

TMA-11

INSTRUCTION TEST
MD-11-DZTMF-D

EP-DZTMF-D-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN USA

This microfiche card contains a grid of frames on the left side, each containing technical data. The data is organized into columns and rows, with some frames containing text and others containing numerical data or diagrams. The frames are arranged in a regular grid pattern, typical of microfiche storage.

B01

TMA,B-11 SUPPLEMENTAL INST TEST MACY11 27(732) 04-NOV-76 11:15 PAGE 2
DZTMFD.SRC

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTMF-D-D
PRODUCT NAME: TMA,B-11/TU10,N,W SUPPLEMENTAL INST. TEST
PROGRAM DATE: AUGUST 1976
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: R. B. BARNES
REVISED BY: RON PLATUKIS/R. SOLER

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, 1976 BY DIGITAL EQUIPMENT CORPORATION

CO1

TMA, B-11 SUPPLEMENTAL INST TEST MACY11 27(732) 04-NOV-76 11:15 PAGE 3
DZTMFD.SRC

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1	ABSTRACT	3
2	REQUIREMENTS	3
3	LOADING PROCEDURE	3
4	STARTING PROCEDURE	3
5	SWITCH SETTINGS	3
6	ERROR PRINTOUT	3
7	OPERATION	3
8	TEST DESCRIPTION	3
9	LISTING	3

1. ABSTRACT

THIS PROGRAM IS INTENDED TO BE USED IN ADDITION TO THE TMA,B-11 INSTRUCTION TEST (MAINDEC-11-DZTMA) TO COMPLETE TESTING OF THE MAG TAPE CONTROLLER. THE PROGRAM CONSISTS OF ONLY FOUR (4) TESTS WHICH CHECK ONLY THE TMA,B-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 OR TMB-11 TAPE CONTROLLER (ONLY)
- E. 1-8 TAPE TRANSPORTS (TU10,N,W)

3. LOADING PROCEDURE

- A. USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE
- B. THIS PROGRAM IS LOADABLE AND CHAINABLE PER XXDP, ACT11, AND SLIDE, IN 8K OF MEMORY. (SEE 7.1)

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 AND ALSO 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 (0-7) 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. 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 TYPE OUT
0=PRINT ALL ERRORS
SW12: 1=INHIBIT ITERATION**(FIRST PASS IS SINGLE 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

5.1 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.

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

GO1

TMA,B-11 SUPPLEMENTAL INST TEST MACY11 27(732) 04-NOV-76 11:15 PAGE 7
DZTMFD.SRC

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 (SW11=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.

7.1 CHAIN MODE RUNS A SINGLE PASS ON DRIVE 0 WITH 7 OR 9 TRACK AT THE STANDARD UNIBUS ADDRESS.

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 THIRTY FIVE FEET OF BLANK TAPE IS PASSED. THE PROCEDURE IS TO PERFORM A WRITE WITH IRG, BACKSPACE, WRITE WITH IRG 105(10) TIMES IN ORDER TO ERASE 35 FEET OF TAPE. AFTER REWIND, ISSUE A READ COMMAND AND OPI SHOULD TIME OUT BEFORE THE FIRST RECORD (35 FEET DOWN TAPE) IS FOUND. THE NOMINAL VALUE FOR OPI IS SEVEN SECONDS (7SEC) OR ABOUT TWENTY-SIX FEET (26 FT) OF TAPE. THIRTY-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 SIXTEEN FEET (16 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 (16 FEET DOWN TAPE). THE SIXTEEN FEET OF TAPE RELECTS THE MINIMUM TOLERANCE FOR OPI.

9. LISTING

289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329

.TITLE TMA,B-11 SUPPLEMENTAL INST TEST
;MAINDEC-11-DZTMF-D-D
;JUNE 76
;R. BARNES
.ENABLE ABS,AMA
.MCALL .SACT11

;REGISTER EQUIVS*****

000000
000001
000002
000003
000004
000005
000006
000007

R0=%0
R1=%1
R2=%2
R3=%3
R4=%4
R5=%5
SP=%6
PC=%7

;TRAP CATCHERS*****

.SBTTL ACT11 HOOKS

;;*****
;HOOKS REQUIRED BY ACT11

001000
000046
001660
000052
000000
001000

\$\$VPC=
.=46
\$ENDAD
.=52
.WORD 0
.=\$\$VPC

;SAVE PC
;;1)SET LOC.46 TO ADDRESS OF \$ENDAD IN .SEOP
;;2)SET LOC.52 TO ZERO
;; RESTORE PC

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

J01

TMA,B-11 SUPPLEMENTAL INST TEST MACY11 27(732) 04-NOV-76 11:15 PAGE 10
DZTMFD.SRC ACT11 HOOKS

330
331
332
333 000060 000060
334 000062 004410
335 000000
336
337
338
339
340 000174 000174
341 000176 000000
342
343
344
345 000200 000200
346 000202 005000 001000
347 000202 000137
348
349 000210 000210
350 000210 012700 000001
351 000214 000137 001006
352
353
354
355 000224 000224
356 000224 004370
357 000226 000340
358

```
;TTY INTERRUPT VECTOR*****  
.=60  
TTINT          ;TTY INTERRUPT HANDLER  
0  
  
;SOFTWARE SWITCH REGISTER LOCATIONS*****  
.=174  
DISPREG:0  
SWREG: 0  
  
;STARTING ADDRESS*****  
.=200  
CLR          R0  
JMP          START          ;PROGRAM START  
  
.=210  
MOV          #1,R0  
JMP          RSTART        ;NO HEADER START  
  
;TMA,B-11 INTERRUPT VECTOR*****  
.=224  
MTINT  
340          ;TAPE INTERRUPT HANDLER
```

K01

359 000600
360
361
362 000600 172520
363 000602 172522
364 000604 172524
365 000606 172526
366 000610 000000
367 000612 000020
368 000614 177760
369 000616 177776
370 000620 177570
371 000622 177570
372 000624 177560
373 000626 177562
374 000630 177564
375 000632 177566
376 000634 000010
377 000636 000004
378

. =500
;CONSTANTS*****
MTS: 172520 ;TAPE STATUS REGISTER
MTC: 172522 ;TAPE COMMAND REGISTER
MTBC: 172524 ;TAPE BYTE COUNTER
MTBA: 172526 ;TAPE BUS ADDRESS
UDES: 0 ;UNIT DESCRIPTION(PRESET FOR UNIT 0)
RCNT: 20 ;RECORD COUNT
CCNT: -20 ;CHARACTER COUNT
PSW: 177776 ;PROCESSOR STATUS
SWR: 177570 ;CONSOLE SWITCH REGISTER
DISPLAY: 177570 ;CONSOLE DISPLAY REGISTER
TKS: 177560 ;TTY READ STATUS
TKB: 177562 ;TTY READ BUFFER
TPS: 177564 ;TTY PUNCH STATUS
TPB: 177566 ;TTY OUTPUT BUFFER
ITAMT: 10 ;NUMBER OF ITERATIONS
STALL: 4 ;READY DELAY MULTIPLIER

```

379                                     ;FLAGS AND COUNTERS*****
380
381 000640 000000      TINF: 0
382 000642 000000      TOB: 00
383 000644 000000      TIB: 00
384 000646 000000      TEMP1: 00
385 000650 000000      TEMP2: 00
386 000652 000000      TEMP3: 00
387 000654 000000      EMADDR: 00
388 000656 000000      ERRAD: 00
389 000660 000000      LTADD: 00
390 000662 000000      ITRLP: 00
391 000664 000000      SPFLG: 00
392 000666 000000      STFLG: 00
393 000670 000000      PCNTR: 00
394 000672 000000      BADR: 00
395 000674 000000      BYTES: 00
396 000676 000000      SCNT: 00
397 000700 000000      FUN: 00
398 000702 000000      ITCNT: 00
399 000704 000000      CRCNT: 00
400 000706 000000      DERFL: 00
401 000710 000000      HDRFL: 00
402 000712 000000      PFLG: 00
403 000714 000000      UNP: 00
404 000716 000000      BCNT: 00
405 000720 000000      COUNT: 00
406 000722 000000      TEMPST: 00
407 000724 000000      RDSW: 0

```

```

408                                     ;TEST ENTRY TABLE*****
409
410
411 000726 000000      TSTTBL: 0
412 000730 000000      0
413 000732 001720      T1AD: LT1
414 000734 001720      T1IAD: LT1
415 000736 002146      T2AD: LT2
416 000740 002146      T2IAD: LT2
417 000742 002370      T3AD: LT3
418 000744 002466      T3IAD: LT3IT
419 000746 002620      T4AD: LT4
420 000750 002716      T4IAD: LT4IT
421 000752 000000      0
422

