TS03

SUPPLEMENTAL INSTRUCTION
CZTSFDO
TEST

AH-9440D-MC
COPYRIGHT ©75-78
FICHE 1 OF 1

APR 1978
digital
MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE: AC-9439D-MC

PRODUCT NAME: CZTSFDO TS03 SPLMTL INSTR

PROGRAM DATE: MARCH 1978

MAINTAINER: DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MMAY APPEAR IN THIS DOCUMENT

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975,1978 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1. ABSTRACT

THIS PROGRAM IS INTENDED TO BE USED IN ADDITION TO
THE TM11/TU10 INSTRUCTION TEST (MAINDEC-11-DZTMA-C)
TO COMPLETE TESTING OF THE TS03 MAGTAPE SYSTEM.
THE PROGRAM CONSISTS OF ONLY FOUR (4) TESTS WHICH
CHECK THE TMA-11 FEATURES OF DATA TRANSFER AT ODD
BYTE STARTING ADDRESS AND OPERATION INCOMPLETE TIME OUT.

2. REQUIREMENTS

A. ANY PDP-11 PROCESSOR
B. 4K OF CORE
C. CONSOLE TTY
D. TMA-11 TAPE CONTROLLER
E. 1 OR 2 TAPE TRANSPORTS

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE

4. STARTING PROCEDURE

THERE ARE TWO (2) STARTING ADDRESSES THAT MAY BE USED;
200(8) AND 210(8).

A. 200(8): STARTING AT THIS ADDRESS WILL CAUSE A PROGRAM
IDENTIFICATION HEADER TO BE PRINTED, A
REQUEST FOR UNIBUS ADDRESS AND VECTOR AND A
REQUEST FOR ENTRY OF THE UNIT NUMBER (TAPE TRANSPORT SELECT).
THE DEFAULT SELECTION OF UNIT ZERO (0) IS DISPLAYED, AND
MAY BE CHANGED TO ANY NUMBER (0-1) OR UNCHANGED BY
TYPING THE DESIRED NUMBER OR A CARRIAGE RETURN.
IF THE SELECTED UNIT IS NOT AVAILABLE, A MESSAGE WILL
BE PRINTED SO STATING, AND THE UNIT SELECT REQUEST REPEATED.

B. 210(8): STARTING AT THIS ADDRESS WILL NOT PRINT THE HEADER OR THE
UNIT SELECT REQUEST AND IS INTENDED AS A RESTART ADDRESS ONLY.

5. THIS PROGRAM HAS BEEN MODIFIED TO RUN ON A PROCESSOR WITH OR WITHOUT A HARDWARE SWITCH REGISTER. WHEN FIRST EXECUTED THE PROGRAM TESTS THE EXISTENCE OF A HARDWARE SWITCH REGISTER. IF NOT FOUND A SOFTWARE SWITCH REGISTER LOCATION (SWREG=LOC. 176) IS DEFAULTED TO. IF THIS IS THE CASE, UPON EXECUTION THE CONTENTS OF THE SWREG ARE DUMPED IN OCTAL ON THE CONSOLE TTY AND ANY CHANGES ARE REQUESTED

(IE) SWR=XXXXXX NEW=

POSSIBLE RESPONSES ARE:

1.	<CR>	IF NO CHANGES ARE TO BE MADE
2.	6 DIGITS 0-7	TO REPRESENT IN OCTAL THE NEW SWITCH REGISTER VALUE ;LAST DIGIT FOLLOWED BY <CR>.
3.	↑U	TO ALLOW REENTERING VALUE IF ERROR IS COMMITTED KEYING IN SWREG VALUE.
4.	<LF>	ONLY VALID FOR ACT-11 SYSTEMS-DO NOT USE

BUILT INTO THE PROGRAM IS THE ABILITY TO DYNAMICALLY CHANGE THE CONTENTS OF SWREG DURING PROGRAM EXECUTION. BY STRIKING ↑G (CNTL G) ON CONSOLE TTY THE OPERATOR SETS A REQUEST FLAG TO CHANGE THE CONTENTS OF SWREG, WHICH IS PROCESSED IN KEY AREAS OF THE PROGRAM CODE (IE) ERROR ROUTINES, AFTER HALTS END OF PASS, AND OTHER APPLICABLE AREAS.

NOTE ANOTHER MODIFICATION TO THE PROGRAM IS THE TESTING OF 1-8 TS03 UNITS UNDER ACT11M.

5.1 CONSOLE SWITCH SETTING

ALL SWITCHES EXCEPT 3-9 ARE USED AND THE NORMAL, OR DEFAULT, RUN IS DONE WITH ALL SWITCHES SET TO ZERO (0).
ALL SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME.

SW15: 1=HALT ON ERROR
0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
SW13: 1=INHIBIT ERROR TIME OUT
0=PRINT ALL ERRORS
SW12: 1=INHIBIT ITERATION
0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
SW11: 1=CONTINUOUS CYCLE
0=HALT AT END OF PASS
SW10: 1=HALT AT END OF CURRENT TEST
0=CONTINUE
SW9-3: NOT USED
SW2-0: SELECT INDIVIDUAL TEST (1-4)** 00 = DO ALL TESTS

6. ERROR PRINTOUTS

THERE ARE THREE (3) TYPES OF ERROR PRINTOUTS WHICH MAY APPEAR: STATUS ERROR, DATA ERROR, POSITION ERROR.

A. STATUS ERROR: ANY READ, WRITE, OR SPACE OPERATION WHICH RESULTS IN SOME BAD STATUS (BIT 15 OF MTC), OR UNEXPECTED BUS ADDRESS, OR INCORRECT BYTE COUNT, WILL BE PRINTED.

B. DATA ERROR: ANY READ OPERATION WHICH RESULTS IN UNEXPECTED DATA WILL BE PRINTED.

C. POSITION ERROR: ANY SPACE OR REWIND OPERATION RESULTING IN UNEXPECTED STATUS WILL BE PRINTED.

EXAMPLES***

1. THE FOLLOWING EXAMPLE SHOWS A TYPICAL STATUS ERROR.

```
TEST1: WRITE FROM ODD BYTE
WRITE ERROR                 THIS PRINT SHOWS THAT WHILE EXECUTING
MTS: 10101                  TEST 1 ON UNIT 2 AT 800 BPI, A WRITE
MTC: 161204                 PARITY ERROR OCCURED.  THE BYTE COUNT
MTBC: 0                     IS ZERO AS IT SHOULD BE AND THE CURRENT
MTCA: 6003 6003             ADDRESS IS AS EXPECTED.
```

2. THE FOLLOWING EXAMPLE SHOWS A TYPICAL DATA ERROR.

```
TEST 2: READ TO ODD BYTE
DATA ERROR                  THIS PRINT SHOWS THAT A SINGLE BIT WAS
CN: 0                       PICKED UP IN BOTH CHARACTER NUMBER ZERO
G: 00000000                 (0) AND CHARACTER NUMBER THREE (3).
B: 01000000
CN: 3
G: 00000011
B: 01000011
```

3. THE FOLLOWING EXAMPLE SHOWS AN ERROR DURING A REWIND OPERATION.

```
TEST4: OPI TOO LONG
REWIND ERROR: NO BOT
```

7. OPERATION

THE PROCEDURES FOR OPERATING THIS PROGRAM ARE QUITE SIMPLE AND REQUIRE ONLY A FEW STEPS:

1. LOAD ADDRESS 200 OR 210
2. SET SWICHES FOR DESIRED TEST SEQUENCE
3. PRESS START

ALL CONSOLE SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME. THE NORMAL OPERATING SEQUENCE IS ALL SWITCHES DOWN (0). THE PROGRAM WILL TAKE APPROXIMATELY 1.25 MINUTES TO RUN; HOWEVER, IF ITERATIONS ARE INHIBITED (SW[illegible]=1), THE PROGRAM WILL RUN IN ABOUT .75 MINUTES. THE END OF PASS IS NOTED BY A PRINTOUT STATING END OF PASS AND THE NUMBER OF THAT PASS.

SINGLE TEST SELECTION: (SW0-SW3)

WHEN SW0-3 ARE SET TO ZERO (0), THE SCHEDULAR WILL EXECUTE ALL TESTS (1-4) IN SEQUENCE AS A SINGLE PASS. IF SW0-3 ARE SET TO SOME NUMBER BETWEEN 1 AND 4, THEN THAT PARTICULAR TEST WILL BE EXECUTED CONTINUOUSLY. THE PROGRAM MAY BE STOPPED AT THE END OF THE CURRENT TEST (EITHER IN SEQUENCE OR SINGLE TEST MODE) BY SETTING SWITCH TEN (SW10) TO A ONE (1). YOU MAY SELECT TEST NUMBERS IN ANY ORDER (UP OR DOWN) BECAUSE EACH TEST IS SELF CONTAINED.

8. TEST DESCRIPTION:

TEST1: WRITE FROM ODD BYTE

THE PURPOSE OF THIS TEST IS TO ASSURE THAT DATA MAY BE TRANSFERRED FROM MEMORY TO TAPE STARTING FROM AN ODD BYTE ADDRESS. THE TEST WILL WRITE A SIX (6) BYTE RECORD FROM AN ODD ADDRESS (WDATA+1) AND READ THAT RECORD BACK INTO AN EVEN ADDRESS (RDATA). NO STATUS ERROR SHOULD OCCUR, AND THE READ DATA SHOULD BE POSITIONED PROPERLY. THE RECORD IS SIX BYES LONG, EACH BYTE IS ITS NUMBER (0,1,2,3,4,5)

TEST2: READ TO ODD BYTE

THE PURPOSE OF THIS TEST IS TO ASSURE THAT DATA MAY BE TRANSFERRED FROM TAPE TO MEMORY STARTING AT AN ODD BYTE ADDRESS. THE PROCEDURE IS THE SAME AS IN TEST ONE (1), EXCEPT THAT THE WRITE IS FROM AN EVEN ADDRESS (WDATA) AND THE READ IS TO AN ODD ADDRESS (RDATA+1).

TEST3: OPI TOO LONG (OPI = BIT 8 OF MTS)

THE PURPOSE OF THIS TEST IS TO ASSURE THAT THE OPI TIMER WILL SHUTDOWN THE DRIVE BEFORE TEN POINT FIVE FEET OF BLANK TAPE IS PASSED. THE PROCEDURE IS TO PERFORM A WRITE WITH IRG, BACKSPACE, WRITE WITH IRG 32(10) TIMES IN ORDER TO ERASE 10.5 FEET OF TAPE. AFTER REWIND, ISSUE A READ COMMAND AND OPI SHOULD TIME OUT BEFORE THE FIRST RECORD (10.5 FEET DOWN TAPE) IS FOUND. THE NOMINAL VALUE FOR OPI IS SEVEN SECONDS (7SEC) OR ABOUT SEVEN POINT FIVE FEET (7.5 FT) OF TAPE. TEN POINT FIVE FEET OF TAPE REFLECTS THE MAXIMUM TOLERANCE FOR OPI.

TEST4: OPI TOO SHORT (OPI = BIT 8 OF MTS)

THE PURPOSE OF THIS TEST IS TO ASSURE THAT THE OPI TIMER WILL NOT SHUTDOWN THE DRIVE BEFORE FOUR FEET (4 FT) OF BLANK TAPE IS PASSED. THE PROCEDURE IS THE SAME AS IN TEST THREE (3), HOWEVER OPI IS NOT EXPECTED BEFORE THE FIRST RECORD IS FOUND (4 FEET DOWN TAPE). THE FOUR FEET OF TAPE RELECTS THE MINIMUM TOLERANCE FOR OPI.

