

TMA-11

TMA,B-11 SPLMTL
CZTMFEO
INSTRUCTION

AH-9415E-MC

COPYRIGHT © 75-78

FICHE 1 OF 1

JUL 1978

digital

MADE IN USA

This microfiche card contains a grid of frames. The first column contains 16 frames, the second column contains 16 frames, and the third column contains 16 frames. Each frame contains technical data, including tables and diagrams. The data is too small to read clearly but appears to be organized into a structured format. The frames are arranged in a 3x16 grid.

IDENTIFICATION

PRODUCT CODE: AC-9414E-MC
PRODUCT NAME: CZTMFEO TMA,B-11 SPLMTL INSTR
PROGRAM DATE: JUNE 1978
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: R. B. BARNES
REVISED BY: RON PLATUKIS/R. SOLER/CLEM WALSH

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

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

1. ABSTRACT

THIS PROGRAM IS INTENDED TO BE USED IN ADDITION TO THE TMA,B-11 INSTRUCTION TEST (ZZ - CZTMA) 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

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

.TITLE CZTMFEO TMA,B-11 SPLMTL INSTR
:ZZ - CZTMFEO
:JUNE 76
:R. BARNES
:ENABLE ABS,AMA

: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*****

```
312  
313 : *****  
314 : MODIFIED JAN 10 1978  
315 :  
316 : ++  
317 : ACT11 AND XXDP HOOKS  
318 : --  
319 :  
320 001000 $SVPC= . ;SAVE PC  
321  
322 000040 000040  
323 000040 000 DRIVE: .=40 .BYTE 0 ;DRIVE # FOR XXDP LOAD MEDIUM  
324 ;ASSEMBLE AS A 0  
325  
326 000041 000041  
327 000041 000 MEDIUM: .=41 .BYTE 0 ;XXDP LOAD MEDIUM  
328 ;ASSEMBLE AS A 0  
329  
330 000042 000042  
331 000042 000000 .=42 .WORD 0 ;AUTO/MAN MODE INDICATOR  
332 ;ASSEMBLE AS A 0  
333  
334 000046 000046  
335 000046 001720 .=46 .WORD $ENDAD ;SET TO $ENDAD IN .SEOP  
336  
337 000052 000052  
338 000052 000000 .=52 .WORD 0 ;CHARACTERISTICS OF PROGRAM  
339 ;ASSEMBLE AS A 0  
340  
341 001000 .=$SVPC ;RESTORE PC  
342  
343 : *****  
344
```

```
345
346      : *****
347      :                               MODIFIED JAN 10 1978
348      :
349      : ++
350      :                               ACT11 AND XXDP MODE INDICATORS
351      : --
352
353 001000 000000 AUTOM: .WORD 0 ;AUTOMATIC MODE INDICATOR
354 001002 000 ACT11M: .BYTE 0 ;ACT11 AUTO MODE INDICATOR
355 001003 000 XXDPM: .BYTE 0 ;XXDP AUTO MODE INDICATOR
356 001004 000 ADUMPM: .BYTE 0 ;ACT11 DUMP MODE INDICATOR
357 001005 000 XDUMPM: .BYTE 0 ;XXDP DUMP MODE INDICATOR
358
359      : *****
360
```

361
362
363
364
365
366

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

```
367 ;TTY INTERRUPT VECTOR*****
368
369      000060 000060      .=60
370 000060 004450      TTINT ;TTY INTERRUPT HANDLER
371 000062 000000      0
372
373
374 ;SOFTWARE SWITCH REGISTER LOCATIONS*****
375
376      000174 000174      .=174
377 000174 000000      DISPREG:0
378 000176 000000      SWREG: 0
379
380
381 ;STARTING ADDRESS*****
382      000200 000200      .=200
383 000200 005000      CLR R0
384 000202 000137 001000      JMP START ;PROGRAM START
385
386      000210 000210      .=210
387 000210 012700 000001      MOV #1,R0
388 000214 000137 001006      JMP RSTART ;NO HEADER START
389
390 ;TMA,B-11 INTERRUPT VECTOR*****
391
392      000224 000224      .=224
393 000224 004430      MTINT ;TAPE INTERRUPT HANDLER
394 000226 000340      340
395
```

```
396          000600          . =600
397                                     ;CONSTANTS*****
398
399 000600 172520 MTS: 172520 ;TAPE STATUS REGISTER
400 000602 172522 MTC: 172522 ;TAPE COMMAND REGISTER
401 000604 172524 MTBC: 172524 ;TAPE BYTE COUNTER
402 000606 172526 MTBA: 172526 ;TAPE BUS ADDRESS
403 000610 000000 UDES: 0 ;UNIT DESCRIPTION(PRESET FOR UNIT 0)
404 000612 000020 RCNT: 20 ;RECORD COUNT
405 000614 177760 CCNT: -20 ;CHARACTER COUNT
406 000616 177776 PSW: 177776 ;PROCESSOR STATUS
407 000620 177570 SWR: 177570 ;CONSOLE SWITCH REGISTER
408 000622 177570 DISPLAY:177570 ;CONSOLE DISPLAY REGISTER
409 000624 177560 TKS: 177560 ;TTY READ STATUS
410 000626 177562 TKB: 177562 ;TTY READ BUFFER
411 000630 177564 TPS: 177564 ;TTY PUNCH STATUS
412 000632 177566 TPB: 177566 ;TTY OUTPUT BUFFER
413 000634 000010 ITAMT: 10 ;NUMBER OF ITERATIONS
414 000636 000004 STALL: 4 ;READY DELAY MULTIPLIER
415
```

416 ;FLAGS AND COUNTERS*****
417
418 000640 000000 TINF: 0
419 000642 000000 TOB: 0
420 000644 000000 TIB: 0
421 000646 000000 TEMP1: 0
422 000650 000000 TEMP2: 0
423 000652 000000 TEMP3: 0
424 000654 000000 EMADDR: 0
425 000656 000000 ERRAD: 0
426 000660 000000 LTADD: 0
427 000662 000000 ITRLP: 0
428 000664 000000 SPFLG: 0
429 000666 000000 STFLG: 0
430 000670 000000 PCNTR: 0
431 000672 000000 BADR: 0
432 000674 000000 BYTES: 0
433 000676 000000 SCNT: 0
434 000700 000000 FUN: 0
435 000702 000000 ITCNT: 0
436 000704 000000 CRCNT: 0
437 000706 000000 DERFL: 0
438 000710 000000 HDRFL: 0
439 000712 000000 PFLG: 0
440 000714 000000 UNP: 0
441 000716 000000 BCNT: 0
442 000720 000000 COUNT: 0
443 000722 000000 TEMPST: 0
444 000724 000000 RDSW: 0