```

MO1

TMA, B-11 SUPPLEMENTAL INST TEST MACY11 27(732) 04-NOV-76 11:15 PAGE 13
 DZTMFD.SRC ACT11 HOOKS

```

423          001000          =1000
424          *****
425          ;PROGRAM START AND HOUSEKEEPING
426          *****
427 001000 012737 177570 000620 START: MOV #177570,SWR ;PRESET TO CONSOLE SWITCHES
428 001006 012777 000340 177602 RSTART: MOV #340,@PSW ;SET PRIORITY
429 001014 012706 000500          MOV #500,SP ;SET STACK POINTER
430 001020 013746 000006          SUSWR: MOV @#6,-(SP) ;SAVE VECTORS
431 001024 013746 000004          MOV @#4,-(SP)
432 001030 012737 001050 000004 MOV #1$,@#4 ;SET UP FOR TIMEOUT
433 001036 022777 177777 177554 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
434 001044 001402          BEQ 2$
435 001046 000407          BR 3$
436 001050 022626          1$: CMP (SP)+,(SP)+ ;ADJUST STACK
437 001052 012737 000176 000620 2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
438 001060 012737 000174 000622 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
439 001066 012637 000004          3$: MOV (SP)+,@#4 ;RESTORE VECTORS
440 001072 012637 000006          MOV (SP)+,@#6
441 001076 122737 000004 000041 CMPB #4,@#41 ;TM-11 MAG TAPE?
442 001104 001006          BNE 4$ ;IF NO, BR
443 001106 012704 006377          MOV #MSG2,R4
444 001112 004737 004656          JSR PC,TTOUT ;LOADER TM-11 MAG TAPE
445 001116 000137 001630          JMP TEND ;END OP
446 001122 012777 010000 177452 4$: MOV #10000,@MTC ;POWER CLEAN
447 001130 005700          TST R0 ;SEE IF SKIP HEADER
448 001132 001116          BNE ST4 ;IF SO: BR
449 001134 012704 005724          MOV #MSG1,R4
450 001140 004737 004656          JSR PC,TTOUT ;PRINT HEADER
451 001144 012704 006012          STO: MOV #MSG2,R4
452 001150 004737 004656          JSR PC,TTOUT ;REQUEST DRIVE NUMBER
453 001154 005037 000610          CLR UDES ;PRESET UNIT 0
454 001160 013703 000610          MOV UDES,R3 ;GET UNIT NUMBER
455 001164 000303          SWAB R3 ;POSITION
456 001166 042703 177770          BIC #177770,R3 ;MASK UNIT NUMBER
457 001172 004737 005054          JSR PC,OCTP ;PRINT CURRENT VALUE
458 001176 005737 000042          TST @#42 ;CHAIN MODE?
459 001202 001404          BEQ 1$ ;IF YES, BR
460 001204 012737 000176 000620 MOV #176,SWR ;STORE SWR
461 001212 000434          BR ST1
462 001214 012705 000652          1$: MOV #TEMP3,R5 ;SET SAVE LOCATION
463 001220 012701 000001          MOV #1,R1 ;SET SIZE OF ENTRY
464 001224 012702 000007          MOV #7,R2 ;SET UPPER LIMIT
465 001230 012703 000000          MOV #0,R3 ;SET LOWER LIMIT
466 001234 004737 004420          JSR PC,TTR ;GO GET UNIT NUMBER
467 001240 000337 000652          SWAB TEMP3 ;POSITION UNIT NUMBER
468 001244 042737 003400 000610 BIC #3400,UDES ;CLEAR OLD NUMBER
469 001252 053737 000652 000610 BIS TEMP3,UDES ;LOAD NEW NUMBER
470 001260 013777 000610 177314 MOV UDES,@MTC ;SELECT UNIT
471 001266 005000          CLR R0
472 001270 022737 000176 000620 CMP #SWREG,SWR
473 001276 001002          BNE ST1
474 001300 004737 005462          JSR PC,CNTLU
475 001304 032777 000100 177266 ST1: BIT #100,@MTC ;SEE IF SELECT REMOTE
476 001312 001013          BNE ST2 ;IF SO: BR
477 001314 005300          DEC R0
478 001316 001372          BNE ST1 ;DELAY FOR SELECT REMOTE

```

479	001320	012704	005356			MOV	#MSG21,R4	
480	001324	004737	004656			JSR	PC,TOUT	;PRINT NOT AVAILABLE
481	001330	005737	000042			TST	#42	;SEE IF CHAIN MODE
482	001334	001703				BEQ	ST0	;IF YES, BR
483	001336	000137	001630			JMP	TEND	;END OP
484	001342	032777	000020	177230	ST2:	BIT	#20,AMTS	;SEE IF 7 CHANNEL
485	001350	001404				BEQ	ST3	;IF NOT: BR
486	001352	052737	040000	000610		BIS	#40000,UDES	;SET TO 800 BPI 7 CHAN
487	001360	000403				BR	ST4	
488	001362	052737	060000	000610	ST3:	BIS	#60000,UDES	;SET TO 800 BPI 9 CHAN
489	001370	000240			ST4:	NOP		
490	001372	012702	000642			MOV	#TOB,R2	;GET START OF TABLE
491	001376	012700	000027			MOV	#27,R0	;SET SIZE OF TABLE
492	001402	005022			ST5:	CLR	(R2)+	;CLEAR TABLE
493	001404	005300				DEC	R0	
494	001406	001375				BNE	ST5	;DO ALL
495	001410	005037	000670			CLR	PCNTR	;CLEAR PASS COUNTER
496								

:TEST SCHEDULAR*****

```

497
498
499
500 001414 000240          TSCD:  NOP
501 001416 005037 000666  CLR      STFLG      ;CLEAR SINGLE TEST FLAG
502 001422 017700 177172  MOV      @SWR,R0   ;GET SWITCH REGISTER
503 001426 042700 177700  BIC      #177700,R0 ;MASK TEST SELECT
504 001432 005700          TST      R0        ;SEE IF SINGLE TEST SELECT
505 001434 001050          BNE      STSCD     ;IF SO: BR
506 001436 012737 000726 000660  MOV      #TSTTBL,LTADD ;GET TABLE START
507 001444 062737 000004 000660  TSCD0:  ADD      #4,LTADD
508 001452 013737 000660 000662  MOV      LTADD,ITRLP ;SET ITERATION ADDRESS
509 001460 062737 000002 000662  ADD      #2,ITRLP
510 001466 005777 177165  TST      @LTADD     ;SEE IF END OF CYCLE
511 001472 001002          BNE      TSCD1    ;IF NOT: BR
512 001474 000137 001630  JMP      TEND      ;GO TO END ROUTINE
513 001500 005037 000710  TSCD1:  CLR      HDRFL   ;CLEAR HEADER FLAG
514 001504 017700 177150  MOV      @LTADD,R0 ;GET TEST ADDRESS
515 001510 000110          JMP      (R0)      ;GO TO TEST
516 001512 004737 005410  TSCD2:  JSR      PC,CKSWR
517 001516 032777 002000 177074  BIT      #2000,@SWR ;SEE IF HALT ON TEST
518 001524 001401          BEQ      TSCD3    ;IF NOT: BR
519 001526 000000          HALT
520 001530 004737 005410  TSCD3:  JSR      PC,CKSWR ;GO TEST FOR ↑G
521 001534 005737 000666  TST      STFLG     ;SEE IF SINGLE TEST
522 001540 001741          BEQ      TSCD0    ;IF NOT: BR
523 001542 017700 177052  MOV      @SWR,R0
524 001546 042700 177750  BIC      #177760,R0 ;GET TEST NUMBER
525 001552 005700          TST      R0        ;SEE IF ALL TESTS
526 001554 001717          BEQ      TSCD     ;IF SO: BR
527 001556 012737 000001 000666  STSCD:  MOV      #1,STFLG ;SET SINGLE TEST FLAG
528 001564 022700 000005  CMP      #5,R0     ;SEE IF EXCEEDED TEST NUMBER
529 001570 003417          BLE      TEND     ;IF SO: BR
530 001572 000241          CLC
531 001574 006100          ROL      R0
532 001576 006100          ROL      R0        ;POSITION NUMBER
533 001600 012737 000726 000660  MOV      #TSTTBL,LTADD ;GET START OF TABLE
534 001606 060037 000660  ADD      R0,LTADD  ;SET POINTER
535 001612 013737 000660 000662  MOV      LTADD,ITRLP
536 001620 062737 000002 000662  ADD      #2,ITRLP  ;SET ITERATION ADDRESS
537 001626 000724          BR       TSCD1    ;GO DO TEST
538 001630 012704 006031  TEND:  MOV      #MSG3,R4
539 001634 004737 004656  JSR      PC,TIOUT  ;PRINT END OF PASS
540 001640 013703 000670  MOV      PCNTR,R3
541 001644 004737 005054  JSR      PC,OC1P  ;PRINT PASS NUMBER
542 001650 013703 000042  MOV      @#42,R3
543 001654 001405          BEQ      HERE
544 001656 000005          RESET
545 001660 004713  SENDAD: JSR      PC,(R3)
546 001662 000240          NOP
547 001664 000240          NOP
548 001666 000240          NOP
549 001670 000240          NOP
550 001672 032777 004000 176720  HERE:  BIT      #4000,@SWR ;SEE IF HALT ON PASS
551 001700 001001          BNE      TENDX   ;IF NOT: BR
552 001702 000000          HALT
553 001704 004737 005410  TENDX:  JSR      PC,CKSWR ;GO TEST FOR ↑G

