

TS03

SUPPLEMENTAL INSTRUCTION
CZTSFDO
TEST

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

APR 1978
digital
MADE IN USA

E.F. 00000001
15-FEB-78 14:04

00010000 780330
CZTSFDD TS03 SPLMTL INSTR

PDP10 411 88HDRICZTSFDSEQ
MACY11 30A(1052 15-FEB-78 14:05 PAGE 1

00010000

780330

.PEM .

IDENTIFICATION

PRODUCT CODE: AC-9439D-MC
 PRODUCT NAME: CZTSFDD TS03 SPLMTL INSTR
 PROGRAM DATE: MARCH 1978
 MAINTAINER: DIAGNOSTIC ENGINEERING

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

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

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

COPYRIGHT (C) 1975, 1978 BY DIGITAL EQUIPMENT CORPORATION

00000001
 15-FEB-78 14:04
 PDP10 411
 MACY11 30A(1052
 15-FEB-78 14:05
 PAGE 1

TABLE OF CONTENTS

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

0215FDC TSO3 SPLMTL INSTR
0215FDC TSO3 15-FEB-78 14:04


```

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

```

LO1

CZTSFDD TSO3 SPLMTL INSTR
CZTSFDD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 11

SEG 00:1

369

```

370
371
372
373 000060 000060
374 000062 004544
375
376
377
378
379
380 000174 000174
381 000176 000000
382
383
384
385 000200 000200
386 000200 005000
387 000202 000167 000572
388
389
390 000210 000210
391 000214 012700 000001
392 000214 000167 000560
393
394
395
396 000224 000224
397 000226 004524 000340
398

```

```

;TTY INTERRUPT VECTOR*****
.=60
TTINT ;TTY INTERRUPT HANDLER
0

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

;STARTING ADDRESS*****
.=200
CLR R0
JMP START ;PROGRAM START

.=210
MOV #1,R0
JMP START ;NO HEADER START

;TMA-11 INTERRUPT VECTOR*****
.=224
MTINT ;TAPE INTERRUPT HANDLER
340

```

```

399          000600
400
401
402 000600 172520
403 000602 172522
404 000604 172524
405 000606 172526
406 000610 000000
407 000612 000020
408 000614 177760
409 000616 177776
410 000620 177570
411 000622 177570
412 000624 177560
413 000626 177562
414 000630 177564
415 000632 177566
416 000634 000010
417 000636 000040
418 000640 172520
419 000642 000224
420

```

```

      =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
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: 40      :READY DELAY MULTIPLIER
REGS:  172520  :UNIBUS ADDRESS
VECT:  224     :VECTOR ADDRESS

```

; FLAGS AND COUNTERS*****

421			
422			
423	000644	000000	TINF: 0
424	000646	000000	TOB: 000000
425	000650	000000	TIB: 000000
426	000652	000000	TEMP1: 000000
427	000654	000000	TEMP2: 000000
428	000656	000000	TEMP3: 000000
429	000660	000000	EMADDR: 000000
430	000662	000000	ERRAD: 000000
431	000664	000000	LTADD: 000000
432	000666	000000	ITRLP: 000000
433	000670	000000	SPFLG: 000000
434	000672	000000	STFLG: 000000
435	000674	000000	PCNTR: 000000
436	000676	000000	BADR: 000000
437	000700	000000	BYTES: 000000
438	000702	000000	SCNT: 000000
439	000704	000000	FUN: 000000
440	000706	000000	ITCNT: 000000
441	000710	000000	CRCNT: 000000
442	000712	000000	DERFL: 000000
443	000714	000000	HDRFL: 000000
444	000716	000000	PFLG: 000000
445	000720	000000	UNP: 000000
446	000722	000000	BCNT: 000000
447	000724	000000	COUNT: 000000
448	000726	000000	TEMPST: 0
449	000730	000000	RDSW: 0

; TEST ENTRY TABLE*****

451			
452			
453	000732	000000	TSTTBL: 0
454	000734	000000	0
455	000736	002064	T1AD: LT1
456	000740	002064	T1IAC: LT1
457	000742	002312	T2AD: LT2
458	000744	002312	T2IAD: LT2
459	000746	002534	T3AD: LT3
460	000750	002632	T3IAD: LT3IT
461	000752	002764	T4AD: LT4
462	000754	003062	T4IAD: LT4IT
463	000756	000000	0
464			


```

465 001000
466 001002
467 001010
468 001014
469 001020
470 001024
471 001030
472 001032
473 001036
474 001042
475 001044
476 001046
477 001052
478 001056
479 001062
480 001066
481 001072
482 001076
483 001102
484 001106
485 001112
486 001116
487 001122
488 001126
489 001132
490 001136
491 001142
492 001146
493 001152
494 001156
495 001162
496 001166
497 001172
498 001176
499 001202
500 001206
501 001212
502 001216
503 001222
504 001224
505 001230
506 001232
507 001234
508 001242
509 001246
510 001252
511 001256
512 001262
513 001264
514 001270

```

```

000240
012777 000340
012706 000500
004767 004524
004767 005160
105767 177752
001404
005067 177636
000167 000256
005700
001402
000167 000416
012704 006410
004767 003730
012704 007041
004767 003720
016703 177542
004767 004106
012705 000640
012701 000006
012702 172700
012703 172300
004767 003426
012704 007063
004767 003654
016703 177500
004767 004042
012705 000642
012701 000003
012702 000250
012703 000150
004767 003362
016700 177444
012720 004524
012710 000340
012701 000004
012702 000600
016700 177416
010022
062700 000002
005301
001373
012777 010000
012704 006444
004767 003540
005067 177332
016703 177326
000303
042703 177770
004767 003714

```

```

.=1000
:*****
:PROGRAM START AND HOUSEKEEPING
:*****

START: NOP
MOV #340, @PSW ; SET PRIORITY
MOV #500, SP ; SET STACK POINTER
JSR PC, SUSWR ; SEE IF NO HARDWARE SWITCH REG.
JSR PC, CKMODE ; CHECK FOR MODE OF OPERATION ++ C.W
TSTB ACT11M ; ACT MODE? ++ C.W
BEQ 1$ ; BRANCH - IF NO ++ C.W
CLR PCNTR ; INIT PASS COUNTER ++ C.W
JMP $ACT ; EXECUTE PROGRAM IN ACT MODE ++ C.W
1$: TST R0 ; SEE IF SKIP HEADER
BEQ 2$ ; BRANCH - IF NO ++ C.W
JMP ST4 ; DO TEST FROM RESTART ++ C.W
2$: MOV #MSG1, R4 ; PRINT HEADER
JSR PC, TTOUT
MOV #MSG22, R4 ; REQUEST UNIBUS ADDRESS
JSR PC, TTOUT
MOV REGS, R3 ; PRINT CURRENT ADDRESS
JSR PC, OCTP ; GET ADDRESS OF ENTRY
MOV #REGS, R5 ; SET SIZE OF ENTRY
MOV #6, R1 ; SET UPPER LIMIT
MOV #172700, R2 ; SET LOWER LIMIT
MOV #172300, R3 ; GO GET RESPONSE
JSR PC, TTR
MOV #MSG23, R4 ; REQUEST VECTOR
JSR PC, TTOUT
MOV VECT, R3 ; PRINT CURRENT VECTOR
JSR PC, OCTP ; GET ADDRESS OF ENTRY
MOV #VECT, R5 ; SET SIZE OF ENTRY
MOV #3, R1 ; SET UPPER LIMIT
MOV #250, R2 ; SET LOWER LIMIT
MOV #150, R3 ; GO GET RESPONSE
JSR PC, TTR ; GET VECTOR
MOV VECT, R0 ; SET INTERRUPT ADDRESS IN VECTOR
MOV #340, (R0) ; SET INTERRUPT PRIORITY
MOV #4, R1 ; SET NUMBER OF REGISTER
MOV #MTS, R2 ; GET FIRST ADDRESS OF TABLE
MOV REGS, R0 ; GET FIRST REGISTER
STOA: MOV R0, (R2)+ ; LOAD TABLE
ADD #2, R0 ; BUMP ADDRESS
DEC R1 ; SEE IF DONE
BNE STOA ; IF NOT: BE
MOV #10000, @MTC ; POWER CLEAR
STO: MOV #MSG2, R4 ; REQUEST UNIT NUMBER
JSR PC, TTOUT ; PRESET UNIT 0
CLR UDES ; GET UNIT NUMBER
MOV UDES, R3 ; GET UNIT NUMBER
SWAB R3 ; POSITION
BIC #177770, R3 ; MASK UNIT NUMBER
JSR PC, OCTP ; PRINT CURRENT VALUE

