUMBER OF BLOCKS
001754' 005002                  1$:     CLR     R2              ; ZERO REG. 2
001756' 060001                          ADD     R0,R1           ; GET # OF BLOCKS LEFT ON TAPE
001760' 022701  000400                  CMP     #256.,R1        ; MORE THAN 256 BLOCK LEFT ?
001764' 003420                          BLE     4$              ; YES
001766' 062702  000400          1$:     ADD     #256.,R2        ; GET TOTAL # OF WORDS LEFT
001772' 005301                          DEC     R1              ; ALL BLOCKS ADDED IN ?
001774' 001374                          BNE     1$              ; NO, KEEP ADDING
001776' 005702                          TST     R2              ; IS NUMBER OF WORDS LEFT ON TAPE NEG. ?
002000' 100404                          BMI     2$              ; YES
002002' 005767  176134                  TST     WBUFSZ          ; IS TRANSFER SIZE NEG. ?
002006' 100411                          BMI     5$              ; YES
002010' 000403                          BR      3$              ; NO, GO CONTINUE
002012' 005767  176124          2$:     TST     WBUFSZ          ; IS TRANSFER SIZE POS. ?
002016' 100003                          BPL     4$              ; YES
002020' 020267  176116          3$:     CMP     R2,WBUFSZ       ; WAS THERE ENOUGH ROOM ?
002024' 002402                          BLT     5$              ; NO, RETURN CK
002026' 005725                  4$:     TST     (R5)+           ; YES, MUST BE A REAL ERROR
002030' 000205                          RTS     R5              ; RETURN AS REAL ERROR
002032' 152767  000001  176215  5$:     BISB    #BIT0,FLAG      ; SET OVERFLOW FLAG
002040' 000205                          RTS     R5              ; RETURN OK
                                ;       -------------------------

                                ;  SUBR TO TYPE PRESENT DRIVE# AND BLOCK#
002042'                         NOW:
                                ;*****************************************************************
                                                                ;CONVERT R1SV TO ASCII AND
                                                                ;STORE AT DNUM
002042' 104421  000000' 000324'         BTOD$,BEGIN,R1SV,DNUM
002050' 002360'
                                ;*****************************************************************
002052' 116767  000306  000247          MOVB    DNUM+4,MSG6+3   ;PUT IN MSG
                                ;*****************************************************************
                                                                ;CONVERT BLOCK TO ASCII AND
                                                                ;STORE AT DNUM
002060' 104421  000000' 000270'         BTOD$,BEGIN,BLOCK,DNUM
002066' 002360'
                                ;*****************************************************************
002070' 116767  000266  000234          MOVB    DNUM+2,MSG6+6   ;PUT IN FIRST DIGIT OF MSG
002076' 116767  000261  000227          MOVB    DNUM+3,MSG6+7.  ;2ND
002104' 116767  000254  000222          MOVB    DNUM+4,MSG6+8.  ;3RD
002112' 000207                          RTS     PC              ;RETURN
```

```
                                ;ROUTINE TO SEQUENTIALLY SELECT A DRIVE FOR TESTING.
002114' 012702  000010          SEQDRV: MOV     #8.,R2          ;WILL CHECK 8 TIMES.
002120' 016701  176200                  MOV     RISV,R1
002124' 005201                  1$:     INC     R1
002126' 042701  177770                  BIC     #177770,R1      ;CLEAR JUNK BITS.
002132' 010167  176166                  MOV     R1,RISV
002136' 136167  002366' 176144          BITB    UNTAB(1),USELCT ;DRIVE AVAILABLE?
002144' 001004                          BNE     2$              ;BR IF YES.
002146' 005302                          DEC     R2              ;CHECKED 8 TIMES?
002150' 001365                          BNE     1$              ;BR IF NOT.
002152' 104410  000000'                 ENDS,BEGIN              ;NO DRIVE AVAILABLE
002156' 110167  176101          2$:     MOVB    R1,UNIT+1       ;SELECTED DRIVE # TO UNIT+1.
002162' 000207                          RTS     PC              ;EXIT.

