

TS03

INSTRUCTION TEST
MD-11-DZTSF-C

EP-DZTSF-C-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1978
digital
MADE IN USA

B01

1500 SUPPLEMENTAL INSTRUCTION TEST
1100
1100

MACY11 27(732) 03-NOV-75 08:48 PAGE 2

1100-10-01-6-0100-

CO1

T803 SUPPLEMENTAL INSTRUCTION TEST
D2TSFC.SRC

MACY11 27(732) 03-NOV-76 08:48 PAGE 3

11

000000
000001
000002
000003
000004
000005
000006
000007

000000
000002
000000
000006
000000
000012
000000
000016
000000
000022
000000
000026
000000
000032
000000

```
.TITLE TS03 SUPPLEMENTAL INSTRUCTION TEST
:MAINDEC-11-DZTSF-C-D
:15 APR 75/MARCH 1976
:R. BARNES/RON PLATUKIS
:ABS
```

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

```
;CONSOLE SWITCHES
```

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

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

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

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

```
. =0
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
.+2
HALT
```

68	000034	000036	.+2
69	000036	000000	HALT
70	000040	000042	.+2
71	000042	000000	HALT
72	000044	000046	.+2
73	000046	000000	HALT
74	000050	000052	.+2
75	000052	000000	HALT
76	000054	000056	.+2
77	000056	000000	HALT
78	000060	000062	.+2
79	000062	000000	HALT
80	000064	000066	.+2
81	000066	000000	HALT
82	000070	000072	.+2
83	000072	000000	HALT
84	000074	000076	.+2
85	000076	000000	HALT
86	000100	000102	.+2
87	000102	000000	HALT
88	000104	000106	.+2
89	000106	000000	HALT
90	000110	000112	.+2
91	000112	000000	HALT
92	000114	000116	.+2
93	000116	000000	HALT
94	000120	000122	.+2
95	000122	000000	HALT
96	000124	000126	.+2
97	000126	000000	HALT
98	000130	000132	.+2
99	000132	000000	HALT
100	000134	000136	.+2
101	000136	000000	HALT
102	000140	000142	.+2
103	000142	000000	HALT
104	000144	000146	.+2
105	000146	000000	HALT
106	000150	000152	.+2
107	000152	000000	HALT
108	000154	000156	.+2
109	000156	000000	HALT
110	000160	000162	.+2
111	000162	000000	HALT
112	000164	000166	.+2
113	000166	000000	HALT
114	000170	000172	.+2
115	000172	000000	HALT
116	000174	000176	.+2
117	000176	000000	HALT
118	000200	000202	.+2
119	000202	000000	HALT
120	000204	000206	.+2
121	000206	000000	HALT
122	000210	000212	.+2
123	000212	000000	HALT

124	000214	000216	.+2
125	000216	000000	HALT
126	000220	000222	.+2
127	000222	000000	HALT
128	000224	000226	.+2
129	000226	000000	HALT
130	000230	000232	.+2
131	000232	000000	HALT
132	000234	000236	.+2
133	000236	000000	HALT
134	000240	000242	.+2
135	000242	000000	HALT
136	000244	000246	.+2
137	000246	000000	HALT
138	000250	000252	.+2
139	000252	000000	HALT
140	000254	000256	.+2
141	000256	000000	HALT
142	000260	000262	.+2
143	000262	000000	HALT
144	000264	000266	.+2
145	000266	000000	HALT
146	000270	000272	.+2
147	000272	000000	HALT
148	000274	000276	.+2
149	000276	000000	HALT
150	000300	000302	.+2
151	000302	000000	HALT
152	000304	000306	.+2
153	000306	000000	HALT
154	000310	000312	.+2
155	000312	000000	HALT
156	000314	000316	.+2
157	000316	000000	HALT
158	000320	000322	.+2
159	000322	000000	HALT
160	000324	000326	.+2
161	000326	000000	HALT
162	000330	000332	.+2
163	000332	000000	HALT
164	000334	000336	.+2
165	000336	000000	HALT
166	000340	000342	.+2
167	000342	000000	HALT
168	000344	000346	.+2
169	000346	000000	HALT
170	000350	000352	.+2
171	000352	000000	HALT
172	000354	000356	.+2
173	000356	000000	HALT
174	000360	000362	.+2
175	000362	000000	HALT
176	000364	000366	.+2
177	000366	000000	HALT
178	000370	000372	.+2
179	000372	000000	HALT

