

# TMB-11

SPLMTL INSTRUCTION  
CZTMFF0

AH-9415F-MC

COPYRIGHT © 75-78  
FICHE 1 OF 1

JAN 1979

**digital**

MADE IN USA

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48
49	50	51	52
53	54	55	56
57	58	59	60
61	62	63	64
65	66	67	68
69	70	71	72
73	74	75	76
77	78	79	80
81	82	83	84
85	86	87	88
89	90	91	92
93	94	95	96
97	98	99	100

IDENTIFICATION

PRODUCT CODE: AC-9414F-MC  
PRODUCT NAME: CZTMFFO TMA,B-11 SPLMTL INSTR  
PROGRAM DATE: DEC 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 MAY 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

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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 (?) 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 UNI 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 C/JT  
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 (CNTRL 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.

THE PROGRAM IS FURTHER MODIFIED TO TEST 1-8 DEVICES IN ACT MODE.

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

%



294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315

.TITLE CZTMFFO TMA,B-11 SPLMTL INSTR  
:ZZ - CZTMFFO  
: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\*\*\*\*\*

```

316
317      : *****
318      :                               MODIFIED JAN 10 1978
319      :
320      : ++
321      :                               ACT11 AND XXDP HOOKS
322      : --
323
324      001000      $SVPC=.      ;SAVE PC
325
326      000040      000040
327      000040      000      DRIVE:  .=40      0      ;DRIVE # FOR XXDP LOAD MEDIUM
328      ;ASSEMBLE AS A 0
329
330      000041      000041
331      000041      000      MEDIUM:  .-41      0      ;XXDP LOAD MEDIUM
332      ;ASSEMBLE AS A 0
333
334      000042      000042
335      000042      000000      .-42      0      ;AUTO/MAN MODE INDICATOR
336      ;ASSEMBLE AS A 0
337
338      000046      000046
339      000046      002160      .=46      $ENDAD      ;SET TO $ENDAD IN .SEOP
340
341      000052      000052
342      000052      000000      .-52      0      ;CHARACTERISTICS OF PROGRAM
343      ;ASSEMBLE AS A 0
344
345      001000      .=$SVPC      ;RESTORE PC
346
347      : *****
348
349      000100      . 100
350
  
```

351  
352 :  
353 : \*\*\*\*\*  
354 :  
355 :  
356 :  
357 :  
358 :  
359 000100 000000 AUTOM: .WORD 0 ;AUTOMATIC MODE INDICATOR  
360 000102 000 ACT11M: .BYTE 0 ;ACT11 AUTO MODE INDICATOR  
361 000103 000 XXDPM: .BYTE 0 ;XXDP AUTO MODE INDICATOR  
362 000104 000 ADUMPM: .BYTE 0 ;ACT11 DUMP MODE INDICATOR  
363 000105 000 XDUMPM: .BYTE 0 ;XXDP DUMP MODE INDICATOR  
364 000106 000000 ACTCNR: .WORD 0 ;ACT MODE PASS COUNTER  
365 000110 000000 TBL: .WORD 0 ;OFFLINE DEVICE POINTER  
366 000112 000000 TEMP4: .WORD 0 ;TEMPO DEVICE POINTER  
367 :  
368 : \*\*\*\*\*  
369 :  
:

370  
371  
372  
373  
374  
375

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

```
376 ;TTY INTERRUPT VECTOR*****
377
378 . =60
379 000060 000060 ;TTY INTERRUPT HANDLER
380 000062 000000
381
382 ;SOFTWARE SWITCH REGISTER LOCATIONS*****
383
384 . =174
385 000174 000174
386 000174 000000
387 000176 000000
388
389 ;STARTING ADDRESS*****
390 . =200
391 000200 000200
392 000200 005000
393 000202 000137 001000
394
395 . =210
396 000210 012700 000001
397 000214 000137 001012
398
399 ;TMA,B-11 INTERRUPT VECTOR*****
400
401 . =224
402 000224 004704 ;TAPE INTERRUPT HANDLER
403 000226 000340
404
```

405 000600  
406  
407  
408 000600 172520  
409 000602 172522  
410 000604 172524  
411 000606 172526  
412 000610 000000  
413 000612 000020  
414 000614 177760  
415 000616 177776  
416 000620 177570  
417 000622 177570  
418 000624 177560  
419 000626 177562  
420 000630 177564  
421 000632 177566  
422 000634 000010  
423 000636 000004  
424

. =600  
: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

```
425 ;FLAGS AND COUNTERS*****
426
427 000640 000000 TINF: 0
428 C 0642 000000 TOB: 0
429 000644 000000 TIB: 0
430 000646 000000 TEMP1: 0
431 000650 000000 TEMP2: 0
432 000652 000000 TEMP3: 0
433 000654 000000 EMADDR: 0
434 000656 000000 ERRAD: 0
435 000660 000000 LTADD: 0
436 000662 000000 ITRLP: 0
437 000664 000000 SPFLG: 0
438 000666 000000 STFLG: 0
439 000670 000000 PCNTR: 0
440 000672 000000 BADR: 0
441 000674 000000 BYTES: 0
442 000676 000000 SCNT: 0
443 000700 000000 FUN: 0
444 000702 000000 ITCNT: 0
445 000704 000000 CRCNT: 0
446 000706 000000 DERFL: 0
447 000710 000000 HDRFL: 0
448 000712 000000 PFLG: 0
449 000714 000000 UNP: 0
450 000716 000000 BCNI: 0
451 000720 000000 COUNT: 0
452 000722 000000 TEMPST: 0
453 000724 000000 RDSW: 0
```

```
454
455 ;TEST ENTRY TABLE*****
456
457
458 000726 000000 TSTTBL: 0
459 000730 000000 0
460 000732 002234 T1AD: LT1
461 000734 002234 T1IAD: LT1
462 000736 002462 T2AD: LT2
463 000740 002462 T2IAD: LT2
464 000742 002704 T3AD: LT3
465 000744 003002 T3IAD: LT3IT
466 000746 003134 T4AD: LT4
467 000750 003232 T4IAD: LT4IT
468 000752 000000 0
469
```

```

470          001000          .-1000
471          :*****
472          :PROGRAM START AND HOUSEKEEPING
473          :*****
474 001000 012737 177570 000620 START: MOV #177570,SWR ;PRESET TO CONSOLE SWITCHES
475 001006 005037 000106          CLR ACTCNR ;INIT PASS COUNTER ++ C.W
476 001012 012777 000340 177576 RSTART: MOV #340,@PSW ;SET PRIORITY
477 001020 012706 000500          MOV #500,SP ;SET STACK POINTER
478 001024 013746 000006          SUSWR: MOV @#6,-(SP) ;SAVE VECTORS
479 001030 013746 000004          MOV @#4,-(SP)
480 001034 012737 001054 000004 MOV #1$,@#4 ;SET UP FOR TIMEOUT
481 001042 022777 177777 177550 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
482 001050 001402          BEQ 2$
483 001052 000412          BR 3$
484 001054 022626          1$: CMP (SP)+,(SP)+ ;ADJUST STACK
485 001056 012737 000176 000620 2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
486 001064 012777 004000 177526 MOV #4000,@SWR ;DEFAULT TO MULTIPLE PASSES ++ C.W
487 001072 012737 000174 000622 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
488 001100 012637 000004          3$: MOV (SP)+,@#4 ;RESTORE VECTORS
489 001104 012637 000006          MOV (SP)+,@#6
490 001110 122737 000004 000041 CMPB #4,@#41 ;TM-11 MAG TAPE?
491 001116 001006          BNE 4$ ;IF NO , BR
492 001120 012704 007014          MOV #MSG22,R4
493 001124 004737 005172          JSR PC,TTOUT ;LOADER TM-11 MAG TAPE
494 001130 000137 002066          JMP TEND ;END OP
495 001134 004737 006240          4$: JSR PC,CKMODE ;CHECK FOR MODE OF OPERATION ++ C.W
496 001140 105737 000102          TSTB ACT11M ;ACT11 MODE? ++ C.W
497 001144 001051          BNE $ACT ;BRANCH - IF YES ++ C.W
498 001146 012777 010000 177426 MOV #10000,@MTC ;POWER CLEAN
499 001154 005700          TST R0 ;SKIP HEADER? ++ C.W
500 001156 001402          BEQ 5$ ;BRANCH - IF NO ++ C.W
501 001160 000137 001620          JMP ST4 ;GO DO TEST ++ C.W
502 001164 012704 006360          5$: MOV #MSG1,R4 ;GET HEADER MESSAGE ++ C.W
503 001170 004737 005172          JSR PC,TTOUT ;PRINT HEADER
504 001174 012704 006421          STO: MOV #MSG2,R4 ;GET REQUEST MESSAGE ++ C.W
505 001200 004737 005172          JSR PC,TTOUT ;REQUEST DRIVE NUMBER
506 001204 005037 000610          CLR UDES ;PRESET UNIT 0
507 001210 013703 000610          MOV UDES,R3 ;GET UNIT NUMBER
508 001214 000303          SWAB R3 ;POSITION
509 001216 042703 177770          BIC #177770,R3 ;MASK UNIT NUMBER
510 001222 004737 005370          JSR PC,OCTP ;PRINT CURRENT VALUE
511 001226 005737 000042          TST @#42 ;CHAIN MODE?
512 001232 001404          BEQ 1$ ;IF NO, BR
513 001234 012737 000176 000620 MOV #176,SWR ;STORE SWR
514 001242 000503          BR ST1
515 001244 012705 000652          1$: MOV #TEMP3,R5 ;SET SAVE LOCATION
516 001250 012701 000001          MOV #1,R1 ;SET SIZE OF ENTRY
517 001254 012702 000007          MOV #7,R2 ;SET UPPER LIMIT
518 001260 012703 000000          MOV #0,R3 ;SET LOWER LIMIT
519 001264 004737 004734          JSR PC,ITR ;GO GET UNIT NUMBER
520 001270 105737 000102          $ACT: TSTB ACT11M ;ACT MODE? ++ C.W
521 001274 001421          BEQ NOACT ;BRANCH IF NO ++ C.W
522 001276 005037 000610          CLR UDES ;INIT DEVICE POINTER ++ C.W
523 001302 005037 000652          CLR TEMP3 ;INIT TEMPO LOCATION ++ C.W
524 001306 005037 000112          CLR TEMP4 ;INIT ++ C.W
525 001312 005037 000110          CLR TBL ;INIT OFFLINE DEVICE POINTER ++ C.W