```

521	001274	012705	000656			MOV	#TEMP3,R5	:SET SAVE LOCATION
522	001300	012701	000001			MOV	#1,R1	:SET SIZE OF ENTRY
523	001304	012702	000007			MOV	#7,R2	:SET UPPER LIMIT
524	001310	012703	000000			MOV	#0,R3	:SET LOWER LIMIT
525	001314	004767	003234			JSR	PC,TTR	:GO GET UNIT NUMBER
526	001320	105767	177456			\$ACT:	TSTB	ACT11M
527	001324	001413				BEQ	NOACT	:ACT MODE? ++ C.W
528	001326	005067	177256			CLR	UDES	:BRANCH - IF NO ++ C.W
529	001332	122767	000010	177250	ST:	CMPB	#10,UDES	:INIT UNIT INDICATOR IN ACT MODE ++ C.W
530	001340	001002				BNE	2\$:ARE ALL UNITS DONE? ++ C.W
531	001342	000167	000410			JMP	\$DONE	:BRANCH - IF NO ++ C.W
532	001346	116767	177236	177302	2\$:	MOVB	UDES,TEMP3	:DO END OF PASS ++ C.W
533	001354	005367	177276			NOACT:	SWAB	:GET UNIT NUMBER IN ACT MODE ++ C.W
534	001360	042767	003400	177222		BIC	#3400,UDES	:POSITION UNIT NUMBER
535	001366	056767	177264	177214		BIS	TEMP3,UDES	:CLEAR OLD NUMBER
536	001374	016777	177210	177200		MOV	UDES,#MTC	:LOAD NEW NUMBER
537	001402	005000				CLR	R0	:SELECT UNIT
538	001404	022767	000176	177206		CMP	#SWREG,SWR	
539	001412	001002				BNE	ST1	
540	001414	004767	004256			JSR	PC,CNTLU	:GO ASK FOR SWR SETTINGS
541	001420	032777	000100	177152	ST1:	BIT	#100,#MTC	:SEE IF SELECT REMOTE
542	001426	001015				BNE	ST3	:IF SO: BR
543	001430	005300				DEC	R0	
544	001432	001372				BNE	ST1	:DELAY FOR SELECT REMOTE
545	001434	105767	177342			TSTB	ACT11M	:ACT MODE? ++ C.W
546	001440	001403				BEQ	1\$:BRANCH - IF NO ++ C.W
547	001442	105267	177142			INCB	UDES	:GET NEXT UNIT ++ C.W
548	001446	000731				BR	ST	:CONTINUE ++ C.W
549	001450	012704	007020		1\$:	MOV	#MSG21,R4	
550	001454	004767	003332			JSR	PC,TTOUT	:PRINT NOT AVAILABLE
551	001460	000670				BR	ST0	:REDO REQUEST
552	001462	052767	060000	177120	ST3:	BIS	#60000,UDES	:SET TO 800 BPI 9 CHAN
553	001470	000240			ST4:	NOP		
554	001472	012777	010000	177102		MOV	#10000,#MTC	:POWER CLEAR
555	001500	012702	000646			MOV	#TOB,R2	:GET START OF TABLE
556	001504	012700	000027			MOV	#27,R0	:SET SIZE OF TABLE
557	001510	005022			ST5:	CLR	(R2)+	:CLEAR TABLE
558	001512	005300				DEC	R0	
559	001514	001375				BNE	ST5	:DO ALL
560	001516	105767	177260			TSTB	ACT11M	:ACT MODE? ++ C.W
561	001522	001002				BNE	TSCD	:BRANCH - IF YES ++ C.W
562	001524	005067	177144			CLR	PCNTR	:CLEAR PASS COUNTER

```

564 ;TEST SCHEDULAR*****
565
566 001530 000240 TSCD: NOP
567 001532 005067 177134 CLR STFLG ;CLEAR SINGLE TEST FLAG
568 001536 017700 177056 MOV QSWR,RO ;GET SWITCH REGISTER
569 001542 042700 177700 BIC #177700,RO ;MASK TEST SELECT
570 001546 005700 TST RO ;SEE IF SINGLE TEST SELECT
571 001550 001046 BNE STSCD ;IF SO: BR
572 001552 012767 000732 177104 MOV #TSTTBL,LTADD ;GET TABLE START
573 001560 062767 000004 177076 TSCDD: ADD #4,LTADD
574 001566 016767 177072 LTADD,ITRLP ;SET ITERATION ADDRESS
575 001574 062767 000002 177064 ADD #2,ITRLP
576 001602 005777 177056 TST QLTADD ;SEE IF END OF CYCLE
577 001606 001002 BNE TSCD1 ;IF NOT: BR
578 001610 000167 000124 JMP TEND ;GO TO END ROUTINE
579 001614 005067 177074 TSCD1: CLR HDRFL ;CLEAR HEADER FLAG
580 001620 017700 177040 MOV QLTADD,RO ;GET TEST ADDRESS
581 001624 000110 JMP (RO) ;GO TO TEST
582 001626 032777 002000 176764 TSCD2: BIT #2000,QSWR ;SEE IF HALT ON TEST
583 001634 001401 BEQ TSCD3 ;IF NOT: BR
584 001636 000000 HALT
585 001640 004767 003760 TSCD3: JSR PC,CKSWR ;TEST FOR IG
586 001644 005767 177022 TST STFLG ;SEE IF SINGLE TEST
587 001650 001743 BEQ TSCDD ;IF NOT: BR
588 001652 017700 176742 MOV QSWR,RO
589 001656 042700 177760 BIC #177760,RO ;GET TEST NUMBER
590 001662 005700 TST RO ;SEE IF ALL TESTS
591 001664 001721 BEQ TSCD ;IF SO: BR
592 001666 012767 000001 176776 STSCD: MOV #1,STFLG ;SET SINGLE TEST FLAG
593 001674 022700 000005 CMP #5,RO ;SEE IF EXCEEDED TEST NUMBER
594 001700 003417 BLE TEND ;IF SO: BR
595 001702 000241 CLC
596 001704 006100 ROL RO
597 001706 006100 ROL RO ;POSITION NUMBER
598 001710 012767 000732 176746 MOV #TSTTBL,LTADD ;GET START OF TABLE
599 001716 060067 176742 ADD RO,LTADD ;SET POINTER
600 001722 016767 176736 MOV LTADD,ITRLP
601 001730 062767 000002 176736 ADD #2,ITRLP ;SET ITERATION ADDRESS
602 001736 000726 BR TSCD1 ;GO DO TEST
603 001740 105767 177036 TEND: TSTB ACT11M ;ACT MODE? ++ C.W
604 001744 001404 BEQ $DONE ;BRANCH - IF NO ++ C.W
605 001746 105267 176636 INCB UDES ;GET NEXT UNIT ++ C.W
606 001752 000167 177354 JMP ST ;AND CONTINUE TESTING ++ C.W
607 001756 012704 006470 $DONE: MOV #MSG3,R4
608 001762 004767 003024 JSR PC,ITOUT ;PRINT END OF PASS
609 001766 016703 176702 MOV PCNTR,R3
610 001772 004767 003212 JSR PC,OCIP ;PRINT PASS NUMBER
611 001776 013700 000042 MOV Q#42,RO ;GET MONITOR ADDRESS ++ C.W
612 002002 001405 BEQ HERE ;BRANCH - IF NOT AUTO MODE ++ C.W
613 002004 000005 RESET ;CLEAR THE WORLD ++ C.W
614 002006 004710 $ENDAD: JSR PC,(RO) ;GO TO MONITOR'S ADDRESS ++ C.W
615 002010 000240 NOP
616 002012 000240 NOP
617 002014 000240 NOP
618 002016 000240 HERE: NOP
619 002020 032777 004000 176572 BIT #4000,QSWR ;SEE IF HALT ON PASS

