```
                .REM    !
```

IDENTIFICATION

PRODUCT CODE: AC-E6820-MC

PRODUCT NAME: CXTAAD0 TA11 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

TAA IS AN IOMOD THAT EXERCISES EITHER OR BOTH DRIVES OF A TA11 CASSETTE UNIT.

2. REQUIREMENTS

HARDWARE: TA11 WITH AT LEAST ONE UNIT LOADED WITH A CASSETTE.

STORAGE:: TAA REQUIRES:
1. DECIMAL WORDS: 518
2. OCTAL WORDS: 1006
3. OCTAL BYTES: 2014

3. PASS DEFINITION

ONE PASS CONSISTS OF 70 BLOCKS WRITTEN, READ, AND CHECKED.

4. EXECUTION TIME

A PASS REQUIRES APPROXIMATELY 1 MINUTE WHEN RUNNING ALONE ON A PDP-11/05.

5. CONFIGURATION REQUIREMENTS

A. DEFAULT PARAMETERS

DEVADR: 177500, VECTOR: 260, BR1: 6, DEVCNT: 2

B. REQUIRED PARAMETERS

NONE

6. DEVICE/OPTION SETUP

EACH CASSETTE DRIVE TO BE TESTED MUST BE LOADED WITH A CASSETTE WORK TAPE WHICH HAS THE WRITE PROTECT HOLES COVERED (WRITE ENABLED).

7. MODULE OPERATION

THE MODULE FIRST CHECKS IF THE LOAD MEDIUM IS CASSETTE. IF IT IS, THE LOAD DRIVE IS DROPPED FROM THE EXERCISE, SO AS NOT TO DESTROY CONTENTS OF LOAD CASSETTE. THE DRIVES ARE THEN REWOUND, AND THE FOLLOWING TEST SEQUENCE PERFORMED:

A. SELECT A DRIVE

B. WRITE A BLOCK OF 128 BYTES

C. REVERSE ONE BLOCK

D. READ BLOCK OF 128 BYTES

E. CHECK THE READ DATA AGAINST WRITE DATA AND REPORT ERRORS. (MAXIMUM OF 3 DATA ERRORS PER BLOCK ARE REPORTED).

F. IF NOT 70 BLOCKS DONE, GOES TO STEP A.

G. IF 70 BLOCKS DONE, REPORT END OF PASS, RESUME AT STEP A.

8. SPECIAL FEATURES

THE MODULE REWINDS A DRIVE WHENEVER IT DETECTS CLEAR LEADER, AND THEN REPEATS THE OPERATION ATTEMPTED WHEN CLEAR LEADER WAS DETECTED. (WRITE OR READ).

ALL SOFT ERRORS ARE REPORTED (TIMING ERROR, BLOCK CHECK, FILE GAP) HOWEVER, A WRITE OR READ COMMAND WILL BE TRIED UP TO 3 TIMES AFTER A SOFT ERROR. IF AFTER 3 TRIES THE PROGRAM DOES NOT SUCCED IN PERFORMING THE OPERATION, AND ERROR IS INDICATED.

HARD ERRORS ARE REPORTED SEPARATELY FROM SOFT ERRORS (OFF-LINE, WRITE-LOCK). A DRIVE IS AUTOMATICALLY DROPPED FROM TEST AFTER A HARD ERROR. THE HARD ERROR IS REPORTED, THE FACT THAT THE DRIVE HAS BEEN DROPPED IS NOT. AFTER ALL DRIVES HAVE BEEN DROPPED, THE MODULE ITSELF IS DROPPED, SINCE THERE ARE NO DRIVES TO TEST. A MODULE DROPPED MESSAGE DOES OCCUR.