445
446 ;TEST ENTRY TABLE*****
447
448 000726 000000 TSTTBL: 0
449 000730 000000 0
450 000732 001760 T1AD: LT1
451 000734 001760 T1IAD: LT1
452 000736 002206 T2AD: LT2
453 000740 002206 T2IAD: LT2
454 000742 002430 T3AD: LT3
455 000744 002526 T3IAD: LT3IT
456 000746 002660 T4AD: LT4
457 000750 002756 T4IAD: LT4IT
458 000752 000000 0
459

```
460          001000          .=1000
461          :*****
462          :PROGRAM START AND HOUSEKEEPING
463          :*****
464 001000 012737 177570 000620 START: MOV #177570,SWR ;PRESET TO CONSOLE SWITCHES
465 001006 012777 000340 177602 RSTART: MOV #340,@PSW ;SET PRIORITY
466 001014 012706 000500          MOV #500,SP ;SET STACK POINTER
467 001020 013746 000006          SUSWR: MOV @#6,-(SP) ;SAVE VECTORS
468 001024 013746 000004          MOV @#4,-(SP)
469 001030 012737 001050 000004          MOV #1$,@#4 ;SET UP FOR TIMEOUT
470 001036 022777 177777 177554          CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
471 001044 001402          BEQ 2$
472 001046 000407          BR 3$
473 001050 022626          1$: CMP (SP)+,(SP)+ ;ADJUST STACK
474 001052 012737 000176 000620 2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
475 001060 012737 000174 000622          MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
476 001066 012637 000004          3$: MOV (SP)+,@#4 ;RESTORE VECTORS
477 001072 012637 000006          MOV (SP)+,@#6
478 001076 122737 000004 000041          CMPB #4,@#41 ;TM-11 MAG TAPE?
479 001104 001006          BNE 4$ ;IF NO , BR
480 001106 012704 006616          MOV #MSG22,R4
481 001112 004737 004716          JSR PC,TTOUT ;LOADER TM-11 MAG TAPE
482 001116 000137 001642          JMP TEND ;END OP
483 001122 012777 010000 177452 4$: MOV #10000,@MTC ;POWER CLEAN
484 001130 004737 005764          JSR PC,CKMODE ;CHECK MODE OF OPERATION ++ C.W
485 001134 105737 001002          TSTB ACT11M ;ACT11 MODE? ++ C.W
486 001140 001066          BNE ST1 ;BRANCH - IF YES ++ C.W
487 001142 005700          TST R0 ;SEE IF SKIP HEADER
488 001144 001116          BNE ST4 ;IF SO: BR
489 001146 012704 006170          MOV #MSG1,R4
490 001152 004737 004716          JSR PC,TTOUT ;PRINT HEADER
491 001156 012704 006231          STO: MOV #MSG2,R4
492 001162 004737 004716          JSR PC,TTOUT ;REQUEST DRIVE NUMBER
493 001166 005037 000610          CLR UDES ;PRESET UNIT 0
494 001172 013703 000610          MOV UDES,R3 ;GET UNIT NUMBER
495 001176 000303          SWAB R3 ;POSITION
496 001200 042703 177770          BIC #177770,R3 ;MASK UNIT NUMBER
497 001204 004737 005114          JSR PC,OCTP ;PRINT CURRENT VALUE
498 001210 005737 000042          TST @#42 ;CHAIN MODE?
499 001214 001404          BEQ 1$ ;IF NO, BR
500 001216 012737 000176 000620          MOV #176,SWR ;STORE SWR
501 001224 000434          BR ST1
502 001226 012705 000652          1$: MOV #TEMP3,R5 ;SET SAVE LOCATION
503 001232 012701 000001          MOV #1,R1 ;SET SIZE OF ENTRY
504 001236 012702 000007          MOV #7,R2 ;SET UPPER LIMIT
505 001242 012703 000000          MOV #0,R3 ;SET LOWER LIMIT
506 001246 004737 004460          JSR PC,TTR ;GO GET UNIT NUMBER
507 001252 000337 000652          SWAB TEMP3 ;POSITION UNIT NUMBER
508 001256 042737 003400 000610          BIC #3400,UDES ;CLEAR OLD NUMBER
509 001264 053737 000652 000610          BIS TEMP3,UDES ;LOAD NEW NUMBER
510 001272 013777 000610 177302          MOV UDES,@MTC ;SELECT UNIT
511 001300 005000          CLR R0
512 001302 022737 000176 000620          CMF #SWREG,SWR
513 001310 001002          BNE ST1
514 001312 004737 005522          JSR PC,CNTLU
515 001316 032777 000100 177254 ST1: BIT #100,@MTS ;SEE IF SELECT REMOTE
```



```
537 ;TEST SCHEDULAR*****
538
539 001426 000240 TSCD: NOP
540 001430 005037 000666 CLR STFLG ;CLEAR SINGLE TEST FLAG
541 001434 017700 177160 MOV @SWR,R0 ;GET SWITCH REGISTER
542 001440 042700 177700 BIC #177700,R0 ;MASK TEST SELECT
543 001444 005700 TST R0 ;SEE IF SINGLE TEST SELECT
544 001446 001050 BNE STSCD ;IF SO: BR
545 001450 012737 000726 000660 MOV #TSTTBL,LTADD ;GET TABLE START
546 001456 062737 000004 000660 TSCD0: ADD #4,LTADD
547 001464 013737 000660 000662 MOV LTADD,IIRLP ;SET ITERATION ADDRESS
548 001472 062737 000002 000662 ADD #2,IIRLP
549 001500 005777 177154 TST @LTADD ;SEE IF END OF CYCLE
550 001504 001002 BNE TSCD1 ;IF NOT: BR
551 001506 000137 001642 JMP TEND ;GO TO END ROUTINE
552 001512 005037 000710 TSCD1: CLR HDRFL ;CLEAR HEADER FLAG
553 001516 017700 177136 MOV @LTADD,R0 ;GET TEST ADDRESS
554 001522 000110 JMP (R0) ;GO TO TEST
555 001524 004737 005450 TSCD2: JSR PC,CKSWR
556 001530 032777 002000 177062 BIT #2000,@SWR ;SEE IF HALT ON TEST
557 001536 001401 BEQ TSCD3 ;IF NOT: BR
558 001540 000000 HALT
559 001542 004737 005450 TSCD3: JSR PC,CKSWR ;GO TEST FOR G
560 001546 005737 000666 TST STFLG ;SEE IF SINGLE TEST
561 001552 001741 BEQ TSCD0 ;IF NOT: BR
562 001554 017700 177040 MOV @SWR,R0
563 001560 042700 177760 BIC #177760,R0 ;GET TEST NUMBER
564 001564 005700 TST R0 ;SEE IF ALL TESTS
565 001566 001717 BEQ TSCD ;IF SO: BR
566 001570 012737 000001 000666 STSCD: MOV #1,STFLG ;SET SINGLE TEST FLAG
567 001576 022700 000005 CMP #5,R0 ;SEE IF EXCEEDED TEST NUMBER
568 001602 003417 BLE TEND ;IF SO: BR
569 001604 000241 CLC
570 001606 006100 ROL R0
571 001610 006100 ROL R0 ;POSITION NUMBER
572 001612 012737 000726 000660 MOV #TSTTBL,LTADD ;GET START OF TABLE
573 001620 060037 000660 ADD R0,LTADD ;SET POINTER
574 001624 013737 000660 000662 MOV LTADD,IIRLP
575 001632 062737 000002 000662 ADD #2,IIRLP ;SET ITERATION ADDRESS
576 001640 000724 BR TSCD1 ;GO DO TEST
577 001642 105737 001002 TEND: TSTB ACT11M ;ACT11 MODE? ++ C.W
578 001646 001410 BEQ $DONE ;BRANCH - IF NO ++ C.W
579 001650 005237 000610 INC UDES ;STEP TO NEXT UNIT ++ C.W
580 001654 022737 000010 000610 CMP #10,UDES ;ALL UNITS TESTED? ++ C.W
581 001662 001402 BEQ $DONE ;BRANCH - IF YES ++ C.W
582 001664 000137 001316 JMP ST1 ;GO TEST NEXT UNIT ++ C.W
583 001670 012704 006250 $DONE: MOV #MSG3,R4
584 001674 004737 004716 JSR PC,TTOUT ;PRINT END OF PASS
585 001700 013703 000670 MOV PCNTR,R3
586 001704 004737 005114 JSR PC,OCTP ;PRINT PASS NUMBER
587 001710 013703 000042 MOV @#42,R3
588 001714 001405 BEQ HERE
589 001716 000005 RESET
590 001720 004713 $ENDAD: JSR PC,(R3)
591 001722 000240 NOP
592 001724 000240 NOP
```

```

593 001726 000240      NOP
594 001730 000240      HERE:  NOP
595 001732 032777 004000 176660  BIT    #4000,@SWR    ;SEE IF HALT ON PASS
596 001740 001001      BNE    TENDX        ;IF NOT: BR
597 001742 000000      HALT
598 001744 004737 005450      TENDX: JSR    PC,CKSWR    ;GO TEST FOR G
599 001750 005237 000670      INC    PCNTR        ;BUMP PASS COUNTER
600 001754 000137 001426      JMP    TSCD         ;RESTART
601
602
603      ;*****
604      ;TEST 1: WRITE FROM ODD BYTE
605      ;
606      ;THIS TEST WILL WRITE A SIX (6) BYTE RECORD
607      ;FROM AN ODD BYTE STARTING ADDRESS. THE RECORD
608      ;WILL BE READ BACK INTO AN EVEN STARTING ADDRESS
609      ;TO TEST FOR PROPER TRANSFER.
610      ;*****
611
612 001760 000240      LT1:  NOP
613 001762 012737 006705 000654  MOV    #LT1MSG,EMADDR ;SET HEADER
614 001770 012702 007056      MOV    #WDATA,R2     ;GET BUFFER START
615 001774 112722 000377      MOVVB  #377,(R2)+    ;INSERT BACKGROUND DATA
616 002000 005000      CLR    R0
617 002002 110022      LT1B: MOVVB  R0,(R2)+
618 002004 005200      INC    R0             ;LOAD WRITE BUFFER (0,1,2,3,4,5)
619 002006 022700 000006      CMP    #6,R0
620 002012 001373      BNE    LT1B
621 002014 004737 003066      JSR    PC,RWIND      ;GO REWIND
622 002020 012737 000004 000700  MOV    #4,FUN         ;SET WRITE FUNCTION CODE
623 002026 012737 007057 000672  MOV    #WDATA+1,BADR ;SET DATA POINTER
624 002034 012737 177772 000674  MOV    #-6,BYTES     ;SET SIZE OF RECORD
625 002042 012737 006475 000656  MOV    #MSG17,ERRAD  ;SET WRITE ERROR CODE
626 002050 004737 003344      JSR    PC,EXEC       ;GO EXECUTE COMMAND
627 002054 000240      LT1C: NOP
628 002056 004737 003574      JSR    PC,ERCHK      ;GO CHECK FOR STATUS ERROR
629 002062 012737 177777 000676  MOV    #-1,SCNT
630 002070 004737 003252      JSR    PC,BKSP       ;GO BACKSPACE ONE RECORD
631 002074 012702 007160      MOV    #RDATA,R2
632 002100 012700 000010      MOV    #10,R0
633 002104 012722 177777      LT1D: MOV    #-1,(R2)+   ;BACKGROUND READ BUFFER
634 002110 005300      DEC    R0
635 002112 001374      BNE    LT1D          ;DO ALL
636 002114 012737 000002 000700  MOV    #2,FUN         ;SET READ FUNCTION CODE
637 002122 012737 007160 000672  MOV    #RDATA,BADR   ;SET READ POINTER
638 002130 012737 177772 000674  MOV    #-6,BYTES     ;SET SIZE OF RECORD
639 002136 012737 006512 000656  MOV    #MSG18,ERRAD  ;SET READ ERROR CODE
640 002144 004737 003344      JSR    PC,EXEC       ;GO DO READ
641 002150 000240      LT1E: NOP
642 002152 004737 003574      JSR    PC,ERCHK      ;GO CHECK ERRORS
643 002156 012701 007057      MOV    #WDATA+1,R1   ;SET EXPT DATA POINTER
644 002162 012702 007160      MOV    #RDATA,R2    ;SET RCVD DATA POINTER
645 002166 012700 000006      MOV    #6,R0         ;SET SIZE OF RECORD
646 002172 004737 004062      JSR    PC,DCHK       ;GO CHECK DATA
647 002176 004737 004352      JSR    PC,ITER       ;GO SEE IF ITERATIONS
648 002202 000137 001524      JMP    TSCD2        ;RETURN TO SCHEDULAR
  