```

620	002026	001001		BNE	TENDX	: IF NOT: BP
621	002030	000000		HALT		
622	002032	105767	176744	TENDX: TS*B	ACT11M	: ACT MODE? ++ C.W
623	002036	001006		BNE	IS	: BRANCH - IF YES ++ C.W
624	002040	004767	003560	JSR	PC,CF,SWR	: TEST FOR IG
625	002044	005267	176624	INC	PCNTR	: BUMP PASS COUNTER
626	002050	000167	177454	JMP	TSCU	: RESTART
627	002054	005267	176614	IS:	PCNTR	: BUMP COUNTER IN ACT MODE ++ C.W
628	002060	000167	177234	JMP	\$ACT	: RESTART IN ACT MODE ++ C.W
629						

```

630
631
632
633
634
635
636
637
638
639 002064 000240
640 002066 012767 007075 176564
641 002074 012702 007352
642 002100 112722 0003
643 002104 005000
644 002106 110022
645 002110 005200
646 002112 022700 0000
647 002116 001373
648 002120 004767 001046
649 002124 012767 000004 176552
650 002132 012767 007353 176536
651 002140 012767 177772 176532
652 002146 012767 006715 176506
653 002154 004767 001262
654 002160 000240
655 002162 004767 001504
656 002166 012767 177777 176506
657 002174 004767 001156
658 002200 012702 007454
659 002204 012700 000010
660 002210 012722 177777
661 002214 005300
662 002216 001374
663 002220 012767 000002 176456
664 002226 012767 007454 176442
665 002234 012767 177772 176436
666 002242 012767 006732 176412
667 002250 004767 001166
668 002254 000240
669 002256 004767 001410
670 002262 012701 007353
671 002266 012702 007454
672 002272 012700 000006
673 002276 004767 001656
674 002302 004767 002146
675 002306 000167 177314

```

```

*****
:TEST 1: WRITE FROM ODD BYTE
:
:THIS TEST WILL WRITE A SIX (6) BYTE RECORD
:FROM AN ODD BYTE STARTING ADDRESS. THE RECORD
:WILL BE READ BACK INTO AN EVEN STARTING ADDRESS
:TO TEST FOR PROPER TRANSFER.
*****
LT1:  NOP
      MOV      #LT1MSG,EMADDR      :SET HEADER
      MOV      #WDATA,R2          :GET BUFFER START
      MOV      #377,(R2)+         :INSERT BACKGROUND DATA
      CLR      R0
      MOV      R0,(R2)+           :LOAD WRITE BUFFER (0.1.2.3.4.5)
      INC      R0
      CMP      #6,R0
      BNE     LT1B
      JSR     PC,PWND              :GO REWIND
      MOV      #4,FUN              :SET WRITE FUNCTION CODE
      MOV      #WDATA+1,BADR       :SET DATA POINTER
      MOV      #-6,BYTES           :SET SIZE OF RECORD
      MCV      #MSG17,ERRAD        :SET WRITE ERROR CODE
      JSR     PC,EXEC              :GO EXECUTE COMMAND
LT1C:  NOP
      JSR     PC,ERCHK             :GO CHECK FOR STATUS ERROR
      MOV      #-1,SCNT
      JSR     PC,BKSP              :GO BACKSPACE ONE RECORD
      MOV      #RDATA,R2
      MOV      #10,R0
      MOV      #-1,(R2)+          :BACKGROUND READ BUFFER
      DEC      R0
      BNE     LT1D
      MOV      #2,FUN              :DO ALL
      MOV      #RDATA,BADR         :SET READ FUNCTION CODE
      MOV      #-6,BYTES           :SET READ POINTER
      MCV      #MSG18,ERRAD        :SET SIZE OF RECORD
      JSR     PC,EXEC              :SET READ ERROR CODE
      JSR     PC,EXEC              :GO DO READ
LT1E:  NOP
      JSR     PC,ERCHK             :GO CHECK ERRORS
      MOV      #WDATA+1,R1         :SET EXPT DATA POINTER
      MOV      #RDATA,R2          :SET RCVD DATA POINTER
      MOV      #6,R0               :SET SIZE OF RECORD
      JSR     PC,DCHK              :GO CHECK DATA
      JSR     PC,ITER              :GO SEE IF ITERATIONS
      JMP     *SCD2                :RETURN TO SCHEDULAR

```

```

676
677
678
679
680
681
682
683
684
685 002312 000240
686 002314 012767 007133 176336
687 002322 012702 007352
688 002326 005000
689 002330 110022
690 002332 005200
691 002334 022700 000006
692 002340 001373
693 002342 004767 000624
694 002346 012767 000004 176330
695 002354 012767 007352 176314
696 002362 012767 177772 176310
697 002370 004767 001046
698 002374 000240
699 002376 012767 006715 176256
700 002404 004767 001262
701 002410 012767 177777 176264
702 002416 004767 000734
703 002422 012702 007454
704 002426 012700 000010
705 002432 012722 177777
706 002436 005300
707 002440 001374
708 002442 012767 000002 176234
709 002450 012767 007455 176220
710 002456 012767 177772 176214
711 002464 004767 000752
712 002470 000240
713 002472 012767 006732 176162
714 002500 004767 001166
715 002504 012701 007352
716 002510 012702 007455
717 002514 012700 000006
718 002520 004767 001434
719 002524 004767 001724
720 002530 000167 177072

```

```

*****
:TEST 2: READ INTO ODD BYTE
:
:THIS TEST WILL WRITE A SIX (6) BYTE RECORD
:FROM AN EVEN BYTE STARTING ADDRESS. THE RECORD
:WILL BE READ BACK INTO AN ODD STARTING ADDRESS
:TO TEST FOR PROPER TRANSFER.
*****
LT2:  NOP
      MOV #LT2MSG,EMADDR ;SET HEADER POINTER
      MOV #WDATA,R2 ;POINT TO START OF WRITE BUFFER
      CLR RO
LT2B: MOVB RO,(R2)+ ;LOAD DATA PATTERN
      INC RO ;BUMP PATTERN
      CMP #6,RO ;SEE IF DONE
      BNE LT2B ;IF NOT: BR
      JSR PC,RWND ;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,RO ;SET SIZE
LT2D:  MOV #-1,(R2)+ ;BACKGROUND POINTER
      DEC RO ;SEE IF DONE
      BNE LT2D ;IF NOT: BR
      MOV #2,FUN ;SET READ FUNCTION CODE
      MOV #RDATA+1,BADR ;SET START OF READ BUFFER
      MOV #-6,BYTES ;SET SIZE OF RECORD
      JSR PC,EXEC ;GO EXECUTE COMMAND
LT2E:  NOP
      MOV #MSG18,ERRAD ;SET ERROR CODE
      JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
      MOV #WDATA,R1 ;POINT TO EXPT DATA
      MOV #RDATA+1,R2 ;POINT TO RCVD DATA
      MOV #1,RO ;SET SIZE OF RECORD
      JSR PC,DCHK ;GO CHECK DATA
      JSR PC,ITER ;GO SEE IF ITERATION
      JMP TSCD2 ;RETURN TO SCHEDULAR

```


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
793
794
795
796
797
798
799
800
801
802
803
804
805
806

002764 000240
002766 012767 007215 175664
002774 004767 000172
003000 012700 000014
003004 012767 000014 175672
003012 012767 007352 175656
003020 012767 177760 175652
003026 012767 006715 175626
003034 004767 000402
003040 004767 000626
003044 012767 177777 175630
003052 004767 000300
003056 005300
003060 001351
003062 000240
003064 004767 000102
003070 012767 000200 175540
003076 012767 007454 175572
003104 012767 177760 175566
003112 012767 000002 175564
003120 012767 006773 175534
003126 004767 000310
003132 004767 000534
003136 000240
003140 012767 000040 175470
003146 012767 000002 175460
003154 004767 001274
003160 012767 000010 175446
003166 000167 176434