```

```

553 001710 005237 000670      INC      PCNTR      ;BUMP PASS COUNTER
554 001714 000137 001414      JMP      TSCD      ;RESTART
555
556
557
558
559
560
561
562
563
564
565
566 001720 000240      LT1:     NOP
567 001722 012737 006466 000654      MOV      #LT1MSG,EMADDR ;SET HEADER
568 001730 012702 006636      MOV      #WDATA,R2      ;GET BUFFER START
569 001734 112722 000377      MOVVB   #377,(R2)+      ;INSERT BACKGROUND DATA
570 001740 005000      CLR      RO
571 001742 110022      LT1B:    MOVVB   RO,(R2)+
572 001744 005200      INC      RO              ;LOAD WRITE BUFFER (0,1,2,3,4,5)
573 001746 022700 000006      CMP      #6,RO
574 001752 001373      BNE      LT1B
575 001754 004737 003026      JSR      PC,RWIND        ;GO REWIND
576 001760 012737 000004 000700      MOV      #4,FUN          ;SET WRITE FUNCTION CODE
577 001766 012737 006637 000672      MOV      #WDATA+1,BADR   ;SET DATA POINTER
578 001774 012737 177772 000674      MOV      #-6,BYTES       ;SET SIZE OF RECORD
579 002002 012737 006256 000656      MOV      #MSG17,ERRAD    ;SET WRITE ERROR CODE
580 002010 004737 003304      JSR      PC,EXEC         ;GO EXECUTE COMMAND
581 002014 000240      LT1C:    NOP
582 002016 004737 003534      JSR      PC,ERCHK        ;GO CHECK FOR STATUS ERROR
583 002022 012737 177777 000676      MOV      #-1,SCNT
584 002030 004737 003212      JSR      PC,BKSP         ;GO BACKSPACE ONE RECORD
585 002034 012702 006740      MOV      #RDATA,R2
586 002040 012700 000010      MOV      #10,RO
587 002044 012722 177777      LT1D:    MOV      #-1,(R2)+      ;BACKGROUND READ BUFFER
588 002050 005300      DEC      RO
589 002052 001374      BNE      LT1D
590 002054 012737 000002 000700      MOV      #2,FUN          ;DO ALL
591 002062 012737 006740 000672      MOV      #RDATA,BADR    ;SET READ FUNCTION CODE
592 002070 012737 177772 000674      MOV      #-6,BYTES       ;SET READ POINTER
593 002076 012737 006273 000656      MOV      #MSG18,ERRAD    ;SET SIZE OF RECORD
594 002104 004737 003304      JSR      PC,EXEC         ;SET READ ERROR CODE
595 002110 000240      LT1E:    NOP
596 002112 004737 003534      JSR      PC,ERCHK        ;GO CHECK ERRORS
597 002116 012701 006637      MOV      #WDATA+1,R1    ;SET EXPT DATA POINTER
598 002122 012702 006740      MOV      #RDATA,R2      ;SET RCVD DATA POINTER
599 002126 012700 000006      MOV      #6,RO          ;SET SIZE OF RECORD
600 002132 004737 004022      JSR      PC,DCHK         ;GO CHECK DATA
601 002136 004737 004312      JSR      PC,ITER        ;GO SEE IF ITERATIONS
602 002142 000137 001512      JMP      TSCD2          ;RETURN TO SCHEDULAR
    
```

```

*****
;TEST 1: WRITE FROM ODD BYTE
;
;THIS TEST WILL WRITE A SIX (6) BYTE RECORD
;FROM AN ODD BYTE STARTING ADDRESS. THE RECORD
;WILL BE READ BACK INTO AN EVEN STARTING ADDRESS
;TO TEST FOR PROPER TRANSFER.
*****
    
```


603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648

002146 000240
002150 012737 006524 000654
002156 012702 006636
002162 005000
002164 110022
002166 005200
002170 022700 000006
002174 001373
002176 004737 003026
002202 012737 000004 000700
002210 012737 006636 000672
002216 012737 177772 000674
002224 004737 003304
002230 000240
002232 012737 006256 000656
002240 004737 003534
002244 012737 177777 000676
002252 004737 003212
002256 012702 006740
002262 012700 000010
002266 012722 177777
002272 005300
002274 001374
002276 012737 000002 000700
002304 012737 006741 000672
002312 012737 177772 000674
002320 004737 003304
002324 000240
002326 012737 006273 000656
002334 004737 003534
002340 012701 006636
002344 012702 006741
002350 012700 000006
002354 004737 004022
002360 004737 004312
002364 000137 001512

```
*****  
: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.  
:*****  
LT2:  NOP  
      MOV    #LT2MSG,EMADDR ;SET HEADER POINTER  
      MOV    #WDATA,R2     ;POINT TO START OF WRITE BUFFER  
      CLR    RO  
LT2B:  MOVB   RO,(R2)+      ;LOAD DATA PATTERN  
      INC    RO             ;BUMP PATTERN  
      CMP    #6,RO         ;SEE IF DONE  
      BNE    LT2B          ;IF NOT: BR  
      JSR    PC,RWIND      ;GO REWIND TO BOT  
      MOV    #4,FUN        ;SET WRITE OP-CODE  
      MOV    #WDATA,BADR   ;SET STARTING ADDRESS  
      MOV    #-6,BYTES     ;SET SIZE OF RECORD  
      JSR    PC,EXEC       ;GO EXECUTE COMMAND  
LT2C:  NOP  
      MOV    #MSG17,ERRAD  ;SET ERROR CODE  
      JSR    PC,ERCHK      ;GO CHECK FOR STATUS ERROR  
      MOV    #-1,SCNT      ;GO BACKSPACE ONE RECORD  
      JSR    PC,BKSP       ;GO BACKSPACE ONE RECORD  
      MOV    #RDATA,R2    ;GET READ BUFFER POINTER  
      MOV    #10,RO       ;SET SIZE  
LT2D:  MOV    #-1,(R2)+   ;BACKGROUND POINTER  
      DEC    RO           ;SEE IF DONE  
      BNE    LT2D        ;IF NOT: BR  
      MOV    #2,FUN       ;SET READ FUNCTION CODE  
      MOV    #RDATA+1,BADR ;SET START OF READ BUFFER  
      MOV    #-6,BYTES   ;SET SIZE OF RECORD  
      JSR    PC,EXEC     ;GO EXECUTE COMMAND  
LT2E:  NOP  
      MOV    #MSG18,ERRAD ;SET ERROR CODE  
      JSR    PC,ERCHK     ;GO CHECK FOR STATUS ERROR  
      MOV    #WDATA,R1   ;POINT TO EXPT DATA  
      MOV    #RDATA+1,R2 ;POINT TO RCVD DATA  
      MOV    #6,RO       ;SET SIZE OF RECORD  
      JSR    PC,DCHK     ;GO CHECK DATA  
      JSR    PC,ITER     ;GO SEE IF ITERATION  
      JMP    TSCD2      ;RETURN TO SCHEDULAR
```