```

```
649  
650  
651  
652  
653  
654  
655  
656  
657  
658 002206 000240  
659 002210 012737 006743 000654  
660 002216 012702 007056  
661 002222 005000  
662 002224 110022  
663 002226 005200  
664 002230 022700 000006  
665 002234 001373  
666 002236 004737 003066  
667 002242 012737 000004 000700  
668 002250 012737 007056 000672  
669 002256 012737 177772 000674  
670 002264 004737 003344  
671 002270 000240  
672 002272 012737 006475 000656  
673 002300 004737 003574  
674 002304 012737 177777 000676  
675 002312 004737 003252  
676 002316 012702 007160  
677 002322 012700 000010  
678 002326 012722 177777  
679 002332 005300  
680 002334 001374  
681 002336 012737 000002 000700  
682 002344 012737 007161 000672  
683 002352 012737 177772 000674  
684 002360 004737 003344  
685 002364 000240  
686 002366 012737 006512 000656  
687 002374 004737 003574  
688 002400 012701 007056  
689 002404 012702 007161  
690 002410 012700 000006  
691 002414 004737 004062  
692 002420 004737 004352  
693 002424 000137 001524  
694
```

;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 R0
LT2B: MOVB R0,(R2)+ ;LOAD DATA PATTERN
INC R0 ;BUMP PATTERN
CMP #6,R0 ;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
JSR PC,BKSP ;GO BACKSPACE ONE RECORD
MOV #RDATA,R2 ;GET READ BUFFER POINTER
MOV #10,R0 ;SET SIZE
LT2D: MOV #-1,(R2)+ ;BACKGROUND POINTER
DEC R0 ;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,R0 ;SET SIZE OF RECORD
JSR PC,DCHK ;GO CHECK DATA
JSR PC,ITER ;GO SEE IF ITERATION
JMP TSCD2 ;RETURN TO SCHEDULAR

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
734
735
736
737
738
739

```

:*****
: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.
:*****

LT3:  NOP
      MOV #LT3MSG,EMADDR ;SET TEST HEADER
      MOV #151,R0 ;SET NUMBER OF WRITE IRG/BACKSPACE
      JSR PC,RWND ;GO REWIND UNIT
LT3A:  MOV #14,FUN ;SET WRITE IRG FUNCTION CODE
      MOV #WDATA,BADR ;SET BUS ADDRESS
      MOV #-20,BYTES ;SET SIZE OF RECORD
      JSR PC,EXEC ;GO EXECUTE COMMAND
LT3B:  MOV #MSG17,ERRAD ;SET ERROR CODE
      JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
      MOV #-1,SCNT
      JSR PC,BKSP ;GO BACKSPACE ONE RECORD
      DEC R0 ;SEE IF DONE ALL
      BNE LT3A ;IF NOT: BR
LT3IT: NOP
      JSR PC,RWND ;GO 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 #MSG19,ERRAD ;SET ERROR CODE
      JSR PC,EXEC ;GO EXECUTE COMMAND
LT3C:  NOP
      MOV #4,STALL ;RESET NORMAL STALL
      BIT #400,@MTS ;SEE IF BTE IS SET
      BNE LT3X ;IF SO: BR
      MOV #1,SPFLG ;SET NO BA PRINT FLAG
      JSR PC,ERPT ;GO PRINT ERROR
      CLR SPFLG ;RESET FLAG
LT3X:  MOV #2,ITAMT ;SET TO TWO (2) ITERATIONS
      JSR PC,ITER ;GO SEE IF ITERATION
      MOV #10,ITAMT ;RESET ITERATIONS
      JMP TSCD2 ;RETURN TO SCHEDULAR
    
```

```
740 ;*****
741 ;TEST 4: OPI TOO SHORT
742 ;
743 ;THIS TEST WILL ERASE APPROXIMATELY SIXTEEN (16) FEET
744 ;OF TAPE BY WRITING WITH IRG, BACKSPACING
745 ;ONE (1) RECORD AND REPEATING THIS SEQUENCE
746 ;50(10) TIMES. TAPE WILL REWIND AND BE READ
747 ;FORWARD. THE FIRST RECORD ON TAPE SHOULD BE
748 ;REACHED BEFORE OPI TIMES OUT.
749 ;*****
750
751 002660 000240 LT4: NOP
752 002662 012737 007025 000654 MOV #LT4MSG,EMADDR ;SET HEADER
753 002670 004737 003066 JSR PC,RWND ;GO REWIND
754 002674 012700 000062 MOV #62,R0 ;SET NUMBER OF WRITE IRG/BACKSPACES
755 002700 012737 000014 000700 LT4A: MOV #14,FUN ;SET WRITE IRG FUNCTION CODE
756 002706 012737 007056 000672 MOV #WDATA,BADR ;SET START OF WRITE BUFFER
757 002714 012737 177760 000674 MOV #-20,BYTES ;SET SIZE OF RECORD
758 002722 012737 006475 000656 MOV #MSG17,ERRAD ;SET ERROR CODE
759 002730 004737 003344 JSR PC,EXEC ;GO EXECUTE COMMAND
760 002734 004737 003574 LT4B: JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
761 002740 012737 177777 000676 MOV #-1,SCNT
762 002746 004737 003252 JSR PC,BKSP ;GO BACKSPACE ONE RECORD
763 002752 005300 DEC R0 ;SEE IF DONE ALL SEQUENCES
764 002754 001351 BNE LT4A ;IF NOT: BR
765 002756 000240 LT4IT: NOP
766 002760 004737 003066 JSR PC,RWND ;REWIND
767 002764 012737 000500 000636 MOV #500,STALL ;SET OPI STALL
768 002772 012737 007160 000672 MOV #RDATA,BADR ;SET START OF READ BUFFER
769 003000 012737 177760 000674 MOV #-20,BYTES ;SET SIZE OF RECORD
770 003006 012737 000002 000700 MOV #2,FUN ;SET READ FUNCTION CODE
771 003014 012737 006551 000656 MOV #MSG20,ERRAD ;SET ERROR CODE
772 003022 004737 003344 JSR PC,EXEC ;GO EXECUTE COMMAND
773 003026 004737 003574 LT4C: JSR PC,ERCHK ;GO CHECK FOR STATUS ERRORS
774 003032 000240 NOP
775 003034 012737 000004 000636 MOV #4,STALL ;RESET NORMAL STALL
776 003042 012737 000002 000634 MOV #2,ITAMT ;SET TO TWO (2) ITERATIONS
777 003050 004737 004352 JSR PC,ITER ;GO SEE IF ITERATIONS
778 003054 012737 000010 000634 MOV #10,ITAMT ;RESET ITERATIONS
779 003062 000137 001524 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

780                                     ;REWIND SUBROUTINE*****
781
782 003066 000240          RWND:  NOP
783 003070 013777 000610 175504    MOV      UDES,@MTC      ;SELECT UNIT
784 003076 032777 000040 175474    BIT      #40,@MTS     ;SEE IF AT BOT
785 003104 001056          BNE     RWNDXX      ;IF SO: BR
786 003106 052777 000017 175466    BIS     #17,@MTC     ;START REWIND
787 003114 105777 175462          1$:  TSTB     @MTC
788 003120 100375          BPL     1$              ;AWAIT CUR
789 003122 032777 000001 175450    RWND1: BIT      #1,@MTS     ;AWAIT TUR
790 003130 001774          BEQ     RWND1
791 003132 032777 000040 175440    BIT      #40,@MTS     ;SEE IF BOT SET
792 003140 001040          BNE     RWNDXX      ;IF SO: BR
793 003142 032777 020000 175450    BIT     #20000,@SWR   ;SEE IF PRINT ERROR
794 003150 001030          BNE     RWNDX       ;IF NOT: BR
795 003152 013704 000654          MOV     EMADDR,R4
796 003156 004737 004716          JSR     PC,TTOUT     ;PRINT HEADER
797 003162 012704 006271          MOV     #MSG4,R4
798 003166 004737 004716          JSR     PC,TTOUT     ;PRINT REWIND ERROR
799 003172 012704 006321          MOV     #MSG5,R4
800 003176 004737 004716          JSR     PC,TTOUT     ;PRINT MTS TAG
801 003202 017703 175372          MOV     @MTS,R3
802 003206 004737 005104          JSR     PC,OCTPE     ;PRINT MTS
803 003212 012704 006330          MOV     #MSG6,R4
804 003216 004737 004716          JSR     PC,TTOUT     ;PRINT MTC TAG
805 003222 017703 175354          MOV     @MTC,R3
806 003226 004737 005104          JSR     PC,OCTPE     ;PRINT MTC
807 003232 005777 175362          RWNDX: TST     @SWR     ;SEE IF HALT ON ERROR
808 003236 100001          BPL     RWNDXX      ;IF NOT: BR
809 003240 000000          HALT
810 003242 004737 005450          RWNDXX: JSR     PC,CKSWR ;GO TEST FOR G
811 003246 000240          NOP
812 003250 000207          RTS     PC          ;RETURN
813
814                                     ;BACKSPACE SUBROUTINE*****
815
816 003252 000240          BKSP:  NOP
817 003254 013777 000610 175320    MOV     UDES,@MTC     ;SELECT UNIT
818 003262 013777 000676 175314    MOV     SCNT,@MTBC    ;SET NUMBER OF RECORDS TO SPACE
819 003270 052777 000013 175304    BIS     #13,@MTC     ;START SPACE REVERSE
820 003276 105777 175300          1$:  TSTB     @MTC
821 003302 100375          BPL     1$              ;AWAIT CUR
822 003304 032777 000001 175266    BKSP1: BIT     #1,@MTS     ;AWAIT TUR
823 003312 001774          BEQ     BKSP1
824 003314 012737 000001 000664    MOV     #1,SPFLG     ;SET SPACE FLAG
825 003322 012737 006357 000656    MOV     #MSG9,ERRAD
826 003330 004737 003574          JSR     PC,ERCHK     ;GO CHECK FOR ERROR
827 003334 005037 000664          CLR     SPFLG       ;CLEAR SPACE FLAG
828 003340 000240          NOP
829 003342 000207          RTS     PC          ;RETURN
830