LT4:
LT4A:
LT4B:
LT4IT:
LT4C:

```
*****
:TEST 4: OPI TOO SHORT
:
:THIS TEST WILL ERASE APPROXIMATELY FOUR (4) FEET
:OF TAPE BY WRITING WITH IRG, BAC SPACING
:ONE (1) RECORD AND REPEATING THIS SEQUENCE
:12(10) TIMES. TAPE WILL REWIND AND BE READ
:FORWARD. THE FIRST RECORD ON TAPE SHOULD BE
:REACHED BEFORE OPI TIMES OUT.
:*****
NOP
MOV #LT4MSG,EMADDR ;SET HEADER
JSR PC,RWND ;GO REWIND
MOV #14,RO ;SET NUMBER OF WRITE IRG/BACKSPACES
LT4A: MOV #14,FUN ;SET WRITE IRG FUNCTION CODE
MOV #WDATA,BADR ;SET START OF WRITE BUFFER
MOV #-20,BYTES ;SET SIZE OF RECORD
MOV #MSG17,ERRAD ;SET ERROR CODE
JSR PC,EXEC ;GO EXECUTE COMMAND
LT4B: JSR PC,ERCHK ;GO CHECK FOR STATUS ERROR
MOV #-1,SCNT
JSR PC,BKSP ;GO BACKSPACE ONE RECORD
DEC RO ;SEE IF DONE ALL SEQUENCES
BNE LT4A ;IF NOT: BR
LT4IT: NOP
JSR PC,RWND ;REWIND
MOV #200,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 #40,STALL ;RESET NJRMAL STALL
MOV #2,ITAMT ;SET TO TWO (2) ITERATIONS
JSR PC,ITER ;GO SEE IF ITERATIONS
MCMV #10,ITAMT ;RESET ITERATIONS
JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

807                                     ;REWIND SUBROUTINE*****
808
809 003172 000240 RWND: NOP
810 003174 016777 175410 175400 MOV UDES,DMTC ;SELECT UNIT
811 003202 032777 000040 175370 BIT #40,DMTS ;SEE IF AT BOT
812 003210 001056 BNE RWNDXX ;IF SO: BR
813 003212 052777 000017 175362 BIS #17,DMTC ;START REWIND
814 003220 105777 175356 IS: TSTB DMTC
815 003224 100375 BPL IS ;AWAIT CUR
816 003226 032777 000001 175344 RWND1: BIT #1,DMTS ;AWAIT TUR
817 003234 001774 BEQ RWND1
818 003236 032777 000040 175334 BIT #40,DMTS ;SEE IF BOT SET
819 003244 001040 BNE RWNDXX ;IF SO: BR
820 003246 032777 020000 175344 BIT #20000,DSWR ;SEE IF PRINT ERROR
821 003254 001030 BNE RWNDX ;IF NOT: BR
822 003256 016704 175376 MOV EMADDR,R4
823 003262 004767 001524 JSR PC,TTOUT ;PRINT HEADER
824 003266 012704 006511 MOV #MSG4,R4
825 003272 004767 001514 JSR PC,TTOUT ;PRINT REWIND ERROR
826 003276 012704 006541 MOV #MSG5,R4
827 003302 004767 001504 JSR PC,TTOUT ;PRINT MTS TAG
828 003306 017703 175266 MOV DMTS,R3
829 003312 004767 001662 JSR PC,OTPE ;PRINT MTS
830 003316 012704 006550 MOV #MSG6,R4
831 003322 004767 001464 JSR PC,TTOUT ;PRINT MTC TAG
832 003326 017703 175250 MOV DMTC,R3
833 003332 004767 001642 JSR PC,OTPE ;PRINT MTC
834 003336 005777 175256 RWNDX: TST DSWR ;SEE IF HALT ON ERROR
835 003342 100001 BPL RWNDXX ;IF NOT: BR
836 003344 000000 HALT
837 003346 004767 002252 RWNDXX: JSR PC,CKSWR ;TEST FOR IG
838 003352 000240 NOP
839 003354 000207 RTS ;RETURN
840
841                                     ;BACKSPACE SUBROUTINE*****
842
843 003356 000240 BKSP: NOP
844 003360 016777 175224 175214 MOV UDES,DMTC ;SELECT UNIT
845 003366 016777 175310 175210 MOV SCNT,DMTBC ;SET NUMBER OF RECORDS TO SPACE
846 003374 052777 000013 175200 BIS #13,DMTC ;START SPACE REVERSE
847 003402 032777 000001 175170 BKSP1: BIT #1,DMTS
848 003410 001774 BEQ BKSP1 ;AWAIT TUR
849 003412 012767 000001 175250 MOV #1,SPFLG ;SET SPACE FLAG
850 003420 012767 006577 175234 MOV #MSG9,ERRAD
851 003426 004767 000240 JSR PC,ERCHK ;GO CHECK FOR ERROR
852 003432 005067 175232 CLR SPFLG ;CLEAR SPACE FLAG
853 003436 000240 NOP
854 003440 000207 RTS ;RETURN
855

```

```

856                                     ;COMMAND EXECUTE SUBROUTINE*****
857
858 003442 000240 EXEC: NOP
859 003444 005005 CLR R5
860 003446 032777 000200 175126 EXEC0: BIT #200, @MTC ;SEE IF CUR
861 003454 001021 BNE EXEC2 ;IF SO: BR
862 003456 005305 DEC R5 ;SEE IF TIMED OUT
863 003460 001372 BNE EXEC0 ;IF NOT: BR
864 003462 005767 175226 TST HDRFL ;SEE IF DONE HEADER
865 003466 001004 BNE EXEC1 ;IF SO: BR
866 003470 016704 175164 MOV EMADDR, R4
867 003474 004767 001312 JSR PC, TOUT ;ELSE PRINT HEADER
868 003500 012704 006622 EXEC1: MOV #MSG10, R4
869 003504 004767 001302 JSR PC, TOUT ;PRINT NOT READY ERROR
870 003510 005777 175104 TST @SWR ;SEE IF HALT ON ERRCR
871 003514 100001 BPL EXEC2 ;IF NOT: BR
872 003516 000000 HALT
873 003520 004767 002100 EXEC2: JSR PC, CKSWR ;TEST FOR IG
874 003524 000240 NOP
875 003526 016777 175056 175046 MOV UDES, @MTC ;SELECT UNIT
876 003534 016777 175136 175044 MOV BADR, @MTBA ;SET BUS MEMORY ADDRESS
877 003542 016777 175132 175034 MOV BYTES, @MTBC ;SET BYTE COUNT
878 003550 016701 175130 MOV FUN, R1 ;GET FUNCTION
879 003554 052701 000101 BIS #101, R1 ;SET IN GO BIT AND INTERRUPT ENABLE
880 003560 050177 175016 BIS R1, @MTC ;LOAD COMMAND+GO+IE
881 003564 000240 NOP
882 003566 005077 175024 CLR @PSW ;ALLOW INTERRUPTS
883 003572 016767 175040 175052 MOV STALL, TEMP1 ;SET READY STALL
884 003600 005001 CLR R1
885 003602 005301 EXEC3: DEC R1
886 003604 001376 BNE EXEC3 ;AWAIT INTERRUPT
887 003606 005367 175040 DEC TEMP1
888 003612 001373 BNE EXEC3
889 003614 032777 020000 174776 BIT #20000, @SWR ;SEE IF PRINT ERROR
890 003622 001013 BNE EXECX ;IF NOT: BR
891 003624 005767 175064 TST HDRFL ;SEE IF DONE HEADER
892 003630 001004 BNE EXEC4 ;IF SO: BR
893 003632 016704 175022 MOV EMADDR, R4
894 003636 004767 001150 JSR PC, TOUT ;PRINT HEADER
895 003642 012704 006637 EXEC4: MOV #MSG11, R4
896 003646 004767 001140 JSR PC, TOUT ;PRINT NO INTERRUPT MESSAGE
897 003652 005777 174742 EXECX: TST @SWR ;SEE IF HALT ON ERROR
898 003656 100001 BPL EXECXX ;IF NOT: BR
899 003660 000000 HALT
900 003662 004767 001736 EXECXX: JSR PC, CKSWR ;TEST FOR IG
901 003666 000240 NOP
902 003670 000207 RTS PC ;RETURN TO CALLER
903