9. LISTING

%

```
                                .TITLE CZTSFDO TSO3 SPLMTL INSTR
                                ;ZZ - CZTSFDO
                                ;15 APR 75/MARCH 1976/JAN 1978
                                ;R. BARNES/RON PLATUKIS/CLEM WALSH
                                .ABS

                                ;*****************************************************************
                                ;NOTE:   PROGRAM HAS BEEN MODIFIED TO RUN WITH OR WITHOUT
                                ;        A HARDWARE SWITCH REGISTER-REFER TO DOCUMENT
                                ;*****************************************************************

                                ;CONSOLE SWITCHES
                                ;
                                ;SW15:   1=HALT ON ERROR
                                ;        0=CONTINUE
                                ;SW14:   1=LOOP ON ERROR (SCOPE)
                                ;        0=CONTINUE
                                ;SW13:   1=DO NOT PRINT ERRORS
                                ;        0=PRINT ALL ERRORS
                                ;SW12:   1=INHIBIT ITERATIONS
                                ;        0=DO ITERATIONS
                                ;SW11:   1=CONTINUOUS CYCLE
                                ;        0=SINGLE PASS
                                ;SW10:   1=HALT AT END OF EACH TEST
                                ;        0=CONTINUE
                                ;SW0-5: SELECT TEST NUMBER:: 00=ALL TESTS

                                ;REGISTER EQUIVS**********

000000                          R0=%0
000001                          R1=%1
000002                          R2=%2
000003                          R3=%3
000004                          R4=%4
000005                          R5=%5
000006                          SP=%6
000007                          PC=%7

                                ;TRAP CATCHERS**********
```

```
                        ;       ********************************************************************
                        ;                       MODIFIED JAN 24 1978
                        ;
                        ;       ++
                        ;               ACT11 AND XXDP HOOKS
                        ;       --
                                
001000                          $SVPC=.                 ;SAVE PC

000040                          .=40
000040  000             DRIVE:  .BYTE   0               ;DRIVE # FOR XXDP LOAD MEDIUM
                                                        ;ASSEMBLE AS A 0

000041                          .=41
000041  000             MEDIUM: .BYTE   0               ;XXDP LOAD MEDIUM
                                                        ;ASSEMBLE AS A 0

000042                          .=42
000042  000000                  .WORD   0               ;AUTO MAN/MODE INDICATOR
                                                        ;ASSEMBLE AS A 0

000046                          .=46
000046  002006                  .WORD   $ENDAD          ;SET TO $ENDAD IN .$EOP

000052                          .=52
000052  000000                  .WORD   0               ;CHARACTERISTICS OF PROGRAM
                                                        ;ASSEMBLE AS A 0

001000                          .=$SVPC                 ;RESTORE PC

                        ;       ********************************************************************
```

```
353
354                             ;       ********************************************************************
355                             ;                       MODIFIED JAN 24 1978
356                             ;
357                             ;       ++
358                             ;               ACT11 AND XXDP MODE INDICATORS
359                             ;       --
360
361  001000  000000                     AUTOM:  .WORD   0                       ;AUTOMATIC MODE INDICATOR
362  001002     000                     ACT11M: .BYTE   0                       ;ACT11 AUTO MODE INDICATOR
363  001003     000                     XXDPM:  .BYTE   0                       ;XXDP AUTO MODE INDICATOR
364  001004     000                     ADUMPM: .BYTE   0                       ;ACT11 DUMP MODE INDICATOR
365  001005     000                     XDUMPM: .BYTE   0                       ;XXDP DUMP MODE INDICATOR
366
367                             ;       ********************************************************************
368
```

```
   369
```

```
                                        ;TTY INTERRUPT VECTOR**********
                 000060                 .=60
000060  004544                          TTINT                   ;TTY INTERRUPT HANDLER
000062  000000                          0

                                        ;SOFTWARE SWITCH REGISTER LOCATIONS*******
                 000174                 .=174
000174  000000                          DISPREG:0
000176  000000                          SWREG:  0

                                        ;STARTING ADDRESS**********
                 000200                 .=200
000200  005000                          CLR     R0
000202  000167  000572                  JMP     START           ;PROGRAM START

                 000210                 .=210
000210  012700  000001                  MOV     #1,R0
000214  000167  000560                  JMP     START           ;NO HEADER START

                                        ;TMA-11 INTERRUPT VECTOR**********
                 000224                 .=224
000224  004524                          MTINT                   ;TAPE INTERRUPT HANDLER
000226  000340                          340
```

```
                        000600                          .=600
                                                        ;CONSTANTS**********
                                                        
000600  172520          MTS:    172520  ;TAPE STATUS REGISTER
000602  172522          MTC:    172522  ;TAPE COMMAND REGISTER
000604  172524          MTBC:   172524  ;TAPE BYTE COUNTER
000606  172526          MTBA:   172526  ;TAPE BUS ADDRESS
000610  000000          UDES:   0       ;UNIT DESCRIPTION
000612  000020          RCNT:   20      ;RECORD COUNT
000614  177760          CCNT:   -20     ;CHARACTER COUNT
000616  177776          PSW:    177776  ;PROCESSOR STATUS
000620  177570          SWR:    177570  ;CONSOLE SWITCH REGISTER
000622  177570          DISPLAY:177570  ;CONSOLE DISPLAY REGISTER
000624  177560          TKS:    177560  ;TTY READ STATUS
000626  177562          TKB:    177562  ;TTY READ BUFFER
000630  177564          TPS:    177564  ;TTY PUNCH STATUS
000632  177566          TPB:    177566  ;TTY OUTPUT BUFFER
000634  000010          ITAMT:  10      ;NUMBER OF ITERATIONS
000636  000040          STALL:  40      ;READY DELAY MULTIPLIER
000640  172520          REGS:   172520  ;UNIBUS ADDRESS
000642  000224          VECT:   224     ;VECTOR ADDRESS
```

```
                                        ;FLAGS AND COUNTERS**********
                                
000644  000000                  TINF:   0
000646  000000                  TOB:    0
000650  000000                  TIB:    0
000652  000000                  TEMP1:  0
000654  000000                  TEMP2:  0
000656  000000                  TEMP3:  0
000660  000000                  EMADDR: 0
000662  000000                  ERRAD:  0
000664  000000                  LTADD:  0
000666  000000                  ITRLP:  0
000670  000000                  SPFLG:  0
000672  000000                  STFLG:  0
000674  000000                  PCNTR:  0
000676  000000                  BADR:   0
000700  000000                  BYTES:  0
000702  000000                  SCNT:   0
000704  000000                  FUN:    0
000706  000000                  ITCNT:  0
000710  000000                  CRCNT:  0
000712  000000                  DERFL:  0
000714  000000                  HDRFL:  0
000716  000000                  PFLG:   0
000720  000000                  UNP:    0
000722  000000                  BCNT:   0
000724  000000                  COUNT:  0
000726  000000                  TEMPST: 0
000730  000000                  RDSW:   0

                                        ;TEST ENTRY TABLE**********

000732  000000                  TSTTBL: 0
000734  000000                          0
000736  002064                  T1AD:   LT1
000740  002064                  T1IAD:  LT1
000742  002312                  T2AD:   LT2
000744  002312                  T2IAD:  LT2
000746  002534                  T3AD:   LT3
000750  002632                  T3IAD:  LT3IT
000752  002764                  T4AD:   LT4
000754  003062                  T4IAD:  LT4IT
000756  000000                          0
```

```
                001000                          .=1000
                                                ;*******************************
                                                ;PROGRAM START AND HOUSEKEEPING
                                                ;
                                                ;*******************************

001000  000240                          START:  NOP
001002  012777  000340  177606                  MOV     #340,@PSW       ;SET PRIORITY
001010  012706  000500                          MOV     #500,SP         ;SET STACK POINTER
001014  004767  004524                          JSR     PC,SUSWR        ;SEE IF NO HARDWARE SWITCH REG.
001020  004767  005160                          JSR     PC,CKMODE       ;CHECK FOR MODE OF OPERATION ++ C.W
001024  105767  177752                          TSTB    ACT11M          ;ACT MODE? ++ C.W
001030  001404                                  BEQ     1$              ;BRANCH - IF NO ++ C.W
001032  005067  177636                          CLR     PCNTR           ;INIT PASS COUNTER ++ C.W
001036  000167  000256                          JMP     $ACT            ;EXECUTE PROGRAM IN ACT MODE ++ C.W
001042  005700                          1$:     TST     R0              ;SEE IF SKIP HEADER
001044  001402                                  BEQ     2$              ;BRANCH - IF NO ++ C.W
001046  000167  000416                          JMP     ST4             ;DO TEST FROM RESTART ++ C.W
001052  012704  006410                  2$:     MOV     #MSG1,R4
001056  004767  003730                          JSR     PC,TTOUT        ;PRINT HEADER
001062  012704  007041                          MOV     #MSG22,R4
001066  004767  003720                          JSR     PC,TTOUT        ;REQUEST UNIBUS ADDRESS
001072  016703  177542                          MOV     REGS,R3
001076  004767  004106                          JSR     PC,OCTP         ;PRINT CURRENT ADDRESS
001102  012705  000640                          MOV     #REGS,R5        ;GET ADDRESS OF ENTRY
001106  012701  000006                          MOV     #6,R1           ;SET SIZE OF ENTRY
001112  012702  172700                          MOV     #172700,R2      ;SET UPPER LIMIT
001116  012703  172300                          MOV     #172300,R3      ;SET LOWER LIMIT
001122  004767  003426                          JSR     PC,TTR          ;GO GET RESPONSE
001126  012704  007063                          MOV     #MSG23,R4
001132  004767  003654                          JSR     PC,TTOUT        ;REQUEST VECTOR
001136  016703  177500                          MOV     VECT,R3
001142  004767  004042                          JSR     PC,OCTP         ;PRINT CURRENT VECTOR
001146  012705  000642                          MOV     #VECT,R5        ;GET ADDRESS OF ENTRY
001152  012701  000003                          MOV     #3,R1           ;SET SIZE OF ENTRY
001156  012702  000250                          MOV     #250,R2         ;SET UPPER LIMIT
001162  012703  000150                          MOV     #150,R3         ;SET LOWER LIMIT
001166  004767  003362                          JSR     PC,TTR          ;GO GET RESPONSE
001172  016700  177444                          MOV     VECT,R0         ;GET VECTOR
001176  012720  004524                          MOV     #MTINT,(R0)+    ;SET INTERRUPT ADDRESS IN VECTOR
001202  012710  000340                          MOV     #340,(R0)       ;SET INTERRUPT PRIORITY
001206  012701  000004                          MOV     #4,R1           ;SET NUMBER OF REGISTER
001212  012702  000600                          MOV     #MTS,R2         ;GET FIRST ADDRESS OF TABLE
001216  016700  177416                          MOV     REGS,R0         ;GET FIRST REGISTER
001222  010022                          STOA:   MOV     R0,(R2)+        ;LOAD TABLE
001224  062700  000002                          ADD     #2,R0           ;BUMP ADDRESS
001230  005301                                  DEC     R1              ;SEE IF DONE
001232  001373                                  BNE     STOA            ;IF NOT: BE
001234  012777  010000  177340                  MOV     #10000,@MTC     ;POWER CLEAR
001242  012704  006444                  STO:    MOV     #MSG2,R4
001246  004767  003540                          JSR     PC,TTOUT        ;REQUEST UNIT NUMBER
001252  005067  177332                          CLR     UDES            ;PRESET UNIT 0
001256  016703  177326                          MOV     UDES,R3         ;GET UNIT NUMBER
001262  000303                                  SWAB    R3              ;POSITION
001264  042703  177770                          BIC     #177770,R3      ;MASK UNIT NUMBER
001270  004767  003714                          JSR     PC,OCTP         ;PRINT CURRENT VALUE
```

```
                                                MOV     #TEMP3,R5       ;SET SAVE LOCATION
                                                MOV     #1,R1           ;SET SIZE OF ENTRY
                                                MOV     #7,R2           ;SET UPPER LIMIT
                                                MOV     #0,R3           ;SET LOWER LIMIT
                                                JSR     PC,TTR          ;GO GET UNIT NUMBER
                                        $ACT:   TSTB    ACT11M          ;ACT MODE? ++ C.W
                                                BEQ     NOACT           ;BRANCH - IF NO ++ C.W
                                                CLR     UDES            ;INIT UNIT INDICATOR IN ACT MODE ++ C.W
                                        ST:     CMPB    #10,UDES        ;ARE ALL UNITS DONE? ++ C.W
                                                BNE     2$              ;BRANCH - IF NO ++ C.W
                                                JMP     $DONE           ;DO END OF PASS ++ C.W
                                        2$:     MOVB    UDES,TEMP3      ;GET UNIT NUMBER IN ACT MODE ++ C.W
                                        NOACT:  SWAB    TEMP3           ;POSITION UNIT NUMBER
                                                BIC     #3400,UDES      ;CLEAR OLD NUMBER
                                                BIS     TEMP3,UDES      ;LOAD NEW NUMBER
                                                MOV     UDES,@MTC       ;SELECT UNIT
                                                CLR     R0
                                                CMP     #SWREG,SWR
                                                BNE     ST1
                                                JSR     PC,CNTLU        ;GO ASK FOR SWR SETTINGS
                                        ST1:    BIT     #100,@MTS       ;SEE IF SELECT REMOTE
                                                BNE     ST3             ;IF SO: BR
                                                DEC     R0
                                                BNE     ST1             ;DELAY FOR SELECT REMOTE
                                                TSTB    ACT11M          ;ACT MODE? ++ C.W
                                                BEQ     1$              ;BRANCH - IF NO ++ C.W
                                                INCB    UDES            ;GET NEXT UNIT ++ C.W
                                                BR      ST              ;CONTINUE ++ C.W
                                        1$:     MOV     #MSG21,R4
                                                JSR     PC,TTOUT        ;PRINT NOT AVAILABLE
                                                BR      ST0             ;REDO REQUEST
                                        ST3:    BIS     #60000,UDES     ;SET TO 800 BPI 9 CHAN
                                        ST4:    NOP