```



```

526 001316 122737 000010 000610 ST:  CMPB  #10,UDES ;ALL UNITS DONE? ++ C.W
527 001324 001002          BNE  1$ ;BRANCH - IF NO ++ C.W
528 001326 000137 002112          JMP  $DONE ;DO END OF PASS ++ C.W
529 001332 113737 000610 000652 1$:  MOVB  UDES,TEMP3 ;GET UNIT NUMBER ++ C.W
530 001340 000337 000652          NOACT: SWAB  TEMP3 ;POSITION UNIT NUMBER
531 001344 105037 000652          CLRB  TEMP3 ;GET RID OF GARBAGE ++ C.W
532 001350 042737 003400 000610 BIC  #3400,UDES ;CLEAR OLD NUMBER
533 001356 053737 000652 000610 BIS  TEMP3,UDES ;LOAD NEW NUMBER
534 001364 013737 000610 000112 MOV  UDES,TEMP4 ;SAVE NUMBER ++ C.W
535 001372 105037 000112          CLRB  TEMP4 ;KEEP BITS 8,9,10 ++ C.W
536 001376 152737 000017 000112 BISB #17,TEMP4 ;SET REWIND COMMAND ++ C.W
537 001404 012777 010000 177170 MOV  #10000,@MTC ;POWER CLEAN
538 001412 013777 000112 177162 MOV  TEMP4,@MTC ;SELECT UNIT
539 001420 005000          CLR  R0
540 001422 105737 000102          TSTB  ACT11M ;ACT MODE? ++ C.W
541 001426 001011          BNE  ST1 ;BRANCH - IF YES ++ C.W
542 001430 022737 000176 000620 CMP  #SWREG,SWR
543 001436 001005          BNE  ST1
544 001440 005737 000042          TST  @#42 ;AUTO MODE? ++ C.W
545 001444 001002          BNE  ST1 ;BRANCH - IF YES ++ C.W
546 001446 004737 005776          JSR  PC,CNTLU
547 001452 032777 000100 177120 ST1: BIT  #100,@MTC ;SEE IF SELECT REMOTE
548 001460 001030          BNE  ST2 ;IF S0: BR
549 001462 005300          DEC  R0
550 001464 001372          BNE  ST1 ;DELAY FOR SELECT REMOTE
551 001466 105737 000102          TSTB  ACT11M ;ACT MODE? ++ C.W
552 001472 001412          BEQ  1$ ;BRANCH - IF NO ++ C.W
553 001474 105237 000610          INCB  UDES ;STEP TO NEXT UNIT ++ C.W
554 001500 105237 000110          INCB  TBL ;OFFLINE UNIT COUNT ++ C.W
555 001504 122737 000010 000110 CMPB  #10,TBL ;ALL UNITS DONE? ++ C.W
556 001512 001301          BNE  ST ;BRANCH - IF NO ++ C.W
557 001514 005037 000110          CLR  TBL ;REINIT OFFLINE UNIT COUNT ++ C.W
558 001520 012704 006764          1$:  MOV  #MSG21,R4
559 001524 004737 005172          JSR  PC,TOUT ;PRINT NOT AVAILABLE
560 001530 005737 000042          TST  @#42 ;SEE IF CHAIN MODE
561 001534 001617          BEQ  ST0 ;IF NO, BR
562 001536 000137 002112          JMP  $DONE ;END OP
563 001542 105737 000102          ST2: TSTB  ACT11M ;ACT MODE? ++ C.W
564 001546 001411          BEQ  1$ ;BRANCH - IF NO ++ C.W
565 001550 032777 000040 177022 2$:  BIT  #40,@MTC ;BOT? ++ C.W
566 001556 001774          BEQ  2$ ;BRANCH - IF NO ++ C.W
567 001560 013737 000610 000112 MOV  UDES,TEMP4 ;SAVE OLD DEVICE NUMBER ++ C.W
568 001566 105037 000610          CLRB  UDES ;KEEP BITS 8,9,10 ++ C.W
569 001572 032777 000020 177000 1$:  BIT  #20,@MTC ;SEE IF 7 CHANNEL
570 001600 001404          BEQ  ST3 ;IF NOT: BR
571 001602 052737 040000 000610 BIS  #40000,UDES ;SET TO 800 BPI 7 CHAN
572 001610 000403          BR   ST4
573 001612 052737 060000 000610 ST3: BIS  #60000,UDES ;SET TO 800 BPI 9 CHAN
574 001620 000240          ST4: NOP
575 001622 012702 000642          MOV  #TOB,R2 ;GET START OF TABLE
576 001626 012700 000027          MOV  #27,R0 ;SET SIZE OF TABLE
577 001632 005022          ST5: CLR  (R2)+ ;CLEAR TABLE
578 001634 005300          DEC  R0
579 001636 001375          BNE  ST5 ;DO ALL
580 001640 105737 000102          TSTB  ACT11M ;ACT MODE? ++ C.W
581 001644 001002          BNE  TSCD ;BRANCH - IF YES ++ C.W
  