649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693

```

;*****
;TEST 3: OPI TOO LONG
;
;THIS TEST WILL ERASE APPROXIMATELY THIRTYFIVE (35)
;FEET OF TAPE BY WRITING WITH IRG, BACKSPACING
;AND REPEATING THE SEQUENCE 105(10) TIMES. TAPE
;WILL REWIND AND A READ FORWARD ISSUED. THE
;OPI TIMER SHOULD SHUTDOWN THE UNIT BEFORE
;REACHING THE FIRST RECORD ON TAPE.
;*****

002370 000240 LT3: NOP
002372 012737 006557 000654 MOV #LT3MSG,EMADDR ;SET TEST HEADER
002400 012700 000151 MOV #151,RO ;SET NUMBER OF WRITE IRG/BACKSPACE
002404 004737 003026 JSR PC,RWIND ;GO REWIND UNIT
002410 012737 000014 000700 LT3A: MOV #14,FUN ;SET WRITE IRG FUNCTION CODE
002416 012737 006636 000672 MOV #WDATA,BADR ;SET BUS ADDRESS
002424 012737 177760 000674 MOV #-20,BYTES ;SET SIZE OF RECORD
002432 004737 003304 JSR PC,EXEC ;GO EXECUTE COMMAND
002436 012737 006256 000656 LT3B: MOV #MSG17,ERRAD ;SET ERROR CODE
002444 004737 003534 JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
002450 012737 177777 000676 MOV #-1,SCNT
002456 004737 003212 JSR PC,BKSP ;GO BACKSPACE ONE RECORD
002462 005300 DEC RO ;SEE IF DONE ALL
002464 001351 BNE LT3A ;IF NOT: BR
002466 000240 LT3IT: NOP
002470 004737 003026 JSR PC,RWIND ;GO REWIND
002474 012737 000500 000636 MOV #500,STALL ;SET OPI STALL
002502 012737 006740 000672 MOV #RDATA,BADR ;SET START OF READ BUFFER
002510 012737 177760 000674 MOV #-20,BYTES ;SET SIZE OF RECORD
002516 012737 000002 000700 MOV #2,FUN ;SET READ FUNCTION CODE
002524 012737 006307 000656 MOV #MSG19,ERRAD ;SET ERROR CODE
002532 004737 003304 JSR PC,EXEC ;GO EXECUTE COMMAND
002536 000240 LT3C: NOP
002540 012737 000004 000636 MOV #4,STALL ;RESET NORMAL STALL
002546 032777 000400 176024 BIT #400,AMTS ;SEE IF BTE IS SET
002554 001007 BNE LT3X ;IF SO: BR
002556 012737 000001 000664 MOV #1,SPFLG ;SET NO BA PRINT FLAG
002564 004737 003612 JSR PC,ERPT ;GO PRINT ERROR
002570 005037 000664 CLR SPFLG ;RESET FLAG
002574 012737 000002 000634 LT3X: MOV #2,ITAMT ;SET TO TWO (2) ITERATIONS
002602 004737 004312 JSR PC,ITER ;GO SEE IF ITERATION
002606 012737 000010 000634 MOV #10,ITAMT ;RESET ITERATIONS
002614 000137 001512 JMP TSC02 ;RETURN TO SCHEDULAR
    
```

694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733

```

002620 000240
002622 012737 006606 000654
002630 004737 003026
002634 012700 000062
002640 012737 000014 000700
002646 012737 006636 000672
002654 012737 177760 000674
002662 012737 006256 000656
002670 004737 003304
002674 004737 003534
002700 012737 177777 000676
002706 004737 003212
002712 005300
002714 001351
002716 000240
002720 004737 003026
002724 012737 000500 000636
002732 012737 006740 000672
002740 012737 177760 000674
002746 012737 000002 000700
002754 012737 006332 000656
002762 004737 003304
002766 004737 003534
002772 000240
002774 012737 000004 000636
003002 012737 000002 000634
003010 004737 004312
003014 012737 000010 000634
003022 000137 001512
    
```

```

LT4:  NOP
      MOV #LT4MSG,EMADDR ;SET HEADER
      JSR PC,RWND ;GO REWIND
      MOV #62,RO ;SET NUMBER OF WRITE IRG/BACKSPACES
LT4A: MOV #14,FUN ;SET WRITE IRG FUNCTION CODE
      MOV #WDATA,BADR ;SET START OF WRITE BUFFER
      MOV #-20,BYTES ;SET SIZE OF RECORD
      MOV #MSG17,ERRAD ;SET ERROR CODE
      JSR PC,EXEC ;GO EXECUTE COMMAND
LT4B: JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
      MOV #-1,SCNT
      JSR PC,BKSP ;GO BACKSPACE ONE RECORD
      DEC RO ;SEE IF DONE ALL SEQUENCES
      BNE LT4A ;IF NOT: BR
LT4IT: NOP
      JSR PC,RWND ;REWIND
      MOV #500,STALL ;SET OPI STALL
      MOV #RDATA,BADR ;SET START OF READ BUFFER
      MOV #-20,BYTES ;SET SIZE OF RECORD
      MOV #2,FUN ;SET READ FUNCTION CODE
      MOV #MSG20,ERRAD ;SET ERROR CODE
      JSR PC,EXEC ;GO EXECUTE COMMAND
LT4C: JSR PC,ERCHK ;GO CHECK FOR STATUS ERRORS
      NOP
      MOV #4,STALL ;RESET NORMAL STALL
      MOV #2,ITAMT ;SET TO TWO (2) ITERATIONS
      JSR PC,ITER ;GO SEE IF ITERATIONS
      MOV #10,ITAMT ;RESET ITERATIONS
      JMP TSCD2 ;RETURN TO SCHEDULAR
    
```

```

:*****
:TEST 4: OPI TOO SHORT
:
:THIS TEST WILL ERASE APPROXIMATELY SIXTEEN (16) FEET
:OF TAPE BY WRITING WITH IRG, BACKSPACING
:ONE (1) RECORD AND REPEATING THIS SEQUENCE
:50(10) TIMES. TAPE WILL REWIND AND BE READ
:FORWARD. THE FIRST RECORD ON TAPE SHOULD BE
:REACHED BEFORE OPI TIMES OUT.
:*****
    
```

```

734                                     ;REWIND SUBROUTINE*****
735
736 003026 000240 RWND: NOP
737 003030 013777 000610 175544 MOV UDES,AMTC ;SELECT UNIT
738 003036 032777 000040 175534 BIT #40,AMTS ;SEE IF AT BOT
739 003044 001056 BNE RWNDXX ;IF SO: BR
740 003046 052777 000017 175526 BIS #17,AMTC ;START REWIND
741 003054 105777 175522 IS: TSTB AMTC
742 003060 100375 BPL IS ;AWAIT CUR
743 003062 032777 000001 175510 RWND1: BIT #1,AMTS ;AWAIT TUR
744 003070 001774 BEQ RWND1
745 003072 032777 000040 175500 BIT #40,AMTS ;SEE IF BOT SET
746 003100 001040 BNE RWNDXX ;IF SO: BR
747 003102 032777 020000 175510 BIT #20000,ASWR ;SEE IF PRINT ERROR
748 003110 001030 BNE RWNDX ;IF NOT: BR
749 003112 013704 000654 MOV EMADDR,R4
750 003116 004737 004656 JSR PC,TTOUT ;PRINT HEADER
751 003122 012704 006052 MOV #MSG4,R4
752 003126 004737 004656 JSR PC,TTOUT ;PRINT REWIND ERROR
753 003132 012704 006102 MOV #MSG5,R4
754 003136 004737 004656 JSR PC,TTOUT ;PRINT MTS TAG
755 003142 017703 175432 MOV AMTS,R3
756 003146 004737 005044 JSR PC,OTPE ;PRINT MTS
757 003152 012704 006111 MOV #MSG6,R4
758 003156 004737 004656 JSR PC,TTOUT ;PRINT MTC TAG
759 003162 017703 175414 MOV AMTC,R3
760 003166 004737 005044 JSR PC,OTPE ;PRINT MTC
761 003172 005777 175422 RWNDX: TST ASWR ;SEE IF HALT ON ERROR
762 003176 100001 BPL RWNDXX ;IF NOT: BR
763 003200 000000 HALT
764 003202 004737 005410 RWNDXX: JSR PC,CKSWR ;GO TEST FOR IG
765 003206 000240 NOP
766 003210 000207 RTS PC ;RETURN
767
768                                     ;BACKSPACE SUBROUTINE*****
769
770 003212 000240 BKSP: NOP
771 003214 013777 000610 175360 MOV UDES,AMTC ;SELECT UNIT
772 003222 013777 000676 175354 MOV SCNT,AMTBC ;SET NUMBER OF RECORDS TO SPACE
773 003230 052777 000013 175344 BIS #13,AMTC ;START SPACE REVERSE
774 003236 105777 175340 IS: TSTB AMTC
775 003242 100375 BPL IS ;AWAIT CUR
776 003244 032777 000001 175326 BKSP1: BIT #1,AMTS ;AWAIT TUR
777 003252 001774 BEQ BKSP1 ;SET SPACE FLAG
778 003254 012737 000001 000664 MOV #1,SPFLG
779 003262 012737 006140 000656 MOV #MSG9,ERRAD
780 003270 004737 003534 JSR PC,ERCHK ;GO CHECK FOR ERROR
781 003274 005037 000664 CLR SPFLG ;CLEAR SPACE FLAG
782 003300 000240 NOP
783 003302 000207 RTS PC ;RETURN
784