                                                MOV     #10000,@MTC     ;POWER CLEAR
                                                MOV     #TOB,R2         ;GET START OF TABLE
                                                MOV     #27,R0          ;SET SIZE OF TABLE
                                        ST5:    CLR     (R2)+           ;CLEAR TABLE
                                                DEC     R0
                                                BNE     ST5             ;DO ALL
                                                TSTB    ACT11M          ;ACT MODE? ++ C.W
                                                BNE     TSCD            ;BRANCH - IF YES ++ C.W
                                                CLR     PCNTR           ;CLEAR PASS COUNTER
```

```
                                                ;TEST SCHEDULAR**********
                                        TSCD:   NOP
001530  000240                                    CLR     STFLG           ;CLEAR SINGLE TEST FLAG
001532  005067  177134                           MOV     @SWR,R0         ;GET SWITCH REGISTER
001536  017700  177056                           BIC     #177700,R0      ;MASK TEST SELECT
001542  042700  177700                           TST     R0              ;SEE IF SINGLE TEST SELECT
001546  005700                                    BNE     STSCD           ;IF SO: BR
001550  001046                                    MOV     #TSTTBL,LTADD   ;GET TABLE START
001552  012767  000732  177104                   ADD     #4,LTADD
001560  062767  000004  177076          TSCD0:   MOV     LTADD,ITRLP     ;SET ITERATION ADDRESS
001566  016767  177072  177072                   ADD     #2,ITRLP
001574  062767  000002  177064                   TST     @LTADD          ;SEE IF END OF CYCLE
001602  005777  177056                           BNE     TSCD1           ;IF NOT: BR
001606  001002                                    JMP     TEND            ;GO TO END ROUTINE
001610  000167  000124                           CLR     HDRFL           ;CLEAR HEADER FLAG
001614  005067  177074          TSCD1:   CLR     @LTADD,R0       ;GET TEST ADDRESS
001620  017700  177040                           JMP     (R0)            ;GO TO TEST
001624  000110                                    BIT     #2000,@SWR      ;SEE IF HALT ON TEST
001626  032777  002000  176764  TSCD2:   BIT     TSCD3           ;IF NOT: BR
001634  001401                                    HALT
001636  000000                                    JSR     PC,CKSWR        ;TEST FOR ↑G
001640  004767  003760          TSCD3:   JSR     STFLG           ;SEE IF SINGLE TEST
001644  005767  177022                           BEQ     TSCD0           ;IF NOT: BR
001650  001743                                    MOV     @SWR,R0
001652  017700  176742                           BIC     #177760,R0      ;GET TEST NUMBER
001656  042700  177760                           TST     R0              ;SEE IF ALL TESTS
001662  005700                                    BEQ     TSCD            ;IF SO: BR
001664  001721                                    MOV     #1,STFLG        ;SET SINGLE TEST FLAG
001666  012767  000001  176776  STSCD:   CMP     #5,R0           ;SEE IF EXCEEDED TEST NUMBER
001674  022700  000005                           BLE     TEND            ;IF SO: BR
001700  003417                                    CLC
001702  000241                                    ROL     R0
001704  006100                                    ROL     R0              ;POSITION NUMBER
001706  006100                                    MOV     #TSTTBL,LTADD   ;GET START OF TABLE
001710  012767  000732  176746                   ADD     R0,LTADD        ;SET POINTER
001716  060067  176742                           MOV     LTADD,ITRLP
001722  016767  176736  176736                   ADD     #2,ITRLP        ;SET ITERATION ADDRESS
001730  062767  000002  176730                   BR      TSCD1           ;GO DO TEST
001736  000726                          TEND:    TSTB    ACT11M          ;ACT MODE? ++ C.W
001740  105767  177036                           BEQ     $DONE           ;BRANCH - IF NO ++ C.W
001744  001404                                    INCB    UDES            ;GET NEXT UNIT ++ C.W
001746  105267  176636                           JMP     ST              ;AND CONTINUE TESTING ++ C.W
001752  000167  177354
001756  012704  006470          $DONE:   MOV     #MSG3,R4
001762  004767  003024                           JSR     PC,TTOUT        ;PRINT END OF PASS
001766  016703  176702                           MOV     PCNTR,R3
001772  004767  003212                           JSR     PC,OCTP         ;PRINT PASS NUMBER
001776  013700  000042                           MOV     @#42,R0         ;GET MONITOR ADDRESS ++ C.W
002002  001405                                    BEQ     HERE            ;BRANCH - IF NOT AUTO MODE ++ C.W
002004  000005                                    RESET                   ;CLEAR THE WORLD ++ C.W
002006  004710                          $ENDAD:  JSR     PC,(R0)         ;GO TO MONITOR'S ADDRESS ++ C.W
002010  000240                                    NOP
002012  000240                                    NOP
002014  000240                                    NOP
002016  000240                          HERE:    NOP
002020  032777  004000  176572                   BIT     #4000,@SWR      ;SEE IF HALT ON PASS
```

```
   620  002026  001001                        BNE     TENDX           ;IF NOT: BR
   621  002030  000000                        HALT
   622  002032  105767  176744        TENDX:  TSTB    ACT11M          ;ACT MODE? ++ C.W
   623  002036  001006                        BNE     1$              ;BRANCH - IF YES ++ C.W
   624  002040  004767  003560                JSR     PC,CKSWR        ;TEST FOR ↑G
   625  002044  005267  176624                INC     PCNTR           ;BUMP PASS COUNTER
   626  002050  000167  177454                JMP     TSCU            ;RESTART
   627  002054  005267  176614        1$:     INC     PCNTR           ;BUMP COUNTER IN ACT MODE ++ C.W
   628  002060  000167  177234                JMP     $ACT            ;RESTART IN ACT MODE ++ C.W
   629
```

```
630                                     ;*************************************
631                                     ;TEST 1: WRITE FROM ODD BYTE
632                                     ;
633                                     ;THIS TEST WILL WRITE A SIX (6) BYTE RECORD
634                                     ;FROM AN ODD BYTE STARTING ADDRESS. THE RECORD
635                                     ;WILL BE READ BACK INTO AN EVEN STARTING ADDRESS
636                                     ;TO TEST FOR PROPER TRANSFER.
637                                     ;*************************************
638
639  002064  000240                 LT1:    NOP
640  002066  012767  007075  176564         MOV     #LT1MSG,EMADDR  ;SET HEADER
641  002074  012702  007352                 MOV     #WDATA,R2       ;GET BUFFER START
642  002100  112722  000377                 MOVB    #377,(R2)+      ;INSERT BACKGROUND DATA
643  002104  005000                         CLR     R0
644  002106  110022                 LT1B:   MOVB    R0,(R2)+
645  002110  005200                         INC     R0              ;LOAD WRITE BUFFER (0,1,2,3,4,5)
646  002112  022700  000006                 CMP     #6,R0
647  002116  001373                         BNE     LT1B
648  002120  004767  001046                 JSR     PC,RWND         ;GO REWIND
649  002124  012767  000004  176552         MOV     #4,FUN          ;SET WRITE FUNCTION CODE
650  002132  012767  007353  176536         MOV     #WDATA+1,BADR   ;SET DATA POINTER
651  002140  012767  177772  176532         MOV     #-6,BYTES       ;SET SIZE OF RECORD
652  002146  012767  006715  176506         MOV     #MSG17,ERRAD    ;SET WRITE ERROR CODE
653  002154  004767  001262                 JSR     PC,EXEC         ;GO EXECUTE COMMAND
654  002160  000240                 LT1C:   NOP
655  002162  004767  001504                 JSR     PC,ERCHK        ;GO CHECK FOR STATUS ERROR
656  002166  012767  177777  176506         MOV     #-1,SCNT
657  002174  004767  001156                 JSR     PC,BKSP         ;GO BACKSPACE ONE RECORD
658  002200  012702  007454                 MOV     #RDATA,R2
659  002204  012700  000010                 MOV     #10,R0
660  002210  012722  177777         LT1D:   MOV     #-1,(R2)+       ;BACKGROUND READ BUFFER
661  002214  005300                         DEC     R0
662  002216  001374                         BNE     LT1D            ;DO ALL
663  002220  012767  000002  176456         MOV     #2,FUN          ;SET READ FUNCTION CODE
664  002226  012767  007454  176442         MOV     #RDATA,BADR     ;SET READ POINTER
665  002234  012767  177772  176436         MOV     #-6,BYTES       ;SET SIZE OF RECORD
666  002242  012767  006732  176412         MOV     #MSG18,ERRAD    ;SET READ ERROR CODE
667  002250  004767  001166                 JSR     PC,EXEC         ;GO DO READ
668  002254  000240                 LT1E:   NOP
669  002256  004767  001410                 JSR     PC,ERCHK        ;GO CHECK ERRORS
670  002262  012701  007353                 MOV     #WDATA+1,R1     ;SET EXPT DATA POINTER
671  002266  012702  007454                 MOV     #RDATA,R2       ;SET RCVD DATA POINTER
672  002272  012700  000006                 MOV     #6,R0           ;SET SIZE OF RECORD
673  002276  004767  001656                 JSR     PC,DCHK         ;GO CHECK DATA
674  002302  004767  002146                 JSR     PC,ITER         ;GO SEE IF ITERATIONS
675  002306  000167  177314                 JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
                                                ;************************
                                                ;TEST 2: READ INTO ODD BYTE
                                                ;
                                                ;THIS TEST WILL WRITE A SIX (6) BYTE RECORD
                                                ;FROM AN EVEN BYTE STARTING ADDRESS. THE RECORD
                                                ;WILL BE READ BACK INTO AN ODD STARTING ADDRESS
                                                ;TO TEST FOR PROPER TRANSFER.
                                                ;************************
002312  000240                        LT2:    NOP
002314  012767  007133  176336                MOV     #LT2MSG,EMADDR  ;SET HEADER POINTER
002322  012702  007352                        MOV     #WDATA,R2       ;POINT TO START OF WRITE BUFFER
002326  005000                                CLR     R0
002330  110022                        LT2B:   MOVB    R0,(R2)+        ;LOAD DATA PATTERN
002332  005200                                INC     R0              ;BUMP PATTERN
002334  022700  000006                        CMP     #6,R0           ;SEE IF DONE
002340  001373                                BNE     LT2B            ;IF NOT: BR
002342  004767  000624                        JSR     PC,RWND         ;GO REWIND TO BOT
002346  012767  000004  176330                MOV     #4,FUN          ;SET WRITE OP-CODE
002354  012767  007352  176314                MOV     #WDATA,BADR     ;SET STARTING ADDRESS
002362  012767  177772  176310                MOV     #-6,BYTES       ;SET SIZE OF RECORD
002370  004767  001046                        JSR     PC,EXEC         ;GO EXECUTE COMMAND
002374  000240                        LT2C:   NOP
002376  012767  006715  176256                MOV     #MSG17,ERRAD    ;SET ERROR CODE
002404  004767  001262                        JSR     PC,ERCHK        ;GO CHECK FOR STATUS ERROR
002410  012767  177777  176264                MOV     #-1,SCNT
002416  004767  000734                        JSR     PC,BKSP         ;GO BACKSPACE ONE RECORD
002422  012702  007454                        MOV     #RDATA,R2       ;GET READ BUFFER POINTER
002426  012700  000010                        MOV     #10,R0          ;SET SIZE
002432  012722  177777                LT2D:   MOV     #-1,(R2)+       ;BACKGROUND POINTER
002436  005300                                DEC     R0              ;SEE IF DONE
002440  001374                                BNE     LT2D            ;IF NOT: BR
002442  012767  000002  176234                MOV     #2,FUN          ;SET READ FUNCTION CODE
002450  012767  007455  176220                MOV     #RDATA+1,BADR   ;SET START OF READ BUFFER
002456  012767  177772  176214                MOV     #-6,BYTES       ;SET SIZE OF RECORD
002464  004767  000752                        JSR     PC,EXEC         ;GO EXECUTE COMMAND
002470  000240                        LT2E:   NOP
002472  012767  006732  176162                MOV     #MSG18,ERRAD    ;SET ERROR CODE
002500  004767  001166                        JSR     PC,ERCHK        ;GO CHECK FOR STATUS ERROR
002504  012701  007352                        MOV     #WDATA,R1       ;POINT TO EXPT DATA
002510  012702  007455                        MOV     #RDATA+1,R2     ;POINT TO RCVD DATA
002514  012700  000006                        MOV     #6,R0           ;SET SIZE OF RECORD
002520  004767  001434                        JSR     PC,DCHK         ;GO CHECK DATA
002524  004767  001724                        JSR     PC,ITER         ;GO SEE IF ITERATION
002530  000167  177072                        JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
                                        ;*************************
                                        ;TEST 3: OPI TOO LONG
                                        ;
                                        ;THIS TEST WILL ERASE APPROXIMATELY TEN POINT FIVE (10.5)
                                        ;FEET OF TAPE BY WRITING WITH IRG, BACKSPACING
                                        ;AND REPEATING THE SEQUENCE 32(10) TIMES. TAPE
                                        ;WILL REWIND AND A READ FORWARD ISSUED.  THE
                                        ;OPI TIMER SHOULD SHUTDOWN THE UNIT BEFORE
                                        ;REACHING THE FIRST RECORD ON TAPE.
                                        ;*************************
002534  000240                  LT3:    NOP
002536  012767  007166  176114          MOV     #LT3MSG,EMADDR  ;SET TEST HEADER
002544  012700  000040                  MOV     #40,R0          ;SET NUMBER OF WRITE IRG/BACKSPACE
002550  004767  000416                  JSR     PC,RWND         ;GO REWIND UNIT
002554  012767  000014  176122  LT3A:   MOV     #14,FUN         ;SET WRITE IRG FUNCTION CODE
002562  012767  007352  176106          MOV     #WDATA,BADR     ;SET BUS ADDRESS
002570  012767  177760  176102          MOV     #-20,BYTES      ;SET SIZE OF RECORD
002576  004767  000640                  JSR     PC,EXEC         ;GO EXECUTE COMMAND
002602  012767  006715  176052  LT3B:   MOV     #MSG17,ERRAD    ;SET ERROR CODE
002610  004767  001056                  JSR     PC,ERCHK        ;GO CHECK FOR STATUS ERROR
002614  012767  177777  176060          MOV     #-1,SCNT
002622  004767  000530                  JSR     PC,BKSP         ;GO BACKSPACE ONE RECORD
002626  005300                          DEC     R0              ;SEE IF DONE ALL
002630  001351                          BNE     LT3A            ;IF NOT: BR
002632  000240                  LT3IT:  NOP
002634  004767  000332                  JSR     PC,RWND         ;GO REWIND
002640  012767  000230  175770          MOV     #230,STALL      ;SET OPI STALL
002646  012767  007454  176022          MOV     #RDATA,BADR     ;SET START OF READ BUFFER
002654  012767  177760  176016          MOV     #-20,BYTES      ;SET SIZE OF RECORD
002662  012767  000002  176014          MOV     #2,FUN          ;SET READ FUNCTION CODE
002670  012767  006746  175764          MOV     #MSG19,ERRAD    ;SET ERROR CODE
002676  004767  000540                  JSR     PC,EXEC         ;GO EXECUTE COMMAND
002702  000240                  LT3C:   NOP
002704  012767  000040  175724          MOV     #40,STALL       ;RESET NORMAL STALL
002712  032777  000400  175660          BIT     #400,@MTS       ;SEE IF BTE IS SET
002720  001007                          BNE     LT3X            ;IF SO: BR
002722  012767  000001  175740          MOV     #1,SPFLG        ;SET NO BA PRINT FLAG
002730  004767  001014                  JSR     PC,ERPT         ;GO PRINT ERROR
002734  005067  175730                  CLR     SPFLG           ;RESET FLAG