```

```

904 ;STATUS ERROR CHECK SUBROUTINE*****
905
906 003672 005777 174704 ERCHK: TST @MTC ;SEE IF ANY EPROR BITS
907 003676 100002 BPL ERCHK1 ;IF NOT: BR
908 003700 000167 000044 JMP ERPT ;ELSE PRINT ERROR
909 003704 005777 174674 ERCHK1: TST @MTBC ;SEE IF BYTE COUNT IS ZERO
910 003710 001402 BEQ ERCHK2 ;IF SO: BR
911 003712 000167 000032 JMP ERPT ;ELSE PRINT ERROR
912 003716 016703 174756 ERCHK2: MOV BYTES,R3
913 003722 005403 NEG R3
914 003724 066703 174746 ADD BADR,R3 ;SET EXPT BUS ADDRESS
915 003730 005767 174734 TST SPFLG ;SEE IF SPACE OPERATION
916 003734 001401 BEQ ERCHK3 ;IF NOT: BR
917 003736 000207 RTS PC
918 003740 020377 174642 ERCHK3: CMP R3,@MTBA ;SEE IF EXPT=RCVD
919 003744 001001 BNE ERPT ;IF NOT: BR
920 003746 000207 RTS PC ;ELSE EXIT
921 003750 000240 ERPT: NOP
922 003752 032777 020000 174640 BIT #20000,@SWR ;SEE IF SHOULD PRINT
923 003760 001067 BNE ERPTX ;IF NOT: BR
924 003762 005767 174726 TST HDRFL ;SEE IF DONE HEADER
925 003766 001006 BNE ERPT1 ;IF SO: BR
926 003770 016704 174664 MOV EMADDR,R4
927 003774 004767 001012 JSR PC,TTOUT ;ELSE PRINT HEADER
928 004000 005267 174710 INC HDRFL ;SET FLAG
929 004004 016704 174652 ERPT1: MOV ERRAD,R4
930 004010 004767 000776 JSR PC,TTOUT ;PRINT ERROR CODE
931 004014 012704 006541 MOV #MSG5,R4
932 004020 004767 000766 JSR PC,TTOUT ;PRINT MTS TAG
933 004024 017703 174550 MOV @M1S,R3
934 004030 004767 001144 JSR PC,OC1PE ;PRINT MTS
935 004034 012704 006550 MOV #MSG6,R4
936 004040 004767 000746 JSR PC,TTOUT ;PRINT MTC TAG
937 004044 017703 174532 MOV @M1C,R3
938 004050 004767 001124 JSR PC,OC1PE ;PRINT MTC
939 004054 012704 006557 MOV #MSG7,R4
940 004060 004767 000726 JSR PC,TTOUT ;PRINT BYTE COUNT TAG
941 004064 017703 174514 MOV @M1BC,R3
942 004070 004767 001114 JSR PC,OC1P ;PRINT BYTE COUNT
943 004074 005767 174570 TST SPFLG ;SEE IF PRINT BA
944 004100 001017 BNE ERPTX ;IF NOT: BR
945 004102 012704 006567 MOV #MSG8,R4
946 004106 004767 000700 JSR PC,TTOUT ;PRINT BUS ADDRESS TAG
947 004112 017703 174470 MOV @M1BA,R3
948 004116 004767 001066 JSR PC,OC1P ;PRINT CURRENT ADDRESS
949 004122 016703 174552 MOV BYTES,R3
950 004126 005403 NEG R3
951 004130 066703 174542 ADD BADR,R3
952 004134 004767 001050 JSR PC,OC1P ;PRINT EXPT ADDRESS
953 004140 005777 174454 ERPTX: TST @SWR ;SEE IF HALT ON ERROR
954 004144 100001 BPL ERPTXX ;IF NOT: BR
955 004146 000000 HALT
956 004150 004767 001450 ERPTXX: JSR PC,CKSWR ;TEST FOR IG
957 004154 000167 000240 JMP SCOPE ;GO SEE IF SCOPE ON ERROR

```

```

;DATA CHECK SUBROUTINE*****
958
959
960 004160 000240          DCHK:  NOP
961 004162 005067 174522  DCHK:  CLR      CRCNT      ;CLEAR COUNTER
962 004166 121112          DCHKO:  CMPB    (R1), (R2)  ;SEE IF EXPT DATA=RCVD DATA
963 004170 001007          DCHKO:  BNE     DCHKE    ;IF NOT: BR
964 004172 005267 174512  DCHK1:  INC     CRCNT      ;BUMP CHARACTER COUNTER
965 004176 122122          DCHK1:  CMPB    (R1)+, (R2)+
966 004200 005300          DCHK1:  DEC     RO        ;SEE IF DONE
967 004202 001371          DCHK1:  BNE     DCHKO    ;IF NOT: BR
968 004204 000167 000150  DCHK1:  JMP     DCHKX    ;ELSE GO TO EXIT ROUTINE
969 004210 000240          DCHKE:  NOP
970 004212 012767 000001 174472  DCHKE:  MOV     #1, DERFL  ;SET ERROR FLAG
971 004220 032777 020000 174372  DCHKE:  BIT     #20000, #SWR ;SEE IF PRINT ERROR
972 004226 001054          DCHKE:  BNE     DCHKX    ;IF NOT: BR
973 004230 005767 174460  DCHKE:  TST    HDRFL     ;SEE IF DONE HEADER
974 004234 001007          DCHKE:  BNE     DCHKE1   ;IF SO: BR
975 004236 016704 174416  DCHKE:  MOV     EMADDR, R4
976 004242 004767 000544  DCHKE:  JSR    PC, TTOUT  ;PRINT HEADER
977 004246 012767 000001 174440  DCHKE:  MOV     #1, HDRFL  ;SET HEADER FLAG
978 004254 012704 006657  DCHKE1: MOV     #MSG12, R4
979 004260 005767 174432  DCHKE1: TST    PFLG      ;SEE IF PRINTED DATA EPROP TAG
980 004264 001004          DCHKE1: BNE     DCHKE2   ;IF SO: BR
981 004266 005267 174424  DCHKE1: INC     PFLG
982 004272 004767 000514  DCHKE1: JSR    PC, TTOUT  ;ELSE PRINT DATA ERROR TAG
983 004276 012704 006673  DCHKE2: MOV     #MSG13, R4
984 004302 004767 000504  DCHKE2: JSR    PC, TTOUT  ;PRINT CHAR NUMBER TAG
985 004306 016703 174376  DCHKE2: MOV     CRCNT, R3
986 004312 004767 000672  DCHKE2: JSR    PC, OCTP   ;PRINT CHAR NUMBER
987 004316 012704 006701  DCHKE2: MOV     #MSG14, R4
988 004322 004767 000464  DCHKE2: JSR    PC, TTOUT  ;PRINT GOOD TAG
989 004326 111103          DCHKE2: MOVB   (R1), R3
990 004330 004767 001102  DCHKE2: JSR    PC, DOUT   ;PRINT GOOD CHARACTER
991 004334 012704 006706  DCHKE2: MOV     #MSG15, R4
992 004340 004767 000446  DCHKE2: JSR    PC, TTOUT  ;PRINT BAD TAG
993 004344 111203          DCHKE2: MOVB   (R2), R3
994 004346 004767 001064  DCHKE2: JSR    PC, DOUT   ;PRINT BAD CHARACTER
995 004352 000240          DCHKE2: NOP
996 004354 000167 177612  DCHKE2: JMP     DCHK1    ;CONTINUE FOR ALL BYTES
997 004360 000240          DCHKX:  NOP
998 004362 005767 174324  DCHKX:  TSI     DERFL     ;SEE IF ANY ERROR
999 004366 001404          DCHKX:  BEQ    DCHKXX    ;IF NOT: BR
1000 004370 005777 174224  DCHKX:  TST    #SWR     ;SEE IF HALT ON ERROR
1001 004374 100001          DCHKX:  BPL    DCHKXX    ;IF NOT: BR
1002 004376 000000          DCHKX:  HALT
1003 004400 004767 001220  DCHKXX: JSR    PC, CKSWR   ;TEST FOR IG
1004 004404 000240          DCHKXX: NOP
1005 004406 005067 174304  DCHKXX: CLR     PFLG      ;CLEAR PRINT FLAG
1006 004412 005067 174274  DCHKXX: CLR    DERFL     ;CLEAR DATA ERROR FLAG
1007 004416 000207          DCHKXX: PC        ;RETURN