```

CZTMFFO TMA,B-11 SPLMTL INSTR MACY11 30A(1052) 16-AUG-78 08:51<sup>E 2</sup> PAGE 17  
CZTMFF.P11 16-AUG-78 08:48

SEQ 0017

582 001646 005037 000670  
583

CLR PCNTR

:CLEAR PASS COUNTER

```

584                                     :TEST SCHEDULAR*****
585
586 001652 000240          TSCD:  NOP
587 001654 005037 000666    CLR      STFLG      ;CLEAR SINGLE TEST FLAG
588 001660 017700 176734    MOV      @SWR,RO    ;GET SWITCH REGISTER
589 001664 042700 177700    BIC      #177700,RO ;MASK TEST SELECT
590 001670 005700          TST      RO         ;SEE IF SINGLE TEST SELECT
591 001672 001050          BNE      STSCD     ;IF SO: BR
592 001674 012737 000726 000660 MOV      #TSTTBL,LTADD ;GET TABLE START
593 001702 062737 000004 000660 TSCD0:  ADD      #4,LTADD
594 001710 013737 000660 000662 MOV      LTADD,IIRLP ;SET ITERATION ADDRESS
595 001716 062737 000002 000662 ADD      #2,IIRLP
596 001724 005777 176730    TST      @LTADD    ;SEE IF END OF CYCLE
597 001730 001002          BNE      TSCD1    ;IF NOT: BR
598 001732 000137 002066    JMP      TEND      ;GO TO END ROUTINE
599 001736 005037 000710    TSCD1:  CLR      HDRFL  ;CLEAR HEADER FLAG
600 001742 017700 176712    MOV      @LTADD,RO ;GET TEST ADDRESS
601 001746 000110          JMP      (RO)      ;GO TO TEST
602 001750 004737 005724          TSCD2:  JSR      PC,CKSWR
603 001754 032777 002000 176636 BIT      #2000,@SWR ;SEE IF HALT ON TEST
604 001762 001401          BEQ      TSCD3    ;IF NOT: BR
605 001764 000000          HALT
606 001766 004737 005724          TSCD3:  JSR      PC,CKSWR ;GO TEST FOR ^G
607 001772 005737 000666    TST      STFLG    ;SEE IF SINGLE TEST
608 001776 001741          BEQ      TSCD0    ;IF NOT: BR
609 002000 017700 176614    MOV      @SWR,RO
610 002004 042700 177760    BIC      #177760,RO ;GET TEST NUMBER
611 002010 005700          TST      RO         ;SEE IF ALL TESTS
612 002012 001717          BEQ      TSCD    ;IF SO: BR
613 002014 012737 000001 000666 STSCD:  MOV      #1,STFLG ;SET SINGLE TEST FLAG
614 002022 022700 000005    CMP      #5,RO    ;SEE IF EXCEEDED TEST NUMBER
615 002026 003417          BLE      TEND     ;IF SO: BR
616 002030 000241          CLC
617 002032 006100          ROL      RO
618 002034 006100          ROL      RO         ;POSITION NUMBER
619 002036 012737 000726 000660 MOV      #TSTTBL,LTADD ;GET START OF TABLE
620 002044 060037 000660    ADD      RO,LTADD ;SET POINTER
621 002050 013737 000660 000662 MOV      LTADD,IIRLP
622 002056 062737 000002 000662 ADD      #2,IIRLP
623 002064 000724          BR      TSCD1    ;GO DO TEST
624 002066 105737 000102          TEND:  TSTB     ACT11M ;ACT11 MODE? ++ C.W
625 002072 001407          BEQ      $DONE   ;BRANCH - IF NO ++ C.W
626 002074 113737 000112 000610 MOVB     TEMP4,UDES ;RESTORE DRIVE NUMBER ++ C.W
627 002102 105237 000610    INCB     UDES    ;GET NEXT UNIT ++ C.W
628 002106 000137 001316    JMP      ST      ;GO TEST NEXT UNIT ++ C.W
629 002112 012704 006440          $DONE:  MOV      #MSG3,R4
630 002116 004737 005172    JSR      PC,TOUT  ;PRINT END OF PASS
631 002122 105737 000102    TSTB     ACT11M  ;ACT MODE: ++ C.W
632 002126 001404          BEQ      1$     ;BRANCH - NO ++ C.W
633 002130 013703 000106    MOV      ACTCNR,R3 ;GET ACT PASS COUNT ++ C.W
634 002134 004737 005370    JSR      PC,OCTP ;PRINT PASS NUMBER
635 002140 013703 000670          1$:   MOV      PCNTR,R3 ;GET PASS NUMBER ++ C.W
636 002144 004737 005370    JSR      PC,OCTP ;PRINT PASS NUMBER ++ C.W
637 002150 013703 000042    MOV      @#4?,R3
638 002154 001405          BEQ      HERE
639 002156 000005          RESET
  
```

```

640 002160 004713          SENDAD: JSR      PC,(R3)
641 002162 000240          NOP
642 002164 000240          NOP
643 002166 000240          NOP
644 002170 032777 004000 176422  HERE·  BIT      #4000,@SWR      ;SEE IF HALT ON PASS
645 002176 001001          BNE     TENDX      ;IF NOT: BR
646 002200 000000          HALT
647 002202 105737 000102  TENDX: TSTB     ACT11M      ;ACT MODE? ++ C.W
648 002206 001006          BNE     1$         ;BRANCH - IF YES ++ C.W
649 002210 004737 005724          JSR     PC,CKSWR    ;GO TEST FOR ^G
650 002214 005237 000670          INC     PCNTR      ;BUMP PASS COUNTER
651 002220 000137 001652          JMP     TSCD       ;RESTART
652 002224 005237 000106          1$:    INC     ACTCNR  ;BUMP PASS COUNTER IN ACT MODE ++ C.W
653 002230 000137 001270          JMP     $ACT       ;RESTART IN ACT MODE ++ C.W
654
655
656
657
658
659
660
661
662
663
664
665 002234 000240          LT1:   NOP
666 002236 012737 007103 000654  MOV     #LT1MSG,EMADDR ;SET HEADER
667 002244 012702 007254          MOV     #WDATA,R2    ;GET BUFFER START
668 002250 112722 000377          MOVB   #377,(R2)+    ;INSERT BACKGROUND DATA
669 002254 005000          CLR     R0
670 002256 110022          LT1B: MOVB   R0,(R2)+
671 002260 005200          INC     R0           ;LOAD WRITE BUFFER (0,1,2,3,4,5)
672 002262 022700 000006          CMP    #6,R0
673 002266 001373          BNE     LT1B
674 002270 004737 003342          JSR     PC,RWIND     ;GO REWIND
675 002274 012737 000004 000700  MOV     #4,FUN       ;SET WRITE FUNCTION CODE
676 002302 012737 007255 000672  MOV     #WDATA+1,BADR ;SET DATA POINTER
677 002310 012737 177772 000674  MOV     #-6,BYTES    ;SET SIZE OF RECORD
678 002316 012737 006664 000656  MOV     #MSG17,ERRAD ;SET WRITE ERROR CODE
679 002324 004737 003620          JSR     PC,EXEC      ;GO EXECUTE COMMAND
680 002330 000240          LT1C: NOP
681 002332 004737 004050          JSR     PC,ERCHK     ;GO CHECK FOR STATUS ERROR
682 002336 012737 177777 000676  MOV     #-1,SCNT
683 002344 004737 003526          JSR     PC,BKSP      ;GO BACKSPACE ONE RECORD
684 002350 012702 007356          MOV     #RDATA,R2
685 002354 012700 000010          MOV     #10,R0
686 002360 012722 177777          LT1D: MOV     #-1,(R2)+   ;BACKGROUND READ BUFFER
687 002364 005300          DEC     R0
688 002366 001374          BNE     LT1D        ;DO ALL
689 002370 012737 000002 000700  MOV     #2,FUN       ;SET READ FUNCTION CODE
690 002376 012737 007356 000672  MOV     #RDATA,BADR  ;SET READ POINTER
691 002404 012737 177772 000674  MOV     #-6,BYTES    ;SET SIZE OF RECORD
692 002412 012737 006701 000656  MOV     #MSG18,ERRAD ;SET READ ERROR CODE
693 002420 004737 003620          JSR     PC,EXEC      ;GO DO READ
694 002424 000240          *1E:  NOP
695 002426 004737 004050          JSR     PC,ERCHK     ;GO CHECK ERRORS