002740  012767  000002  175666  LT3X:   MOV     #2,ITAMT        ;SET TO TWO (2) ITERATIONS
002746  004767  001502                  JSR     PC,ITER         ;GO SEE IF ITERATION
002752  012767  000010  175654          MOV     #10,ITAMT       ;RESET ITERATIONS
002760  000167  176642                  JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
                                                ;************************
                                                ;TEST 4: OPI TOO SHORT
                                                ;
                                                ;THIS TEST WILL ERASE APPROXIMATELY FOUR (4) FEET
                                                ;OF TAPE BY WRITING WITH IRG, BACKSPACING
                                                ;ONE (1) RECORD AND REPEATING THIS SEQUENCE
                                                ;12(10) TIMES.  TAPE WILL REWIND AND BE READ
                                                ;FORWARD.  THE FIRST RECORD ON TAPE SHOULD BE
                                                ;REACHED BEFORE OPI TIMES OUT.
                                                ;************************
002764  000240                          LT4:    NOP
002766  012767  007215  175664                  MOV     #LT4MSG,EMADDR  ;SET HEADER
002774  004767  000172                          JSR     PC,RWND         ;GO REWIND
003000  012700  000014                          MOV     #14,R0          ;SET NUMBER OF WRITE IRG/BACKSPACES
003004  012767  000014  175672          LT4A:   MOV     #14,FUN         ;SET WRITE IRG FUNCTION CODE
003012  012767  007352  175656                  MOV     #WDATA,BADR     ;SET START OF WRITE BUFFER
003020  012767  177760  175652                  MOV     #-20,BYTES      ;SET SIZE OF RECORD
003026  012767  006715  175626                  MOV     #MSG17,ERRAD    ;SET ERROR CODE
003034  004767  000402                          JSR     PC,EXEC         ;GO EXECUTE COMMAND
003040  004767  000626                  LT4B:   JSR     PC,ERCHK        ;GO CHECK FOR STATUS ERROR
003044  012767  177777  175630                  MOV     #-1,SCNT
003052  004767  000300                          JSR     PC,BKSP         ;GO BACKSPACE ONE RECORD
003056  005300                                  DEC     R0              ;SEE IF DONE ALL SEQUENCES
003060  001351                                  BNE     LT4A            ;IF NOT: BR
003062  000240                          LT4IT:  NOP
003064  004767  000102                          JSR     PC,RWND         ;REWIND
003070  012767  000200  175540                  MOV     #200,STALL      ;SET OPI STALL
003076  012767  007454  175572                  MOV     #RDATA,BADR     ;SET START OF READ BUFFER
003104  012767  177760  175566                  MOV     #-20,BYTES      ;SET SIZE OF RECORD
003112  012767  000002  175564                  MOV     #2,FUN          ;SET READ FUNCTION CODE
003120  012767  006773  175534                  MOV     #MSG20,ERRAD    ;SET ERROR CODE
003126  004767  000310                          JSR     PC,EXEC         ;GO EXECUTE COMMAND
003132  004767  000534                  LT4C:   JSR     PC,ERCHK        ;GO CHECK FOR STATUS ERRORS
003136  000240                                  NOP
003140  012767  000040  175470                  MOV     #40,STALL       ;RESET NORMAL STALL
003146  012767  000002  175460                  MOV     #2,ITAMT        ;SET TO TWO (2) ITERATIONS
003154  004767  001274                          JSR     PC,ITER         ;GO SEE IF ITERATIONS
003160  012767  000010  175446                  MOV     #10,ITAMT       ;RESET ITERATIONS
003166  000167  176434                          JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
                                                ;REWIND SUBROUTINE**********
                                
003172  000240                          RWND:   NOP
003174  016777  175410  175400                  MOV     UDES,@MTC       ;SELECT UNIT
003202  032777  000040  175370                  BIT     #40,@MTS        ;SEE IF AT BOT
003210  001056                                  BNE     RWNDXX          ;IF SO: BR
003212  052777  000017  175362                  BIS     #17,@MTC        ;START REWIND
003220  105777  175356                  1$:     TSTB    @MTC
003224  100375                                  BPL     1$              ;AWAIT CUR
003226  032777  000001  175344          RWND1:  BIT     #1,@MTS         ;AWAIT TUR
003234  001774                                  BEQ     RWND1
003236  032777  000040  175334                  BIT     #40,@MTS        ;SEE IF BOT SET
003244  001040                                  BNE     RWNDXX          ;IF SO: BR
003246  032777  020000  175344                  BIT     #20000,@SWR     ;SEE IF PRINT ERROR
003254  001030                                  BNE     RWNDX           ;IF NOT: BR
003256  016704  175376                          MOV     EMADDR,R4
003262  004767  001524                          JSR     PC,TTOUT        ;PRINT HEADER
003266  012704  006511                          MOV     #MSG4,R4
003272  004767  001514                          JSR     PC,TTOUT        ;PRINT REWIND ERROR
003276  012704  006541                          MOV     #MSG5,R4
003302  004767  001504                          JSR     PC,TTOUT        ;PRINT MTS TAG
003306  017703  175266                          MOV     @MTS,R3
003312  004767  001662                          JSR     PC,OCTPE        ;PRINT MTS
003316  012704  006550                          MOV     #MSG6,R4
003322  004767  001464                          JSR     PC,TTOUT        ;PRINT MTC TAG
003326  017703  175250                          MOV     @MTC,R3
003332  004767  001642                          JSR     PC,OCTPE        ;PRINT MTC
003336  005777  175256                  RWNDX:  TST     @SWR            ;SEE IF HALT ON ERROR
003342  100001                                  BPL     RWNDXX          ;IF NOT: BR
003344  000000                                  HALT
003346  004767  002252                  RWNDXX: JSR     PC,CKSWR        ;TEST FOR ↑G
003352  000240                                  NOP
003354  000207                                  RTS     PC              ;RETURN

                                                ;BACKSPACE SUBROUTINE**********

003356  000240                          BKSP:   NOP
003360  016777  175224  175214                  MOV     UDES,@MTC       ;SELECT UNIT
003366  016777  175310  175210                  MOV     SCNT,@MTBC      ;SET NUMBER OF RECORDS TO SPACE
003374  052777  000013  175200                  BIS     #13,@MTC        ;START SPACE REVERSE
003402  032777  000001  175170  BKSP1:  BIT     #1,@MTS
003410  001774                                  BEQ     BKSP1           ;AWAIT TUR
003412  012767  000001  175250                  MOV     #1,SPFLG        ;SET SPACE FLAG
003420  012767  006577  175234                  MOV     #MSG9,ERRAD
003426  004767  000240                          JSR     PC,ERCHK        ;GO CHECK FOR ERROR
003432  005067  175232                          CLR     SPFLG           ;CLEAR SPACE FLAG
003436  000240                                  NOP
003440  000207                                  RTS     PC              ;RETURN
```

```
                                                ;COMMAND EXECUTE SUBROUTINE**********

003442  000240                        EXEC:   NOP
003444  005005                                CLR     R5
003446  032777  000200  175126        EXEC0:  BIT     #200,@MTC       ;SEE IF CUR
003454  001021                                BNE     EXEC2           ;IF SO: BR
003456  005305                                DEC     R5              ;SEE IF TIMED OUT
003460  001372                                BNE     EXEC0           ;IF NOT: BR
003462  005767  175226                        TST     HDRFL           ;SEE IF DONE HEADER
003466  001004                                BNE     EXEC1           ;IF SO: BR
003470  016704  175164                        MOV     EMADDR,R4
003474  004767  001312                        JSR     PC,TTOUT        ;ELSE PRINT HEADER
003500  012704  006622                EXEC1:  MOV     #MSG10,R4
003504  004767  001302                        JSR     PC,TTOUT        ;PRINT NOT READY ERROR
003510  005777  175104                        TST     @SWR            ;SEE IF HALT ON ERROR
003514  100001                                BPL     EXEC2           ;IF NOT: BR
003516  000000                                HALT
003520  004767  002100                EXEC2:  JSR     PC,CKSWR        ;TEST FOR ↑G
003524  000240                                NOP
003526  016777  175056  175046                MOV     UDES,@MTC       ;SELECT UNIT
003534  016777  175136  175044                MOV     BADR,@MTBA      ;SET BUS MEMORY ADDRESS
003542  016777  175132  175034                MOV     BYTES,@MTBC     ;SET BYTE COUNT
003550  016701  175130                        MOV     FUN,R1          ;GET FUNCTION
003554  052701  000101                        BIS     #101,R1         ;SET IN GO BIT AND INTERRUPT ENABLE
003560  050177  175016                        BIS     R1,@MTC         ;LOAD COMMAND+GO+IE
003564  000240                                NOP
003566  005077  175024                        CLR     @PSW            ;ALLOW INTERRUPTS
003572  016767  175040  175052                MOV     STALL,TEMP1     ;SET READY STALL
003600  005001                                CLR     R1
003602  005301                        EXEC3:  DEC     R1
003604  001376                                BNE     EXEC3           ;AWAIT INTERRUPT
003606  005367  175040                        DEC     TEMP1
003612  001373                                BNE     EXEC3
003614  032777  020000  174776                BIT     #20000,@SWR     ;SEE IF PRINT ERROR
003622  001013                                BNE     EXECX           ;IF NOT: BR
003624  005767  175064                        TST     HDRFL           ;SEE IF DONE HEADER
003630  001004                                BNE     EXEC4           ;IF SO: BR
003632  016704  175022                        MOV     EMADDR,R4
003636  004767  001150                        JSR     PC,TTOUT        ;PRINT HEADER
003642  012704  006637                EXEC4:  MOV     #MSG11,R4
003646  004767  001140                        JSR     PC,TTOUT        ;PRINT NO INTERRUPT MESSAGE
003652  005777  174742                EXECX:  TST     @SWR            ;SEE IF HALT ON ERROR
003656  100001                                BPL     EXECXX          ;IF NOT: BR
003660  000000                                HALT
003662  004767  001736                EXECXX: JSR     PC,CKSWR        ;TEST FOR ↑G
003666  000240                                NOP
003670  000207                                RTS     PC              ;RETURN TO CALLER
```

```
                                                ;STATUS ERROR CHECK SUBROUTINE**********
                                                
003672  005777  174704          ERCHK:  TST     @MTC            ;SEE IF ANY ERROR BITS
003676  100002                          BPL     ERCHK1          ;IF NOT: BR
003700  000167  000044                  JMP     ERPT            ;ELSE PRINT ERROR
003704  005777  174674          ERCHK1: TST     @MTBC           ;SEE IF BYTE COUNT IS ZERO
003710  001402                          BEQ     ERCHK2          ;IF SO: BR
003712  000167  000032                  JMP     ERPT            ;ELSE PRINT ERROR
003716  016703  174756          ERCHK2: MOV     BYTES,R3
003722  005403                          NEG     R3
003724  066703  174746                  ADD     BADR,R3         ;SET EXPT BUS ADDRESS
003730  005767  174734                  TST     SPFLG           ;SEE IF SPACE OPERATION
003734  001401                          BEQ     ERCHK3          ;IF NOT: BR
003736  000207                          RTS     PC
003740  020377  174642          ERCHK3: CMP     R3,@MTBA        ;SEE IF EXPT=RCVD
003744  001001                          BNE     ERPT            ;IF NOT: BR
003746  000207                          RTS     PC              ;ELSE EXIT
003750  000240                  ERPT:   NOP
003752  032777  020000  174640          BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
003760  001067                          BNE     ERPTX           ;IF NOT: BR
003762  005767  174726                  TST     HDRFL           ;SEE IF DONE HEADER
003766  001006                          BNE     ERPT1           ;IF SO: BR
003770  016704  174664                  MOV     EMADDR,R4
003774  004767  001012                  JSR     PC,TTOUT        ;ELSE PRINT HEADER
004000  005267  174710                  INC     HDRFL           ;SET FLAG
004004  016704  174652          ERPT1:  MOV     ERRAD,R4
004010  004767  000776                  JSR     PC,TTOUT        ;PRINT ERROR CODE
004014  012704  006541                  MOV     #MSG5,R4
004020  004767  000766                  JSR     PC,TTOUT        ;PRINT MTS TAG
004024  017703  174550                  MOV     @MTS,R3
004030  004767  001144                  JSR     PC,OCTPE        ;PRINT MTS
004034  012704  006550                  MOV     #MSG6,R4
004040  004767  000746                  JSR     PC,TTOUT        ;PRINT MTC TAG
004044  017703  174532                  MOV     @MTC,R3
004050  004767  001124                  JSR     PC,OCTPE        ;PRINT MTC
004054  012704  006557                  MOV     #MSG7,R4
004060  004767  000726                  JSR     PC,TTOUT        ;PRINT BYTE COUNT TAG
004064  017703  174514                  MOV     @MTBC,R3
004070  004767  001114                  JSR     PC,OCTP         ;PRINT BYTE COUNT
004074  005767  174570                  TST     SPFLG           ;SEE IF PRINT BA
004100  001017                          BNE     ERPTX           ;IF NOT: BR
004102  012704  006567                  MOV     #MSG8,R4
004106  004767  000700                  JSR     PC,TTOUT        ;PRINT BUS ADDRESS TAG
004112  017703  174470                  MOV     @MTBA,R3
004116  004767  001066                  JSR     PC,OCTP         ;PRINT CURRENT ADDRESS
004122  016703  174552                  MOV     BYTES,R3
004126  005403                          NEG     R3
004130  066703  174542                  ADD     BADR,R3
004134  004767  001050                  JSR     PC,OCTP         ;PRINT EXPT ADDRESS
004140  005777  174454          ERPTX:  TST     @SWR            ;SEE IF HALT ON ERROR
004144  100001                          BPL     ERPTXX          ;IF NOT: BR
004146  000000                          HALT
004150  004767  001450          ERPTXX: JSR     PC,CKSWR        ;TEST FOR ↑G
004154  000167  000240                  JMP     SCOPE           ;GO SEE IF SCOPE ON ERROR
```

```
                                                ;DATA CHECK SUBROUTINE**********

004160  000240                        DCHK:   NOP
004162  005067  174522                        CLR     CRCNT           ;CLEAR COUNTER
004166  121112                        DCHKO:  CMPB    (R1),(R2)       ;SEE IF EXPT DATA=RCVD DATA
004170  001007                                BNE     DCHKE           ;IF NOT: BR
004172  005267  174512                DCHK1:  INC     CRCNT           ;BUMP CHARACTER COUNTER
004176  122122                                CMPB    (R1)+,(R2)+
004200  005300                                DEC     R0              ;SEE IF DONE
004202  001371                                BNE     DCHKO           ;IF NOT: BR
004204  000167  000150                        JMP     DCHKX           ;ELSE GO TO EXIT ROUTINE
004210  000240                        DCHKE:  NOP
004212  012767  000001  174472                MOV     #1,DERFL        ;SET ERROR FLAG
004220  032777  020000  174372                BIT     #20000,@SWR     ;SEE IF PRINT ERROR
004226  001054                                BNE     DCHKX           ;IF NOT: BR
004230  005767  174460                        TST     HDRFL           ;SEE IF DONE HEADER
004234  001007                                BNE     DCHKE1          ;IF SO: BR
004236  016704  174416                        MOV     EMADDR,R4
004242  004767  000544                        JSR     PC,TTOUT        ;PRINT HEADER
004246  012767  000001  174440                MOV     #1,HDRFL        ;SET HEADER FLAG
004254  012704  006657                DCHKE1: MOV     #MSG12,R4
004260  005767  174432                        TST     PFLG            ;SEE IF PRINTED DATA ERROR TAG