```

```

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

```



```

1103 ;TTY ENTRY ERROR SUBROUTINE*****
1104
1105 004722 012704 006713 TINNER: MOV #MSG16,F4
1106 004726 004767 000060 JSR PC,TTOUT ;PRINT?
1107 004732 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
1108 004736 000207 RTS PC ;REDO VALUE ENTRY
1109
1110 ;TTY READ SUBROUTINE*****
1111
1112 004740 005077 173660 TTIN: CLR @TKS
1113 004744 005077 173656 CLR @TKB
1114 004750 005067 173674 CLR TIB
1115 004754 005277 173644 INC @TKS
1116 004760 105777 173640 TTINI: TSTB @TKS
1117 004764 100375 BPL TTINI
1118 004766 017767 173634 173654 MOV @TKB,TIB
1119 004774 105777 173630 TTIN2: TSTB @TPS
1120 005000 100375 BPL TTIN2
1121 005002 116777 173642 173622 MOVB TIB,@TPB
1122 005010 000207 RTS PC
1123
1124 ;TTY OUTPUT SUBROUTINE*****
1125
1126 005012 112467 173630 TTOUT: MOVB (R4)+,TOB
1127 005016 122767 000043 173622 CMPB #43,TOB
1128 005024 001452 BEQ TEX
1129 005026 122767 000045 173612 CMPB #45,TOB
1130 005034 001407 BEQ TCRLF
1131 005036 122767 000041 173602 CMPB #41,TOB
1132 005044 001443 BEQ TBELL
1133 005046 004767 000064 JSR PC,TOG
1134 005052 000757 BR TTOUT
1135 005054 112767 000015 173564 TCRLF: MOVB #15,TOB
1136 005062 004767 000050 JSR PC,TOG
1137 005066 012703 000004 MOV #4,R3
1138 005072 005067 173550 TCRLFA: CLR TOB
1139 005076 004767 000034 JSR PC,TOG
1140 005102 005303 DEC R3
1141 005104 001372 BNE TCRLFA ;DC FILLERS
1142 005106 112767 000012 173532 MOVB #12,TOB
1143 005114 004767 000016 JSR PC,TOG
1144 005120 105767 173604 TSTB RDSW
1145 005124 100401 BMI IS
1146 005126 000731 BR TTOUT
1147 005130 005067 173574 IS: CLR RDSW
1148 005134 000406 BR TEX
1149 005136 105777 173466 TOG: TSTB @TPS
1150 005142 100375 BPL TOG
1151 005144 116777 173476 173460 MOVB TOB,@TPB
1152 005152 000207 RTS PC
1153 005154 012703 000002 TEX: MOV #2,R3
1154 005160 012767 000007 173460 TBELL: MOV #7,TOB
1155 005166 004767 177744 JSR PC,TOG
1156 005172 005303 DEC R3
1157 005174 001371 BNE TBELA
1158 005176 000705 BR TTOUT

```

```

:OCTAL OUTPUT SUBROUTINE*****
1159
1160
1161 005200 012767 000001 300226 OCTPE: MOV #1,OFL
1162 005206 000402 BR OCTPE1
1163 005210 005067 000220 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
1164 005214 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
1165 005216 001007 BNE OCTPD ;IF NOT ZERO: BR
1166 005220 005767 000210 TST OFL ;SEE IF PRINT ALL 0
1167 005224 001004 BNE OCTPD ;IF SO: BR
1168 005226 004767 000162 JSR PC,OCTPG1 ;ELSE PRINT ZERO
1169 005232 000167 000120 JMP OCTP3 ;SPACE AND EXIT
1170 005236 032704 100000 OCTPD: BIT #100000,R4 ;SEE IF MSD = 1
1171 005242 001406 BEQ OCTP1 ;IF NOT: BR
1172 005244 012704 000001 MOV #1,R4
1173 005250 004767 000116 JSR PC,OCTPG ;PRINT 1
1174 005254 000167 000006 JMP OCTP2
1175 005260 005004 OCTP1: CLR R4
1176 005262 004767 000104 JSR PC,OCTPG ;PRINT 0
1177 005266 010304 OCTP2: MOV R3,R4
1178 005270 006004 ROR R4
1179 005272 006004 ROR R4
1180 005274 006004 ROR R4 ;POSITION DIGIT
1181 005276 006004 ROR R4
1182 005300 000304 SWAB R4
1183 005302 004767 000064 JSR PC,OCTPG ;PRINT DIGIT 2
1184 005306 010304 MOV R3,R4
1185 005310 006004 ROR R4
1186 005312 000304 SWAB R4
1187 005314 004767 000052 JSR PC,OCTPG ;PRINT DIGIT 3
1188 005320 010304 MOV R3,R4
1189 005322 006104 ROL R4
1190 005324 006104 ROL R4
1191 005326 000304 SWAB R4
1192 005330 004767 000036 JSR PC,OCTPG ;PRINT DIGIT 4
1193 005334 010304 MOV R3,R4
1194 005336 006004 ROR R4
1195 005340 006004 ROR R4
1196 005342 006004 ROR R4
1197 005344 004767 000022 JSR PC,OCTPG
1198 005350 010304 MOV R3,R4
1199 005352 004767 000014 JSR PC,OCTPG ;PRINT DIGIT 5
1200 005356 012767 000240 173262 OCTP3: MOV #240,T09
1201 005364 004767 177546 JSR PC,T09 ;PRINT SPACE
1202 005370 000207 RTS PC ;EXIT

```

```

1203
1204
1205
1206 005372 042704 177770
1207 005376 001004
1208 005400 005767 000030
1209 005404 001001
1210 005406 000207
1211 005410 005267 000020
1212 005414 052704 000260
1213 005420 010467 173222
1214 005424 004767 177506
1215 005430 010304
1216 005432 000207
1217 005434 000000
1218
1219
1220
1221 005436 005067 173204
1222 005442 012704 000010
1223 005446 110367 173174
1224 005452 105777 173152
1225 005456 100375
1226 005460 132767 000200 173160
1227 005466 001404
1228 005470 012777 000061 173134
1229 005476 000403
1230 005500 012777 000060 173124
1231 005506 006167 173134
1232 005512 005304
1233 005514 001356
1234 005516 000207
1235 005520 016703 173132
1236 005524 000303
1237 005526 004767 177704
1238 005532 016703 173120
1239 005536 004767 177674
1240 005542 000207
1241
1242
1243
1244
1245 005544 013746 000006
1246 005550 013746 000004
1247 005554 012737 005574 000004
1248 005562 022777 177777 173030
1249 005570 001402
1250 005572 000407
1251 005574 022626
1252 005576 012767 000176 173014
1253 005604 012767 000174 173010
1254 005612 012637 000004
1255 005616 012637 000006
1256 005622 000207
1257
1258

```

```

: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
      MOV #10,R4 ;SET NUMBER TO PRINT
      MOV R3,TOB
DOUT1: TSTB #TPB
      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
      JSP PC,DOUT
      RTS PC

SUSWR: MOV @#6,-(SP) ;SAVE VECTORS
      MOV @#4,-(SP)
      MOV #15,@#4 ;SET UP FOR TIMEOUT
      CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
      BEQ 2$
      BR 3$
1$: CMP (SP)+,(SP)+ ;ADJUST STACK
2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
3$: MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
      MOV (SP)+,@#4 ;RESTORE VECTORS
      MOV (SP)+,@#6
      RTS PC