```

```

831 ;COMMAND EXECUTE SUBROUTINE*****
832
833 003344 000240 EXEC: NOP
834 003346 005005 CLR R5
835 003350 032777 000200 175224 EXEC0: BIT #200,@MTC ;SEE IF CUR
836 003356 001021 BNE EXEC2 ;IF SO: BR
837 003360 005305 DEC R5 ;SEE IF TIMED OUT
838 003362 001372 BNE EXEC0 ;IF NOT: BR
839 003364 005737 000710 TST HDRFL ;SEE IF DONE HEADER
840 003370 001004 BNE EXEC1 ;IF SO: BR
841 003372 013704 000654 MOV EMADDR,R4
842 003376 004737 004716 JSR PC,TTOUT ;ELSE PRINT HEADER
843 003402 012704 006402 EXEC1: MOV #MSG10,R4
844 003406 004737 004716 JSR PC,TTOUT ;PRINT NOT READY ERROR
845 003412 005777 175202 TST @SWR ;SEE IF HALT ON ERROR
846 003416 100001 BPL EXEC2 ;IF NOT: BR
847 003420 000000 HALT
848 003422 004737 005450 EXEC2: JSR PC,CKSWR ;GO TEST FOR G
849 003426 000240 NOP
850 003430 013777 000610 175144 MOV UDES,@MTC ;SELECT UNIT
851 003436 013777 000672 175142 MOV BADR,@MTBA ;SET BUS MEMORY ADDRESS
852 003444 013777 000674 175132 MOV BYTES,@MTC ;SET BYTE COUNT
853 003452 013701 000700 MOV FUN,R1 ;GET FUNCTION
854 003456 052701 000101 BIS #101,R1 ;SET IN GO BIT AND INTERRUPT ENABLE
855 003462 050177 175114 BIS R1,@MTC ;LOAD COMMAND+GO+IE
856 003466 000240 NOP
857 003470 005077 175122 CLR @PSW ;ALLOW INTERRUPTS
858 003474 013737 000636 000646 MOV STALL,TEMP1 ;SET READY STALL
859 003502 005001 CLR R1
860 003504 005301 EXEC3: DEC R1
861 003506 001376 BNE EXEC3 ;AWAIT INTERRUPT
862 003510 005337 000646 DEC TEMP1
863 003514 001373 BNE EXEC3
864 003516 032777 020000 175074 BIT #20000,@SWR ;SEE IF PRINT ERROR
865 003524 001013 BNE EXECX ;IF NOT: BR
866 003526 005737 000710 TST HDRFL ;SEE IF DONE HEADER
867 003532 001004 BNE EXEC4 ;IF SO: BR
868 003534 013704 000654 MOV EMADDR,R4
869 003540 004737 004716 JSR PC,TTOUT ;PRINT HEADER
870 003544 012704 006417 EXEC4: MOV #MSG11,R4
871 003550 004737 004716 JSR PC,TTOUT ;PRINT NO INTERRUPT MESSAGE
872 003554 005777 175040 EXECX: TST @SWR ;SEE IF HALT ON ERROR
873 003560 100001 BPL EXECXX ;IF NOT: BR
874 003562 000000 HALT
875 003564 004737 005450 EXECXX: JSR PC,CKSWR ;GO TEST FOR G
876 003570 000240 NOP
877 003572 000207 RTS PC ;RETURN TO CALLER
878
  
```



```
879 ;STATUS ERROR CHECK SUBROUTINE*****
880
881 003574 005777 175002 ERCHK: TST @MTC ;SEE IF ANY ERROR BITS
882 003600 100002 BPL ERCHK1 ;IF NOT: BR
883 003602 000137 003652 JMP ERPT ;ELSE PRINT ERROR
884 003606 005777 174772 ERCHK1: TST @MTBC ;SEE IF BYTE COUNT IS ZERO
885 003612 001402 BEQ ERCHK2 ;IF SO: BR
886 003614 000137 003652 JMP ERPT ;ELSE PRINT ERROR
887 003620 013703 000674 ERCHK2: MOV BYTES,R3
888 003624 005403 NEG R3
889 003626 063703 000672 ADD BADR,R3 ;SET EXPT BUS ADDRESS
890 003632 005737 000664 TST SPFLG ;SEE IF SPACE OPERATION
891 003636 001401 BEQ ERCHK3 ;IF NOT: BR
892 003640 000207 RTS PC
893 003642 020377 174740 ERCHK3: CMP R3,@MTBA ;SEE IF EXPT=RCVD
894 003646 001001 BNE ERPT ;IF NOT: BR
895 003650 000207 RTS PC ;ELSE EXIT
896 003652 000240 ERPT: NOP
897 003654 032777 020000 174736 BIT #20000,@SWR ;SEE IF SHOULD PRINT
898 003662 001067 BNE ERPTX ;IF NOT: BR
899 003664 005737 000710 TST HDRFL ;SEE IF DONE HEADER
900 003670 001006 BNE ERPT1 ;IF SO: BR
901 003672 013704 000654 MOV EMADDR,R4
902 003676 004737 004716 JSR PC,TTOUT ;ELSE PRINT HEADER
903 003702 005237 000710 INC HDRFL ;SET FLAG
904 003706 013704 000656 ERPT1: MOV ERRAD,R4
905 003712 004737 004716 JSR PC,TTOUT ;PRINT ERROR CODE
906 003716 012704 006321 MOV #MSG5,R4
907 003722 004737 004716 JSR PC,TTOUT ;PRINT MTS TAG
908 003726 017703 174646 MOV @MTS,R3
909 003732 004737 005104 JSR PC,OCTPE ;PRINT MTS
910 003736 012704 006330 MOV #MSG6,R4
911 003742 004737 004716 JSR PC,TTOUT ;PRINT MTC TAG
912 003746 017703 174630 MOV @MTC,R3
913 003752 004737 005104 JSR PC,OCTPE ;PRINT MTC
914 003756 012704 006337 MOV #MSG7,R4
915 003762 004737 004716 JSR PC,TTOUT ;PRINT BYTE COUNT TAG
916 003766 017703 174612 MOV @MTBC,R3
917 003772 004737 005114 JSR PC,OCTP ;PRINT BYTE COUNT
918 003776 005737 000664 TST SPFLG ;SEE IF PRINT BA
919 004002 001017 BNE ERPTX ;IF NOT: BR
920 004004 012704 006347 MOV #MSG8,R4
921 004010 004737 004716 JSR PC,TTOUT ;PRINT BUS ADDRESS TAG
922 004014 017703 174566 MOV @MTBA,R3
923 004020 004737 005114 JSR PC,OCTP ;PRINT CURRENT ADDRESS
924 004024 013703 000674 MOV BYTES,R3
925 004030 005403 NEG R3
926 004032 063703 000672 ADD BADR,R3
927 004036 004737 005114 JSR PC,OCTP ;PRINT EXPT ADDRESS
928 004042 005777 174552 ERPTX: TST @SWR ;SEE IF HALT ON ERROR
929 004046 100001 BPL ERPTXX ;IF NOT: BR
930 004050 000000 HALT
931 004052 004737 005450 ERPTXX: JSR PC,CKSWR ;GO TEST FOR G
932 004056 000137 004322 JMP SCOPE ;GO SEE IF SCOPE ON ERROR
```

```

933                                     ;DATA CHECK SUBROUTINE*****
934
935 004062 000240      DCHK:  NOP
936 004064 005037 000704  CLR      CRCNT      ;CLEAR COUNTER
937 004070 121112      DCHK0: CMPB      (R1),(R2) ;SEE IF EXPT DATA=RCVD DATA
938 004072 001007      BNE      DCHKE      ;IF NOT: BR
939 004074 005237 000704  DCHK1: INC      CRCNT      ;BUMP CHARACTER COUNTER
940 004100 122122      CMPB      (R1)+,(R2)+
941 004102 005300      DEC      R0          ;SEE IF DONE
942 004104 001371      BNE      DCHK0      ;IF NOT: BR
943 004106 000137 004262  JMP      DCHKX      ;ELSE GO TO EXIT ROUTINE
944 004112 000240      DCHKE: NOP
945 004114 012737 000001 000706  MOV      #1,DERFL ;SET ERROR FLAG
946 004122 032777 020000 174470  BIT      #20000,@SWR ;SEE IF PRINT ERROR
947 004130 001054      BNE      DCHKX      ;IF NOT: BR
948 004132 005737 000710  TST      HDRFL      ;SEE IF DONE HEADER
949 004136 001007      BNE      DCHKE1     ;IF SO: BR
950 004140 013704 000654  MOV      EMADDR,R4
951 004144 004737 004716  JSR      PC,TTOUT   ;PRINT HEADER
952 004150 012737 000001 000710  MOV      #1,HDRFL   ;SET HEADER FLAG
953 004156 012704 006437  DCHKE1: MOV      #MSG12,R4
954 004162 005737 000712  TST      PFLG      ;SEE IF PRINTED DATA ERROR TAG
955 004166 001004      BNE      DCHKE2     ;IF SO: BR
956 004170 005237 000712  INC      PFLG
957 004174 004737 004716  JSR      PC,TTOUT   ;ELSE PRINT DATA ERROR TAG
958 004200 017704 006453  DCHKE2: MOV      #MSG13,R4
959 004204 004737 004716  JSR      PC,TTOUT   ;PRINT CHAR NUMBER TAG
960 004210 013703 000704  MOV      CRCNT,R3
961 004214 004737 005114  JSR      PC,OCTP    ;PRINT CHAR NUMBER
962 004220 012704 006461  MOV      #MSG14,R4
963 004224 004737 004716  JSR      PC,TTOUT   ;PRINT GOOD TAG
964 004230 111103  MOVVB   (R1),R3
965 004232 004737 005342  JSR      PC,DOUT    ;PRINT GOOD CHARACTER
966 004236 012704 006466  MOV      #MSG15,R4
967 004242 004737 004716  JSR      PC,TTOUT   ;PRINT BAD TAG
968 004246 111203  MOVVB   (R2),R3
969 004250 004737 005342  JSR      PC,DOUT    ;PRINT BAD CHARACTER
970 004254 000240  NOP
971 004256 000137 004074  JMP      DCHK1      ;CONTINUE FOR ALL BYTES
972 004262 000240      DCHKX: NOP
973 004264 005737 000706  TST      DERFL      ;SEE IF ANY ERROR
974 004270 001404      BEQ      DCHKXX     ;IF NOT: BR
975 004272 005777 174322  TST      @SWR      ;SEE IF HALT ON ERROR
976 004276 100001      BPL      DCHKXX     ;IF NOT: BR
977 004300 000000  HALT
978 004302 004737 005450  DCHKXX: JSR      PC,CKSWR ;GO TEST FOR G
979 004306 000240  NOP
980 004310 005037 000712  CLR      PFLG      ;CLEAR PRINT FLAG
981 004314 005037 000706  CLR      DERFL     ;CLEAR DATA ERROR FLAG
982 004320 000207  RTS      PC         ;RETURN