004264  001004                                BNE     DCHKE2          ;IF SO: BR
004266  005267  174424                        INC     PFLG
004272  004767  000514                        JSR     PC,TTOUT        ;ELSE PRINT DATA ERROR TAG
004276  012704  006673                DCHKE2: MOV     #MSG13,R4
004302  004767  000504                        JSR     PC,TTOUT        ;PRINT CHAR NUMBER TAG
004306  016703  174376                        MOV     CRCNT,R3
004312  004767  000672                        JSR     PC,OCTP         ;PRINT CHAR NUMBER
004316  012704  006701                        MOV     #MSG14,R4
004322  004767  000464                        JSR     PC,TTOUT        ;PRINT GOOD TAG
004326  111103                                MOVB    (R1),R3
004330  004767  001102                        JSR     PC,DOUT         ;PRINT GOOD CHARACTER
004334  012704  006706                        MOV     #MSG15,R4
004340  004767  000446                        JSR     PC,TTOUT        ;PRINT BAD TAG
004344  111203                                MOVB    (R2),R3
004346  004767  001064                        JSR     PC,DOUT         ;PRINT BAD CHARACTER
004352  000240                                NOP
004354  000167  177612                        JMP     DCHK1           ;CONTINUE FOR ALL BYTES
004360  000240                        DCHKX:  NOP
004362  005767  174324                        TST     DERFL           ;SEE IF ANY ERROR
004366  001404                                BEQ     DCHKXX          ;IF NOT: BR
004370  005777  174224                        TST     @SWR            ;SEE IF HALT ON ERROR
004374  100001                                BPL     DCHKXX          ;IF NOT: BR
004376  000000                                HALT
004400  004767  001220                DCHKXX: JSR     PC,CKSWR        ;TEST FOR ^G
004404  000240                                NOP
004406  005067  174304                        CLR     PFLG            ;CLEAR PRINT FLAG
004412  005067  174274                        CLR     DERFL           ;CLEAR DATA ERROR FLAG
004416  000207                                RTS     PC              ;RETURN
```

```
1008                                                ;SCOPE LOOP ON ERROR SUBROUTINE**********
1009
1010  004420  004767  001200          SCOPE:  JSR     PC,CKSWR        ;TEST FOR ↑G
1011  004424  000240                          NOP
1012  004426  032777  040000  174164          BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
1013  004434  001001                          BNE     SCOPE1          ;IF SO: BR
1014  004436  000207                          RTS     PC              ;ELSE EXIT
1015  004440  000240                  SCOPE1: NOP
1016  004442  005726                          TST     (SP)+           ;RESET STACK
1017  004444  000240                          NOP
1018  004446  017703  174212                  MOV     @LTADD,R3
1019  004452  000113                          JMP     (R3)            ;LOOP ON ERROR
1020
1021                                                ;TEST ITERATION SUBROUTINE**********
1022
1023  004454  000240                  ITER:   NOP
1024  004456  004767  001142                  JSR     PC,CKSWR        ;TEST FOR ↑G
1025  004462  032777  010000  174130          BIT     #10000,@SWR     ;SEE IF ITERATIONS
1026  004470  001403                          BEQ     ITER1           ;IF SO: BR
1027  004472  005067  174210          ITER0:  CLR     ITCNT           ;CLEAR ITERATION COUNTER
1028  004476  000207                          RTS     PC              ;ELSE EXIT
1029  004500  005267  174202          ITER1:  INC     ITCNT           ;BUMP COUNTER
1030  004504  026767  174176  174122          CMP     ITCNT,ITAMT     ;SEE IF DONE ALL
1031  004512  001767                          BEQ     ITER0           ;IF SO: BR
1032  004514  005726                          TST     (SP)+           ;RESET STACK
1033  004516  017700  174144                  MOV     @ITRLP,R0       ;SET ITERATION POINTER
1034  004522  000110                          JMP     (R0)            ;GO ITERATE
1035
1036                                                ;MAG TAPE INTERRUPT HANDLER**********
1037
1038  004524  000240                  MTINT:  NOP
1039  004526  022626                          CMP     (SP)+,(SP)+     ;RESET STACK POINTER
1040  004530  042777  000100  174044          BIC     #100,@MTC       ;CLEAR INTERRUPT ENABLE
1041  004536  000240                          NOP
1042  004540  000240                          NOP
1043  004542  000207                          RTS     PC              ;RETURN TO CALLER
1044
1045                                                ;TTY INTERRUPT HANDLER**********
1046
1047  004544  000240                  TTINT:  NOP
1048  004546  000240                          NOP
1049  004550  000240                          NOP
1050  004552  000002                          RTI
1051
```

```
1052                                    ;*************************************************************
1053                                    ;TTY ENTRY SUBROUTINE:
1054                                    ;
1055                                    ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
1056                                    ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
1057                                    ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
1058                                    ;LIMITS.  ALL RESPONSE MUST BE TYPED IN OCTAL
1059                                    ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
1060                                    ;THE CALLING ROUTINE.
1061                                    ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
1062                                    ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
1063                                    ;MAY BE REENTERED.
1064                                    ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
1065                                    ;MAY BE TERMINATED AT LESS THAN SIZ BY TYPING A
1066                                    ;CARRIAGE RETURN
1067                                    ;*************************************************************
1068
1069  004554  005067  174072       TTR:    CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
1070  004560  005000                        CLR     R0
1071  004562  004767  000152       TTR0:   JSR     PC,TTIN         ;GO READ CHARACTER
1072  004566  122767  000215  174054       CMPB    #215,TIB        ;SEE IF CR
1073  004574  001005                        BNE     TTR1            ;IF NOT: BR
1074  004576  005767  174050               TST     TEMP1           ;SEE IF FIRST CHARACTER
1075  004602  001446                        BEQ     TTR5            ;IF SO: BR
1076  004604  000167  000066               JMP     TTR2            ;ELSE GO LOAD VALUE
1077  004610  122767  000260  174032 TTR1:  CMPB    #260,TIB        ;SEE IF CHAR IS LESS THAN 0
1078  004616  101402                        BLOS    TTR1A           ;IF NOT: BR
1079  004620  000167  000076               JMP     TINER           ;ELSE GO TO ERROR
1080  004624  122767  000270  174016 TTR1A: CMPB    #270,TIB        ;SEE IF CHAR IS GREATER THAN 7
1081  004632  101002                        BHI     TTR1B           ;IF NOT: BR
1082  004634  000167  000062               JMP     TINER           ;ELSE GO TO ERROR
1083  004640  005267  174006       TTR1B:  INC     TEMP1           ;SET FIRST CHARACTER FLAG
1084  004644  000241                        CLC
1085  004646  006100                        ROL     R0
1086  004650  000241                        CLC
1087  004652  006100                        ROL     R0              ;SHIFT 3 LEFT
1088  004654  000241                        CLC
1089  004656  006100                        ROL     R0
1090  004660  042767  177770  173762       BIC     #177770,TIB     ;STRIP ASCII
1091  004666  056700  173756               BIS     TIB,R0          ;LOAD CHARACTER
1092  004672  005301                        DEC     R1              ;SEE IF DONE
1093  004674  001332                        BNE     TTR0            ;IF NOT: BR
1094  004676  020002               TTR2:   CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
1095  004700  101402                        BLOS    TTR3            ;IF OT: BR
1096  004702  000167  000014               JMP     TINER           ;ELSE GO TO ERROR
1097  004706  020300               TTR3:   CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
1098  004710  101402                        BLOS    TTR4            ;IF NOT: BR
1099  004712  000167  000004               JMP     TINER           ;ELSE GO TO ERROR
1100  004716  010015               TTR4:   MOV     R0,(R5)         ;LOAD VALUE
1101  004720  000207               TTR5:   RTS     PC              ;EXIT
1102
```

```
                                                ;TTY ENTRY ERROR SUBROUTINE**********
                                                
004722  012704  006713          TINER:  MOV     #MSG16,R4
004726  004767  000060                  JSR     PC,TTOUT        ;PRINT?
004732  162716  000020                  SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
004736  000207                          RTS     PC              ;REDO VALUE ENTRY
                                                
                                                ;TTY READ SUBROUTINE**********
                                                
004740  005077  173660          TTIN:   CLR     @TKS
004744  005077  173656                  CLR     @TKB
004750  005067  173674                  CLR     TIB
004754  005277  173644                  INC     @TKS
004760  105777  173640          TTIN1:  TSTB    @TKS
004764  100375                          BPL     TTIN1
004766  017767  173634  173654          MOV     @TKB,TIB
004774  105777  173630          TTIN2:  TSTB    @TPS
005000  100375                          BPL     TTIN2
005002  116777  173642  173622          MOVB    TIB,@TPB
005010  000207                          RTS     PC
                                                
                                                ;TTY OUTPUT SUBROUTINE**********
                                                
005012  112467  173630          TTOUT:  MOVB    (R4)+,TOB
005016  122767  000043  173622          CMPB    #43,TOB
005024  001452                          BEQ     TEX
005026  122767  000045  173612          CMPB    #45,TOB
005034  001407                          BEQ     TCRLF
005036  122767  000041  173602          CMPB    #41,TOB
005044  001443                          BEQ     TBELL
005046  004767  000064                  JSR     PC,TOG
005052  000757                          BR      TTOUT
005054  112767  000015  173564  TCRLF:  MOVB    #15,TOB
005062  004767  000050                  JSR     PC,TOG
005066  012703  000004                  MOV     #4,R3
005072  005067  173550          TCRLFA: CLR     TOB
005076  004767  000034                  JSR     PC,TOG
005102  005303                          DEC     R3
005104  001372                          BNE     TCRLFA          ;DO FILLERS
005106  112767  000012  173532          MOVB    #12,TOB
005114  004767  000016                  JSR     PC,TOG
005120  105767  173604                  TSTB    RDSW
005124  100401                          BMI     1$
005126  000731                          BR      TTOUT
005130  005067  173574          1$:     CLR     RDSW
005134  000406                          BR      TEX
005136  105777  173466          TOG:    TSTB    @TPS
005142  100375                          BPL     TOG
005144  116777  173476  173460          MOVB    TOB,@TPB
005152  000207                  TEX:    RTS     PC
005154  012703  000002          TBELL:  MOV     #2,R3
005160  012767  000007  173460  TBELA:  MOV     #7,TOB
005166  004767  177744                  JSR     PC,TOG
005172  005303                          DEC     R3
005174  001371                          BNE     TBELA
005176  000705                          BR      TTOUT
```

```
1159                                            ;OCTAL OUTPUT SUBROUTINE**********
1160
1161  005200  012767  000001  000226  OCTPE:   MOV     #1,OFL
1162  005206  000402                           BR      OCTPE1
1163  005210  005067  000220          OCTP:    CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
1164  005214  010304                  OCTPE1:  MOV     R3,R4           ;SEE IF NUMBER IS ZERO
1165  005216  001007                           BNE     OCTP0           ;IF NOT ZERO: BR
1166  005220  005767  000210                   TST     OFL             ;SEE IF PRINT ALL 0
1167  005224  001004                           BNE     OCTP0           ;IF SO: BR
1168  005226  004767  000162                   JSR     PC,OCTPG1       ;ELSE PRINT ZERO
1169  005232  000167  000120                   JMP     OCTP3           ;SPACE AND EXIT
1170  005236  032704  100000          OCTP0:   BIT     #100000,R4      ;SEE IF MSD = 1
1171  005242  001406                           BEQ     OCTP1           ;IF NOT: BR
1172  005244  012704  000001                   MOV     #1,R4
1173  005250  004767  000116                   JSR     PC,OCTPG        ;PRINT 1
1174  005254  000167  000006                   JMP     OCTP2
1175  005260  005004                  OCTP1:   CLR     R4
1176  005262  004767  000104                   JSR     PC,OCTPG        ;PRINT 0
1177  005266  010304                  OCTP2:   MOV     R3,R4
1178  005270  006004                           ROR     R4
1179  005272  006004                           ROR     R4
1180  005274  006004                           ROR     R4              ;POSITION DIGIT
1181  005276  006004                           ROR     R4
1182  005300  000304                           SWAB    R4
1183  005302  004767  000064                   JSR     PC,OCTPG        ;PRINT DIGIT 2
1184  005306  010304                           MOV     R3,R4
1185  005310  006004                           ROR     R4
1186  005312  000304                           SWAB    R4
1187  005314  004767  000052                   JSR     PC,OCTPG        ;PRINT DIGIT 3
1188  005320  010304                           MOV     R3,R4
1189  005322  006104                           ROL     R4
1190  005324  006104                           ROL     R4
1191  005326  000304                           SWAB    R4
1192  005330  004767  000036                   JSR     PC,OCTPG        ;PRINT DIGIT 4
1193  005334  010304                           MOV     R3,R4
1194  005336  006004                           ROR     R4
1195  005340  006004                           ROR     R4
1196  005342  006004                           ROR     R4
1197  005344  004767  000022                   JSR     PC,OCTPG
1198  005350  010304                           MOV     R3,R4
1199  005352  004767  000014                   JSR     PC,OCTPG        ;PRINT DIGIT 5
1200  005356  012767  000240  173262  OCTP3:   MOV     #240,TOB
1201  005364  004767  177546                   JSR     PC,TOG          ;PRINT SPACE
1202  005370  000207                           RTS     PC              ;EXIT
```

```
1203
1204                                        ;OCTAL PRINT SUBROUTINE**********
1205
1206  005372  042704  177770        OCTPG:  BIC     #177770,R4
1207  005376  001004                        BNE     OCTPGO
1208  005400  005767  000030                TST     OFL
1209  005404  001001                        BNE     OCTPGO
1210  005406  000207                        RTS     PC
1211  005410  005267  000020        OCTPGO: INC     OFL
1212  005414  052704  000260        OCTPG1: BIS     #260,R4
1213  005420  010467  173222                MOV     R4,TOB
1214  005424  004767  177506                JSR     PC,TOG
1215  005430  010304                        MOV     R3,R4
1216  005432  000207                        RTS     PC
1217  005434  000000        OFL:    0                       ;FIRST CHAR FLAG
1218
1219                                        ;DATA CHARACTER OUTPUT SUBROUTINE**********
1220
1221  005436  005067  173204        DOUT:   CLR     TOB
1222  005442  012704  000010                MOV     #10,R4          ;SET NUMBER TO PRINT
1223  005446  110367  173174                MOVB    R3,TOB
1224  005452  105777  173152        DOUT1:  TSTB    @TPS
1225  005456  100375                        BPL     DOUT1