```

696	002432	012701	007255	MOV	#WDATA+1,R1	:SET EXPT DATA POINTER
697	002436	012702	007356	MOV	#RDATA,R2	:SET RCVD DATA POINTER
698	002442	012700	000006	MOV	#6,R0	:SET SIZE OF RECORD
699	002446	004737	004336	JSR	PC,DCHK	:GO CHECK DATA
700	002452	004737	004626	JSR	PC,ITER	:GO SEE IF ITERATIONS
701	002456	000137	001750	JMF	TSCD2	:RETURN TO SCHEDULAR

```

702
703
704
705
706
707
708
709
710
711 002462 000240
712 002464 012737 007141 000654
713 002472 012702 007254
714 002476 005000
715 002500 110022
716 002502 005200
717 002504 022700 000006
718 002510 001373
719 002512 004737 003342
720 002516 012737 000004 000700
721 002524 012737 007254 000672
722 002532 012737 177772 000674
723 002540 004737 003620
724 002544 000240
725 002546 012737 006664 000656
726 002554 004737 004050
727 002560 012737 177777 000676
728 002566 004737 003526
729 002572 012702 007356
730 002576 012700 000010
731 002602 012722 177777
732 002606 005300
733 002610 001374
734 002612 012737 000002 000700
735 002620 012737 007357 000672
736 002626 012737 177772 000674
737 002634 004737 003620
738 002640 000240
739 002642 012737 006701 000656
740 002650 004737 004050
741 002654 012701 007254
742 002660 012702 007357
743 002664 012700 000006
744 002670 004737 004336
745 002674 004737 004626
746 002700 000137 001750
747

:*****
: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: MOV#B 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
    
```

748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790  
791  
792

```

:*****
: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,RWIND ;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,RWIND ;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
  
```

793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832

003134 000240  
 003136 012737 007223 000654  
 003144 004737 003342  
 003150 012700 000062  
 003154 012737 000014 000700  
 003162 012737 007254 000672  
 003170 012737 177760 000674  
 003176 012737 006664 000656  
 003204 004737 003620  
 003210 004737 004050  
 003214 012737 177777 000676  
 003222 004737 003526  
 003226 005300  
 003230 001351  
 003232 000240  
 003234 004737 003342  
 003240 012737 000500 000636  
 003246 012737 007356 000672  
 003254 012737 177760 000674  
 003262 012737 000002 000700  
 003270 012737 006740 000656  
 003276 004737 003620  
 003302 004737 004050  
 003306 000240  
 003310 012737 000004 000636  
 003316 012737 000002 000634  
 003324 004737 004626  
 003330 012737 000010 000634  
 003336 000137 001750

```

LT4:  NOP
      MOV #LT4MSG,EMADDR ;SET HEADER
      JSR PC,RWIND ;GO REWIND
      MOV #62,R0 ;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 R0 ;SEE IF DONE ALL SEQUENCES
      BNE LT4A ;IF NOT: BR
LT4IT: NOP
      JSR PC,RWIND ;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.
:*****
  
```



```

833                                     ;REWIND SUBROUTINE*****
834
835 003342 000240 RWND: NOP
836 003344 013777 000610 175230 MOV UDES,@MTC ;SELECT UNIT
837 003352 032777 000040 175220 BIT #40,@MTS ;SEE IF AT BOT
838 003360 001056 BNE RWNDXX ;IF SO: BR
839 003362 052777 000017 175212 BIS #17,@MTC ;START REWIND
840 003370 105777 175206 1$: TSTB @MTC
841 003374 100375 BPL 1$ ;AWAIT CUR
842 003376 032777 000001 175174 RWND1: BIT #1,@MTS ;AWAIT TUR
843 003404 001774 BEQ RWND1
844 003406 032777 000040 175164 BIT #40,@MTS ;SEE IF BOT SET
845 003414 001040 BNE RWNDXX ;IF SO: BR
846 003416 032777 020000 175174 BIT #20000,@SWR ;SEE IF PRINT ERROR
847 003424 001030 BNE RWNDX ;IF NOT: BR
848 003426 013704 000654 MOV EMADDR,R4
849 003432 004737 005172 JSR PC,TTOUT ;PRINT HEADER
850 003436 012704 006460 MOV #MSG4,R4
851 003442 004737 005172 JSR PC,TTOUT ;PRINT REWIND ERROR
852 003446 012704 006510 MOV #MSG5,R4
853 003452 004737 005172 JSR PC,TTOUT ;PRINT MTS TAG
854 003456 017703 175116 MOV @MTS,R3
855 003462 004737 005360 JSR PC,OCTPE ;PRINT MTS
856 003466 012704 006517 MOV #MSG6,R4
857 003472 004737 005172 JSR PC,TTOUT ;PRINT MTC TAG
858 003476 017703 175100 MOV @MTC,R3
859 003502 004737 005360 JSR PC,OCTPE ;PRINT MTC
860 003506 005777 175106 RWNDX: TST @SWR ;SEE IF HALT ON ERROR
861 003512 100001 BPL RWNDXX ;IF NOT: BR
862 003514 000000 HALT
863 003516 004737 005724 RWNDXX: JSR PC,CKSWR ;GO TEST FOR ^G
864 003522 000240 NOP
865 003524 000207 RTS ;RETURN
866
867                                     ;BACKSPACE SUBROUTINE*****
868
869 003526 000240 BKSP: NOP
870 003530 013777 000610 175044 MOV UDES,@MTC ;SELECT UNIT
871 003536 013777 000676 175040 MOV SCNT,@MTBC ;SET NUMBER OF RECORDS TO SPACE
872 003544 052777 000013 175030 BIS #13,@MTC ;START SPACE REVERSE
873 003552 105777 175024 1$: TSTB @MTC
874 003556 100375 BPL 1$ ;AWAIT CUR
875 003560 032777 000001 175012 BKSP1: BIT #1,@MTS ;AWAIT TUR
876 003566 001774 BEQ BKSP1
877 003570 012737 000001 000664 MOV #1,SPFLG ;SET SPACE FLAG
878 003576 012737 006546 000656 MOV #MSG9,ERRAD
879 003604 004737 004050 JSR PC,ERCHK ;GO CHECK FOR ERROR
880 003610 005037 000664 CLR SPFLG ;CLEAR SPACE FLAG
881 003614 000240 NOP
882 003616 000207 RTS ;RETURN
883