180	000374	000376	.+2
181	000376	000000	HALT
182	000400	000402	.+2
183	000402	000000	HALT
184	000404	000406	.+2
185	000406	000000	HALT
186	000410	000412	.+2
187	000412	000000	HALT
188	000414	000416	.+2
189	000416	000000	HALT
190	000420	000422	.+2
191	000422	000000	HALT
192	000424	000426	.+2
193	000426	000000	HALT
194	000430	000432	.+2
195	000432	000000	HALT
196	000434	000436	.+2
197	000436	000000	HALT
198	000440	000442	.+2
199	000442	000000	HALT
200	000444	000446	.+2
201	000446	000000	HALT
202	000450	000452	.+2
203	000452	000000	HALT
204	000454	000456	.+2
205	000456	000000	HALT
206	000460	000462	.+2
207	000462	000000	HALT
208	000464	000466	.+2
209	000466	000000	HALT
210	000470	000472	.+2
211	000472	000000	HALT
212	000474	000476	.+2
213	000476	000000	HALT
214	000500	000502	.+2
215	000502	000000	HALT
216	000504	000506	.+2
217	000506	000000	HALT
218	000510	000512	.+2
219	000512	000000	HALT
220	000514	000516	.+2
221	000516	000000	HALT
222	000520	000522	.+2
223	000522	000000	HALT
224	000524	000526	.+2
225	000526	000000	HALT
226	000530	000532	.+2
227	000532	000000	HALT
228	000534	000536	.+2
229	000536	000000	HALT
230	000540	000542	.+2
231	000542	000000	HALT
232	000544	000546	.+2
233	000546	000000	HALT
234	000550	000552	.+2
235	000552	000000	HALT

236	000554	000556	.+2
237	000556	000000	HALT
238	000560	000562	.+2
239	000562	000000	HALT
240	000564	000566	.+2
241	000566	000000	HALT
242	000570	000572	.+2
243	000572	000000	HALT
244	000574	000576	.+2
245	000576	000000	HALT
246	000600	000602	.+2
247	000602	000000	HALT
248	000604	000606	.+2
249	000606	000000	HALT
250	000610	000612	.+2
251	000612	000000	HALT
252	000614	000616	.+2
253	000616	000000	HALT
254	000620	000622	.+2
255	000622	000000	HALT
256	000624	000626	.+2
257	000626	000000	HALT
258	000630	000632	.+2
259	000632	000000	HALT
260	000634	000636	.+2
261	000636	000000	HALT
262	000640	000642	.+2
263	000642	000000	HALT
264	000644	000646	.+2
265	000646	000000	HALT
266	000650	000652	.+2
267	000652	000000	HALT
268	000654	000656	.+2
269	000656	000000	HALT
270	000660	000662	.+2
271	000662	000000	HALT
272	000664	000666	.+2
273	000666	000000	HALT
274	000670	000672	.+2
275	000672	000000	HALT
276	000674	000676	.+2
277	000676	000000	HALT
278	000700	000702	.+2
279	000702	000000	HALT
280	000704	000706	.+2
281	000706	000000	HALT
282	000710	000712	.+2
283	000712	000000	HALT
284	000714	000716	.+2
285	000716	000000	HALT
286	000720	000722	.+2
287	000722	000000	HALT
288	000724	000726	.+2
289	000726	000000	HALT
290	000730	000732	.+2
291	000732	000000	HALT

292	000734	000736	.+2
293	000736	000000	HALT
294	000740	000742	.+2
295	000742	000000	HALT
296	000744	000746	.+2
297	000746	000000	HALT
298	000750	000752	.+2
299	000752	000000	HALT
300	000754	000756	.+2
301	000756	000000	HALT
302	000760	000762	.+2
303	000762	000000	HALT
304	000764	000766	.+2
305	000766	000000	HALT
306	000770	000772	.+2
307	000772	000000	HALT
308	000774	000776	.+2
309	000776	000000	HALT
310			

```

311                                     ;TTY INTERRUPT VECTOR*****
312
313                                     .=60
314 000060 000060                       TTINT           ;TTY INTERRUPT HANDLER
315 000062 000000                       0
316
317
318                                     ;SOFTWARE SWITCH REGISTER LOCATIONS*****
319
320                                     .=174
321 000174 000000                       DISPREG:0
322 000176 000000                       SWREG: 0
323
324                                     ;STARTING ADDRESS*****
325
326                                     .=200
327 000200 005000                       CLR           RO
328 000202 000167 000572                 JMP           START           ;PROGRAM START
329
330                                     .=210
331 000210 012700 000001                 MOV          #1,RO
332 000214 000167 000560                 JMP           START           ;NO HEADER START
333
334                                     ;TMA-11 INTERRUPT VECTOR*****
335
336                                     .=224
337 000224 004342                       MTINT
338 000226 000340                       340           ;TAPE INTERRUPT HANDLER
339