9. OPERATION OPTIONS:

A. MODULE LOCATION DVID1 (14) MAY BE CHANGED TO EXERCISE ANY COMBINATION OF UNITS 0-1. BIT0=DRV0, BIT1=DRV1.

B. IF DVID1=0 AT RUNTINE THE MODULE WILL BE DROPPED FROM THE EXERCISE.

10. NON-STANDARD PRINTOUT

ALL PRINTOUTS HAVE STANDARD SIGNIFICANCE.

```
                        ;
000000'                 IOMOD   <TAAD  >,177500,260,6,0,2,70.,7
000000'                 MODULE  140000,TAAD  ,177500,260,6,0,2,70.,7
                        .TITLE  TAAD DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                                ;*****************************************************************
000000'                         BEGIN:
000000' 040524  042101     040  MODNAM: .ASCII  /TAAD / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177500                  ADDR:   177500+0                ;1ST DEVICE ADDR.
000010' 000260                  VECTOR: 260+0                   ;1ST DEVICE VECTOR.
000012'    300                  BR1:    .BYTE   PRTY6+0         ;1ST BR LEVEL.
000013'    000                  BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000003                  DVID1:  2+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' 000250'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000106                  ICONT:  70.                     ;# OF ITERATIONS PER PASS=70.
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000436'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000007                  IDNUM:  7                       ;MODULE IDENTIFICATION NUMBER=7
```