```

```

884                                     ;COMMAND EXECUTE SUBROUTINE*****
885
886 003620 000240 EXEC: NOP
887 003622 005005 CLR R5
888 003624 032777 000200 174750 EXEC0: BIT #200,@MTC ;SEE IF CLK
889 003632 001021 BNE EXEC2 ;IF SO: BR
890 003634 005305 DEC R5 ;SEE IF TIMED OUT
891 003636 001372 BNE EXEC0 ;IF NOT: BR
892 003640 005737 000710 TST HDRFL ;SEE IF DONE HEADER
893 003644 001004 BNE EXEC1 ;IF SO: BR
894 003646 013704 000654 MOV EMADDR,R4
895 003652 004737 005172 JSR PC,TTOUT ;ELSE PRINT HEADER
896 003656 012704 006571 EXEC1: MOV #MSG10,R4
897 003662 004737 005172 JSR PC,TTOUT ;PRINT NOT READY ERROR
898 003666 005777 174726 TST @SWR ;SEE IF HALT ON ERROR
899 003672 100001 BPL EXEC2 ;IF NOT: BR
900 003674 000000 HALT
901 003676 004737 005724 EXEC2: JSR PC,CKSWR ;GO TEST FOR ^G
902 003702 000240 NOP
903 003704 013777 000610 174670 MOV UDES,@MTC ;SELECT UNIT
904 003712 013777 000672 174666 MOV BADR,@MTBA ;SET BUS MEMORY ADDRESS
905 003720 013777 000674 174656 MOV BYTES,@MTBC ;SET BYTE COUNT
906 003726 013701 000700 MOV FUN,R1 ;GET FUNCTION
907 003732 052701 000101 BIS #101,R1 ;SET IN GO BIT AND INTERRUPT ENABLE
908 003736 050177 174640 BIS R1,@MTC ;LOAD COMMAND+GO+IE
909 003742 000240 NOP
910 003744 005077 174646 CLR @PSW ;ALLOW INTERRUPTS
911 003750 013737 000636 000646 MOV STALL,TEMP1 ;SET READY STALL
912 003756 005001 CLR R1
913 003760 005301 EXEC3: DEC R1
914 003762 001376 BNE EXEC3 ;AWAIT INTERRUPT
915 003764 005337 000646 DEC TEMP1
916 003770 001373 BNE EXEC3
917 003772 032777 020000 174620 BIT #20000,@SWR ;SEE IF PRINT ERROR
918 004000 001013 BNE EXECX ;IF NOT: BR
919 004002 005737 000710 TST HDRFL ;SEE IF DONE HEADER
920 004006 001004 BNE EXEC4 ;IF SO: BR
921 004010 013704 000654 MOV EMADDR,R4
922 004014 004737 005172 JSR PC,TTOUT ;PRINT HEADER
923 004020 012704 006606 EXEC4: MOV #MSG11,R4
924 004024 004737 005172 JSR PC,TTOUT ;PRINT NO INTERRUPT MESSAGE
925 004030 005777 174564 EXECX: TST @SWR ;SEE IF HALT ON ERROR
926 004034 100001 BPL EXECXX ;IF NOT: BR
927 004036 000000 HALT
928 004040 004737 005724 EXECXX: JSR PC,CKSWR ;GO TEST FOR ^G
929 004044 000240 NOP
930 004046 000207 RTS PC ;RETURN TO CALLER
931

```

```

932                                     ;STATUS ERROR CHECK SUBROUTINE*****
933
934 004050 005777 174526 ERCHK: TST @MTC ;SEE IF ANY ERROR BITS
935 004054 100002 BPL ERCHK1 ;IF NOT: BR
936 004056 000137 004126 JMP ERPT ;ELSE PRINT ERROR
937 004062 005777 174516 ERCHK1: TST @MTBC ;SEE IF BYTE COUNT IS ZERO
938 004066 001402 BEQ ERCHK2 ;IF SO: BR
939 004070 000137 004126 JMP ERPT ;ELSE PRINT ERROR
940 004074 013703 000674 ERCHK2: MOV BYTES,R3
941 004100 005403 NEG R3
942 004102 063703 000672 ADD BADR,R3 ;SET EXPT BUS ADDRESS
943 004106 005737 000664 TST SPFLG ;SEE IF SPACE OPERATION
944 004112 001401 BEQ ERCHK3 ;IF NOT: BR
945 004114 000207 RTS PC
946 004116 020377 174464 ERCHK3: CMP R3,@MTBA ;SEE IF EXPT=RCVD
947 004122 001001 BNE ERPT ;IF NOT: BR
948 004124 000207 RTS PC ;ELSE EXIT
949 004126 000240 ERPT: NOP
950 004130 032777 020000 174462 BIT #20000,@SWR ;SEE IF SHOULD PRINT
951 004136 001067 BNE ERPTX ;IF NOT: BR
952 004140 005737 000710 TST HDRFL ;SEE IF DONE HEADER
953 004144 001006 BNE ERPT1 ;IF SO: BR
954 004146 013704 000654 MOV EMADDR,R4
955 004152 004737 005172 JSR PC,TTOUT ;ELSE PRINT HEADER
956 004156 005237 000710 INC HDRFL ;SET FLAG
957 004162 013704 000656 ERPT1: MOV ERRAD,R4
958 004166 004737 005172 JSR PC,TTOUT ;PRINT ERROR CODE
959 004172 012704 006510 MOV #MSG5,R4
960 004176 004737 005172 JSR PC,TTOUT ;PRINT MTS TAG
961 004202 017703 174372 MOV @MTS,R3
962 004206 004737 005360 JSR PC,OCTPE ;PRINT MTS
963 004212 012704 006517 MOV #MSG6,R4
964 004216 004737 005172 JSR PC,TTOUT ;PRINT MTC TAG
965 004222 017703 174354 MOV @MTC,R3
966 004226 004737 005360 JSR PC,OCTPF ;PRINT MTC
967 004232 012704 006526 MOV #MSG7,R4
968 004236 004737 005172 JSR PC,TTOUT ;PRINT BYTE COUNT TAG
969 004242 017703 174336 MOV @MTBC,R3
970 004246 004737 005370 JSR PC,OCTP ;PRINT BYTE COUNT
971 004252 005737 000664 TST SPFLG ;SEE IF PRINT BA
972 004256 001017 BNE ERPTX ;IF NOT: BR
973 004260 012704 006536 MOV #MSG8,R4
974 004264 004737 005172 JSR PC,TTOUT ;PRINT BUS ADDRESS TAG
975 004270 017703 174312 MOV @MTBA,R3
976 004274 004737 005370 JSR PC,OCTP ;PRINT CURRENT ADDRESS
977 004300 013703 000674 MOV BYTES,R3
978 004304 005403 NEG R3
979 004306 063703 000672 ADD BADR,R3
980 004312 004737 005370 JSR PC,OCTP ;PRINT EXPT ADDRESS
981 004316 005777 174276 ERPTX: TST @SWR ;SEE IF HALT ON ERROR
982 004322 100001 BPL ERPTXX ;IF NOT: BR
983 004324 000000 HALT
984 004326 004737 005724 ERPTXX: JSR PC,CKSWR ;GO TEST FOR ^G
985 004332 000137 004576 JMP SCOPE ;GO SEE IF SCOPE ON ERROR
  
```

```

986                                     :DATA CHECK SUBROUTINE*****
987
988 004336 000240      DCHK:  NOP
989 004340 005037 000704  CLR      CRCNT      :CLEAR COUNTER
990 004344 121112      DCHK0:  CMPB      (R1),(R2)  :SEE IF EXPT DATA=RCVD DATA
991 004346 001007      BNE      DCHKE      :IF NOT: BR
992 004350 005237 000704  DCHK1:  INC      CRCNT      :BUMP CHARACTER COUNTER
993 004354 122122      CMPB      (R1)+,(R2)+
994 004356 005300      DEC      R0          :SEE IF DONE
995 004360 001371      BNE      DCHK0      :IF NOT: BR
996 004362 000137 004536  JMP      DCHKX      :ELSE GO TO EXIT ROUTINE
997 004366 000240      DCHKE:  NOP
998 004370 012737 000001 000706  MOV      #1,DERFL  :SET ERROR FLAG
999 004376 032777 020000 174214  BIT      #20000,@SWR :SEE IF PRINT ERROR
1000 004404 001054      BNE      DCHKX      :IF NOT: BR
1001 004406 005737 000710  TST      HDRFL      :SEE IF DONE HEADER
1002 004412 001007      BNE      DCHKE1     :IF SO: BR
1003 004414 013704 000654  MOV      EMADDR,R4
1004 004420 004737 005172  JSR      PC,TTOUT   :PRINT HEADER
1005 004424 012737 000001 000710  MOV      #1,HDRFL  :SET HEADER FLAG
1006 004432 012704 006626  DLHKE1:  MOV      #MSG12,R4
1007 004436 005737 000712  TST      PFLG      :SEE IF PRINTED DATA ERROR TAG
1008 004442 001004      BNE      DCHKE2     :IF SO: BR
1009 004444 005237 000712  INC      PFLG
1010 004450 004737 005172  JSR      PC,TTOUT   :ELSE PRINT DATA ERROR TAG
1011 004454 012704 006642  DCHKE2:  MOV      #MSG13,R4
1012 004460 004737 005172  JSR      PC,TTOUT   :PRINT CHAR NUMBER TAG
1013 004464 013703 000704  MOV      CRCNT,R3
1014 004470 004737 005370  JSR      PC,OCTP    :PRINT CHAR NUMBER
1015 004474 012704 006650  MOV      #MSG14,R4
1016 004500 004737 005172  JSR      PC,TTOUT   :PRINT GOOD TAG
1017 004504 111103      MOVB     (R1),R3
1018 004506 004737 005616  JSR      PC,DOUT    :PRINT GOOD CHARACTER
1019 004512 012704 006655  MOV      #MSG15,R4
1020 004516 004737 005172  JSR      PC,TTOUT   :PRINT BAD TAG
1021 004522 111203      MOVB     (R2),R3
1022 004524 004737 005616  JSR      PC,DOUT    :PRINT BAD CHARACTER
1023 004530 000240      NOP
1024 004532 000137 004350  JMP      DCHK1      :CONTINUE FOR ALL BYTES
1025 004536 000240      DCHKX:  NOP
1026 004540 005737 000706  TST      DERFL      :SEE IF ANY ERROR
1027 004544 001404      BEQ     DCHKXX      :IF NOT: BR
1028 004546 005777 174046  TST      @SWR      :SEE IF HALT ON ERROR
1029 004552 100001      BPL     DCHKXX      :IF NOT: BR
1030 004554 000000      HALT
1031 004556 004737 005724  DCHKXX:  JSR      PC,CKSWR :GO TEST FOR ^G
1032 004562 000240      NOP
1033 004564 005037 000712  CLR      PFLG      :CLEAR PRINT FLAG
1034 004570 005037 000706  CLR      DERFL      :CLEAR DATA ERROR FLAG
1035 004574 000207      RTS      PC        :RETURN
  