```

```
983 ;SCOPE LOOP ON ERROR SUBROUTINE*****
984
985 004322 000240 SCOPE: NOP
986 004324 032777 040000 174266 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
987 004332 001001 BNE SCOPE1 ;IF SO: BR
988 004334 000207 RTS PC ;ELSE EXIT
989 004336 000240 SCOPE1: NOP
990 004340 005726 TST (SP)+ ;RESET STACK
991 004342 000240 NOP
992 004344 017703 174310 MOV @LTADD,R3
993 004350 000113 JMP (R3) ;LOOP ON ERROR
994
995 ;TEST ITERATION SUBROUTINE*****
996
997 004352 000240 ITER: NOP
998 004354 004737 005450 JSR PC,CKSWR
999 004360 032777 010000 174232 BIT #10000,@SWR ;SEE IF ITERATIONS
1000 004366 001403 BEQ ITER1 ;IF SO: BR
1001 004370 005037 000702 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
1002 004374 000207 RTS PC ;ELSE EXIT
1003 004376 005737 000670 ITER1: TST PCNTR ;SEE IF FIRST PASS
1004 004402 001772 BEQ ITER0 ;IF SO: BR
1005 004404 005237 000702 INC ITCNT ;BUMP COUNTER
1006 004410 023737 000702 000634 CMP ITCNT,ITAMT ;SEE IF DONE ALL
1007 004416 001764 BEQ ITER0 ;IF SO: BR
1008 004420 005726 TST (SP)+ ;RESET STACK
1009 004422 017700 174234 MOV @ITRLP,R0 ;SET ITERATION POINTER
1010 004426 000110 JMP (R0) ;GO ITERATE
1011
1012 ;MAG TAPE INTEPRUPT HANDLER*****
1013
1014 004430 000240 MTINT: NOP
1015 004432 022626 CMP (SP)+,(SP)+ ;RESET STACK POINTER
1016 004434 042777 000100 174140 BIC #100,@MTC ;CLEAR INTERRUPT ENABLE
1017 004442 000240 NOP
1018 004444 000240 NOP
1019 004446 000207 RTS PC ;RETURN
1020
1021 ;TTY INTERRUPT HANDLER*****
1022
1023 004450 000240 TTINT: NOP
1024 004452 000240 NOP
1025 004454 000240 NOP
1026 004456 000002 RTI
1027
```

```
1028 ;*****  
1029 ;TTY ENTRY SUBROUTINE:  
1030 ;  
1031 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION  
1032 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED  
1033 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND  
1034 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL  
1035 ; (0-7) AND MUST FALL WITHIN THE LIMITS SET BY  
1036 ;THE CALLING ROUTINE.  
1037 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,  
1038 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE  
1039 ;MAY BE REENTERED.  
1040 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND  
1041 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A  
1042 ;CARRIAGE RETURN  
1043 ;*****  
1044  
1045 004460 005037 000646 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
1046 004464 005000 CLR RO  
1047 004466 004737 004644 TTR0: JSR PC,TTIN ;GO READ CHARACTER  
1048 004472 122737 000215 000644 CMPB #215,TIB ;SEE IF CR  
1049 004500 001005 BNE TTR1 ;IF NOT: BR  
1050 004502 005737 000646 TST TEMP1 ;SEE IF FIRST CHARACTER  
1051 004506 001446 BEQ TTR5 ;IF SO: BR  
1052 004510 000137 004602 JMP TTR2 ;ELSE GO LOAD VALUE  
1053 004514 122737 000260 000644 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0  
1054 004522 101402 BLOS TTR1A ;IF NOT: BR  
1055 004524 000137 004626 JMP TINNER ;ELSE GO TO ERROR  
1056 004530 122737 000270 000644 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7  
1057 004536 101002 BHI TTR1B ;IF NOT: BR  
1058 004540 000137 004626 JMP TINNER ;ELSE GO TO ERROR  
1059 004544 005237 000646 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG  
1060 004550 000241 CLC  
1061 004552 006100 ROL RO  
1062 004554 000241 CLC  
1063 004556 006100 ROL RO ;SHIFT 3 LEFT  
1064 004560 000241 CLC  
1065 004562 006100 ROL RO  
1066 004564 042737 177770 000644 BIC #177770,TIB ;STRIP ASCII  
1067 004572 053700 000644 BIS TIB,RO ;LOAD CHARACTER  
1068 004576 005301 DEC R1 ;SEE IF DONE  
1069 004600 001332 BNE TTR0 ;IF NOT: BR  
1070 004602 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT  
1071 004604 101402 BLOS TTR3 ;IF OT: BR  
1072 004606 000137 004626 JMP TINNER ;ELSE GO TO ERROR  
1073 004612 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT  
1074 004614 101402 BLOS TTR4 ;IF NOT: BR  
1075 004616 000137 004626 JMP TINNER ;ELSE GO TO ERROR  
1076 004622 010015 TTR4: MOV RO,(R5) ;LOAD VALUE  
1077 004624 000207 TTR5: RTS PC ;EXIT  
1078
```

```

1079 ;TTY ENTRY ERROR SUBROUTINE*****
1080
1081 004626 012704 006473 TINER: MOV #MSG16,R4
1082 004632 004737 004716 JSR PC,TTOUT ;PRINT?
1083 004636 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
1084 004642 000207 RTS PC ;REDO VALUE ENTRY
1085
1086 ;TTY READ SUBROUTINE*****
1087
1088 004644 005077 173754 TTIN: CLR @TKS
1089 004650 005077 173752 CLR @TKB
1090 004654 005037 000644 CLR TIB
1091 004660 005277 173740 INC @TKS
1092 004664 105777 173734 TTIN1: TSTB @TKS
1093 004670 100375 BPL TTIN1
1094 004672 017737 173730 000644 MOV @TKB,TIB
1095 004700 105777 173724 TTIN2: TSTB @TPS
1096 004704 100375 BPL TTIN2
1097 004706 113777 000644 173716 MOVB TIB,@TPB
1098 004714 000207 RTS PC
1099
1100 ;TTY OUTPUT SUBROUTINE*****
1101
1102 004716 112437 000642 TTOUT: MOVB (R4)+,TOB
1103 004722 122737 000043 000642 CMPB #43,TOB
1104 004730 001452 BEQ TEX
1105 004732 122737 000045 000642 CMPB #45,TOB
1106 004740 001407 BEQ TCRLF
1107 004742 122737 000041 000642 CMPB #41,TOB
1108 004750 001443 BEQ TBELL
1109 004752 004737 005042 JSR PC,TOG
1110 004756 000757 BR TTOUT
1111 004760 112737 000015 000642 TCRLF: MOVB #15,TOB
1112 004766 004737 005042 JSR PC,TOG
1113 004772 012703 000004 MOV #4,R3
1114 004776 005037 000642 TCRLFA: CLR TOB
1115 005002 004737 005042 JSR PC,TOG
1116 005006 005303 DEC R3
1117 005010 001372 BNE TCRLFA ;DO FILLERS
1118 005012 112737 000012 000642 MOVB #12,TOB
1119 005020 004737 005042 JSR PC,TOG
1120 005024 105737 000724 TSTB RDSW
1121 005030 100401 BMI 1$
1122 005032 000731 BR TTOUT
1123 005034 005037 000724 1$: CLR RDSW
1124 005040 000406 BP TEX
1125 005042 105777 173562 TOG: TSTB @TPS
1126 005046 100375 BPL TOG
1127 005050 113777 000642 173554 MOVB TOB,@TPB
1128 005056 000207 RTS PC
1129 005060 012703 000002 TBELL: MOV #2,R3
1130 005064 012737 000007 000642 TBELA: MOV #7,TOB
1131 005072 004737 005042 JSR PC,TOG
1132 005076 005303 DEC R3
1133 005100 001371 BNE TBELA
1134 005102 000705 BR TTOUT