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

```
                                ;FUNCTIONAL DEFINITION FOR TA11
                                INTRU=100
                                UNIT1=400               ;SELECT UNIT 1
                                GO=1
                                FREWND=16+INTRU+GO+20   ;REWND & GO
                                FREAD=4+INTRU+GO                ;READ & GO
                                FWRITE=2+INTRU+GO       ;WRITE & GO
                                FBKSP=10+INTRU+GO+20    ;BACKSPACE & GO
                                ;
                                BDONE=40                ;DONE
                                BERROR=100000           ;ERROR
                                BCLRLDR=20000           ;CLEAR LEADER
                                BILBS=20                ;ILBS BIT
                                ;PARAMETER BLOCK
                                ;
                                ;
                                        .BYTE   OPEN
                                RWCMND: .BYTE   OPEN            ;HOLDS READ/WRITE COMMAND.
                                CMND:   .BYTE   OPEN            ;HOLDS CURRENT COMMAND.
                                UNIT:   .BYTE   OPEN            ;HOLDS CURRENT UNIT #.
                                        .EVEN
                                TWBUF:  OPEN                    ;WRITE BUFFER POINTER
                                BUF:    OPEN                    ;HOLDS XFR BUFFER ADDR.
                                FORK:   OPEN                    ;POINTER TO ACTIVE SERVICE ROUTINE
                                TMPBUF: OPEN                    ;HOLDS CURRENT BUFFER ADDR.
                                BYTCNT: OPEN                    ;HOLDS CURRENT TRANSFER COUNT.
                                TRYCTR: OPEN                    ;HOLDS MAX TRY COUNT.
                                TMPCSR: OPEN                    ;HOLDS CSR CONTENTS
                                TTADB:  OPEN                    ;HOLDS TADB ADDRESS.

                                ;REGISTER USAGE
                                ;R0     =
                                ;R1     =
                                ;R2     =
                                ;R3     =
                                ;R4     =AVAILABLE UNIT STATUS
                                ;R5     =CONTAINS TA11 CSR ADDR.
```



```
                                ;MODULE CODE STARTS HERE
                                ;
000250' 012767  000100  177636  START:  MOV     #64.,WDTO       ;64 WORDS TO MEM PER ITERATION
000256' 012767  000100  177632          MOV     #64.,WDFR       ;64 WORDS FROM MEM PER ITERATION
000264' 012767  000405  177626          MOV     #261.,INTR      ;261 INTERRUPTS PER ITERATION
000272' 016705  177510                  MOV     ADDR,R5         ;GET TA11 ADDRESS.
000276' 010567  177576                  MOV     R5,CSRA         ;GET TA11 ADDR TO CSRA NOW.
000302' 010567  177740                  MOV     R5,TTADB        ;SET UP TADB ADDR.
000306' 062767  000002  177732          ADD     #2,TTADB
000314' 016700  177470                  MOV     VECTOR,R0       ;GET THE VECTOR ADDRESS
000320' 012720  001052'                 MOV     #TAINTR,(0)+    ;POINT THE INTERRUPTS TO TAINTR
000324' 016720  177462                  MOV     BR1,(0)+        ;SET UP THE PRIORITY
000330' 016704  177460                  MOV     DVID1,R4        ;SAVE DEVICE SELECTION PARAMETERS
000334' 122737  000005  000041          CMPB    #5,@#41         ;LOAD MEDIUM CASSETTE?
000342' 001011                          BNE     1$              ;BR IF NOT
000344' 123727  000040  000000          CMPB    @#40,#0         ;LOAD UNIT 0?
000352' 001003                          BNE     3$              ;NO, UNIT 1
000354' 042704  000001                  BIC     #1,R4           ;DON'T TEST UNIT 0
000360' 000402                          BR      1$
000362' 042704  000002          3$:     BIC     #2,R4           ;DON'T TEST UNIT 1
                                ;REWIND BOTH UNITS
000366' 004767  000112          1$:     JSR     PC,SEQDRV       ;SELECT UNIT
000372' 004767  000154                  JSR     PC,REWIND       ;REWIND IT.
000376' 000240                          NOP                     ;ERROR RETURN.
000400' 004767  000100                  JSR     PC,SEQDRV       ;SELECT OTHER UNIT.
000404' 004767  000142                  JSR     PC,REWIND       ;REWIND IT TOO.
000410' 000240                          NOP                     ;ERROR RETURN.

                                ; LOAD WRITE BUFFER AREA
000412' 012701  000040                  MOV     #32.,R1         ; SET COUNTER
000416' 012700  001414'                 MOV     #WTBUF,R0       ; GET WRITE BUFFER ADDRESS
000422' 012720  125125          2$:     MOV     #125125,(R0)+   ; LOAD WRITE BUFFER AREA
000426' 012720  000377                  MOV     #000377,(R0)+   ;
000432' 005301                          DEC     R1              ; DONE ?
000434' 001372                          BNE     2$              ; NO, KEEP LOADING

                                ;TEST SEQUENCE
000436' 012767  001414' 177564  RESTRT: MOV     #WTBUF,TWBUF    ;GET WRITE BUFFER ADDR.
000444' 004767  000034          1$:     JSR     PC,SEQDRV       ;SELECT A DRIVE.
000450' 004767  000202                  JSR     PC,WRITE        ;WRITE A BLOCK.
000454' 000773                          BR      1$              ;ERROR RETURN.
000456' 004767  000132                  JSR     PC,REVBLK       ;BACKSPACE A BLOCK.
000462' 000770                          BR      1$              ;ERROR RETURN.
000464' 004767  000204                  JSR     PC,READ         ;READ THE BLOCK.
000470' 000765                          BR      1$              ;ERROR RETURN.
000472' 004767  000630                  JSR     PC,CKDAT        ;CHECK THE DATA.
000476' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
000502' 000760                          BR      1$              ;BR IF NOT