```

```

1036                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
1037
1038 004576 000240 SCOPE: NOP
1039 004600 032777 040000 174012 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
1040 004606 001001 BNE SCOPE1 ;IF SO: BR
1041 004610 000207 RTS PC ;ELSE EXIT
1042 004612 000240 SCOPE1: NOP
1043 004614 005726 TST (SP)+ ;RESET STACK
1044 004616 000240 NOP
1045 004620 017703 174034 MOV @LTADD,R3
1046 004624 000113 JMP (R3) ;LOOP ON ERROR
1047
1048                                     ;TEST ITERATION SUBROUTINE*****
1049
1050 004626 000240 ITER: NOP
1051 004630 004737 005724 JSR PC,CKSWR
1052 004634 032777 010000 173756 BIT #10000,@SWR ;SEE IF ITERATIONS
1053 004642 001403 BEQ ITER1 ;IF SO: BR
1054 004644 005037 000702 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
1055 004650 000207 RTS PC ;ELSE EXIT
1056 004652 005737 000670 ITER1: TST PCNTR ;SEE IF FIRST PASS
1057 004656 001772 BEQ ITER0 ;IF SO: BR
1058 004660 005237 000702 INC ITCNT ;BUMP COUNTER
1059 004664 023737 000702 000634 CMP ITCNT,ITAMT ;SEE IF DONE ALL
1060 004672 001764 BEQ ITER0 ;IF SO: BR
1061 004674 005726 TST (SP)+ ;RESET STACK
1062 004676 017700 173760 MOV @ITRLP,R0 ;SET ITERATION POINTER
1063 004702 000110 JMP (R0) ;GO ITERATE
1064
1065                                     ;MAG TAPE INTERRUPT HANDLER*****
1066
1067 004704 000240 MTINT: NOP
1068 004706 022626 CMP (SP)+,(SP)+ ;RESET STACK POINTER
1069 004710 042777 000100 173664 BIC #100,@MTC ;CLEAR INTERRUPT ENABLE
1070 004716 000240 NOP
1071 004720 000240 NOP
1072 004722 000207 RTS PC ;RETURN
1073
1074                                     ;TTY INTERRUPT HANDLER*****
1075
1076 004724 000240 TTINT: NOP
1077 004726 000240 NOP
1078 004730 000240 NOP
1079 004732 000002 RTI
1080
    
```

```

1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098 004734 005037 000646 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
1099 004740 005000 CLR R0
1100 004742 004737 005120 TTR0: JSR PC,TTR0 ;GO READ CHARACTER
1101 004746 122737 000215 000644 CMPB #215,TIB ;SEE IF CR
1102 004754 001005 BNE TTR1 ;IF NOT: BR
1103 004756 005737 000646 TST TEMP1 ;SEE IF FIRST CHARACTER
1104 004762 001446 BEQ TTR5 ;IF SO: BR
1105 004764 000137 005056 JMP TTR2 ;ELSE GO LOAD VALUE
1106 004770 122737 000260 000644 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0
1107 004776 101402 BLOS TTR1A ;IF NOT: BR
1108 005000 000137 005102 JMP TTR1A ;ELSE GO TO ERROR
1109 005004 122737 000270 000644 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7
1110 005012 101002 BHI TTR1B ;IF NOT: BR
1111 005014 000137 005102 JMP TTR1B ;ELSE GO TO ERROR
1112 005020 005237 000646 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
1113 005024 000241 CLC
1114 005026 006100 ROL R0
1115 005030 000241 CLC
1116 005032 006100 ROL R0 ;SHIFT 3 LEFT
1117 005034 000241 CLC
1118 005036 006100 ROL R0
1119 005040 042737 177770 000644 BIC #177770,TIB ;STRIP ASCII
1120 005046 053700 000644 BIS TIB,R0 ;LOAD CHARACTER
1121 005052 005301 DEC R1 ;SEE IF DONE
1122 005054 001332 BNE TTR0 ;IF NOT: BR
1123 005056 020002 TR2: CMP R0,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
1124 005060 101402 BLOS TTR3 ;IF OT: BR
1125 005062 000137 005102 JMP TTR3 ;ELSE GO TO ERROR
1126 005066 020300 TTR3: CMP R3,R0 ;SEE IF BELOW MINIMUM LIMIT
1127 005070 101402 BLOS TTR4 ;IF NOT: BR
1128 005072 000137 005102 JMP TTR4 ;ELSE GO TO ERROR
1129 005076 010015 TTR4: MOV R0,(R5) ;LOAD VALUE
1130 005100 000207 TTR5: RTS PC ;EXIT
1131