002164' 002222'                 MP1:    MSG1
002166' 002324'                         MSG6
002170' 177777                          177777
002172' 002244'                 MP2:    MSG2
002174' 002324'                         MSG6
002176' 177777                          177777
002200' 002263'                 MP3:    MSG3
002202' 002324'                         MSG6
002204' 177777                          177777
002206' 002304'                 MP4:    MSG4
002210' 002324'                         MSG6
002212' 177777                          177777
002214' 002336'                 MP7:    MSG7
002216' 002324'                         MSG6
002220' 177777                          177777
002222' 020040  046102  041517  MSG1:   .ASCIZ  '  BLOCK NOT FOUND'
002230' 020113  047516  020124
002236' 047506  047125  000104
002244' 020040  051040  052105  MSG2:   .ASCIZ  '   RETRY LIMIT'
002252' 054522  046040  046511
002260' 052111  000             MSG3:   .ASCIZ  '   DRIVE DROPPED'
002263'    040  020040  051104
002270' 053111  020105  051104
002276' 050117  042520  000104
002304' 020040  041040  047514  MSG4:   .ASCIZ  '   BLOCK MISSED'
002312' 045503  046440  051511
002320' 042523  000104
002324' 042040  020124  041040  MSG6:   .ASCIZ  ' DT  B   '
002332' 020040  020040          MSG7:   .ASCIZ  '   SELECTION ERR'
002336' 020040  051440  046105
002344' 041505  044524  047117
002352' 042440  051122     000
                002360'                 .EVEN
002360' 000003                  DNUM:   .BLKW   3.              ;RESERVE 6 BYTES FOR BTOD
002366'    001     002     004  UNTAB:  .BYTE   BIT0,BIT1,BIT2,BIT3,BIT4,BIT5,BIT6,BIT7
002371'    010     020     040
002374'    100     200
002376' 000400                  RBUF:   .BLKW   256.
                003376'         .=.     .END
                C00001
```



```
ACSR     000102R          192#    491*
ADDR     000006R          156#    260
ADDR22=  001000           221#
ASB      000106R          196#
ASTAT    000104R          194#    463     468     473     492*
AWAS     000110R          197#
AQ       001344R          417     419#    484
A1       001374R          411     425#
BACK     000726R          333     339#
BATMP    000266R          238#    350*    358*    361
BEGIN    000000R          152#    289     298     318     321     369     402     405     408     415     423     447     458
                          466     475     479     482     536     543     560
BIT0  =  000001           221#    323     325     526     598
BIT1  =  000002           221#    598
BIT10 =  002000           221#    225
BIT11 =  004000           221#    223     227
BIT12 =  010000           221#    222     397
BIT13 =  020000           221#
BIT14 =  040000           221#
BIT15 =  100000           221#
BIT2  =  000004           221#    598
BIT3  =  000010           221#    294     334     598
BIT4  =  000020           221#    598
BIT5  =  000040           221#    598
BIT6  =  000100           221#    224     598
BIT7  =  000200           221#    598
BIT8  =  000400           221#
BIT9  =  001000           221#    226
BLK1     000320R          251#
BLK2     000322R          252#
BLOCK    000270R          239#    276*    292     301*    302     304     377     382     391     500     543
BM    =  002000           225#    463
BREAK$=  104407           221#
BR1      000012R          160#    273
BR2      000013R          161#
BTOD$ =  104421           221#    536     543
CDATA$=  104412           221#    318
CDERCT   000144R          211#
CDWDCT   000146R          212#
CMND     000264R          237#    349*    357*    362*    363*    435
COMMON   001040R          353     361#    426     489
CONFIG   000056R          180#
CONT     001176R          383     387#    392     394     399     433     437     440
CSRA     000100R          190#    263*
DATCK$=  104411           221#
DATER$=  104404           221#
DATM  =  001000           225#    466
DIRIND   002254R          230#    294*    332     334*    339*    367     432     438
DNUM     002360R          473*    477     536     539     543     546     547     548     597#
DROP     001326R          414#    487
DRTRYC   000256R          232#    311*    485     488*
DTER     001472R          376     444     452#
DTINT    001122R          272     372#
DTREV    001234R          385     396#    455
DVID1    000014R          162#    274
EABITS   000312R          248#    352*    360*    363
```

```
ENDIT$= 104413          221#    321
END$  = 104410          221#    560
ERRTYP  000106R         195#    406*    465*    470*    472*    480*
EXIT$ = 104400          221#    369
FLAG    000255R         231#    275*    323     325*    526*
FREE    000150R         213#
GETPA$= 104415          221#    289
GOBK    000670R         295     303     307     332#
GWBUF$= 104414          221#    298
HRDCNT  000044R         175#
HRDER$= 104405          221#    475     482
HRDPAS  000050R         177#
ICONT   000036R         172#
ICOUNT  000040R         173#
IDNUM   000122R         202#
IE    = 000100          224#
ILO   = 010000          222#
IMODX.= 000000          214#    299
INIT    000030R         169#
INTR    000120R         201#    259*
INTSW   000252R         228#    365*    372     436*    454
LARGER  001206R         379     390#
LOWER   001150R         381#
MAP22$= 104416          221#
MAYBE   001746R         337*    342*    503#
MODNAM  000000R         156#
MODSP   000252R         170     219#
MP1     002164R         405     564#
MP2     002172R         415     567#
MP3     002200R         423     570#
MP4     002206R         466     573#
MP7     002214R         479     576#
MSGN$ = 104403          221#    405     415     423     466     479
MSGS$ = 104402          221#
MSGS  = 104401          221#
MSG1    002222R         564     579#
MSG2    002244R         567     582#
MSG3    002263R         570     585#
MSG4    002304R         573     588#
MSG6    002324R         539*    546*    547*    548*    565     568     571     574     577     591#
MSG7    002336R         576     593#
NOW     002342R         404     462     532#
NULL  = 000000          221#    406     475     482
OPEN  = 000000          157     163     164     165     166     183     184     185     186     187     188     189     190
                        192     194     196     197     199     200     201     204     205     207     208     210     211
                        212     213     221#    228     229     230     231     232     233     236     237     238     239
                        240     241     242     243     244     245     246     247     248     249     250     251     252
                        253
OTOA$ = 104420          221#
PASCNT  000034R         171#
PIRQ$ = 000004          221#    402     447     458
POPSP = 005726          221#
POPSP2= 022626          221#
PRTV  = 000000          221#
PRTY0 = 000000          161     221#
PRTY1 = 000040          221#
```