```

```
                                        ;ROUTINE TO SELECT A DRIVE.
000504' 012700  000002          SEQDRV: MOV     #2,R0           ;WILL TRY TWICE.
000510' 105167  177513          1$:     COMB    UNIT            ;TRY OTHER UNIT.
000514' 142767  000376  177505          BICB    #376,UNIT       ;NEED ONE BIT ONLY.
000522' 001407                          BEQ     3$              ;BR IF 0 (UNIT 0).
000524' 032704  000002                  BIT     #BIT1,R4        ;UNIT 1 AVAILABLE?
000530' 001007                          BNE     4$              ;BR IF YES.
000532' 005300                  2$:     DEC     R0              ;TRIED TWICE?
000534' 001365                          BNE     1$              ;BR IF NOT.
000536' 104410  000000'                 ENDS,BEGIN              ;NO. DROP MODULE. NO DRIVES AVAILABLE.
000542' 032704  000001          3$:     BIT     #BIT0,R4        ;UNIT 0 AVAILABLE?
000546' 001771                          BEQ     2$              ;BR IF NOT.
000550' 000207                  4$:     RTS     PC              ;EXIT. GOT A DRIVE.

                                        ;ROUTINE TO REWIND SELECTED DRIVE.
000552' 012767  000576' 177454  REWIND: MOV     #1$,FORK        ;
000560' 112767  000137  177440          MOVB    #FREWND,CMND    ;SET UP REWIND COMMAND IN COMMAND WORD.
000566' 016715  177434                  MOV     CMND,(5)        ;ISSUE COMMAND.
000572' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
000576' 004767  000342          1$:     JSR     PC,CKER         ;CHECK FOR ERRORS.
000602' 000240                          NOP                     ;HARD ERROR.
000604' 000402                          BR      2$              ;SOFT ERROR.
000606' 062716  000002                  ADD     #2,(6)          ;OK RETURN.
000612' 000207                  2$:     RTS     PC              ;DONE. EXIT.

                                        ;ROUTINE TO BACKSPACE A BLOCK.
000614' 012767  000640' 177412  REVBLK: MOV     #1$,FORK        ;
000622' 112767  000131  177376          MOVB    #FBKSP,CMND     ;SET UP REV BLK COMMAND IN COMMAND WORD.
000630' 016715  177372                  MOV     CMND,(5)        ;ISSUE COMMAND.
000634' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
000640' 004767  000300          1$:     JSR     PC,CKER         ;CHECK FOR ERRORS.
000644' 000240                          NOP                     ;HARD ERROR.
000646' 000402                          BR      2$              ;SOFT ERROR.
000650' 062716  000002                  ADD     #2,(6)          ;OK RETURN.
000654' 000207                  2$:     RTS     PC              ;DONE. EXIT.
```



```
                                        ;READ/WRITE ROUTINE.
000656' 112767  000103  177341  WRITE:  MOVB    #FWRITE,RWCMND  ;SET UP WRITE COMMAND IN COMMAND WORD.
000664' 016767  177340  177340          MOV     TWBUF,BUF
000672' 000406                          BR      RWCOM
000674' 112767  000105  177323  READ:   MOVB    #FREAD,RWCMND   ;SET UP READ COMMAND IN COMMAND WORD.
000702' 012767  001614' 177322          MOV     #RDBUF,BUF
000710' 012767  000003  177324  RWCOM:  MOV     #3,TRYCTR       ;WILL TRY 3 TIMES.
000716' 016767  177310  177312  1$:     MOV     BUF,TMPBUF      ;LOAD TEMP BUFF.
000724' 012767  000756' 177302          MOV     #2$,FORK        ;POINT FORK TO 2$
000732' 012767  000200  177300          MOV     #128.,BYTCNT    ;WILL READ/WRITE 128 BYTES.
000740' 116767  177261  177260          MOVB    RWCMND,CMND     ;MOVE RW COMMAND TO COMMAND WORD.
000746' 016715  177254                  MOV     CMND,(5)        ;ISSUE COMMAND.
000752' 104400  000000'                 EXIT$,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
000756' 004767  000162          2$:     JSR     PC,CKER         ;CHECK FOR ERRORS.
000762' 000403                          BR      3$              ;HARD ERROR.
000764' 000403                          BR      4$              ;SOFT ERROR.
000766' 062716  000002                  ADD     #2,(6)          ;OK RETURN.
000772' 000207                  3$:     RTS     PC              ;DONE, EXIT.
000774' 032767  020000  177242  4$:     BIT     #20000,TMPCSR   ;WAS IT CLEAR LEADER?
001002' 001404                          BEQ     5$              ;BR IF NOT.
001004' 004767  177542                  JSR     PC,REWIND       ;REWIND CASSETTE.
001010' 000770                          BR      3$              ;ERROR RETURN.
001012' 000741                          BR      1$
001014' 005367  177222          5$:     DEC     TRYCTR          ;TRIED 3 TIMES?
001020' 001404                          BEQ     6$              ;BR IF YES.
001022' 004767  177566                  JSR     PC,REVBLK       ;NO,REVERSE BLOCK.
001026' 000761                          BR      3$              ;ERROR RETURN.
001030' 000732                          BR      1$              ;GO READ AGAIN.
001032' 004767  000252          6$:     JSR     PC,ERRSET       ;COMMON ERROR SETUP.
001036' 005067  177044                  CLR     ERRTYP          ;UNKNOWN ERROR CODE
                                        ;*****************************************************************
001042' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;FAILED TO READ/WRITE WITHIN 3 TRIES
                                        ;*****************************************************************
001050' 000207                          RTS     PC              ;ERROR RETURN.

                                        ;CASSETTE INTERRRUPT HANDLER
001052' 105777  176730          TAINTR: TSTB    #ADDR           ;TRANSFER REQUEST?
001056' 100405                          BMI     TREQ            ;YES = PIRQ    1$
                                        ;---------------------------------------------------------------
001060' 000004  000000' 001066'         PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;---------------------------------------------------------------
001066' 000177  177142          1$:     JMP     #FORK           ;GO TO SERVICE CURRENT SEQUENCE