```

```

1135
1136                ;OCTAL OUTPUT SUBROUTINE*****
1137
1138 005104 012737 000001 005340 OCTPE: MOV    #1,OFL
1139 005112 000402                BR      OCTPE1
1140 005114 005037 005340        OCTP:  CLR    OFL                ;CLEAR FLAG FOR LEADING ZERO
1141 005120 010304                OCTPE1: MOV   R3,R4                ;SEE IF NUMBER IS ZERO
1142 005122 001007                BNE    OCTP0                ;IF NOT ZERO: BR
1143 005124 005737 005340        TST    OFL                ;SEE IF PRINT ALL 0
1144 005130 001004                BNE    OCTP0                ;IF SO: BR
1145 005132 004737 005320        JSR    PC,OCTPG1            ;ELSE PRINT ZERO
1146 005136 000137 005262        JMP    OCTP3                ;SPACE AND EXIT
1147 005142 032704 100000        OCTP0: BIT   #100000,R4        ;SEE IF MSD = 1
1148 005146 001406                BEQ    OCTP1                ;IF NOT: BR
1149 005150 012704 000001        MOV    #1,R4
1150 005154 004737 005276        JSR    PC,OCTPG            ;PRINT 1
1151 005160 000137 005172        JMP    OCTP2
1152 005164 005004                OCTP1: CLR    R4
1153 005166 004737 005276        JSR    PC,OCTPG            ;PRINT 0
1154 005172 010304                OCTP2: MOV   R3,R4
1155 005174 006004                ROR    R4
1156 005176 006004                ROR    R4
1157 005200 006004                ROR    R4                ;POSITION DIGIT
1158 005202 006004                ROR    R4
1159 005204 000304                SWAB   R4
1160 005206 004737 005276        JSR    PC,OCTPG            ;PRINT DIGIT 2
1161 005212 010304                MOV    R3,R4
1162 005214 006004                ROR    R4
1163 005216 000304                SWAB   R4
1164 005220 004737 005276        JSR    PC,OCTPG            ;PRINT DIGIT 3
1165 005224 010304                MOV    R3,R4
1166 005226 006104                ROL    R4
1167 005230 006104                ROL    R4
1168 005232 000304                SWAB   R4
1169 005234 004737 005276        JSR    PC,OCTPG            ;PRINT DIGIT 4
1170 005240 010304                MOV    R3,R4
1171 005242 006004                ROR    R4
1172 005244 006004                ROR    R4
1173 005246 006004                ROR    R4
1174 005250 004737 005276        JSR    PC,OCTPG
1175 005254 010304                MOV    R3,R4
1176 005256 004737 005276        JSR    PC,OCTPG            ;PRINT DIGIT 5
1177 005262 012737 000240 000642 OCTP3: MOV   #240,TOB
1178 005270 004737 005042        JSR    PC,TOG                ;PRINT SPACE
1179 005274 000207                RTS    PC                    ;EXIT
  
```

```

1180
1181                ;OCTAL PRINT SUBROUTINE*****
1182
1183 005276 042704 177770      OCTPG: BIC      #177770,R4
1184 005302 001004              BNE      OCTPG0
1185 005304 005737 005340      TST      OFL
1186 005310 001001              BNE      OCTPG0
1187 005312 000207              RTS      PC
1188 005314 005237 005340      OCTPG0: INC      OFL
1189 005320 052704 000260      OCTPG1: BIS      #260,R4
1190 005324 010437 000642      MOV      R4,TOB
1191 005330 004737 005042      JSR      PC,TOG
1192 005334 010304              MOV      R3,R4
1193 005336 000207              RTS      PC
1194 005340 000000      OFL:    0                ;FIRST CHAR FLAG
1195
1196                ;DATA CHARACTER OUTPUT SUBROUTINE*****
1197
1198 005342 005037 000642      DOUT:   CLR      TOB
1199 005346 012704 000010      MOV      #10,R4                ;SET NUMBER TO PRINT
1200 005352 110337 000642      MOV      R3,TOB
1201 005356 105777 173246      DOUT1:  TSTB    @TPS
1202 005362 100375              BPL      DOUT1
1203 005364 132737 000200 000642  BITB    #200,TOB
1204 005372 001404              BEQ      DOUT2
1205 005374 012777 000061 173230  MOV      #061,@TPB
1206 005402 000403              BR       DOUT3
1207 005404 012777 000060 173220  DOUT2:  MOV      #060,@TPB
1208 005412 006137 000642      DOUT3:  ROL      TOB
1209 005416 005304              DEC      R4
1210 005420 001356              BNE      DOUT1
1211 005422 000207              RTS      PC
1212 005424 013703 000652      DOUTD:  MOV      TEMP3,R3
1213 005430 000303              SWAB    R3
1214 005432 004737 005342      JSR      PC,DOUT
1215 005436 013703 000652      MOV      TEMP3,R3
1216 005442 004737 005342      JSR      PC,DOUT
1217 005446 000207              RTS      PC
1218
1219
1220                ;SUBROUTINE TO CHANGE CONTENTS OF SOFTWARE SWITCH REGISTER
1221 005450 022737 000176 000620  CKSWR:  CMP      #SWREG,SWR                ;SOFTWARE SWITCH REG PRESENT
1222 005456 001041              BNE      OUT                    ;NO, GET OUT
1223 005460 105777 173140      TSTB    @TKS                    ;YES, WAIT FOR
1224 005464 100036              BPL      OUT                    ;READY, GET CHARACTER
1225 005466 017737 173134 000644  MOV      @TKB,TIB                ;AND STRIP OFF
1226 005474 042737 177600 000644  BIC      #177600,TIB                ;THE GARBAGE
1227 005502 022737 000007 000644  CMP      #7,TIB                    ;IS IT A < G>
1228 005510 001024              BNE      OUT
1229 005512 012704 006652      MOV      #SCNTG,R4
1230 005516 004737 004716      JSR      PC,TTOUT
1231 005522 012704 006656      CNTLU: MOV      #SMSWR,R4
1232 005526 004737 004716      JSR      PC,TTOUT
1233 005532 017703 173062      MOV      @SWR,R3
1234 005536 004737 005104      JSR      PC,OCTPE
1235 005542 012704 006670      MOV      #SMNEW,R4