```

K01

TS03 SUPPLEMENTAL INSTRUCTION TEST
DZTSFC.SRC

MACY11 27(732) 03-NOV-76 08:48 PAGE 11

340 000600
341
342
343 000600 172520
344 000602 172522
345 000604 172524
346 000606 172526
347 000610 000000
348 000612 000020
349 000614 177760
350 000616 177776
351 000620 177570
352 000622 177570
353 000624 177560
354 000626 177562
355 000630 177564
356 000632 177566
357 000634 000010
358 000636 000040
359 000640 172520
360 000642 000224
361

. =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

362
363
364 000644 000000
365 000646 000000
366 000650 000000
367 000652 000000
368 000654 000000
369 000656 000000
370 000660 000000
371 000662 000000
372 000664 000000
373 000666 000000
374 000670 000000
375 000672 000000
376 000674 000000
377 000676 000000
378 000700 000000
379 000702 000000
380 000704 000000
381 000706 000000
382 000710 000000
383 000712 000000
384 000714 000000
385 000716 000000
386 000720 000000
387 000722 000000
388 000724 000000
389 000726 000000
390 000730 000000

;FLAGS AND COUNTERS*****

TINF: 0
TOB: 00
TIB: 00
TEMP1: 00
TEMP2: 00
TEMP3: 00
EMADDR: 00
ERRAD: 00
LTADD: 00
ITRLP: 00
SPFLG: 00
STFLG: 00
PCNTR: 00
BADR: 00
BYTES: 00
SCNT: 00
FUN: 00
ITCNT: 00
CRCNT: 00
DERFL: 00
HDRFL: 00
PFLG: 00
UNP: 00
BCNT: 00
COUNT: 00
TEMPST: 00
RDSW: 0

391
392
393
394 000732 000000
395 000734 000000
396 000736 001702
397 000740 001702
398 000742 002130
399 000744 002130
400 000746 002352
401 000750 002450
402 000752 002602
403 000754 002700
404 000756 000000
405