001072' 005367  177142          TREQ:   DEC     BYTCNT          ;WHEN COUNT NEG,, DONE
001076' 100416                          BMI     4$              ;GO TO LAST BYTE SEQUENCE
001100' 032777  000004  176700          BIT     #4,@ADDR        ;CHECK READ OR WRITE FUNCTION
001106' 001004                          BNE     1$              ;ONE = GO TO PROCESS READ
001110' 117777  177122  177130          MOVB    @TMPBUF,@TTADB  ;WRITE BYTE
001116' 000403                          BR      2$
001120' 117777  177122  177110  1$:     MOVB    @TTADB,@TMPBUF  ;READ BYTE
001126' 005267  177104          2$:     INC     TMPBUF          ;UPDATE ADDR.
001132' 000002                          RTI                     ;RETURN FROM INT.
001134' 052777  000020  176644  4$:     BIS     #20,@ADDR       ;INITIATE LAST BYTE SEQUENCE
001142' 000002                          RTI                     ;EXIT INTERRUPT SEQUENCE.
```

```
                                ;ROUTINE TO CHECK FOR ERRORS.
001144' 032715  000040          CKER:   BIT     #40,(5)         ;DONE BIT SET?
001150' 001011                          BNE     1$              ;BR IF YES.
001152' 004767  000132                  JSR     PC,ERRSET       ;NO, ERROR SETUP.
001156' 012767  000003  176722          MOV     #3,ERRTYP       ;CONTROLLER NOT READY
                                        ;***************************************************************
001164' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DONE/XFR REQUEST NOT SET
                                        ;***************************************************************
001172' 000424                          BR      4$
001174' 005715                  1$:     TST     (5)             ;ANY ERRORS?
001176' 100020                          BPL     3$              ;BR IF NOT.
001200' 004767  000104                  JSR     PC,ERRSET       ;DO ERROR SETUP.
001204' 032767  011000  177032          BIT     #11000,TMPCSR   ;WAS IT HARD ERROR?
001212' 001017                          BNE     6$              ;BR IF YES.
001214' 032767  020000  177022          BIT     #20000,TMPCSR   ;CLEAR LEADER?
001222' 001010                          BNE     4$              ;BR IF YES, DO NOT REPORT HARD ERROR.
001224' 005067  176656                  CLR     ERRTYP          ;UNKNOWN SOFT ERROR
                                        ;COULD BE TIMING,BLOCK,ETC
                                        ;***************************************************************
001230' 104406  000000' 000000          SOFERS,BEGIN,NULL       ;TA11 SOFT ERROR
                                        ;***************************************************************
001236' 000402                          BR      4$
001240' 062716  000002          3$:     ADD     #2,(6)          ;OK RETURN SETUP.
001244' 062716  000002          4$:     ADD     #2,(6)          ;SOFT ERROR RETURN SET UP.
001250' 000207                  5$:     RTS     PC              ;RETURN TO SEQUENCE
001252' 012767  000006  176626  6$:     MOV     #6,ERRTYP       ;OFF LINE OR WRITE PROTECTED
                                        ;***************************************************************
001260' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;TA11 HARD ERROR
                                        ;***************************************************************
001266' 105767  176735                  TSTB    UNIT            ;DETERMINE UNIT NUMBER.
001272' 001003                          BNE     7$              ;BR IF UNIT 1.
001274' 042704  000001                  BIC     #BIT0,R4        ;DROP UNIT 0.
001300' 000763                          BR      5$
001302' 042704  000002          7$:     BIC     #BIT1,R4        ;DROP UNIT 1
001306' 000760                          BR      5$

001310' 011567  176566          ERRSET: MOV     (5),ACSR        ;SAVE CSR CONTENTS IN ACSR,
001314' 011567  176724                  MOV     (5),TMPCSR      ;AND TMPCSR.
001320' 012715  000020                  MOV     #BIL0S,(5)      ;STOP MOTION.
001324' 000207                          RTS     PC              ;EXIT.
```



```
                                ;DATA CHECK ROUTINE, REPORTS UP TO 3 ERRORS PER BLOCK CHECKED.
001326' 016700  176676          CKDAT:  MOV     TWBUF,R0        ;WRITE BUFFER ADDR TO R0,
001332' 012701  001614'                 MOV     #RDBUF,R1       ;READ BUFFER ADDR TO R1,
001336' 012702  000200                  MOV     #128.,R2        ;WILL CHECK 128 BYTES,
001342' 012703  000003                  MOV     #3,R3           ;WILL REPORT UP TO 3 ERRORS,
001346' 121011                  1$:     CMPB    (0),(1)         ;CHECK A SET OF BYTES.
001350' 001414                          BEQ     2$              ;BR IF SAME, GOOD,
001352' 010067  176524                  MOV     R0,SBADR        ;SAVE GOOD ADDR,
001356' 010167  176522                  MOV     R1,WASADR       ;SAVE BAD ADDR,
001362' 111067  176520                  MOVB    (0),ASB         ;SAVE GOOD BYTE,
001366' 111167  176516                  MOVB    (1),AWAS        ;SAVE BAD BYTE,
                                        ;***************************************************************
001372' 104404  000000'                 DATERS,BEGIN            ;DATA ERROR!!!
                                        ;***************************************************************
001376' 005303                          DEC     R3              ;3 ERRORS REPORTED?
001400' 001404                          BEQ     3$              ;BR IF YES, QUIT,
001402' 005302                  2$:     DEC     R2              ;ALL BYTES CHECKED?
001404' 001402                          BEQ     3$              ;BR IF YES, QUIT,
001406' 122021                          CMPB    (0)+,(1)+       ;UPDATE BYTE ADDRESSES,
001410' 000756                          BR      1$              ;DO IT AGAIN,
001412' 000207                  3$:     RTS     PC              ;EXIT.