```

1236	005546	004737	004716			JSR	PC,TTOUT	
1237	005552	005037	000722			CLR	@#TEMPST	
1238	005556	004737	005564			JSR	PC,\$READ	
1239	005562	000207			OUT:	RTS	PC	;GO READ A LINE ;RETURN TO MAIN BODY OF PROGRAM
1240								
1241	005564	005037	000722		\$READ:	CLR	TEMPST	
1242	005570	012737	000007	000720		MOV	#7,COUNT	
1243	005576	004737	004644		1\$:	JSR	PC,TTIN	;GO READ A CHARACTER
1244	005602	042737	177600	000644		BIC	#177600,TIB	;STRIP OFF GARBAGE
1245	005610	122737	000025	000644		CMPB	#25,TIB	;IS IT A U?
1246	005616	001002				BNE	2\$;BRANCH IF NOT
1247	005620	005726			3\$:	TST	(SP)+	;POP THE STACK
1248	005622	000737				BR	CNTLU	;START OVER
1249	005624	122737	000015	000644	2\$:	CMPB	#15,TIB	;IS IT A <CR>?
1250	005632	001013				BNE	4\$;BRANCH IF NOT
1251	005634	012737	000200	000724		MOV	#200,RDSW	
1252	005642	004737	004760			JSR	PC,TCRLF	;ECHO IT WITH <LF>
1253	005646	022737	000007	000720		CMP	#7,COUNT	;WAS IT FIRST CHARACTER
1254	005654	001037				BNE	7\$;CHANGE SWR IF NOT FIRST ONE
1255	005656	005726			8\$:	TST	(SP)+	;POP THE STACK
1256	005660	000740				BR	OUT	;GET OUT
1257	005662	122737	000060	000644	4\$:	CMPB	#60,TIB	
1258	005670	003004				BGT	5\$	
1259	005672	122737	000067	000644		CMPB	#67,TIB	
1260	005700	002005				BGE	6\$	
1261	005702	012704	006700		5\$:	MOV	#\$QUEST,R4	
1262	005706	004737	004716			JSR	PC,TTOUT	
1263	005712	000742				BR	3\$;START OVER IF NOT LEGAL CHARACTER
1264	005714	006337	000722		6\$:	ASL	TEMPST	
1265	005720	006337	000722			ASL	TEMPST	
1266	005724	006337	000722			ASL	TEMPST	
1267	005730	142737	000060	000644		BICB	#60,TIB	;GET NITTY-GRITTY
1268	005736	153737	000644	000722		BISB	TIB,TEMPST	
1269	005744	005337	000720			DEC	COUNT	;ONLY WANT 6 DIGITS
1270	005750	001754				BEQ	5\$	
1271	005752	000711				BR	1\$	
1272	005754	013777	000722	172636	7\$:	MOV	TEMPST,@SWR	;CHANGE SWITCH REGISTER CONTENTS
1273	005762	000735				BR	8\$	
1274								


```

1275
1276 : *****
1277 :
1278 :
1279 : ++
1280 :
1281 : --
1282 :
1283 005764 005037 001000 CKMODE: CLR AUTOM ;INIT AUTOMATIC MODE INDICATOR
1284 005770 105037 001002 CLRB ACT11M ;INIT ACT11 AUTO MODE INDICATOR
1285 005774 105037 001003 CLRB XXDPM ;INIT XXDP AUTO MODE INDICATOR
1286 006000 105037 001004 CLRB ADUMPM ;INIT ACT11 DUMP MODE INDICATOR
1287 006004 105037 001005 CLRB XDUMPM ;INIT XXDP DUMP MODE INDICATOR
1288 006010 005737 000042 TST @#42 ;AUTO MODE?
1289 006014 001425 BEQ 2$ ;BRANCH -IF NO
1290 006016 005237 001000 INC AUTOM ;SET AUTO MODE INDICATOR
1291 006022 032737 020000 000052 BIT #20000,@#52 ;MANUAL INTERVENTION?
1292 006030 001402 BEQ 6$ ;BRANCH - IF NO
1293 006032 000137 006112 JMP ABORT ;ABORT THE PROGRAM
1294 006036 023737 000042 000046 6$: CMP @#42,@#46 ;ACT11 MODE?
1295 006044 001403 BEQ 1$ ;BRANCH - IF YES
1296 006046 105237 001003 INCB XXDPM ;INDICATE XXDP AUTO MODE
1297 006052 000416 BR 5$ ;AND EXIT
1298 006054 105237 001002 1$: INCB ACT11M ;INDICATE ACT11 AUTO MODE
1299 006060 012777 114000 172532 MOV #114000,@SWR ;SET SWITCH REGISTER
1300 006066 000410 BR 5$ ;AND EXIT
1301 006070 105737 000041 2$: TSTB @#41 ;MAN/MODE VIA ACT11/PAPER TAPE?
1302 006074 001003 BNE 3$ ;BRANCH - IF NOT
1303 006076 105237 001004 INCB ADUMPM ;INDICATE MAN/MODE VIA ACT11/PAPER TAPE
1304 006102 000402 BR 5$ ;AND EXIT
1305 006104 105237 001005 3$: INCB XDUMPM ;INDICATE MANUAL MODE VIA XXDP
1306 006110 000207 5$: RTS PC ;RETURN
1307
1308 : *****
1309

```

1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329

```
: *****  
:                               MODIFIED JAN 10 1978  
:                               *****  
:                               ++  
:                               CHECK FOR DUMP MODE OR AUTOMATIC/ACT11-XXDP MODE  
:                               --  
ABORT: RESET                               ;CLEAR THE WORLD  
        MOV #MSG0,R4                       ;GET THE ABORT MESSAGE  
        JSR PC,TIOUT                        ;PRINT ABORT MESSAGE  
        TSTB XXDPM                          ;XXDP AUTO MODE?  
        BEQ 1$                              ;BRANCH - IF NO  
        MOV @#42,R0                         ;GET MONITOR EXIT ADDRESS  
        CLR @#42                            ;USE AS ABORT FLAG  
        JSR PC,R0                           ;EXIT TO XXDP MONITOR  
1$:     BR .                                ;AND HANG  
: *****
```

CZTMFEO TMA,B-11 SPLMTL INSTR
CZTMFE.P11 23-JAN-78 10:22

MACY11 30A(1052) 23-JAN-78 10:23 ¹3
PAGE 34

SEQ 0034

1330 ;MESSAGE ABLE*****
1331 006146 022445 051120 043517 MSG0: .ASCII /%PROGRAM ABORTED#/
1332 006154 040522 020115 041101
1333 006162 051117 042524 021504
1334 006170 022445 055103 046524 MSG1: .ASCII /%CZTMFEO TMA,B-11 SPLMTL INSTR #/
1335 006176 042506 020060 046524
1336 006204 026101 026502 030461
1337 006212 051440 046120 052115
1338 006220 020114 047111 052123
1339 006226 020122 043
1340 006231 045 047125 052111 MSG2: .ASCII /%UNIT NUMBER: #/
1341 006236 047040 046525 042502
1342 006244 035122 021440
1343 006250 022445 042441 042116 MSG3: .ASCII /%END OF PASS: #/
1344 006256 047440 020106 040520
1345 006264 051523 020072 043
1346 006271 045 020441 042522 MSG4: .ASCII /%REWIND ERROR: NO BOT#/
1347 006276 044527 042116 042440
1348 006304 051122 051117 020072
1349 006312 047516 041040 052117
1350 006320 043
1351 006321 045 052115 035123 MSG5: .ASCII /%MTS: #/
1352 006326 021440
1353 006330 046445 041524 020072 MSG6: .ASCII /%MTC: #/
1354 006336 043
1355 006337 045 052115 041502 MSG7: .ASCII /%MTBC: #/
1356 006344 020072 043
1357 006347 045 052115 040503 MSG8: .ASCII /%MTCA: #/
1358 006354 020072 043
1359 006357 045 020441 040502 MSG9: .ASCII /%BACKSPACE ERROR#/
1360 006364 045503 050123 041501
1361 006372 020105 051105 047522
1362 006400 021522
1363 006402 020445 047041 052117 MSG10: .ASCII /%NOT READY#/
1364 006410 051040 040505 054504
1365 006416 043
1366 006417 045 020441 047516 MSG11: .ASCII /%NO INTERRUPT#/
1367 006424 044440 052116 051105
1368 006432 052522 052120 043
1369 006437 045 040504 040524 MSG12: .ASCII /%DATA ERROR#/
1370 006444 042440 051122 051117
1371 006452 043
1372 006453 045 047103 020072 MSG13: .ASCII /%CN: #/
1373 006460 043
1374 006461 045 035107 021440 MSG14: .ASCII /%G: #/
1375 006466 041045 020072 043 MSG15: .ASCII /%B: #/
1376 006473 077 043 MSG16: .ASCII /?#/
1377 006475 045 051127 052111 MSG17: .ASCII /%WRITE ERROR#/
1378 006502 020105 051105 047522
1379 006510 021522
1380 006512 051045 040505 020104 MSG18: .ASCII /%READ ERROR#/
1381 006520 051105 047522 021522
1382 006526 047045 020117 050117 MSG19: .ASCII /%NO OPI IN 35 FEET#/
1383 006534 020111 047111 031440
1384 006542 020065 042506 052105
1385 006550 043

1386	006551	045	050117	020111	MSG20:	.ASCII	/ZOPI WITHIN 16 FEET#/
1387	006556	044527	044124	047111			
1388	006564	030440	020066	042506			
1389	006572	052105	043				
1390	006575	040	020441	047516	MSG21:	.ASCII	/ !!NOT AVAILABLE#/
1391	006602	020124	053101	044501			
1392	006610	040514	046102	021505			
1393	006616	041445	047101	047516	MSG22:	.ASCII	/XCANNOT TEST LOAD MEDIUM!!%#/
1394	006624	020124	042524	052123			
1395	006632	046040	040517	020104			
1396	006640	042515	044504	046525			
1397	006646	020441	021445				
1398	006652	057045	021507		\$CNTG:	.ASCII	/X G#/
1399	006656	022445	020441	053523	\$MSWR:	.ASCII	/X%!!SWR= #/
1400	006664	036522	021440				
1401	006670	020040	042516	036527	\$MNEW:	.ASCII	/ NEW= #/
1402	006676	021440					
1403	006700	037445	022445	043	\$QUEST:	.ASCII	/X?X%#/
1404							
1405							
1406							
1407	006705	045	052045	051505	LT1MSG:	.ASCII	/X%TEST 1: WRITE FROM ODD BYTE#/
1408	006712	020124	035061	053440			
1409	006720	044522	042524	043040			
1410	006726	047522	020115	042117			
1411	006734	020104	054502	042524			
1412	006742	043					
1413	006743	045	052045	051505	LT2MSG:	.ASCII	/X%TEST 2: READ TO ODD BYTE#/
1414	006750	020124	035062	051040			
1415	006756	040505	020104	047524			
1416	006764	047440	042104	041040			
1417	006772	052131	021505				
1418	006776	022445	042524	052123	LT3MSG:	.ASCII	/X%TEST 3: OPI TOO LONG#/
1419	007004	031440	020072	050117			
1420	007012	020111	047524	020117			
1421	007020	047514	043516	043			
1422	007025	045	052045	051505	LT4MSG:	.ASCII	/X%TEST 4: OPI TOO SHORT#/
1423	007032	020124	035064	047440			
1424	007040	044520	052040	047517			
1425	007046	051440	047510	052122			
1426	007054	043					
1427		007056					.EVEN
1428							
1429	007056	177777			WDATA:	-1	
1430		007160					.=. +100
1431	007160	000000			RDATA:	0	
1432		000001					.END