```

```

;COMMAND EXECUTE SUBROUTINE*****
785
786
787 003304 000240          EXEC:  NOP
788 003306 005005          CLR      R5
789 003310 032777 000200 175264 EXEC0:  BIT      #200, @MTC      ;SEE IF CUR
790 003316 001021          BNE     EXEC2          ;IF SO: BR
791 003320 005305          DEC      R5            ;SEE IF TIMED OUT
792 003322 001372          BNE     EXEC0          ;IF NOT: BR
793 003324 005737 000710  TST     HDRFL         ;SEE IF DONE HEADER
794 003330 001004          BNE     EXEC1          ;IF SO: BR
795 003332 013704 000654  MOV     EMADDR, R4
796 003336 004737 004656  JSR     PC, TTOUT     ;ELSE PRINT HEADER
797 003342 012704 006163  EXEC1:  MOV     #MSG10, R4
798 003346 004737 004656  JSR     PC, TTOUT     ;PRINT NOT READY ERROR
799 003352 005777 175242  TST     @SWR          ;SEE IF HALT ON ERROR
800 003356 100001          BPL     EXEC2          ;IF NOT: BR
801 003360 000000          HALT
802 003362 004737 005410  EXEC2:  JSR     PC, CKSWR ;GO TEST FOR ↑G
803 003366 000240          NOP
804 003370 013777 000610 175204 MOV     UDES, @MTC    ;SELECT UNIT
805 003376 013777 000672 175202 MOV     BADR, @MTBA   ;SET BUS MEMORY ADDRESS
806 003404 013777 000674 175172 MOV     BYTES, @MTBC ;SET BYTE COUNT
807 003412 013701 000700  MOV     FUN, R1       ;GET FUNCTION
808 003416 052701 000101  BIS     #101, R1      ;SET IN GO BIT AND INTERRUPT ENABLE
809 003422 050177 175154  BIS     R1, @MTC     ;LOAD COMMAND+GO+IE
810 003426 000240          NOP
811 003430 005077 175162  CLR     @PSW          ;ALLOW INTERRUPTS
812 003434 013737 000636 000646 MOV     STALL, TEMP1  ;SET READY STALL
813 003442 005001          CLR     R1
814 003444 005301          EXEC3:  DEC     R1
815 003446 001376          BNE     EXEC3          ;AWAIT INTERRUPT
816 003450 005337 000646  DEC     TEMP1
817 003454 001373          BNE     EXEC3
818 003456 032777 020000 175134 BIT     #20000, @SWR  ;SEE IF PRINT ERROR
819 003464 001013          BNE     EXECX          ;IF NOT: BR
820 003466 005737 000710  TST     HDRFL         ;SEE IF DONE HEADER
821 003472 001004          BNE     EXEC4          ;IF SO: BR
822 003474 013704 000654  MOV     EMADDR, R4
823 003500 004737 004656  JSR     PC, TTOUT     ;PRINT HEADER
824 003504 012704 006200  EXEC4:  MOV     #MSG11, R4
825 003510 004737 004656  JSR     PC, TTOUT     ;PRINT NO INTERRUPT MESSAGE
826 003514 005777 175100  EXECX:  TST     @SWR          ;SEE IF HALT ON ERROR
827 003520 100001          BPL     EXECXX          ;IF NOT: BR
828 003522 000000          HALT
829 003524 004737 005410  EXECXX: JSR     PC, CKSWR ;GO TEST FOR ↑G
830 003530 000240          NOP
831 003532 000207          RTS     PC            ;RETURN TO CALLER
832
    
```

```

;STATUS ERROR CHECK SUBROUTINE*****
833
834
835 003534 005777 175042 ERCHK: TST JMTC ;SEE IF ANY ERROR BITS
836 003540 100002 BPL ERCHK1 ;IF NOT: BR
837 003542 000137 003612 JMP ERPT ;ELSE PRINT ERROR
838 003546 005777 175032 ERCHK1: TST JMTBC ;SEE IF BYTE COUNT IS ZERO
839 003552 001402 BEQ ERCHK2 ;IF SO: BR
840 003554 000137 003612 JMP ERPT ;ELSE PRINT ERROR
841 003560 013703 000674 ERCHK2: MOV BYTES,R3
842 003564 005403 NEG R3
843 003566 063703 000672 ADD BADR,R3 ;SET EXPT BUS ADDRESS
844 003572 005737 000664 TST SPFLG ;SEE IF SPACE OPERATION
845 003576 001401 BEQ ERCHK3 ;IF NOT: BR
846 003600 000207 RTS PC
847 003602 020377 175000 ERCHK3: CMP R3,JMTBA ;SEE IF EXPT=RCVD
848 003606 001001 BNE ERPT ;IF NOT: BR
849 003610 000207 RTS PC ;ELSE EXIT
850 003612 000240 ERPT: NOP
851 003614 032777 020000 174776 BIT #20000,JSWR ;SEE IF SHOULD PRINT
852 003622 001067 BNE ERPTX ;IF NOT: BR
853 003624 005737 000710 TST HDRFL ;SEE IF DONE HEADER
854 003630 001006 BNE ERPT1 ;IF SO: BR
855 003632 013704 000654 MOV EMADDR,R4
856 003636 004737 004656 JSR PC,TTOUT ;ELSE PRINT HEADER
857 003642 005237 000710 INC HDRFL ;SET FLAG
858 003646 013704 000656 ERPT1: MOV ERRAD,R4
859 003652 004737 004656 JSR PC,TTOUT ;PRINT ERROR CODE
860 003656 012704 006102 MOV #MSG5,R4 ;PRINT MTS TAG
861 003662 004737 004656 JSR PC,TTOUT
862 003666 017703 174706 MOV JMTS,R3 ;PRINT MTS
863 003672 004737 005044 JSR PC,OCPE
864 003676 012704 006111 MOV #MSG6,R4 ;PRINT MTC TAG
865 003702 004737 004656 JSR PC,TTOUT
866 003706 017703 174670 MOV JMTC,R3 ;PRINT MTC
867 003712 004737 005044 JSR PC,OCPE
868 003716 012704 006120 MOV #MSG7,R4 ;PRINT BYTE COUNT TAG
869 003722 004737 004656 JSR PC,TTOUT
870 003726 017703 174652 MOV JMTBC,R3 ;PRINT BYTE COUNT
871 003732 004737 005054 JSR PC,OCPE ;SEE IF PRINT BA
872 003736 005737 000664 TST SPFLG ;IF NOT: BR
873 003742 001017 BNE ERPTX
874 003744 012704 006130 MOV #MSG8,R4 ;PRINT BUS ADDRESS TAG
875 003750 004737 004656 JSR PC,TTOUT
876 003754 017703 174626 MOV JMTBA,R3 ;PRINT CURRENT ADDRESS
877 003760 004737 005054 JSR PC,OCPE
878 003764 013703 000674 MOV BYTES,R3
879 003770 005403 NEG R3
880 003772 063703 000672 ADD BADR,R3
881 003776 004737 005054 JSR PC,OCPE ;PRINT EXPT ADDRESS
882 004002 005777 174612 ERPTX: TST JSWR ;SEE IF HALT ON ERROR
883 004006 100001 BPL ERPTXX ;IF NOT: BR
884 004010 000000 HALT
885 004012 004737 005410 ERPTXX: JSR PC,CKSWR ;GO TEST FOR IG
886 004016 000137 004262 JMP SCOPE ;GO SEE IF SCOPE ON ERROR
    
```

```

887                                     ;DATA CHECK SUBROUTINE*****
888
889 004022 000240          DCHK:  NOP          ;CLEAR COUNTER
890 004024 005037 000704  DCHK:  CLR          CRCNT      ;SEE IF EXPT DATA=RCVD DATA
891 004030 121112          DCHK0: CMPB       (R1),(R2)  ;IF NOT: BR
892 004032 001007          DCHK1: BNE        DCHKE     ;BUMP CHARACTER COUNTER
893 004034 005237 000704  DCHK1: INC        CRCNT
894 004040 122122          DCHK1: CMPB       (R1)+,(R2)+
895 004042 005300          DCHK1: DEC        RD          ;SEE IF DONE
896 004044 001371          DCHK1: BNE        DCHK0     ;IF NOT: BR
897 004046 000137 004222  DCHK1: JMP        DCHKX     ;ELSE GO TO EXIT ROUTINE
898 004052 000240          DCHKE: NOP
899 004054 012737 000001 000706  DCHKE: MOV        #1,DERFL   ;SET ERROR FLAG
900 004062 032777 020000 174530  DCHKE: BIT        #20000,DSWR  ;SEE IF PRINT ERROR
901 004070 001054          DCHKE: BNE        DCHKX     ;IF NOT: BR
902 004072 005737 000710  DCHKE: TST        HDRFL   ;SEE IF DONE HEADER
903 004076 001007          DCHKE: BNE        DCHKE1    ;IF SO: BR
904 004100 013704 000654  DCHKE: MOV        EMADDR,R4
905 004104 004737 004656  DCHKE: JSR        PC,TTOUT  ;PRINT HEADER
906 004110 012737 000001 000710  DCHKE: MOV        #1,HDRFL   ;SET HEADER FLAG
907 004116 012704 006220  DCHKE1: MOV        #MSG12,R4
908 004122 005737 000712  DCHKE1: TST        PFLG     ;SEE IF PRINTED DATA ERROR TAG
909 004126 001004          DCHKE1: BNE        DCHKE2    ;IF SO: BR
910 004130 005237 000712  DCHKE1: INC        PFLG
911 004134 004737 004656  DCHKE2: JSR        PC,TTOUT  ;ELSE PRINT DATA ERROR TAG
912 004140 012704 006234  DCHKE2: MOV        #MSG13,R4
913 004144 004737 004656  DCHKE2: JSR        PC,TTOUT  ;PRINT CHAR NUMBER TAG
914 004150 013703 000704  DCHKE2: MOV        CRCNT,R3
915 004154 004737 005054  DCHKE2: JSR        PC,OC1P   ;PRINT CHAR NUMBER
916 004160 012704 006242  DCHKE2: MOV        #MSG14,R4
917 004164 004737 004656  DCHKE2: JSR        PC,TTOUT  ;PRINT GOOD TAG
918 004170 111103          DCHKE2: MOVB       (R1),R3
919 004172 004737 005302  DCHKE2: JSR        PC,DOUT   ;PRINT GOOD CHARACTER
920 004176 012704 006247  DCHKE2: MOV        #MSG15,R4
921 004202 004737 004656  DCHKE2: JSR        PC,TTOUT  ;PRINT BAD TAG
922 004206 111203          DCHKE2: MOVB       (R2),R3
923 004210 004737 005302  DCHKE2: JSR        PC,DOUT   ;PRINT BAD CHARACTER
924 004214 000240          DCHKE2: NOP
925 004216 000137 004034  DCHKE2: JMP        DCHK1
926 004222 000240          DCHKX: NOP
927 004224 005737 000706  DCHKX: TST        DERFL   ;SEE IF ANY ERROR
928 004230 001404          DCHKX: BEQ        DCHKXX   ;IF NOT: BR
929 004232 005777 174362  DCHKX: TST        DSWR     ;SEE IF HALT ON ERROR
930 004236 100001          DCHKX: BPL        DCHKXX   ;IF NOT: BR
931 004240 000000          DCHKXX: HALT
932 004242 004737 005410  DCHKXX: JSR        PC,CKSWR  ;GO TEST FOR 1G
933 004246 000240          DCHKXX: NOP
934 004250 005037 000712  DCHKXX: CLR        PFLG     ;CLEAR PRINT FLAG
935 004254 005037 000706  DCHKXX: CLR        DERFL   ;CLEAR DATA ERROR FLAG
936 004260 000207          DCHKXX: RTS         PC      ;RETURN
    
```

```

937                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
938
939 004262 000240 SCOPE: NOP
940 004264 032777 040000 174326 BIT #40000, @SWR ;SEE IF LOOP ON ERROR
941 004272 001001 BNE SCOPE1 ;IF SO: BR
942 004274 000207 RTS PC ;ELSE EXIT
943 004276 000240 SCOPE1: NOP
944 004300 005726 TST (SP)+ ;RESET STACK
945 004302 000240 NOP
946 004304 017703 174350 MOV @LTADD, R3
947 004310 000113 JMP (R3) ;LOOP ON ERROR
948
949                                     ;TEST ITERATION SUBROUTINE*****
950
951 004312 000240 ITER: NOP
952 004314 004737 005410 JSR PC, CKSWR
953 004320 032777 010000 174272 BIT #10000, @SWR ;SEE IF ITERATIONS
954 004326 001403 BEQ ITER1 ;IF SO: BR
955 004330 005037 000702 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
956 004334 000207 RTS PC ;ELSE EXIT
957 004336 005737 000670 ITER1: TST PCNTR ;SEE IF FIRST PASS
958 004342 001772 BEQ ITER0 ;IF SO: BR
959 004344 005237 000702 INC ITCNT ;BUMP COUNTER
960 004350 023737 000702 000634 CMP ITCNT, ITAMT ;SEE IF DONE ALL
961 004356 001764 BEQ ITER0 ;IF SO: BR
962 004360 005726 TST (SP)+ ;RESET STACK
963 004362 017700 174274 MOV @ITRLP, R0 ;SET ITERATION POINTER
964 004366 000110 JMP (R0) ;GO ITERATE
965
966                                     ;MAG TAPE INTERRUPT HANDLER*****
967
968 004370 000240 MTINT: NOP
969 004372 022626 000100 174200 CMP (SP)+, (SP)+ ;RESET STACK POINTER
970 004374 042777 BIC #100, @MTC ;CLEAR INTERRUPT ENABLE
971 004402 000240 NOP
972 004404 000240 NOP
973 004406 000207 RTS PC ;RETURN
974
975                                     ;TTY INTERRUPT HANDLER*****
976
977 004410 000240 TTINT: NOP
978 004412 000240 NOP
979 004414 000240 NOP
980 004416 000002 RTI
981