001414' 000200                  WTBUF:  .BLKB   128.            ; WRITE BUFFER AREA
001614' 000200                  RDBUF:  .BLKB   128.            ;INPUT BUFFER AREA
        000001                          .END
```

```
ACSR    000102R          192#    422*
ADDR    000006R          158#    252     369     377     384*
ADDR22= 001000           210#
ASB     000106R          196#    435*
ASTAT   000104R          194#
AWAS    000110R          197#    436*
BCLRLD= 020000           221#
BDONE = 000040           219#
BEGIN   000000R          155#    293     305     314     325     345     364     372     392     405     413     438
BERROR= 100000           220#
BILBS = 000020           222#    424
BIT0  = 000001           210#    306     417
BIT1  = 000002           210#    301     419
BIT10 = 002000           210#
BIT11 = 004000           210#
BIT12 = 010000           210#
BIT13 = 020000           210#
BIT14 = 040000           210#
BIT15 = 100000           210#
BIT2  = 000004           210#
BIT3  = 000010           210#
BIT4  = 000020           210#
BIT5  = 000040           210#
BIT6  = 000100           210#
BIT7  = 000200           210#
BIT8  = 000400           210#
BIT9  = 001000           210#
BREAK$= 104407           210#
BR1     000012R          160#    258
BR2     000013R          161#
BTOD$ = 104421           210#
BUF     000232R          232#    335*    338*    340
BYTCNT  000240R          235#    342*    375*
CDATA$= 104412           210#
CKDAT   001326R          292     427#
CKER    001144R          315     326     346     387#
CMND    000226R          228#    312*    313     323*    324     343*    344
CONFIG  000056R          180#
CSRA    000100R          190#    253*
DATCK$= 104411           210#
DATER$= 104404           210#    438
DVID1   000014R          162#    259
ENDIT$= 104413           210#    293
END$  = 104410           210#    305
ERRSET  001310R          361     389     397     422#
ERRTYP  000106R          195#    362*    390*    402*    411*
EXIT$ = 104400           210#    314     325     345
FBK3P = 000131           217#    323
FORK    000234R          233#    311*    322*    341*    374
FREAD = 000105           215#    337
FREWND= 000137           214#    312
FWRITE= 000103           216#    334
GETPA$= 104415           210#
GO    = 000001           213#    214     215     216     217
GWBUF$= 104414           210#
HRDCNT  000044R          175#
```



```
HRDER$= 104405           210#    364     392     413
HRDPA$  000050R          177#
ICONT   000036R          172#
ICOUNT  000040R          173#
IDNUM   000122R          202#
INIT    000030R          169#
INTR    000120R          201#    251*
INTRU = 000100           211#    214     215     216     217
MAP22$= 104416           210#
MODNAM  000000R          156#
MOD3P   000224R          170     208#
MSGN$ = 104403           210#
MSG8$ = 104402           210#
MSG$  = 104401           210#
NULL  = 000000           210#    364     392     405     413
OPEN  = 000000           157     163     164     165     166     183     184     185     186     187     188     189     190
                         192     194     196     197     199     200     201     210#    226     227     228     229     231
                         232     233     234     235     236     237     238
OTOA$ = 104420           210#
PASCNT  000034R          171#
PIRQ$ = 000004           210#    372
POPSP = 005726           210#
POPSP2= 022626           210#
PRTY  = 000000           210#
PRTY0 = 000000           161     210#
PRTY1 = 000040           210#
PRTY2 = 000100           210#
PRTY3 = 000140           210#
PRTY4 = 000200           210#
PRTY5 = 000240           210#
PRTY6 = 000300           160     210#
PRTY7 = 000340           210#
PS    = 177776           210#
PSW   = 177776           210#
PUSH  = 005746           210#
PUSH2 = 024646           210#
RAND$ = 104417           210#
RANNUM  000052R          179#
RDBUF   001614R          338     428     449#
READ    000674R          290     337#
RESTRT  000436R          198     284#
RES1    000056R          181#
RES2    000060R          182#
REVBLK  000614R          288     322#    358
REWIND  000352R          269     272     311#    353
RSTRT   000112R          198#
RWCMND  000225R          227#    334*    337*    343
RWCOM   000710R          336     339#
S8ADR   000102R          191#    433*
SEQDRV  000504R          268     271     285     297#
SOFCNT  000042R          174#
SOFER$= 104406           210#    405
SOFPA$  000046R          176#
SPOINT  000032R          170#
SP8IZ = 000040             1#    203
SR1     000016R          163#
```

```
SR2      000020R        164#
SR3      000022R        165#
SR4      000024R        166#
START    000250R        169     249#
STAT     000026R        168#
SVR0     000062R        183#
SVR1     000064R        184#
SVR2     000066R        185#
SVR3     000070R        186#
SVR4     000072R        187#
SVR5     000074R        188#
SVR6     000076R        189#
SYSCNT   000052R        178#
TAINTR   001052R        257     369#
TMPBUF   000236R        234#    340*    379     381*    382*
TMPCSR   000244R        237#    351     398     400     423*
TREQ     001072R        370     375#
TRPDFD=  000022         210#
TRYCTR   000242R        236#    339*    356*
TTADS    000246R        238#    254*    255*    379*    381
TWBUF    000230R        231#    284*    335     427
UNIT     000227R        229#    298*    299*    415
UNIT1 =  000400         212#
VECTOR   000010R        159#    256
WASADR   000104R        193#    434*
WDFR     000116R        200#    250*
WDTO     000114R        199#    249*
WRITE    000656R        286     334#
WTBUF    001414R        277     284     448#
XFLAG    000005R        157#
.     =  002014R        448#    449#

. ABS.   000000     000
         002014     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XTAAD0,XTAAD0/SOL/CRF:SYM=DDXCOM,XTAAD0
 RUN-TIME: 1 1 .2 SECONDS
 RUN-TIME RATIO: 12/3=3.8
 CORE USED:  7K  (13 PAGES)
```