;TEST ENTRY TABLE*****

TSTTBL: 0
0
T1AD: LT1
T1IAD: LT1
T2AD: LT2
T2IAD: LT2
T3AD: LT3
T3IAD: LT3IT
T4AD: LT4
T4IAD: LT4IT
0

```

406          001000          . =1000
407          ;*****
408          ;PROGRAM START AND HOUSEKEEPING
409          ;
410          ;*****
411
412 001000 000240          START:  NOP
413 001002 012777 000340 177606  MOV    #340, @PSW      ;SET PRIORITY
414 001010 012706 000500          MOV    #500, SP       ;SET STACK POINTER
415 001014 004767 004342          JSR    PC, @SUSWR    ;SEE IF NO HARDWARE SWITCH REG.
416 001020 005700          TST    RO             ;SEE IF SKIP HEADER
417 001022 001163          BNE    ST4           ;IF SO: BR
418 001024 012704 006022          MOV    #MSG1, R4
419 001030 004767 003574          JSR    PC, TTOUT     ;PRINT HEADER
420 001034 012704 006461          MOV    #MSG22, R4
421 001040 004767 003564          JSR    PC, TTOUT     ;REQUEST UNIBUS ADDRESS
422 001044 016703 177570          MOV    REGS, R3
423 001050 004767 003752          JSR    PC, OCTP      ;PRINT CURRENT ADDRESS
424 001054 012705 000640          MOV    #REGS, R5     ;GET ADDRESS OF ENTRY
425 001060 012701 000006          MOV    #6, R1        ;SET SIZE OF ENTRY
426 001064 012702 172700          MOV    #172700, R2   ;SET UPPER LIMIT
427 001070 012703 172300          MOV    #172300, R3   ;SET LOWER LIMIT
428 001074 004767 003272          JSR    PC, TTR       ;GO GET RESPONSE
429 001100 012704 006503          MOV    #MSG23, R4
430 001104 004767 003520          JSR    PC, TTOUT     ;REQUEST VECTOR
431 001110 016703 177526          MOV    VECT, R3
432 001114 004767 003706          JSR    PC, OCTP      ;PRINT CURRENT VECTOR
433 001120 012705 000642          MOV    #VECT, R5     ;GET ADDRESS OF ENTRY
434 001124 012701 000003          MOV    #3, R1        ;SET SIZE OF ENTRY
435 001130 012702 000250          MOV    #250, R2      ;SET UPPER LIMIT
436 001134 012703 000150          MOV    #150, R3      ;SET LOWER LIMIT
437 001140 004767 003226          JSR    PC, TTR       ;GO GET RESPONSE
438 001144 016700 177472          MOV    VECT, RO      ;GET VECTOR
439 001150 012720 004342          MOV    #MTINT, (RO)+ ;SET INTERRUPT ADDRESS IN VECTOR
440 001154 012710 000340          MOV    #340, (RO)    ;SET INTERRUPT PRIORITY
441 001160 012701 000004          MOV    #4, R1        ;SET NUMBER OF REGISTER
442 001164 012702 000600          MOV    #MTS, R2      ;GET FIRST ADDRESS OF TABLE
443 001170 016700 177444          MOV    REGS, RO      ;GET FIRST REGISTER
444 001174 010022          STOA:  MOV    RO, (R2)+ ;LOAD TABLE
445 001176 062700 000002          ADD    #2, RO        ;BUMP ADDRESS
446 001202 005301          DEC    R1             ;SEE IF DONE
447 001204 001373          BNE    STOA          ;IF NOT: BE
448 001206 012777 010000 177366  MOV    #10000, @MTC  ;POWER CLEAR
449 001214 012704 006064          STO:  MOV    #MSG2, R4
450 001220 004767 003404          JSR    PC, TTOUT     ;REQUEST UNIT NUMBER
451 001224 005067 177360          CLR    UDES          ;PRESET UNIT 0
452 001230 016703 177354          MOV    UDES, R3      ;GET UNIT NUMBER
453 001234 000303          SWAB   R3            ;POSITION
454 001236 042703 177770          BIC    #177770, R3   ;MASK UNIT NUMBER
455 001242 004767 003560          JSR    PC, OCTP      ;PRINT CURRENT VALUE
456 001246 012705 000656          MOV    #TEMP3, R5    ;SET SAVE LOCATION
457 001252 012701 000001          MOV    #1, R1        ;SET SIZE OF ENTRY
458 001256 012702 000007          MOV    #7, R2        ;SET UPPER LIMIT
459 001262 012703 000000          MOV    #0, R3        ;SET LOWER LIMIT
460 001266 004767 003100          JSR    PC, TTR       ;GO GET UNIT NUMBER
461 001272 000367 177360          SWAB   TEMP3        ;POSITION UNIT NUMBER