1226  005460  132767  000200  173160        BITB    #200,TOB
1227  005466  001404                        BEQ     DOUT2
1228  005470  012777  000061  173134        MOV     #061,@TPB
1229  005476  000403                        BR      DOUT3
1230  005500  012777  000060  173124  DOUT2: MOV     #060,@TPB
1231  005506  006167  173134        DOUT3:  ROL     TOB
1232  005512  005304                        DEC     R4
1233  005514  001356                        BNE     DOUT1
1234  005516  000207                        RTS     PC
1235  005520  016703  173132        DOUTD:  MOV     TEMP3,R3
1236  005524  000303                        SWAB    R3
1237  005526  004767  177704                JSR     PC,DOUT
1238  005532  016703  173120                MOV     TEMP3,R3
1239  005536  004767  177674                JSR     PC,DOUT
1240  005542  000207                        RTS     PC
1241
1242
1243
1244
1245  005544  013746  000006        SUSWR:  MOV     @#6,-(SP)       ;SAVE VECTORS
1246  005550  013746  000004                MOV     @#4,-(SP)
1247  005554  012737  005574  000004        MOV     #1$,@#4         ;SET UP FOR TIMEOUT
1248  005562  022777  177777  173030        CMP     #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
1249  005570  001402                        BEQ     2$
1250  005572  000407                        BR      3$
1251  005574  022626                1$:     CMP     (SP)+,(SP)+     ;ADJUST STACK
1252  005576  012767  000176  173014  2$:   MOV     #SWREG,SWR      ;POINT TO SOFTWARE SWITCH REG
1253  005604  012767  000174  173010        MOV     #DISPREG,DISPLAY        ;POINT TO SOFT DISPLAY REG
1254  005612  012637  000004        3$:     MOV     (SP)+,@#4       ;RESTORE VECTORS
1255  005616  012637  000006                MOV     (SP)+,@#6
1256  005622  000207                        RTS     PC
1257
1258
```

```
1259  005624  022767  000176  172766  CKSWR:  CMP     #SWREG,SWR              ;SOFTWARE SWITCH REG PRESENT
1260  005632  001041                          BNE     OUT                     ;NO, GET OUT
1261  005634  105777  172764                  TSTB    @TKS                    ;YES,WAIT FOR
1262  005640  100036                          BPL     OUT                     ;READY, GET CHARACTER
1263  005642  017767  172760  173000          MOV     @TKB,TIB                ;AND STRIP OFF
1264  005650  042767  177600  172772          BIC     #177600,TIB             ;THE GARBAGE
1265  005656  022767  000007  172764          CMP     #7,TIB          ;IS IT A ^G>
1266  005664  001024                          BNE     OUT
1267  005666  012704  007245                  MOV     #$CNTG,R4
1268  005672  004767  177114                  JSR     PC,TTOUT
1269  005676  012704  007251          CNTLU:  MOV     #$MSWR,R4
1270  005702  004767  177104                  JSR     PC,TTOUT
1271  005706  017703  172706                  MOV     @SWR,R3
1272  005712  004767  177262                  JSR     PC,OCTPE
1273  005716  012704  007263                  MOV     #$MNEW,R4
1274  005722  004767  177064                  JSR     PC,TTOUT
1275  005726  005037  000726                  CLR     @#TEMPST
1276  005732  004767  000002                  JSR     PC,$READ                ;GO READ A LINE
1277  005736  000207                  OUT:    RTS     PC                      ;RETURN TO MAIN BODY OF PROGRAM
1278
1279  005740  005067  172762          $READ:  CLR     TEMPST
1280  005744  012767  000007  172752          MOV     #7,COUNT
1281  005752  004767  176762          1$:     JSR     PC,TTIN         ;GO  READ A CHARACTER
1282  005756  042767  177600  172664          BIC     #177600,TIB     ;STRIP OFF GARBAGE
1283  005764  122767  000025  172656          CMPB    #25,TIB         ;IS IT A ^U?
1284  005772  001002                          BNE     11$             ;BRANCH IF NOT
1285  005774  005726                  3$:     TST     (SP)+           ;POP THE STACK
1286  005776  000737                          BR CNTLU                ;START OVER
1287  006000  122767  000012  172642  11$:    CMPB    #12,TIB         ;IS IT A <LF>?
1288  006006  001016                          BNE 2$                  ;BRANCH IF NOT
1289  006010  005767  172032                  TST 46                  ;TEST FOR ACT-11 HOOKS
1290  006014  001406                          BEQ     10$             ;NO HOOKS-<LF> WAS MISTAKE
1291  006016  005726                          TST     (SP)+           ;POP THE STACK
1292  006020  016716  172022                  MOV     46,(SP)         ;MAKE ACT-11 ADDRESS
1293  006024  062716  000010                  ADD     #10,(SP)
1294  006030  000742                          BR      OUT             ;GO TO THAT  ADDRESS
1295  006032  012704  007273          10$:    MOV     #$NOACT,R4      ;LET OPR KNOW IT
1296  006036  004767  176750                  JSR     PC,TTOUT        ;AND RETURN
1297  006042  000754                          BR      3$              ;FOR ANOTHER TRY
1298  006044  122767  000015  172576  2$:     CMPB    #15,TIB         ;IS IT A <CR>?
1299  006052  001013                          BNE     4$              ;BRANCH IF NOT
1300  006054  012767  000200  172646          MOV     #200,RDSW
1301  006062  004767  176766                  JSR     PC,TCRLF        ;ECHO IT WITH <LF>
1302  006066  022767  000007  172630          CMP     #7,COUNT        ;WAS IT FIRST CHARACTER
1303  006074  001037                          BNE     7$              ;CHANGE SWR IF NOT FIRST ONE
1304  006076  005726                  8$:     TST     (SP)+           ;POP THE STACK
1305  006100  000716                          BR      OUT             ;GET OUT
1306  006102  122767  000060  172540  4$:     CMPB    #60,TIB
1307  006110  003004                          BGT     5$
1308  006112  122767  000067  172530          CMPB    #67,TIB
1309  006120  003005                          BGT     6$
1310  006122  012704  007345          5$:     MOV     #$QUEST,R4
1311  006126  004767  176660                  JSR     PC,TTOUT
1312  006132  000720                          BR      3$              ;START OVER IF NOT LEGAL CHARACTER
1313  006134  006367  172566          6$:     ASL     TEMPST
1314  006140  006367  172562                  ASL     TEMPST
```

```
1315  006144  006367  172556                    ASL     TEMPST
1316  006150  142767  000060  172472            BICB    #60,TIB         ;GET NITTY-GRITTY
1317  006156  156767  172466  172542            BISB    TIB,TEMPST
1318  006164  005367  172534                    DEC     COUNT           ;ONLY WANT 6   DIGITS
1319  006170  001754                            BEQ     5$
1320  006172  000667                            BR      1$
1321  006174  016777  172526  172416  7$:       MOV     TEMPST,@SWR     ;CHANGE SWITCH REGISTER CONTENTS
1322  006202  000735                            BR      8$
1323
```

```
1324
1325                                    ;       ************************************************************************
1326                                    ;                               MODIFIED JAN 24 1978
1327
1328                                    ;       ++
1329                                    ;               CHECK FOR DUMP MODE OR AUTOMATIC/ACT11-XXDP MODE
1330                                    ;       --
1331
1332  006204  005067  172570           CKMODE: CLR     AUTOM                   ;INIT AUTOMATIC MODE INDICATOR
1333  006210  105067  172566                   CLRB    ACT11M                  ;INIT   ACT11 AUTO MODE INDICATOR
1334  006214  105067  172563                   CLRB    XXDPM                   ;INIT XXDP AUTO MODE INDICATOR
1335  006220  105067  172560                   CLRB    ADUMPM                  ;INIT ACT11 DUMP MODE INDICATOR
1336  006224  105067  172555                   CLRB    XDUMPM                  ;INIT XXDP DUMP MODE INDICATOR
1337  006230  005737  000042                   TST     @#42                    ;AUTO MODE?
1338  006234  001425                           BEQ     2$                      ;BRANCH - IF NO
1339  006236  005267  172536                   INC     AUTOM                   ;SET AUTO MODE INDICATOR
1340  006242  032737  020000  000052           BIT     #20000,@#52             ;MANUAL INTERVENTION?
1341  006250  001402                           BEQ     6$                      ;BRANCH - IF NO
1342  006252  000167  000054                   JMP     ABORT                   ;ABORT THE PROGRAM
1343  006256  023737  000042  000046  6$:      CMP     @#42,@#46               ;ACT11 MODE?
1344  006264  001403                           BEQ     1$                      ;BRANCH - IF YES
1345  006266  105267  172511                   INCB    XXDPM                   ;INDICATE XXDP AUTO MODE
1346  006272  000416                           BR      5$                      ;AND EXIT
1347  006274  105267  172502           1$:     INCB    ACT11M                  ;INDICATE ACT11 AUTO MODE
1348  006300  012777  177777  172312           MOV     #177777,@SWR            ;SET SWITCH REGISTER
1349  006306  000410                           BR      5$                      ;AND EXIT
1350  006310  105737  000041           2$:     TSTB    @#41                    ;MAN/MODE VIA ACT11/PAPER TAPE?
1351  006314  001003                           BNE     3$                      ;BRANCH - IF NOT
1352  006316  105267  172462                   INCB    ADUMPM                  ;INDICATE MAN/MODE VIA ACT11/PAPER TAPE
1353  006322  000402                           BR      5$                      ;AND EXIT
1354  006324  105267  172455           3$:     INCB    XDUMPM                  ;INDICATE MANUAL MODE VIA XXDP
1355  006330  000207                   5$:     RTS     PC                      ;RETURN
1356
1357                                    ;       ************************************************************************
1358
```

```
1359
1360                                    ;       ********************************************************************
1361                                    ;                       MODIFIED JAN 24 1978
1362                                    ;
1363                                    ;       ++
1364                                    ;               CHECK FOR DUMP MODE OR AUTOMATIC/ACT11-XXDP MODE
1365                                    ;       --
1366
1367  006332  000005                    ABORT:  RESET                           ;CLEAR THE WORLD
1368  006334  012704  006366                   MOV     #MSGO,R4               ;GET THE  MESSAGE
1369  006340  004767  176446                   JSR     PC,TTOUT               ;PRINT ABORT MESSAGE
1370  006344  105767  172433                   TSTB    XXDPM                  ;XXDP AUTO MODE?
1371  006350  001405                           BEQ     1$                     ;BRANCH - IF NO
1372  006352  013700  000042                   MOV     @#42,R0                ;GET MONITOR EXIT ADDRESS
1373  006356  005037  000042                   CLR     @#42                   ;USE AS ABORT FLAG
1374  006362  004700                           JSR     PC,R0                  ;EXIT TO XXDP MONITOR
1375  006364  000777                1$:        BR      .                      ;AND HANG
1376
1377                                    ;       ********************************************************************
1378
```

```
1379
```

```
1380                                                    ;MESSAGE TABLE**********
1381
1382  006366  022445  051120  043517  MSG0:   .ASCII  /%%PROGRAM ABORTED#/
1383  006374  040522  020115  041101
1384  006402  051117  042524  021504
1385  006410  022445  055103  051524  MSG1:   .ASCII  /%%CZTSFD0 TS03 SPLMTL INSTR#/
1386  006416  042106  020060  051524
1387  006424  031460  051440  046120
1388  006432  052115  020114  047111
1389  006440  052123  021522
1390  006444  042445  052116  051105  MSG2:   .ASCII  /%ENTER UNIT NUMBER:#/
1391  006452  052440  044516  020124
1392  006460  052516  041115  051105
1393  006466  021472
1394  006470  022445  042441  042116  MSG3:   .ASCII  /%%!END OF PASS: #/
1395  006476  047440  020106  040520
1396  006504  051523  020072     043
1397  006511     045  020441  042522  MSG4:   .ASCII  /%!!REWIND ERROR: NO BOT#/
1398  006516  044527  042116  042440
1399  006524  051122  051117  020072
1400  006532  047516  041040  052117
1401  006540     043
1402  006541     045  052115  035123  MSG5:   .ASCII  /%MTS: #/
1403  006546  021440
1404  006550  046445  041524  020072  MSG6:   .ASCII  /%MTC: #/
1405  006556     043
1406  006557     045  052115  041502  MSG7:   .ASCII  /%MTBC: #/
1407  006564  020072     043
1408  006567     045  052115  040503  MSG8:   .ASCII  /%MTCA: #/
1409  006574  020072     043
1410  006577     045  020441  040502  MSG9:   .ASCII  /%!!BACKSPACE ERROR#/
1411  006604  045503  050123  041501
1412  006612  020105  051105  047522
1413  006620  021522
1414  006622  020445  047041  052117  MSG10:  .ASCII  /%!!NOT READY#/
1415  006630  051040  040505  054504
1416  006636     043
1417  006637     045  020441  047516  MSG11:  .ASCII  /%!!NO INTERRUPT#/
1418  006644  044440  052116  051105
1419  006652  052522  052120     043
1420  006657     045  040504  040524  MSG12:  .ASCII  /%DATA ERROR#/
1421  006664  042440  051122  051117
1422  006672     043
1423  006673     045  047103  020072  MSG13:  .ASCII  /%CN: #/
1424  006700     043
1425  006701     045  035107  021440  MSG14:  .ASCII  /%G: #/
1426  006706  041045  020072     043  MSG15:  .ASCII  /%B: #/
1427  006713     077     043         MSG16:  .ASCII  /?#/
1428  006715     045  051127  052111  MSG17:  .ASCII  /%WRITE ERROR#/
1429  006722  020105  051105  047522
1430  006730  021522
1431  006732  051045  040505  020104  MSG18:  .ASCII  /%READ ERROR#/
1432  006740  051105  047522  021522
1433  006746  047045  020117  050117  MSG19:  .ASCII  /%NO OPI IN 10.5 FEET#/
1434  006754  020111  047111  030440
1435  006762  027060  020065  042506
```

```
1436  006770  052105     043
1437  006773     045  050117  020111  MSG20:  .ASCII  /%OPI WITHIN 4.0 FEET#/
1438  007000  044527  044124  047111
1439  007006  032040  030056  043040
1440  007014  042505  021524
1441  007020  020440  047041  052117  MSG21:  .ASCII  / !!NOT AVAILABLE#/
1442  007026  040440  040526  046111
1443  007034  041101  042514     043
1444  007041     045  042522  044507  MSG22:  .ASCII  /%REGISTER START: #/
1445  007046  052123  051105  051440
1446  007054  040524  052122  020072
1447  007062     043
1448  007063     045  042526  052103  MSG23:  .ASCII  /%VECTOR: #/
1449  007070  051117  020072     043
1450
1451                                        ;TEST HEADER**********
1452
1453  007075     045  052045  051505  LT1MSG: .ASCII  /%%TEST 1: WRITE FROM ODD BYTE#/
1454  007102  020124  035061  053440
1455  007110  044522  042524  043040
1456  007116  047522  020115  042117
1457  007124  020104  054502  042524
1458  007132     043
1459  007133     045  052045  051505  LT2MSG: .ASCII  /%%TEST 2: READ TO ODD BYTE#/
1460  007140  020124  035062  051040
1461  007146  040505  020104  047524
1462  007154  047440  042104  041040
1463  007162  052131  021505
1464  007166  022445  042524  052123  LT3MSG: .ASCII  /%%TEST 3: OPI TOO LONG#/
1465  007174  031440  020072  050117
1466  007202  020111  047524  020117
1467  007210  047514  043516     043
1468  007215     045  052045  051505  LT4MSG: .ASCII  /%%TEST 4: OPI TOO SHORT#/
1469  007222  020124  035064  047440
1470  007230  044520  052040  047517
1471  007236  051440  047510  052122
1472  007244     043
1473
1474  007245     045  047536     043  $CNTG:  .ASCII  /%↑G#/
1475  007251     045  020445  051441  $MSWR:  .ASCII  /%%!!SWR= #/
1476  007256  051127  020075     043
1477  007263     040  047040  053505  $MNEW:  .ASCII  /  NEW= #/
1478  007270  020075     043
1479  007273     045  047045  020117  $NOACT: .ASCII  /%%NO ACT-11 HOOKS <LF> =INVALID --RETRY%%#/
1480  007300  041501  026524  030461
1481  007306  044040  047517  051513
1482  007314  036040  043114  020076
1483  007322  044475  053116  046101
1484  007330  042111  026440  051055
1485  007336  052105  054522  022445
1486  007344     043
1487  007345     045  022477  021445  $QUEST: .ASCII  /%?%%#/
1488                                        .EVEN
1489
1490  007352  177777                  WDATA:  -1
1491          007454                          .=.+100
```

```
1492  007454  000000                    RDATA:  0
1493          000001                            .END
```