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**

* * * L P T S P L   R U N   L O G * * *

```
15:24:07 LPDAT  [LPTLSJ  LPTSPL VERSION 102(2263)/3(61) RUNNING ON LPT630, 12-DEC-78 15:24:07]
15:24:07 LPDAT  [LPTSJS  STARTING JOB XTAAD0, SEQ #7128, REQUEST CREATED AT 12-DEC-78 15:24:07]
15:24:16 LPMSG  [LPTSTF  STARTING FILE DSKZ:XTAAD0.SEQ<057>[400,2704]]
15:25:41 LPMSG  [LPTFPF  FINISHED PRINTING FILE DSKZ:XTAAD0.SEQ<057>[400,2704]]
15:25:41 LPSUM  SPOOLER RUNTIME 1 SECONDS, 15 KCS, 47 DISK READS, 15 PAGES PRINTED
```

/TO:ML21-4:DAVIES -- DISTRIBUTION TO ML21-4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**

/TO:ML21-4:DAVIES -- DISTRIBUTION TO ML21-4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**

/TO:ML21-4:DAVIES -- DISTRIBUTION TO ML21-4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**

/TO:ML21-4:DAVIES -- DISTRIBUTION TO ML21-4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**

/TO:ML21-4:DAVIES -- DISTRIBUTION TO ML21-4, SLOT 134

END USER DAVIES,TOM [400,2704] JOB XTAAD0 SEQ. 7128 DATE 12-DEC-78 15:25:41 MONITOR IPC-F 603 [7A0] **END**