ABORT	006112	1293	1318#										
ACT11M	001002	354#	485	577	1284*	1298*							
ADUMPM	001004	356#	1286*	1303*									
AUTOM	001000	353#	1283*	1290*									
BADR	000672	431#	623*	637*	668*	682*	711*	723*	756*	768*	851	889	926
BCNT	000716	441#											
BKSP	003252	630	675	717	762	816#							
BKSP1	003304	822#	823										
BYTES	000674	432#	624*	638*	669*	683*	712*	724*	757*	769*	852	887	924
CCNT	000614	405#											
CKMODE	005764	484	1283#										
CKSWR	005450	555	559	598	810	848	875	931	978	998	1221#		
CNTLU	005522	514	1231#	1248									
COUNT	000720	442#	1242*	1253	1269*								
CRCNT	000704	436#	936*	939*	960								
DCHK	004062	646	691	935#									
DCHKE	004112	938	944#										
DCHKE1	004156	949	953#										
DCHKE2	004200	955	958#										
DCHKX	004262	943	947	972#									
DCHKXX	004302	974	976	978#									
DCHKO	004070	937#	942										
DCHK1	004074	939#	971										
DERFL	000706	437#	945*	973	981*								
DISPLA	000622	408#	475*										
DISPRE	000174	377#	475										
DOUT	005342	965	969	1198#	1214	1216							
DOUTD	005424	1212#											
DOUT1	005356	1201#	1202	1210									
DOUT2	005404	1204	1207#										
DOUT3	005412	1206	1208#										
DRIVE	000040	323#											
EMADDR	000654	424#	613*	659*	707*	752*	795	841	868	901	950		
ERCHK	003574	628	642	673	687	715	760	773	826	881#			
ERCHK1	003606	882	884#										
ERCHK2	003620	885	887#										
ERCHK3	003642	891	893#										
ERPT	003652	733	883	886	894	896#							
ERPTX	004042	898	919	928#									
ERPTXX	004052	929	931#										
ERPT1	003706	900	904#										
ERRAD	000656	425#	625*	639*	672*	686*	714*	726*	758*	771*	825*	904	
EXEC	003344	626	640	670	684	713	727	759	772	833#			
EXECX	003554	865	872#										
EXECXX	003564	873	875#										
EXECO	003350	835#	838										
EXEC1	003402	840	843#										
EXEC2	003422	836	846	848#									
EXEC3	003504	860#	861	863									
EXEC4	003544	867	870#										
FUN	000700	434#	622*	636*	667*	681*	710*	725*	755*	770*	853		
HDRFL	000710	438#	552*	839	866	899	903*	948	952*				
HERE	001730	588	594#										
ITAMT	000634	413#	735*	737*	776*	778*	1006						
ITCNT	000702	435#	1001*	1005*	1006								
ITER	004352	647	692	736	777	997#							

ITERO	004370	1001#	1004	1007															
ITER1	004376	1000	1003#																
ITRLP	000662	427#	547*	548*	574*	575*	1009												
LTADD	000660	426#	545*	546*	547	549	553	572*	573*	574	992								
LT1	001760	450	451	612#															
LT1B	002002	617#	620																
LT1C	002054	627#																	
LT1D	002104	633#	635																
LT1E	002150	641#																	
LT1MSG	006705	613	1407#																
LT2	002206	452	453	658#															
LT2B	002224	662#	665																
LT2C	002270	671#																	
LT2D	002326	678#	680																
LT2E	002364	685#																	
LT2MSG	006743	659	1413#																
LT3	002430	454	706#																
LT3A	002450	710#	719																
LT3B	002476	714#																	
LT3C	002576	728#																	
LT3IT	002526	455	720#																
LT3MSG	006776	707	1418#																
LT3X	002634	731	735#																
LT4	002660	456	751#																
LT4A	002700	755#	764																
LT4B	002734	760#																	
LT4C	003026	773#																	
LT4IT	002756	457	765#																
LT4MSG	007025	752	1422#																
MEDIUM	000041	327#																	
MSG0	006146	1319	1331#																
MSG1	006170	489	1334#																
MSG10	006402	843	1363#																
MSG11	006417	870	1366#																
MSG12	006437	953	1369#																
MSG13	006453	958	1372#																
MSG14	006461	962	1374#																
MSG15	006466	966	1375#																
MSG16	006473	1081	1376#																
MSG17	006475	625	672	714	758	1377#													
MSG18	006512	639	686	1380#															
MSG19	006526	726	1382#																
MSG2	006231	491	1340#																
MSG20	006551	771	1386#																
MSG21	006575	519	1390#																
MSG22	006616	480	1393#																
MSG3	006250	583	1343#																
MSG4	006271	797	1346#																
MSG5	006321	799	906	1351#															
MSG6	006330	803	910	1353#															
MSG7	006337	914	1355#																
MSG8	006347	920	1357#																
MSG9	006357	825	1359#																
MTBA	000606	402#	851*	893	922														
MTBC	000604	401#	818*	852*	884	916													
MTC	000602	400#	483*	510*	783*	786*	787	805	817*	819*	820	835	850*	855*					

MTINT	004430	881	912	1016*																
MTS	000600	393	1014#																	
OCTP	005114	399#	515	524	730	784	789	791	801	822	908									
OCTPE	005104	497	586	917	923	927	961	1140#												
OCTPE1	005120	802	806	909	913	1138#	1234													
OCTPG	005276	1139	1141#																	
OCTPGO	005314	1150	1153	1160	1164	1169	1174	1176	1183#											
OCTPG1	005320	1184	1186	1188#																
OCTPO	005142	1145	1189#																	
OCTP1	005164	1142	1144	1147#																
OCTP2	005172	1148	1152#																	
OCTP3	005262	1151	1154#																	
OFL	005340	1146	1177#																	
OUT	005562	1138*	1140*	1143	1185	1188*	1194#													
PCNTR	000670	1222	1224	1228	1239#	1256														
PFLG	000712	430#	535*	585	599*	1003														
PSW	000616	439#	954	956*	980*															
RCNT	000612	406#	465*	857*																
RDATA	007160	404#																		
RDSW	000724	631	637	644	676	682	689	723	768	1431#										
RSTART	001006	444#	1120	1123*	1251*															
RWND	003066	388	465#																	
RWNDX	003232	621	666	709	721	753	766	782#												
RWNDXX	003242	794	807#																	
RWND1	003122	785	792	808	810#															
SCNT	000676	789#	790																	
SCOPE	004322	433#	629*	674*	716*	761*	818													
SCOPE1	004336	932	985#																	
SPFLG	000664	987	989#																	
STALL	000636	428#	732*	734*	824*	827*	890	918												
START	001000	414#	722*	729*	767*	775*	858													
STFLG	000666	384	464#																	
STSCD	001570	429#	540*	560	566*															
STO	001156	544	566#																	
ST1	001316	491#	522																	
ST2	001354	486	501	513	515#	518	582													
ST3	001374	516	524#																	
ST4	001402	525	528#																	
ST5	001414	488	527	529#																
SUSWR	001020	532#	534																	
SWR	000620	467#																		
SWREG	00017-	407#	464*	470	474*	500*	512	541	556	562	595	793	807	845						
TBELA	0050	864	872	897	928	946	975	986	999	1221	1233	1272*	1299*							
TBELL	00506	378#	474	512	1221															
TCRLF	004760	1130#	1133																	
TCRLFA	004776	1108	1129#																	
TEMPST	000722	1106	1111#	1252																
TEMP1	000646	1114#	1117																	
TEMP2	000650	443#	1237*	1241*	1264*	1265*	1266*	1268*	1272											
TEMP3	000652	421#	858*	862*	1045*	1050	1059*													
TEND	001642	422#																		
TENDX	001744	423#	502	507*	509	1212	1215													
TEX	005056	482	523	551	568	577#														
TIB	000644	596	598#																	
		1104	1124	1128#																
		420#	1048	1053	1056	1066*	1067	1090*	1094*	1097	1225*	1226*	1227	1244*						

. ABS. 007162 000

ERRORS DETECTED: 0

CZTMFE,CZTMFE.SEQ/CRF/SOL/NL:TOC=CZTMFE.SML/ML,CZTMFE.P11
RUN-TIME: 13.4 SECONDS
RUN-TIME RATIO: 55/5=9.9
CORE USED: 7K (13 PAGES)

DOCUMENT PAGES: 40