```



```

; TEST SCHEDULAR*****
486      001424 000240
487      001426 005067 177240
488      001432 017700 177162
489      001436 042700 177700
490      001442 005700
491      001444 001046
492      001446 012767 000732 177210
493      001454 062767 000004 177202
494      001462 016767 177176 177176
495      001470 062767 000002 177170
496      001476 005777 177162
497      001502 001002
498      001504 000167 000124
499      001510 005067 177200
500      001514 017700 177144
501      001520 000110
502      001522 032777 002000 177070
503      001530 001401
504      001532 000000
505      001534 004767 003702
506      001540 005767 177126
507      001544 001743
508      001546 017700 177046
509      001552 042700 177760
510      001556 005700
511      001560 001721
512      001562 012767 000001 177102
513      001570 022700 000005
514      001574 003417
515      001576 000241
516      001600 006100
517      001602 006100
518      001604 012767 000732 177052
519      001612 060067 177046
520      001616 016767 177042 177042
521      001624 062767 000002 177034
522      001632 000726
523      001634 012704 006110
524      001640 004767 002764
525      001644 016703 177024
526      001650 004767 003152
527      001654 032777 004000 176736
528      001662 001001
529      001664 000000
530      001666 004767 003550
531      001672 005267 176776
532      001676 000167 177522

TSCD:   NOP
        CLR      STFLG      ;CLEAR SINGLE TEST FLAG
        MOV      @SWR,R0    ;GET SWITCH REGISTER
        BIC      #177700,R0 ;MASK TEST SELECT
        TST      R0        ;SEE IF SINGLE TEST SELECT
        BNE     STSCD      ;IF SO: BR
        MOV      #TSTTBL,LTADD ;GET TABLE START
TSCD0:  ADD      #4,LTADD
        MOV      LTADD,ITRLP ;SET ITERATION ADDRESS
        ADD      #2,ITRLP
        TST      @LTADD     ;SEE IF END OF CYCLE
        BNE     TSCD1      ;IF NOT: BR
        JMP      TEND       ;GO TO END ROUTINE
TSCD1:  CLR      HDRFL      ;CLEAR HEADER FLAG
        MOV      @LTADD,R0  ;GET TEST ADDRESS
        JMP      (R0)       ;GO TO TEST
TSCD2:  BIT      #2000,@SWR  ;SEE IF HALT ON TEST
        BEQ     TSCD3      ;IF NOT: BR
        HALT
TSCD3:  JSR      PC,CKSWR    ;TEST FOR 1G
        TST      STFLG     ;SEE IF SINGLE TEST
        BEQ     TSCD0      ;IF NOT: BR
        MOV      @SWR,R0
        BIC      #177760,R0 ;GET TEST NUMBER
        TST      R0        ;SEE IF ALL TESTS
        BEQ     TSCD       ;IF SO: BR
STSCD:  MOV      #1,STFLG   ;SET SINGLE TEST FLAG
        CMP      #5,R0     ;SEE IF EXCEEDED TEST NUMBER
        BLE     TEND       ;IF SO: BR
        CLC
        ROL      R0
        ROL      R0
        MOV      #TSTTBL,LTADD ;POSITION NUMBER
        ADD      R0,LTADD    ;GET START OF TABLE
        MOV      LTADD,ITRLP ;SET POINTER
        ADD      #2,ITRLP
        BR      TSCD1      ;SET ITERATION ADDRESS
        MOV      #MSG3,R4   ;GO DO TEST
TEND:   JSR      PC,TTOUT   ;PRINT END OF PASS
        MOV      PCNTR,R3
        JSR      PC,OC1P    ;PRINT PASS NUMBER
        BIT      #4000,@SWR ;SEE IF HALT ON PASS
        BNE     TENDX      ;IF NOT: BR
        HALT
TENDX:  JSR      PC,CKSWR   ;TEST FOR 1G
        INC      PCNTR     ;BUMP PASS COUNTER
        JMP      TSCD      ;RESTART

```

536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581

001702 000240
001704 012767 006515 176746
001712 012702 006772
001716 112722 000377
001722 005000
001724 110022
001726 005200
001730 022700 000006
001734 001373
001736 004767 001046
001742 012767 000004 176734
001750 012767 006773 176720
001756 012767 177772 176714
001764 012767 006335 176670
001772 004767 001262
001776 000240
002000 004767 001504
002004 012767 177777 176670
002012 004767 001156
002016 012702 007074
002022 012700 000010
002026 012722 177777
002032 005300
002034 001374
002036 012767 000002 176640
002044 012767 007074 176624
002052 012767 177772 176620
002060 012767 006352 176574
002066 004767 001166
002072 000240
002074 004767 001410
002100 012701 006773
002104 012702 007074
002110 012700 000006
002114 004767 001656
002120 004767 002146
002124 000167 177372

```
*****  
: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  
      MOVVB    #377,(R2)+     ;INSERT BACKGROUND DATA  
      CLR      R0  
LT1B: MOVVB    R0,(R2)+  
      INC      R0              ;LOAD WRITE BUFFER (0,1,2,3,4,5)  
      CMP      #6,R0  
      BNE     LT1B  
      JSR     PC,RWIND         ;GO REWIND  
      MOV      #4,FUN         ;SET WRITE FUNCTION CODE  
      MOV      #WDATA+1,BADR  ;SET DATA POINTER  
      MOV      #-6,BYTES      ;SET SIZE OF RECORD  
      MOV      #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  
LT1D: MOV      #-1,(R2)+     ;BACKGROUND READ BUFFER  
      DEC     R0  
      BNE     LT1D          ;DO ALL  
      MOV      #2,FUN         ;SET READ FUNCTION CODE  
      MOV      #RDATA,BADR   ;SET READ POINTER  
      MOV      #-6,BYTES      ;SET SIZE OF RECORD  
      MOV      #MSG18,ERRAD  ;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     TSCD2         ;RETURN TO SCHEDULAR
```