```

```

1132                                     :TTY ENTRY ERROR SUBROUTINE*****
1133
1134 005102 012704 006662          T1NER: MOV    #MSG16,R4
1135 005106 004737 005172          .      JSR    PC,TTOUT          :PRINT?
1136 005112 162716 000020          .      SUB    #20,(SP)          :RESET SP TO START OF VALUE ROUTINE
1137 005116 000207                .      RTS    PC              :REDO VALUE ENTRY
1138
1139                                     :TTY READ SUBROUTINE*****
1140
1141 005120 005077 173500          TTIN:  CLR    @TKS
1142 005124 005077 173476          .      CLR    @TKB
1143 005130 005037 000644          .      CLR    TIB
1144 005134 005277 173464          .      INC    @TKS
1145 005140 105777 173460          TTIN1: TSTB   @TKS
1146 005144 100375                .      BPL    TTIN1
1147 005146 017737 173454 000644 .      MOV    @TKB,TIB
1148 005154 105777 173450          TTIN2: TSTB   @TPS
1149 005160 100375                .      BPL    TTIN2
1150 005162 113777 000644 173442 .      MOVB   TIB,@TPB
1151 005170 000207                .      RTS    PC
1152
1153                                     :TTY OUTPUT SUBROUTINE*****
1154
1155 005172 112437 000642          TTOUT: MOVB   (R4)+,TOB
1156 005176 122737 000043 000642 .      CMPB   #43,TOB
1157 005204 001452                .      BEQ   TEX
1158 005206 122737 000045 000642 .      CMPB   #45,TOB
1159 005214 001407                .      BEQ   TCRLF
1160 005216 122737 000041 000642 .      CMPB   #41,TOB
1161 005224 001443                .      BEQ   TBELL
1162 005226 004737 005316          .      JSR   PC,TOG
1163 005232 000757                .      BR    TTOUT
1164 005234 112737 000015 000642 TCRLF: MOVB   #15,TOB
1165 005242 004737 005316          .      JSR   PC,TOG
1166 005246 012703 000004          .      MOV   #4,R3
1167 005252 005037 000642          TCRLFA: CLR    TOB
1168 005256 004737 005316          .      JSR   PC,TOG
1169 005262 005303                .      DEC   R3
1170 005264 001372                .      BNE   TCRLFA          ;DO FILLERS
1171 005266 112737 000012 000642 .      MOVB   #12,TOB
1172 005274 004737 005316          .      JSR   PC,TOG
1173 005300 105737 000724          .      TSTB  RDSW
1174 005304 100401                .      BMI   1$
1175 005306 000731                .      BR    TTOUT
1176 005310 005037 000724          1$:   CLR    RDSW
1177 005314 000406                .      BR    TEX
1178 005316 105777 173306          TOG:   TSTB   @TPS
1179 005322 100375                .      BPL   TOG
1180 005324 113777 000642 173300 .      MOVB   TOB,@TPB
1181 005332 000207                .      RTS   PC
1182 005334 012703 000002          TEX:   MOV    #2,R3
1183 005340 012737 000007 000642 TBELL: MOV    #7,TOB
1184 005346 004737 005316          TBELA: JSR    PC,TOG
1185 005352 005303                .      DEC   R3
1186 005354 001371                .      BNE   TBELA
1187 005356 000705                .      BR    TTOUT
  
```

```

1188
1189
1190 ;OCTAL OUTPUT SUBROUTINE*****
1191 005360 012737 000001 005614 OCTPE: MOV #1,OFL
1192 005366 000402 BR OCTPE1
1193 005370 005037 005614 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
1194 005374 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
1195 005376 001007 BNE OCTP0 ;IF NOT ZERO: BR
1196 005400 005737 005614 TST OFL ;SEE IF PRINT ALL 0
1197 005404 001004 BNE OCTP0 ;IF SO: BR
1198 005406 004737 005574 JSR PC,OCTPG1 ;ELSE PRINT ZERO
1199 005412 000137 005536 JMP OCTP3 ;SPACE AND EXIT
1200 005416 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
1201 005422 001406 BEQ OCTP ;IF NOT: BR
1202 005424 012704 000001 MOV #1,R4
1203 005430 004737 005552 JSR PC,OCTPG ;PRINT 1
1204 005434 000137 005446 JMP OCTP2
1205 005440 005004 OCTP1: CLR R4
1206 005442 004737 005552 JSR PC,OCTPG ;PRINT 0
1207 005446 010304 OCTP2: MOV R3,R4
1208 005450 006004 ROR R4
1209 005452 006004 ROR R4
1210 005454 006004 ROR R4 ;POSITION DIGIT
1211 005456 006004 ROR R4
1212 005460 000304 SWAB R4
1213 005462 004737 005552 JSR PC,OCTPG ;PRINT DIGIT 2
1214 005466 010304 MOV R3,R4
1215 005470 006004 ROR R4
1216 005472 000304 SWAB R4
1217 005474 004737 005552 JSR PC,OCTPG ;PRINT DIGIT 3
1218 005500 010304 MOV R3,R4
1219 005502 006104 ROL R4
1220 005504 006104 ROL R4
1221 005506 000304 SWAB R4
1222 005510 004737 005552 JSR PC,OCTPG ;PRINT DIGIT 4
1223 005514 010304 MOV R3,R4
1224 005516 006004 ROR R4
1225 005520 006004 ROR R4
1226 005522 006004 ROR R4
1227 005524 004737 005552 JSR PC,OCTPG
1228 005530 010304 MOV R3,R4
1229 005532 004737 005552 JSR PC,OCTPG ;PRINT DIGIT 5
1230 005536 012737 000240 000642 OCTP3: MOV #240,TOB
1231 005544 004737 005316 JSR PC,TOB ;PRINT SPACE
1232 005550 000207 RTS PC ;EXIT
  
```



```

1233
1234
1235
1236 005552 042704 177770
1237 005556 001004
1238 005560 005737 005614
1239 005564 001001
1240 005566 000207
1241 005570 005237 005614
1242 005574 052704 000260
1243 005600 010437 000642
1244 005604 004737 005316
1245 005610 010304
1246 005612 000207
1247 005614 000000
1248
1249
1250
1251 005616 005037 000642
1252 005622 012704 000010
1253 005626 110337 000642
1254 005632 105777 172772
1255 005636 100375
1256 005640 132737 000200 000642
1257 005646 001404
1258 005650 012777 000061 172754
1259 005656 000403
1260 005660 012777 000060 172744
1261 005666 006137 000642
1262 005672 005304
1263 005674 001356
1264 005676 000207
1265 005700 013703 000652
1266 005704 000303
1267 005706 004737 005616
1268 005712 013703 000652
1269 005716 004737 005616
1270 005722 000207
1271
1272
1273
1274 005724 022737 000176 000620
1275 005732 001041
1276 005734 105777 172664
1277 005740 100036
1278 005742 017737 172660 000644
1279 005750 042737 177600 000644
1280 005756 022737 000007 000644
1281 005764 001024
1282 005766 012704 007050
1283 005772 004737 005172
1284 005776 012704 007054
1285 006002 004737 005172
1286 006006 017703 172606
1287 006012 004737 005360
1288 006016 012704 007066

;OCTAL PRINT SUBROUTINE*****
OCTPG: BIC #177770,R4
        BNE OCTPG0
        TST OFL
        BNE OCTPG0
        RTS PC
OCTPG0: INC OFL
OCTPG1: BIS #260,R4
        MOV R4,TOB
        JSR PC,TOG
        MOV R3,R4
        RTS PC
OFL: 0 ;FIRST CHAR FLAG

;DATA CHARACTER OUTPUT SUBROUTINE*****
DOUT: CLR TOB ;SET NUMBER TO PRINT
       MOV #10,R4
       MOV R3,TOB
DOUT1: TSTB @TPS
       BPL DOUT1
       BITB #200,TOB
       BEQ DOUT2
       MOV #061,@TPB
       BR DOUT3
DOUT2: MOV #060,@TPB
DOUT3: ROL TOB
       DEC R4
       BNE DOUT1
       RTS PC
DOUTD: MOV TEMP3,R3
       SWAB R3
       JSR PC,DOUT
       MOV TEMP3,R3
       JSR PC,DOUT
       RTS PC

;SUBROUTINE TO CHANGE CONTENTS OF SOFTWARE SWITCH REGISTER
;SOFTWARE SWITCH REG PRESENT
;NO, GET OUT
;YES, WAIT FOR
;READY, GET CHARACTER
;AND STRIP OFF
;THE GARBAGE
;IS IT A <^G>
CKSWR: CMP #SWREG,SWR
       BNE OUT
       TSTB @TKS
       BPL OUT
       MOV @TKB,TIB
       BIC #177600,TIB
       CMP #7,TIB
       BNE OUT
       MOV #SCNTG,R4
       JSR PC,TTOUT
CNTLU: MOV #SMSWR,R4
       JSR PC,TTOUT
       MOV @SWR,R3
       JSR PC,OCTPE
       MOV #SMNEW,R4
  