```
PRTY2 = 000100          221#
PRTY3 = 000140          221#
PRTY4 = 000200          221#
PRTY5 = 000240          221#
PRTY6 = 000300          160     221#
PRTY7 = 000340          221#
PS    = 177776          221#
PSW   = 177776          221#
PUSH  = 005746          221#
PUSH2 = 024646          221#
RAND$ = 104417          221#
RANNUM  000054R         179#
RBUF    002376R         203     601#
RBUFEA  000130R         205#    360
RBUFFA  000126R         204#    318     358
RBUFSZ  000132R         206#    290
RBUFVA  000124R         203#    289
RDATAF  001010R         316     357#
READ    000630R         316#
RESTRT  000504R         198     278     288#
RES1    000056R         181#
RES2    000060R         182#
REV   = 004000          227#    384     393     396     430
REVCNT  000253R         229#    364*    398*
ROOM    001742R         336*    341*    460     500#
RSTRT   000112R         198#
RTLMT   000260R         235#    410     485
RISV    000324R         253#    536     552     555*
SAME    001402R         378     430#
SBADR   000102R         191#
SEGB    000542R         293     297#    324     338     343
SEGC    000646R         313     317     320#
SELE  = 004000          223#    477     551#
SEQDRV  002114R         309
SOFCNT  000042R         174#
SOFER$= 104406          221#    408
SOFPAS  000046R         176#
SPOINT  000032R         170#
SPSIZ = 000040            1#    214
SRCHG   001434R         431     438#
SRTRYC  000257R         233#    310*    410     425*
SR1     000016R         163#    416
SR2     000020R         164#
SR3     000022R         165#
SR4     000024R         166#
START   000326R         169     257#
STAT    000026R         168#
STEP    000560R         301#    335*    340*
STOP    001702R         400     456     491#
STOP1   001732R         445     495#
SVR0    000062R         183#
SVR1    000064R         184#
SVR2    000066R         185#
SVR3    000070R         186#
SVR4    000072R         187#
SVR5    000074R         188#
```

```
SVR6    000076R           189#
SYSCNT  000052R           178#
TCBA    000300R           243#    268*    361*    494
TCBAS   000306R           246#    494*
TCCM    000274R           241#    264*    368*    375     384     387*    393     430     443     491     495*
TCDT    000302R           244#    270*    377     381*    382     390*    391
TCST    000272R           240#    261*    452     492
TCWC    000276R           242#    266*    351*    359*    493
TCWCS   000304R           245#    493*
TRPDFD= 000022            221#
UNIT    000262R           236#    366*    367*    368     387     396*    397*    419     435*    561*
UNTAB   002366R           421     556     598#
USELCT  000310R           247#    274*    286*    422*    556
VECTOR  000010R           159#    271
WASADR  000104R           193#
WBUFEA  000136R           208#    352
WBUFPA  000134R           207#    350
WBUFRQ  000140R           209#
WBUFSZ  000142R           210#    299     517     520     522
WCNT1   000314R           249#    299*    300*    351
WCNT2   000316R           250#    290*    291*    359
WDATAF  000756R           312     349#
WDFR    000116R           200#    258#
WDTO    000114R           199#    257*
WRITE   000622R           312#
XFLAG   000005R           157#
XFR     001420R           435#    439
XFRA    001464R           447     449#
XFRINT  001444R           373     443#
X1      000606R           309#
.     = 003376R           319     596#    597#    601#    602#

. ABS.  000000      000
        003376      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XTCAGO,XTCAGO/SOL/CRF:SYM=DDXCOM,XTCAGO
 RUN-TIME:  1  1  .3 SECONDS
 RUN-TIME RATIO: 15/3=4.1
 CORE USED:  7K  (13 PAGES)
```