582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627

002130 000240
002132 012767 006553 176520
002140 012702 006772
002144 005000
002146 110022
002150 005200
002152 022700 000006
002156 001373
002160 004767 000624
002164 012767 000004 176512
002172 012767 006772 176476
002200 012767 177772 176472
002206 004767 001046
002212 000240
002214 012767 006335 176440
002222 004767 001262
002226 012767 177777 176446
002234 004767 000734
002240 012702 007074
002244 012700 000010
002250 012722 177777
002254 005300
002256 001374
002260 012767 000002 176416
002266 012767 007075 176402
002274 012767 177772 176376
002302 004767 000752
002306 000240
002310 012767 006352 176344
002316 004767 001166
002322 012701 006772
002326 012702 007075
002332 012700 000006
002336 004767 001434
002342 004767 001724
002346 000167 177150

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

```

628
629
630
631
632
633
634
635
636
637
638
639 002352 000240          LT3:  NOP
640 002354 012767 006606 176276  MOV      #LT3MSG,EMADDR ;SET TEST HEADER
641 002362 012700 000040          MOV      #40,R0        ;SET NUMBER OF WRITE IRG/BACKSPACE
642 002366 004767 000416          JSR      PC,RWIND      ;GO REWIND UNIT
643 002372 012767 000014 176304  LT3A:  MOV      #14,FUN       ;SET WRITE IRG FUNCTION CODE
644 002400 012767 006772 176270  MOV      #WDATA,BADR   ;SET BUS ADDRESS
645 002406 012767 177760 176264  MOV      #-20,BYTES    ;SET SIZE OF RECORD
646 002414 004767 000640          JSR      PC,EXEC       ;GO EXECUTE COMMAND
647 002420 012767 006335 176234  LT3B:  MOV      #MSG17,ERRAD ;SET ERROR CODE
648 002426 004767 001056          JSR      PC,ERCHK      ;GO CHECK FOR STATUS ERROR
649 002432 012767 177777 176242  MOV      #-1,SCNT      ;
650 002440 004767 000530          JSR      PC,BKSP       ;GO BACKSPACE ONE RECORD
651 002444 005300          DEC      R0            ;SEE IF DONE ALL
652 002446 001351          BNE      LT3A         ;IF NOT: BR
653 002450 000240          LT3IT: NOP
654 002452 004767 000332          JSR      PC,RWIND      ;GO REWIND
655 002456 012767 000230 176152  MOV      #230,STALL    ;SET OPI STALL
656 002464 012767 007074 176204  MOV      #RDATA,BADR   ;SET START OF READ BUFFER
657 002472 012767 177760 176200  MOV      #-20,BYTES    ;SET SIZE OF RECORD
658 002500 012767 000002 176176  MOV      #2,FUN        ;SET READ FUNCTION CODE
659 002506 012767 006366 176146  MOV      #MSG19,ERRAD  ;SET ERROR CODE
660 002514 004767 000540          JSR      PC,EXEC       ;GO EXECUTE COMMAND
661 002520 000240          LT3C:  NOP
662 002522 012767 000040 176106  MOV      #40,STALL     ;RESET NORMAL STALL
663 002530 032777 000400 176042  BIT      #400,AMTS     ;SEE IF BTE IS SET
664 002536 001007          BNE      LT3X         ;IF SO: BR
665 002540 012767 000001 176122  MOV      #1,SPFLG      ;SET NO BA PRINT FLAG
666 002546 004767 001014          JSR      PC,ERPT       ;GO PRINT ERROR
667 002552 005067 176112          CLR      SPFLG         ;RESET FLAG
668 002556 012767 000002 176050  LT3X:  MOV      #2,ITAMT     ;SET TO TWO (2) ITERATIONS
669 002564 004767 001502          JSR      PC,ITER       ;GO SEE IF ITERATION
670 002570 012767 000010 176036  MOV      #10,ITAMT    ;RESET ITERATIONS
671 002576 000167 176720          JMP      TSCD2        ;RETURN TO SCHEDULAR
672