```



```

982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999 004420 005037 000646 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
1000 004424 005000 CLR RO
1001 004426 004737 004604 TTR0: JSR PC,TTIN ;GO READ CHARACTER
1002 004432 122737 000215 000644 CMPB #215,TIB ;SEE IF CR
1003 004440 001005 BNE TTR1 ;IF NOT: BR
1004 004442 005737 000646 TST TEMP1 ;SEE IF FIRST CHARACTER
1005 004446 001446 BEQ TTR5 ;IF SO: BR
1006 004450 000137 004542 JMP TTR2 ;ELSE GO LOAD VALUE
1007 004454 122737 000260 000644 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0
1008 004462 101402 BLOS TTR1A ;IF NOT: BR
1009 004464 000137 004566 JMP T1NER ;ELSE GO TO ERROR
1010 004470 122737 000270 000644 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7
1011 004476 101002 BHI TTR1B ;IF NOT: BR
1012 004500 000137 004566 JMP T1NER ;ELSE GO TO ERROR
1013 004504 005237 000646 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
1014 004510 000241 CLC
1015 004512 006100 ROL RO
1016 004514 000241 CLC
1017 004516 006100 ROL RO ;SHIFT 3 LEFT
1018 004520 000241 CLC
1019 004522 006100 ROL RO
1020 004524 042737 177770 000644 BIC #177770,TIB ;STRIP ASCII
1021 004532 053700 000644 BIS TIB,RO ;LOAD CHARACTER
1022 004536 005301 DEC R1 ;SEE IF DONE
1023 004540 001332 BNE TTR0 ;IF NOT: BR
1024 004542 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
1025 004544 101402 BLOS TTR3 ;IF OT: BR
1026 004546 000137 004566 JMP T1NER ;ELSE GO TO ERROR
1027 004552 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
1028 004554 101402 BLOS TTR4 ;IF NOT: BR
1029 004556 000137 004566 JMP T1NER ;ELSE GO TO ERROR
1030 004562 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
1031 004564 000207 TTR5: RTS PC ;EXIT
1032

```

```

*****
;TTY ENTRY SUBROUTINE:
;THIS SUBROUTINE IS USED BY THE TEST CONDITION
;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
;AT THE TTY AND CHECK THEM FOR LEGALITY AND
;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
;THE CALLING ROUTINE.
;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
;MAY BE REENTERED.
;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
;CARRIAGE RETURN
*****