```

```

1259 005624 022767 000176 172766 CKSWR: CMP      #SWREG,SWR      :SOFTWARE SWITCH REG PRESENT
1260 005632 001041          BNE      OUT        :NO GET OUT
1261 005634 105777 172764      TSTB     #TKS          :YES WAIT FOR
1262 005640 100036          BPL      OUT        :READY GET CHARACTER
1263 005642 017767 172760 173000 MOV      #TKB,TIB   :AND STRIP OFF
1264 005650 042767 177600 172772 BIC      #177600,TIB :THE GARBAGE
1265 005656 022767 000007 172764 CMP      #7,TIB     :IS IT A 'IG
1266 005664 001024          BNE      OUT
1267 005666 012704 007245      MOV      #SCNTG,R4
1268 005672 004767 177114      JSR     PC,TTOUT
1269 005676 012704 007251 CNTLU: MOV      #MSWR,R4
1270 005702 004767 177104      JSR     PC,TTOUT
1271 005706 017703 172706      MOV      #SWR,R3
1272 005712 004767 177262      JSR     PC,CTPE
1273 005716 012704 007263      MOV      #MNEW,R4
1274 005722 004767 177064      JSR     PC,TTOUT
1275 005726 005037 000726      CLR     #TEMPST
1276 005732 004767 000002      JSR     PC,$READ
1277 005736 000207          OUT:    RTS
1278
1279 005740 005067 172762 $READ: CLR     TEMPST
1280 005744 012767 000007 172752 MOV      #7,COUNT
1281 005752 004767 176762 1$:     JSR     PC,TTIN
1282 005756 042767 177600 172664 BIC      #177600,TIB :GO READ A CHARACTER
1283 005764 122767 000025 172656 CMPB     #25,TIB   :STRIP OFF GARBAGE
1284 005772 001002          BNE     11$
1285 005774 005726          TST     (SP)+
1286 005776 000737          BR     CNTLU
1287 006000 122767 000012 172642 11$:  CMPB     #12,TIB
1288 006006 001016          BNE     2$
1289 006010 005757 172032          TST     46
1290 006014 001406          BEQ     10$
1291 006016 005726          TST     (SP)+
1292 006020 016716 172022          MOV     46,(SP)
1293 006024 062716 000010          ADD     #10,(SP)
1294 006030 000742          BR     OUT
1295 006032 012704 007273 10$:  MOV      #NOACT,R4
1296 006036 004767 176750          JSR     PC,TTOUT
1297 006042 000754          BR     3$
1298 006044 122767 000015 172576 2$:  CMPB     #15,TIB
1299 006052 001013          BNE     4$
1300 006054 012767 000200 172646      MOV      #200,RDSW
1301 006062 004767 176766          JSR     PC,TCRLF
1302 006066 022767 000007 172630      CMP      #7,COUNT
1303 006074 001037          BNE     7$
1304 006076 005726          TST     (SP)+
1305 006100 000716          BR     OUT
1306 006102 122767 000060 172540 4$:  CMPB     #60,TIB
1307 006110 003004          BGT     5$
1308 006112 122767 000067 172530      CMPB     #67,TIB
1309 006120 003005          BGT     6$
1310 006122 012704 007345 5$:  MOV      #SQUEST,R4
1311 006126 004767 176660          JSR     PC,TTOUT
1312 006132 000720          BR     3$
1313 006134 006367 172566 6$:  ASL     TEMPST
1314 006140 006367 172562          ASL     TEMPST
:START OVER IF NOT LEGAL CHARACTER

```

1315	006144	006367	172556			ASL	TEMPST	
1316	006150	142767	000060	172472		BICB	#60,TIB	:GET NITTY-GRITTY
1317	006156	156767	172466	172542		BISB	TIB,TEMPST	
1318	006164	005367	172534			DEC	COUNT	:ONLY WANT 6 DIGITS
1319	006170	001754				BFC	5\$	
1320	006172	000667				BR	1\$	
1321	006174	016777	172526	172416	7\$:	MOV	TEMPST,2SWR	:CHANGE SWITCH REGISTER CONTENTS
1322	006202	000735				BR	8\$	

1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378

006332 000005
006334 012704 006366
006340 004767 176446
006344 105767 172433
006350 001405
006352 013700 000042
006356 005037 000042
006362 004700
006364 000777

```

: *****
:                               MODIFIED JAN 24 1978
:
: ++
:                               CHECK FOR DUMP MODE OR AUTOMATIC/ACT11-XXDP MODE
: --
ABORT: RESET                               ; CLEAR THE WORLD
MOV      #MSGD,R4                          ; GET THE MESSAGE
JSR      PC,TROUT                          ; PRINT ABORT MESSAGE
TSTB    XXDPM                              ; XXDP AUTO MODE?
BEQ      IS                                ; BRANCH - IF NO
MOV      @#42,R0                            ; GET MONITOR EXIT ADDRESS
CLR      @#42                              ; USE AS ABORT FLAG
JSR      PC,R0                              ; EXIT TO XXDP MONITOR
IS:     BR      .                            ; AND HANG
: *****

```


K03

CZTSFDD TSO3 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 36

SEC 0036

1379

```

;MESSAGE TABLE*****
1380
1381
1382 006366 022445 051120 043517 MSG0: .ASCII /%PROGFAM ABORTED#/
1383 006374 040522 020115 041101
1384 006402 051117 042524 021504
1385 006410 022445 055103 051524 MSG1: .ASCII %CZTSFDD TSO3 SPLMTL INSTR#
1386 006416 042106 020060 051524
1387 006424 031460 051440 046120
1388 006432 052115 020114 047111
1389 006440 052123 021522
1390 006444 042445 052116 051105 MSG2: .ASCII /%ENTER UNIT NUMBER: #
1391 006452 052440 044516 020124
1392 006460 052516 041115 051105
1393 006466 021472
1394 006470 022445 042441 042116 MSG3: .ASCII /%!END OF PASS: #/
1395 006476 047440 020106 040520
1396 006504 051523 020072 043
1397 006511 045 020441 042522 MSG4: .ASCII /%!!REWIND ERROR: NO BOT#/
1398 006516 044527 042116 042440
1399 006524 051122 051117 020072
1400 006532 047516 041040 052117
1401 006540 043
1402 006541 045 052115 035123 MSG5: .ASCII /%MTS: #/
1403 006546 021440
1404 006550 046445 041524 020072 MSG6: .ASCII /%MTC: #/
1405 006556 043
1406 006557 045 052115 041502 MSG7: .ASCII /%MTBC: #/
1407 006564 020072 043
1408 006567 045 052115 040503 MSG8: .ASCII /%MTCA: #/
1409 006574 020072 043
1410 006577 045 020441 040502 MSG9: .ASCII /%!!BACKSPACE ERROR#/
1411 006604 045503 050123 041501
1412 006612 020105 051105 047522
1413 006620 021522
1414 006622 020445 047041 052117 MSG10: .ASCII /%!!NOT READY#/
1415 006630 051040 040505 054504
1416 006636 043
1417 006637 045 020441 047516 MSG11: .ASCII /%!!NO INTERRUPT#/
1418 006644 044440 052116 051105
1419 006652 052522 052120 043
1420 006657 045 040504 040524 MSG12: .ASCII /%DATA ERROR#/
1421 006664 042440 051122 051117
1422 006672 043
1423 006673 045 047103 020072 MSG13: .ASCII /%CN: #/
1424 006700 043
1425 006701 045 035107 021440 MSG14: .ASCII /%G: #/
1426 006706 041045 020072 043 MSG15: .ASCII /%B: #/
1427 006713 077 043 MSG16: .ASCII /?#/
1428 006715 045 051127 052111 MSG17: .ASCII /%WRITE ERROR#/
1429 006722 020105 051105 047522
1430 006730 021522
1431 006732 051045 040505 020104 MSG18: .ASCII /%READ ERROR#/
1432 006740 051105 047522 021522
1433 006746 047045 020117 050117 MSG19: .ASCII /%NO OPI IN 10.5 FEET#/
1434 006754 020111 047111 030440
1435 006762 027060 020065 042506

```

M03

CZTSFDD TS03 SPLMTL INSTR
CZTSFDD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 38