```



1328  
 1329  
 1330  
 1331  
 1332  
 1333  
 1334  
 1335  
 1336  
 1337  
 1338  
 1339  
 1340  
 1341  
 1342  
 1343  
 1344  
 1345  
 1346  
 1347  
 1348  
 1349  
 1350  
 1351  
 1352  
 1353  
 1354  
 1355  
 1356  
 1357  
 1358  
 1359  
 1360

```

: *****
:                                     MODIFIED JAN 10 1978
:
: ++
:                                     CHECK FOR DUMP MODE OR AUTOMATIC /ACT11-XXDP MODE
: --
:
CKMODE: CLR      AUTOM      ;INIT AUTOMATIC MODE INDICATOR
          CLRB     ACT11M    ;INIT ACT11 AUTO MODE INDICATOR
          CLRB     XXDPM     ;INIT XXDP AUTO MODE INDICATOR
          CLRB     ADUMPM    ;INIT ACT11 DUMP MODE INDICATOR
          CLRB     XDUMPM    ;INIT XXDP DUMP MODE INDICATOR
          TST      @#42      ;AUTO MODE?
          BEQ      2$        ;BRANCH -IF NO
          INC      AUTOM     ;SET AUTO MODE INDICATOR
          CMP      @#42,@#46 ;ACT11 MODE?
          BEQ      1$        ;BRANCH - IF YES
          INCB     XXDPM     ;INDICATE XXDP AUTO MODE
          BR       5$        ;AND EXIT
1$:      INCB     ACT11M    ;INDICATE ACT11 AUTO MODE
          MOV      #176,SWR  ;SET SOFTWARE SWITCH IN ACT MODE
          MOV      #124000,@SWR ;SET SWITCH REGISTER
          BR       5$        ;AND EXIT
2$:      TSTB     @#41      ;MAN/MODE VIA ACT11/PAPER TAPE?
          BNE     3$        ;BRANCH - IF NOT
          INCB     ADUMPM    ;INDICATE MAN/MODE VIA ACT11/PAPER TAPE
          BR       5$        ;AND EXIT
3$:      INCB     XDUMPM    ;INDICATE MANUAL MODE VIA XXDP
5$:      RTS      PC       ;RETURN
:
: *****
  
```



1417	006764	020045	047041	020117	MSG21: .ASCII /% !NO UNIT(S) AVAILABLE#/
1418	006772	047125	052111	051450	
1419	007000	020051	053101	044501	
1420	007006	040514	046102	021505	
1421	007014	041445	047101	047516	MSG22: .ASCII /%CANNOT TEST LOAD MEDIUM..%#/
1422	007022	020124	042524	052123	
1423	007030	046040	040517	020104	
1424	007036	042515	044504	046525	
1425	007044	020441	021445		
1426	007050	057045	021507		\$CNTG: .ASCII /%^G#/
1427	007054	022445	020441	053523	\$MSWR: .ASCII /%%..SWR= #/
1428	007062	036522	021440		
1429	007066	020040	042516	036527	\$MNEW: .ASCII / NEW= #/
1430	007074	021440			
1431	007076	037445	022445	043	\$QUEST: .ASCII /%?%#/ ;TEST HEADER*****
1432					
1433					
1434					
1435	007103	045	052045	051505	LT1MSG: .ASCII /%%TEST 1: WRITE FROM ODD BYTE#/
1436	007110	020124	035061	053440	
1437	007116	044522	042524	043040	
1438	007124	047522	020115	042117	
1439	007132	020104	054502	042524	
1440	007140	043			
1441	007141	045	052045	051505	LT2MSG: .ASCII /%%TEST 2: READ TO ODD BYTE#/
1442	007146	020124	035062	051040	
1443	007154	040505	020104	047524	
1444	007162	047440	042104	041040	
1445	007170	052131	021505		
1446	007174	022445	042524	052123	LT3MSG: .ASCII /%%TEST 3: OPI TOO LONG#/
1447	007202	031440	020072	050117	
1448	007210	020111	047524	020117	
1449	007216	047514	043516	043	
1450	007223	045	052045	051505	LT4MSG: .ASCII /%%TEST 4: OPI TOO SHORT#/
1451	007230	020124	035064	047440	
1452	007236	044520	052040	047517	
1453	007244	051440	047510	052122	
1454	007252	043			
1455		007254			.EVEN
1456					
1457	007254	177777			WDATA: -1
1458		007356			.=.+100
1459	007356	000000			RDATA: 0
1460		000001			.END

CZTMFFO TMA,B-11 SPLMTL INSTR  
 CZTMFF.P11 16-AUG-78 08:48

MACY11 30A(1052) 16-AUG-78 08:51 PAGE 38  
 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0037

ACTCNR	000106	364#	475*	633	652*								
ACT11M	000102	360#	496	520	540	551	563	580	624	631	647	1337*	1348*
ADUMPM	000104	362#	1339*	1354*									
AUTOM	000100	359#	1336*	1343*									
BADR	000672	440#	676*	690*	721*	735*	764*	776*	809*	821*	904	942	979
BCNT	000716	450#											
BKSP	003526	683	728	770	815	869#							
BKSP1	003560	875#	876										
BYTES	000674	441#	677*	691*	722*	736*	765*	777*	810*	822*	905	940	977
CCNT	000614	414#											
CKMODE	006240	495	1336#										
CKSWR	005724	602	606	649	863	901	928	984	1031	1051	1274#		
CNTLU	005776	546	1284#	1301									
COUNT	000720	451#	1295*	1306	1322*								
CRCNT	000704	445#	989*	992*	1013								
DCHK	004336	699	744	988#									
DCHKE	004366	991	997#										
DCHKE1	004432	1002	1006#										
DCHKE2	004454	1008	1011#										
DCHKX	004536	996	1000	1025#									
DCHKXX	004556	1027	1029	1031#									
DCHKO	004344	990#	995										
DCHK1	004350	992#	1024										
DERFL	000706	446#	998*	1026	1034*								
DISPLA	000622	417#	487*										
DISPRE	000174	386#	487										
DOUT	005616	1018	1022	1251#	1267	1269							
DOUTD	005700	1265#											
DOUT1	005632	1254#	1255	1263									
DOUT2	005660	1257	1260#										
DOUT3	005666	1259	1261#										
DRIVE	000040	327#											
EMADDR	000654	433#	666*	712*	760*	805*	848	894	921	954	1003		
ERCHK	004050	681	695	726	740	768	813	826	879	934#			
ERCHK1	004062	935	937#										
ERCHK2	004074	938	940#										
ERCHK3	004116	944	946#										
ERPT	004126	786	936	939	947	949#							
ERPTX	004316	951	972	981#									
ERPTXX	004326	982	984#										
ERPT1	004162	953	957#										
ERRAD	000656	434#	678*	692*	725*	739*	767*	779*	811*	824*	878*	957	
EXEC	003620	679	693	723	737	766	780	812	825	886#			
EXECX	004030	918	925#										
EXECXX	004040	926	928#										
EXECO	003624	888#	891										
EXEC1	003656	893	896#										
EXEC2	003676	889	899	901#									
EXEC3	003760	913#	914	916									
EXEC4	004020	920	923#										
FUN	000700	443#	675*	689*	720*	734*	763*	778*	808*	823*	906		
HDRFL	000710	447#	599*	892	919	952	956*	1001	1005*				
HERE	002170	638	644#										
ITAMT	000634	422#	788*	790*	829*	831*	1059						
ITCNT	000702	444#	1054*	1058*	1059								
ITER	004626	700	745	789	830	1050#							









\$QUEST	007076	1314	1431#															
\$READ	006040	1291	1294#															
\$SVPC	- 001000	324#	345															
.	- 007360	315#	324	326#	330#	334#	338#	341#	345#	349#	378#	385#	391#	395#				
		401#	405#	470#	1455#	1458#												

. ABS. 007360 000

ERRORS DETECTED: 0

CZTMFF,CZTMFF.SEQ/SOL/ML:TOC-CZTMFF.SML/ML,CZTMFF.P11  
RUN-TIME: 13.5 SECONDS  
RUN-TIME RATIO: 41/6-6.6  
CORE USED: 7K (13 PAGES)

DOCUMENT PAGES: 41