ABORT	006332	1342	1367#										
ACT11M	001002	362#	476	526	545	560	603	622	1333*	1347*			
ADUMPM	001004	364#	1335*	1352*									
AUTOM	001000	361#	1332*	1339*									
BADR	000676	436#	650*	664*	695*	709*	738*	750*	783*	795*	876	914	951
BCNT	000722	446#											
BKSP	003356	657	702	744	789	843#							
BKSP1	003402	847#	848										
BYTES	000700	437#	651*	665*	696*	710*	739*	751*	784*	796*	877	912	949
CCNT	000614	408#											
CKMODE	006204	475	1332#										
CKSWR	005624	585	624	837	873	900	956	1003	1010	1024	1259#		
CNTLU	005676	540	1269#	1286									
COUNT	000724	447#	1280*	1302	1318*								
CRCNT	000710	441#	961*	964*	985								
DCHK	004160	673	718	960#									
DCHKE	004210	963	969#										
DCHKE1	004254	974	978#										
DCHKE2	004276	980	983#										
DCHKX	004360	968	972	997#									
DCHKXX	004400	999	1001	1003#									
DCHK0	004166	962#	967										
DCHK1	004172	964#	996										
DERFL	000712	442#	970*	998	1006*								
DISPLA	000622	411#	1253*										
DISPRE	000174	380#	1253										
DOUT	005436	990	994	1221#	1237	1239							
DOUTD	005520	1235#											
DOUT1	005452	1224#	1225	1233									
DOUT2	005500	1227	1230#										
DOUT3	005506	1229	1231#										
DRIVE	000040	331#											
EMADDR	000660	429#	640*	686*	734*	779*	822	866	893	926	975		
ERCHK	003672	655	669	700	714	742	787	800	851	906#			
ERCHK1	003704	907	909#										
ERCHK2	003716	910	912#										
ERCHK3	003740	916	918#										
ERPT	003750	760	908	911	919	921#							
ERPTX	004140	923	944	953#									
ERPTXX	004150	954	956#										
ERPT1	004004	925	929#										
ERRAD	000662	430#	652*	666*	699*	713*	741*	753*	785*	798*	850*	929	
EXEC	003442	653	667	697	711	740	754	786	799	858#			
EXECX	003652	890	897#										
EXECXX	003662	898	900#										
EXEC0	003446	860#	863										
EXEC1	003500	865	868#										
EXEC2	003520	861	871	873#									
EXEC3	003602	885#	886	888									
EXEC4	003642	892	895#										
FUN	000704	439#	649*	663*	694*	708*	737*	752*	782*	797*	879		
HDRFL	000714	443#	579*	864	891	924	928*	973	977*				
HERE	002016	612	618#										
ITAMT	000634	416#	762*	764*	803*	805*	1030						
ITCNT	000706	440#	1027*	1029*	1030								
ITER	004454	674	719	763	804	1023#							

```
ITERO   004472  1027*  1031
ITERI   004500  1026   1029*
ITRLP   000666   432*   574*   575*   600*   601*  1033
LTADD   000664   431*   572*   573*   574    576    580    598*   599*   600   1018
LT1     002064   455    456    639*
LT1B    002106   644*   647
LT1C    002160   654*
LT1D    002210   660*   662
LT1E    002254   668*
LT1MSG  007075   640   1453*
LT2     002312   457    458    685*
LT2B    002330   689*   692
LT2C    002374   698*
LT2D    002432   705*   707
LT2E    002470   712*
LT2MSG  007133   686   1459*
LT3     002534   459    733*
LT3A    002554   737*   746
LT3B    002602   741*
LT3C    002702   755*
LT3IT   002632   460    747*
LT3MSG  007165   734   1464*
LT3X    002740   758    762*
LT4     002764   461    778*
LT4A    003004   782*   791
LT4B    003040   787*
LT4C    003132   800*
LT4IT   003062   462    792*
LT4MSG  007215   779   1468*
MEDIUM  000041   335*
MSG0    006366  1368   1382*
MSG1    006410   483   1385*
MSG10   006622   868   1414*
MSG11   006637   895   1417*
MSG12   006657   978   1420*
MSG13   006673   983   1423*
MSG14   006701   987   1425*
MSG15   006706   991   1426*
MSG16   006713  1105   1427*
MSG17   006715   652    699    741    785   1428*
MSG18   006732   666    713   1431*
MSG19   006746   753   1433*
MSG2    006444   514   1390*
MSG20   006773   798   1437*
MSG21   007020   549   1441*
MSG22   007041   485   1444*
MSG23   007063   494   1448*
MSG3    006470   607   1394*
MSG4    006511   824   1397*
MSG5    006541   826    931   1402*
MSG6    006550   830    935   1404*
MSG7    006557   939   1406*
MSG8    006567   945   1408*
MSG9    006577   850   1410*
MTBA    000606   405*   876*   918    947
MTBC    000604   404*   845*   877*   909    941
```