```

DZTMFD.SRC ACT11 HOOKS

```

1033 ;TTY ENTRY ERROR SUBROUTINE*****
1034
1035 004566 012704 006254 T1NER: MOV #MSG16,R4
1036 004572 004737 004656 JSR PC,TTOUT ;PRINT?
1037 004576 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
1038 004602 000207 RTS PC ;REDO VALUE ENTRY
1039
1040 ;TTY READ SUBROUTINE*****
1041
1042 004604 005077 174014 TTIN: CLR @TKS
1043 004610 005077 174012 CLR @TKB
1044 004614 005037 000644 CLR TIB
1045 004620 005277 174000 INC @TKS
1046 004624 105777 173774 TTIN1: TSTB @TKS
1047 004630 100375 BPL TTIN1
1048 004632 017737 173770 000644 MOV @TKB,TIB
1049 004640 105777 173764 TTIN2: TSTB @TPS
1050 004644 100375 BPL TTIN2
1051 004646 113777 000644 173756 MOVB TIB,@TPB
1052 004654 000207 RTS PC
1053
1054 ;TTY OUTPUT SUBROUTINE*****
1055
1056 004656 112437 000642 TTOUT: MOVB (R4)+,TOB
1057 004662 122737 000043 000642 CMPB #43,TOB
1058 004670 001452 BEQ TEX
1059 004672 122737 000045 000642 CMPB #45,TOB
1060 004700 001407 BEQ TCRLF
1061 004702 122737 000041 000642 CMPB #41,TOB
1062 004710 001443 BEQ TBELL
1063 004712 004737 005002 JSR PC,TOG
1064 004716 000757 BR TTOUT
1065 004720 112737 000015 000642 TCRLF: MOVB #15,TOB
1066 004726 004737 005002 JSR PC,TOG
1067 004732 012703 000004 MOV #4,R3
1068 004736 005037 000642 TCRLFA: CLR TOB
1069 004742 004737 005002 JSR PC,TOG
1070 004746 005303 DEC R3
1071 004750 001372 BNE TCRLFA ;DO FILLERS
1072 004752 112737 000012 000642 MOVB #12,TOB
1073 004760 004737 005002 JSR PC,TOG
1074 004764 105737 000724 TSTB RDSW
1075 004770 100401 BMI IS
1076 004772 000731 BR TTOUT
1077 004774 005037 000724 IS: CLR RDSW
1078 005000 000406 BR TEX
1079 005002 105777 173622 TOG: TSTB @TPS
1080 005006 100375 BPL TOG
1081 005010 113777 000642 173614 MOVB TOB,@TPB
1082 005016 000207 RTS PC
1083 005020 012703 000002 TBELL: MOV #2,R3
1084 005024 012737 000007 000642 TBELA: MOV #7,TOB
1085 005032 004737 005002 JSR PC,TOG
1086 005036 005303 DEC R3
1087 005040 001371 BNE TBELA
1088 005042 000705 BR TTOUT

```

```

1089
1090                                     ;OCTAL OUTPUT SUBROUTINE*****
1091
1092 005044 012737 000001 005300 OCTPE: MOV #1,OFL
1093 005052 000402 BR OCTPE1
1094 005054 005037 005300 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
1095 005060 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
1096 005062 001007 BNE OCTP0 ;IF NOT ZERO: BR
1097 005064 005737 005300 TST OFL ;SEE IF PRINT ALL 0
1098 005070 001004 BNE OCTP0 ;IF SO: BR
1099 005072 004737 005260 JSR PC,OCTPG1 ;ELSE PRINT ZERO
1100 005076 000137 005222 JMP OCTP3 ;SPACE AND EXIT
1101 005102 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
1102 005106 001406 BEQ OCTP1 ;IF NOT: BR
1103 005110 012704 000001 MOV #1,R4
1104 005114 004737 005236 JSR PC,OCTPG ;PRINT 1
1105 005120 000137 005132 JMP OCTP2
1106 005124 005004 OCTP1: CLR R4 ;PRINT 0
1107 005126 004737 005236 JSR PC,OCTPG
1108 005132 010304 OCTP2: MOV R3,R4
1109 005134 006004 ROR R4
1110 005136 006004 ROR R4
1111 005140 006004 ROR R4 ;POSITION DIGIT
1112 005142 006004 ROR R4
1113 005144 000304 SWAB R4
1114 005146 004737 005236 JSR PC,OCTPG ;PRINT DIGIT 2
1115 005152 010304 MOV R3,R4
1116 005154 006004 ROR R4
1117 005156 000304 SWAB R4
1118 005160 004737 005236 JSR PC,OCTPG ;PRINT DIGIT 3
1119 005164 010304 MOV R3,R4
1120 005166 006104 ROL R4
1121 005170 006104 ROL R4
1122 005172 000304 SWAB R4
1123 005174 004737 005236 JSR PC,OCTPG ;PRINT DIGIT 4
1124 005200 010304 MOV R3,R4
1125 005202 006004 ROR R4
1126 005204 006004 ROR R4
1127 005206 006004 ROR R4
1128 005210 004737 005236 JSR PC,OCTPG
1129 005214 010304 MOV R3,R4
1130 005216 004737 005236 JSR PC,OCTPG ;PRINT DIGIT 5
1131 005222 012737 000240 000642 OCTP3: MOV #240,TOB
1132 005230 004737 005002 JSR PC,TOG ;PRINT SPACE
1133 005234 000207 RTS PC ;EXIT
    
```

```

1134
1135
1136
1137 005236 042704 177770      OCTPG:  BIC      #177770,R4
1138 005242 001004                BNE      OCTPG0
1139 005244 005737 005300      TST      OFL
1140 005250 001001                BNE      OCTPG0
1141 005252 000207                RTS      PC
1142 005254 005237 005300      OCTPG0:  INC      OFL
1143 005260 052704 000260      OCTPG1:  BIS      #260,R4
1144 005264 010437 000642      MOV      R4,TOB
1145 005270 004737 005002      JSR      PC,TOG
1146 005274 010304                MOV      R3,R4
1147 005276 000207                RTS      PC
1148 005300 000000      OFL:      0                ;FIRST CHAR FLAG
1149
1150                ;DATA CHARACTER OUTPUT SUBROUTINE*****
1151
1152 005302 005037 000642      DOUT:    CLR      TOB
1153 005306 012704 000010                MOV      #10,R4                ;SET NUMBER TO PRINT
1154 005312 110337 000642      MOV      R3,TOB
1155 005316 105777 173306      DOUT1:   TSTB     @TPS
1156 005322 100375                BPL      DOUT1
1157 005324 132737 000200 000642      BITB     #200,TOB
1158 005332 001404                BEQ      DOUT2
1159 005334 012777 000061 173270      MOV      #061,@TPB
1160 005342 000403                SR      DOUT3
1161 005344 012777 000060 173260      DOUT2:   MOV      #060,@TPB
1162 005352 006137 000642      DOUT3:   ROL      TOB
1163 005356 005304                DEC      R4
1164 005360 001356                BNE      DOUT1
1165 005362 000207                RTS      PC
1166 005364 013703 000652      DOUTD:   MOV      TEMP3,R3
1167 005370 000303                SWAB    R3
1168 005372 004737 005302      JSR      PC,DOUT
1169 005376 013703 000652      MOV      TEMP3,R3
1170 005402 004737 005302      JSR      PC,DOUT
1171 005406 000207                RTS      PC
1172
1173
1174                ;SUBROUTINE TO CHANGE CONTENTS OF SOFTWARE SWITCH REGISTER
1175 005410 022737 000176 000620  CKSWR:  CMP      #SWREG,SWR                ;SOFTWARE SWITCH REG PRESENT
1176 005416 001041                BNE      OUT                    ;NO, GET OUT
1177 005420 105777 173200      TSTB     @TKS                    ;YES, WAIT FOR
1178 005424 100036                BPL      OUT                    ;READY, GET CHARACTER
1179 005426 017737 173174 000644      MOV      @TKB,TIB                ;AND STRIP OFF
1180 005434 042737 177600 000644      BIC      #177600,TIB                ;THE GARBAGE
1181 005442 022737 000007 000644      CMP      #7,TIB                    ;IS IT A <IG>
1182 005450 001024                BNE      OUT
1183 005452 012704 006433      MOV      #SCNTG,R4
1184 005456 004737 004656      JSR      PC,TTOUT
1185 005462 012704 006437      CNTLU:  MOV      #SMSWR,R4
1186 005466 004737 004656      JSR      PC,TTOUT
1187 005472 017703 173122      MOV      @SWR,R3
1188 005476 004737 005044      JSR      PC,OCTPE
1189 005502 012704 006451      MOV      #SMNEW,R4