SEQ 0038

1436	006770	052105	043		
1437	006773	045	050117	020111	MSG20: .ASCII /%OPI WITHIN 4.0 FEET#
1438	007000	044527	044124	04111	
1439	007006	032040	030056	043040	
1440	007014	042505	021524		
1441	007020	020440	047041	052117	MSG21: .ASCII / !!NOT AVAILABLE#
1442	007026	040440	040526	046111	
1443	007034	041101	042514	043	
1444	007041	045	042522	044507	MSG22: .ASCII /%REGISTER START: #
1445	007048	052123	051105	051440	
1446	007054	040524	052122	020072	
1447	007062	043			
1448	007063	045	042526	052103	MSG23: .ASCII /%VECTOR: #
1449	007070	051117	020072	043	
1450					
1451					:TEST HEADER*****
1452					
1453	007075	045	052045	051505	LT1MSG: .ASCII /%TEST 1: WRITE FROM ODD BYTE#
1454	007102	020124	035061	053440	
1455	007110	044522	042524	043040	
1456	007116	047522	020115	042117	
1457	007124	020104	054502	042524	
1458	007132	043			
1459	007133	045	052045	051505	LT2MSG: .ASCII /%TEST 2: READ TO ODD BYTE#
1460	007140	020124	035062	051040	
1461	007146	040505	020104	047524	
1462	007154	047440	042104	041040	
1463	007162	052131	021505		
1464	007166	022445	042524	052123	LT3MSG: .ASCII /%TEST 3: OPI TOO LONG#
1465	007174	031440	020072	050117	
1466	007202	020111	047524	020117	
1467	007210	047514	043516	043	
1468	007215	045	052045	051505	LT4MSG: .ASCII /%TEST 4: OPI TOO SHORT#
1469	007222	020124	035064	047440	
1470	007230	044520	052040	047517	
1471	007236	051440	047510	052122	
1472	007244	043			
1473					
1474	007245	045	047536	043	\$CNTG: .ASCII /%TG#
1475	007251	045	020445	051441	\$MSWR: .ASCII /%!!SWR= #
1476	007256	051127	020075	043	
1477	007263	040	047040	053505	\$MNEW: .ASCII / NEW= #
1478	007270	020075	043		
1479	007273	045	047045	020117	\$NOACT: .ASCII /%NO ACT-11 HOOKS (LF) =INVALID --RETRY%#
1480	007300	041501	026524	030461	
1481	007306	044040	047517	051513	
1482	007314	036040	043114	020076	
1483	007322	044475	053116	046101	
1484	007330	042111	026440	051055	
1485	007336	052105	054522	022445	
1486	007344	043			
1487	007345	045	022477	021445	\$QUEST: .ASCII /%?%#
1488					.EVEN
1489					
1490	007352	177777			WDATA: -1
1491		007454			.=.+100

CZTSF00 T503 SPLMTL INSTR
CZTSFD.P11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 39

SEG 0039

1492 007454 000000
1493 000001

RDATA: 0
.END

ABORT	006332	1342	1367*										
ACT11M	001002	362*	476	526	545	560	603	622	1333*	1347*			
ADUMPM	001004	364*	1335*	1352*									
AUTOM	001000	361*	1332*	1339*									
BADR	000676	436*	650*	664*	695*	709*	738*	750*	783*	795*	876	914	951
BCNT	000722	446*											
BKSP	003356	657	702	744	789	843*							
BKSP1	003402	847*	848										
BYTES	000700	437*	651*	665*	696*	710*	739*	751*	784*	796*	877	912	949
CCNT	000614	408*											
CKMODE	006204	475	1332*										
CKSWR	005624	585	624	837	873	900	956	1003	1010	1024	1259*		
CNTLU	005676	540	1269*	1286									
COUNT	000724	447*	1280*	1302	1318*								
CRCNT	000710	441*	961*	964*	985								
DCHK	004160	673	718	960*									
DCHKE	004210	963	969*										
DCHKE1	004254	974	978*										
DCHKE2	004276	980	983*										
DCHKX	004360	968	972	997*									
DCHKXX	004400	939	1001	1003*									
DCHKO	004166	962*	967										
DCHK1	004172	964*	996										
DERFL	000712	442*	970*	998	1006*								
DISPLA	000622	411*	1253*										
DISPRE	000174	380*	1253										
DOLT	005436	990	994	1221*	1237	1239							
DOUTD	005520	1235*											
DOUT1	005452	1224*	1225	1233									
DOUT2	005500	1227	1230*										
DOUT3	005506	1229	1231*										
DRIVE	000040	331*											
EMADDR	000660	429*	640*	686*	734*	779*	822	866	893	926	975		
ERCHK	003672	655	669	700	714	742	787	800	851	906*			
ERCHK1	003704	907	909*										
ERCHK2	003716	910	912*										
ERCHK3	003740	916	918*										
ERPT	003750	760	908	911	919	921*							
ERPTX	004140	923	944	953*									
ERPTXX	004150	954	956*										
ERPT1	004004	925	929*										
ERRAD	000662	430*	652*	666*	699*	713*	741*	753*	785*	798*	850*	929	
EXEC	003442	653	667	697	711	740	754	786	799	858*			
EXECX	003652	890	897*										
EXECXX	003662	898	900*										
EXECO	003446	860*	863										
EXEC1	003500	865	868*										
EXEC2	003520	861	871	873*									
EXEC3	003602	885*	886	888									
EXEC4	003642	892	895*										
FUN	000704	439*	649*	663*	694*	708*	737*	752*	782*	797*	879		
HDRFL	000714	443*	579*	864	891	924	928*	973	977*				
HERE	002016	612	618*										
ITAMT	000634	416*	762*	764*	803*	805*	1030						
ITCNT	000706	440*	1027*	1029*	1030								
ITER	004454	674	719	763	804	1023*							

ITER0	004172	1027#	1031									
ITER1	004500	1026#	1029#									
ITRLP	000666	432#	574*	575*	600*	601*	1033					
LTADD	000664	431#	572*	573*	574	576	580	598*	599*	600	1018	
LT1	002064	455#	456	639#								
LT1B	002106	644#	647									
LT1C	002187	654#										
LT1D	002210	660#	662									
LT1E	002254	668#										
LT1MSG	007075	640	1453#									
LT2	002312	457	458	685#								
LT2B	002330	689#	692									
LT2C	002374	698#										
LT2D	002432	705#	707									
LT2E	002470	712#										
LT2MSG	007133	686	1459#									
LT3	002534	459	733#									
LT3A	002554	737#	746									
LT3B	002602	741#										
LT3C	002702	755#										
LT3IT	002632	460	747#									
LT3MSG	007163	734	1464#									
LT3X	002740	758	762#									
LT4	002764	461	778#									
LT4A	003004	782#	791									
LT4B	003040	787#										
LT4C	003132	800#										
LT4IT	003062	462	792#									
LT4MSG	007215	779	1468#									
MEDIUM	000041	335#										
MSG0	006366	1368	1382#									
MSG1	006410	483	1385#									
MSG10	006622	868	1414#									
MSG11	006637	895	1417#									
MSG12	006657	978	1420#									
MSG13	006673	983	1423#									
MSG14	006701	987	1425#									
MSG15	006706	991	1426#									
MSG16	006713	1105	1427#									
MSG17	006715	652	699	741	785	1428#						
MSG18	006732	666	713	1431#								
MSG19	006746	753	1433#									
MSG2	006444	514	1390#									
MSG20	006773	798	1437#									
MSG21	007020	549	1441#									
MSG22	007041	485	1444#									
MSG23	007063	494	1448#									
MSG3	006470	607	1394#									
MSG4	006511	824	1397#									
MSG5	006541	826	931	1402#								
MSG6	006550	830	935	1404#								
MSG7	006557	939	1406#									
MSG8	006567	945	1408#									
MSG9	006577	850	1410#									
MTBA	000606	405#	876*	918	947							
MTBC	000604	404#	845*	877*	909	941						

CZTSFD0 TSO3 SPLMTL INSTR
CZTSFD.F11 15-FEB-78 14:04

MACY11 30A(1052) 15-FEB-78 14:05 PAGE 45
CROSS REFERENCE TABLE -- USER SYMBOLS

SEC 0044

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

. ABS. 007456 000

ERRORS DETECTED: 0

CZTSFD,CZTSFD.SEG CRF SOL NL:TOC=CZTSFD.F11

RUN-TIME: 13.5 SECONDS

RUN-TIME RATIO: 96.6=15.3

CORE USED: 7K (13 PAGES)

DOCUMENT PAGES: 44