MTC	000602	403#	513*	536*	554*	810*	813*	814	832	844*	846*	860	875*	880*
		906	937	1040*										
MTINT	004524	396	504	1038#										
MTS	000600	402#	507	541	757	811	816	818	828	847	933			
NOACT	001354	527	533#											
OCTP	005210	488	497	520	610	942	948	952	986	1163#				
OCTPE	005200	829	833	934	938	1161#	1272							
OCTPE1	005214	1162	1164#											
OCTPG	005372	1173	1176	1183	1187	1192	1197	1199	1206#					
OCTPGO	005410	1207	1209	1211#										
OCTPG1	005414	1168	1212#											
OCTPO	005236	1165	1167	1170#										
OCTP1	005260	1171	1175#											
OCTP2	005266	1174	1177#											
OCTP3	005356	1169	1200#											
OFL	005434	1161*	1163*	1166	1208	1211*	1217#							
OUT	005736	1260	1262	1266	1277#	1294	1305							
PCNTR	000674	435#	478*	562*	609	625*	627*							
PFLG	000716	444#	979	981*	1005*									
PSW	000616	409#	472*	882*										
RCNT	000612	407#												
RDATA	007454	658	664	671	703	709	716	750	795	1492#				
RDSW	000730	449#	1144	1147*	1300*									
REGS	000640	418#	487	489	508									
RWND	003172	648	693	736	748	780	793	809#						
RWNDX	003336	821	834#											
RWNDXX	003346	812	819	835	837#									
RWND1	003226	816#	817											
SCNT	000702	438#	656*	701*	743*	788*	845							
SCOPE	004420	957	1010#											
SCOPE1	004440	1013	1015#											
SPFLG	000670	433#	759*	761*	849*	852*	915	943						
ST	001332	529#	548	606										
STALL	000636	417#	749*	756*	794*	802*	883							
START	001000	387	391	471#										
STFLG	000672	434#	567*	586	592*									
STSCD	001666	571	592#											
STO	001242	514#	551											
STOA	001222	509#	512											
ST1	001420	539	541#	544										
ST3	001462	542	552#											
ST4	001470	482	553#											
ST5	001510	557#	559											
SUSWR	005544	474	1245#											
SWR	000620	410#	538	568	582	588	619	820	834	870	889	897	922	953
		971	1000	1012	1025	1248	1252*	1259	1271	1321*	1348*			
SWREG	000176	381#	538	1252	1259									
TBELA	005160	1154#	1157											
TBELL	005154	1132	1153#											
TCRLF	005054	1130	1135#	1301										
TCRLFA	005072	1138#	1141											
TEMPST	000726	448#	1275*	1279*	1313*	1314*	1315*	1317*	1321					
TEMP1	000652	426#	883*	887*	1069*	1074	1083*							
TEMP2	000654	427#												
TEMP3	000656	428#	521	532*	533*	535	1235	1238						
TEND	001740	578	594	603#										

```
TENDX   002032   620    622#
TEX     005152  1128   1148   1152#
TIB     000650   425#   1072   1077   1080   1090*  1091   1114*  1118*  1121   1263*  1264*  1265   1282*
                1283   1287   1298   1306   1308   1316*  1317
TINER   004722  1079   1082   1096   1099   1105#
TINF    000644   423#
TKB     000626   413#   1113*  1118   1263
TKS     000624   412#   1112*  1115*  1116   1261
TOB     000646   424#    555   1126*  1127   1129   1131   1135*  1138*  1142*  1151   1154*  1200*  1213*
                1221*  1223*  1226   1231*
TOG     005136  1133   1136   1139   1143   1149#  1150   1155   1201   1214
TPB     000632   415#   1121*  1151*  1228*  1230*
TPS     000630   414#   1119   1149   1224
TSCD    001530   561    566#   591    626
TSCDO   001560   573#   587
TSCD1   001614   577    579#   602
TSCD2   001626   582#   675    720    765    806
TSCD3   001640   583    585#
TSTTBL  000732   453#   572    598
TTIN    004740  1071   1112#  1281
TTINT   004544   373   1047#
TTIN1   004760  1116#  1117
TTIN2   004774  1119#  1120
TTOUT   005012   484    486    495    515    550    608    823    825    827    831    867    869    894
                 896    927    930    932    936    940    946    976    982    984    988    992   1106
                1126#  1134   1146   1158   1268   1270   1274   1296   1311   1369
TTR     004554   493    502    525   1069#
TTRO    004562  1071#  1093
TTR1    004610  1073   1077#
TTR1A   004624  1078   1080#
TTR1B   004640  1081   1083#
TTR2    004676  1076   1094#
TTR3    004706  1095   1097#
TTR4    004716  1098   1100#
TTR5    004720  1075   1101#
T1AD    000736   455#
T1IAD   000740   456#
T2AD    000742   457#
T2IAD   000744   458#
T3AD    000746   459#
T3IAD   000750   460#
T4AD    000752   461#
T4IAD   000754   462#
UDES    000610   406#   516*   517    528*   529    532    534*   535*   536    547*   552*   605*   810
                 844    875
UNP     000720   445#
VECT    000642   419#   496    498    503
WDATA   007352   641    650    670    687    695    715    738    783   1490#
XDUMPM  001005   365#  1336*  1354*
XXDPM   001003   363#  1334*  1345*  1370
$ACT    001320   479    526#   628
$CNTG   007245  1267   1474#
$DONE   001756   531    604    607#
$ENDAD  002006   343    614#
$MNEW   007263  1273   1477#
$MSWR   007251  1269   1475#
```

F04


```
$NOACT  007273            1295    1479#
$QUEST  007345            1310    1487#
$READ   005740            1276    1279#
$SVPC = 001000             328#    349
.     = 007456             319#    328     330#    334#    338#    342#    345#    349#    372#    379#    385#    389#    395#
                           399#    465#   1375    1491#

. ABS.  007456     000

 ERRORS DETECTED:  0

 CZTSFD,CZTSFD.SEQ/CRF/SOL/NL:TOC=CZTSFD.P11
 RUN-TIME:  1  3  .5 SECONDS
 RUN-TIME RATIO: 96/6=15.3
 CORE USED:  7K  (13 PAGES)

 DOCUMENT PAGES:  44
```