```



```

1229                                     ;MESSAGE TABLE*****
1230
1231 005724 022445 046524 026101 MSG1: .ASCII /%TMA,B-11 :TU10,N,W SUPPLEMENTAL INST TEST (DZTMF-D)#/
1232 005732 026502 030461 035040
1233 005740 052524 030061 047054
1234 005746 053454 051440 050125
1235 005754 046120 046505 047105
1236 005762 040524 020114 047111
1237 005770 052123 052040 051505
1238 005776 020124 042050 052132
1239 006004 043115 042055 021451
1240 006012 052445 044516 020124 MSG2: .ASCII /%UNIT NUMBER: #/
1241 006020 052516 041115 051105
1242 006026 020072 043
1243 006031 045 020445 047105 MSG3: .ASCII /%!END OF PASS: #/
1244 006036 020104 043117 050040
1245 006044 051501 035123 021440
1246 006052 020445 051041 053505 MSG4: .ASCII /%!!REWIND ERROR: NO BOT#/
1247 006060 047111 020104 051105
1248 006066 047522 035122 047040
1249 006074 020117 047502 021524
1250 006102 046445 051524 020072 MSG5: .ASCII /%MTS: #/
1251 006110 043
1252 006111 045 052115 035103 MSG6: .ASCII /%MTC: #/
1253 006116 021440
1254 006120 046445 041124 035103 MSG7: .ASCII /%MTBC: #/
1255 006126 021440
1256 006130 046445 041524 035101 MSG8: .ASCII /%MTCA: #/
1257 006136 021440
1258 006140 020445 041041 041501 MSG9: .ASCII /%!!BACKSPACE ERROR#/
1259 006146 051513 040520 042503
1260 006154 042440 051122 051117
1261 006162 043
1262 006163 045 020441 047516 MSG10: .ASCII /%!!NOT READY#/
1263 006170 020124 042522 042101
1264 006176 021531
1265 006200 020445 047041 020117 MSG11: .ASCII /%!!NO INTERRUPT#/
1266 006206 047111 042524 051122
1267 006214 050125 021524
1268 006220 042045 052101 020101 MSG12: .ASCII /%DATA ERROR#/
1269 006226 051105 047522 021522
1270 006234 041445 035116 021440 MSG13: .ASCII /%CN: #/
1271 006242 043445 020072 043 MSG14: .ASCII /%G: #/
1272 006247 045 035102 021440 MSG15: .ASCII /%B: #/
1273 006254 021477 MSG16: .ASCII /?#/
1274 006256 053445 044522 042524 MSG17: .ASCII /%WRITE ERROR#/
1275 006264 042440 051122 051117
1276 006272 043
1277 006273 045 042522 042101 MSG18: .ASCII /%READ ERROR#/
1278 006300 042440 051122 051117
1279 006306 043
1280 006307 045 047516 047440 MSG19: .ASCII /%NO OPI IN 35 FEET#/
1281 006314 044520 044440 020116
1282 006322 032463 043040 042505
1283 006330 021524
1284 006332 047445 044520 053440 MSG20: .ASCII /%OPI WITHIN 16 FEET#/

```

E03

TMA, B-11 SUPPLEMENTAL INST TEST MACY11 27(732) 04-NOV-76 11:15 PAGE 31
DZTMFD.SRC ACT11 HOOKS

1285	006340	052111	044510	020116	
1286	006346	033061	043040	042505	
1287	006354	021524			
1288	006356	020440	047041	052117	MSG21: .ASCII / !!NOT AVAILABLE#/
1289	006364	040440	040526	046111	
1290	006372	041101	042514	043	
1291	006377	045	040503	047116	MSG22: .ASCII /%CANNOT TEST LOAD MEDIUM!%!#/
1292	006404	052117	052040	051505	
1293	006412	020124	047514	042101	
1294	006420	046440	042105	052511	
1295	006426	020515	022441	043	
1296	006433	045	043536	043	\$CNTG: .ASCII /%IG#/
1297	006437	045	020445	051441	\$MSWR: .ASCII /%!!SWR= #/
1298	006444	051127	020075	043	
1299	006451	040	047040	053505	\$MNEW: .ASCII / NEW= #/
1300	006456	020075	043		
1301	006461	045	022477	021445	\$QUEST: .ASCII /%?%#/
1302					
1303					;↑TEST HEADER*****
1304					
1305	006466	022445	042524	052123	LT1MSG: .ASCII /%TEST 1: WRITE FROM ODD BYTE#/
1306	006474	030440	020072	051127	
1307	006502	052111	020105	051106	
1308	006510	046517	047440	042104	
1309	006516	041040	052131	021505	
1310	006524	022445	042524	052123	LT2MSG: .ASCII /%TEST 2: READ TO ODD BYTE#/
1311	006532	031040	020072	042522	
1312	006540	042101	052040	020117	
1313	006546	042117	020104	054502	
1314	006554	042524	043		
1315	006557	045	052045	051505	LT3MSG: .ASCII /%TEST 3: OPI TOO LONG#/
1316	006564	020124	035063	047440	
1317	006572	044520	052040	047517	
1318	006600	046040	047117	021507	
1319	006606	022445	042524	052123	LT4MSG: .ASCII /%TEST 4: OPI TOO SHORT#/
1320	006614	032040	020072	050117	
1321	006622	020111	047524	020117	
1322	006630	044123	051117	021524	
1323					.EVEN
1324					
1325	006636	177777			WDATA: -1
1326		006740			.=.+100
1327	006740	000000			RDATA: 0
1328		000001			.END

T3AD	000742	417#																
T3IAD	000744	418#																
T4AD	000746	419#																
T4IAD	000750	420#																
UDES	000610	366#	453*	454	469*	469*	470	486*	488*	737	771	804						
UNP	000714	403#																
WDATA	006636	568	577	597	614	622	642	665	710	1325#								
\$CNTG	006433	1183	1296#															
\$ENDAD	001660	319	544#															
\$MNEW	006451	1189	1299#															
\$MSWR	006437	1185	1297#															
\$GLEST	006461	1215	1301#															
\$READ	005524	1192	1195#															
\$SVPC =	001000	317#	322															
.	= 006742	312#	317	318#	320#	322#	332#	339#	345#	349#	355#	359#	423#	1326#				

COMMEN 1#
 ENDCOM 1#
 ESCAPE 1#
 GETPRI 1#
 GETSWR 1#
 MULT 1#
 NEWTST 1#
 POP 1#
 PUSH 1#
 REPORT 1#
 SETPRI 1#
 SETUP 1#
 SKIP 1#
 SLASH 1#
 STARS 1#
 SWRSU 1#
 TYPBIN 1#
 TYPDEC 1#
 TYPNAM 1#
 TYPNUM 1#
 TYPOCS 1#
 TYPOCT 1#
 TYPTXT 1#
 \$\$ESCA 1#
 \$\$NEWT 1#
 \$\$SKIP 1#
 .EQUAT 1#
 .HEADE 1#
 .KT11 1#
 .SETUP 1#
 .SWRHI 1#
 .SACT1 1#
 .SAPT8 1#
 .SAPTH 1#
 .SAPTY 1#
 .SASTA 1#
 .SCATC 1#
 .SCMTA 1#
 .SDB2D 1#
 .SDB20 1#
 .SDIV 1#
 .SEOP 1#
 .SERRO 1#
 .SERRT 1#
 .SMULT 1#
 .SPOWE 1#
 .SRAND 1#
 .SRDDE 1#
 .SRDOC 1#
 .SREAD 1#
 .SR2AZ 1#
 .SSAVE 1#
 .SSB2D 1#
 .SSB20 1#
 .SSCOP 1#
 .SSIZE 1#

315

295# 313

L03

TMA,B-11 SUPPLEMENTAL INST TEST MACY11 27(732) 04-NOV-76 11:15 PAGE 40
DZTMFD.SRC CROSS REFERENCE TABLE -- MACRO NAMES

.SSUPR	1*
.STRAP	1*
.STYPB	1*
.STYPD	1*
.STYPE	1*
.STYPO	1*
.S40CA	1*
.1170	1*

	705	719	728	736	765	770	782	787	803	810	830	850	889	898	924
	926	933	939	943	945	951	968	971	972	977	978	979			
RESET	543														
ROL	530	531	1015	1017	1019	1120	1121	1162							
ROR	1109	1110	1111	1112	1116	1125	1126	1127							
RTI	980														
RTS	766	783	831	846	849	936	942	956	973	1031	1038	1052	1082	1133	1141
	1147	1165	1171	1193											
SUB	1037														
SWAB	455	467	1113	1117	1122	1167									
TST	447	458	481	503	509	520	524	761	793	799	820	826	835	838	844
	853	872	882	902	908	927	929	944	957	962	1004	1097	1139	1201	1209
TSTB	741	774	1046	1049	1074	1079	1155	1177							
.ASCII	1231	1240	1243	1246	1250	1252	1254	1256	1258	1262	1265	1268	1270	1271	1272
	1273	1274	1277	1280	1284	1288	1291	1296	1297	1299	1301	1305	1310	1315	1319
.ENABL	1	294													
.END	1328														
.ENDC	316	320	322												
.EVEN	1323														
.IF	315	318	320												
.IFF	316	320	322												
.LIST	1	288	312												
.MACRO	1														
.MCALL	295														
.NLIST	1	288	312												
.REM	4														
.REPT	312														
.SBTTL	313														
.TITLE	290														
.WORD	321														

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*, DZTMFD.SEG/SOL/CRF/PAGNUM/NL: TOC=DZTMFD.SML, DZTMFD.SRC
 RUN-TIME: 23 27 2 SECONDS
 RUN-TIME RATIO: 117/52=2.2
 CORE USED: 32K (63 PAGES)