```

673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712

```

002602 000240
002604 012767 006635 176046
002612 004767 000172
002616 012700 000014
002622 012767 000014 176054
002630 012767 006772 176040
002636 012767 177760 176034
002644 012767 006335 176010
002652 004767 000402
002656 004767 000626
002662 012767 177777 176012
002670 004767 000300
002674 005300
002676 001351
002700 000240
002702 004767 000102
002706 012767 000200 175722
002714 012767 007074 175754
002722 012767 177760 175750
002730 012767 000002 175746
002736 012767 006413 175716
002744 004767 000310
002750 004767 000534
002754 000240
002756 012767 000040 175652
002764 012767 000002 175642
002772 004767 001274
002776 012767 000010 175630
003004 000167 176512
    
```

```

:*****
:TEST 4: OPI TOO SHORT
:
:THIS TEST WILL ERASE APPROXIMATELY FOUR (4) FEET
:OF TAPE BY WRITING WITH IRG, BACKSPACING
:ONE (1) RECORD AND REPEATING THIS SEQUENCE
:12(10) TIMES. TAPE WILL REWIND AND BE READ
:FORWARD. THE FIRST RECORD ON TAPE SHOULD BE
:REACHED BEFORE OPI TIMES OUT.
:*****
LT4:  NOP
      MOV #LT4MSG,EMADDR ;SET HEADER
      JSR PC,RWIND ;GO REWIND
      MOV #14,RD ;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 RD ;SEE IF DONE ALL SEQUENCES
      BNE LT4A ;IF NOT: BR
LT4IT: NOP
      JSR PC,RWIND ;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 NORMAL STALL
      MOV #2,ITAMT ;SET TO TWO (2) ITERATIONS
      JSR PC,ITER ;GO SEE IF ITERATIONS
      MOV #10,ITAMT ;RESET ITERATIONS
      JMP TSC02 ;RETURN TO SCHEDULAR
    
```



```

762                                     ;COMMAND EXECUTE SUBROUTINE*****
763
764 003260 000240 EXEC: NOP
765 003262 005005 CLR R5
766 003264 032777 000200 175310 EXEC0: BIT #200,AMTC ;SEE IF CUR
767 003272 001021 BNE EXEC2 ;IF SO: BR
768 003274 005305 DEC R5 ;SEE IF TIMED OUT
769 003276 001372 BNE EXEC0 ;IF NOT: BR
770 003300 005767 175410 TST HDRFL ;SEE IF DONE HEADER
771 003304 001004 BNE EXEC1 ;IF SO: BR
772 003306 016704 175346 MOV EMADDR,R4
773 003312 004767 001312 JSR PC,TTOUT ;ELSE PRINT HEADER
774 003316 012704 006242 EXEC1: MOV #MSG10,R4
775 003322 004767 001302 JSR PC,TTOUT ;PRINT NOT READY ERROR
776 003326 005777 175266 TST ASWR ;SEE IF HALT ON ERROR
777 003332 100001 BPL EXEC2 ;IF NOT: BR
778 003334 000000 HALT
779 003336 004767 002100 EXEC2: JSR PC,CKSWR ;TEST FOR IG
780 003342 000240 NOP
781 003344 016777 175240 175230 MOV UDES,AMTC ;SELECT UNIT
782 003352 016777 175320 175226 MOV BADR,AMTBA ;SET BUS MEMORY ADDRESS
783 003360 016777 175314 175216 MOV BYTES,AMTBC ;SET BYTE COUNT
784 003366 016701 175312 MOV FUN,R1 ;GET FUNCTION
785 003372 052701 000101 BIS #101,R1 ;SET IN GO BIT AND INTERRUPT ENABLE
786 003376 050177 175200 BIS R1,AMTC ;LOAD COMMAND+GO+IE
787 003402 000240 NOP
788 003404 005077 175206 CLR APSW ;ALLOW INTERRUPTS
789 003410 016767 175222 175234 MOV STALL,TEMP1 ;SET READY STALL
790 003416 005001 CLR R1
791 003420 005301 EXEC3: DEC R1
792 003422 001376 BNE EXEC3 ;AWAIT INTERRUPT
793 003424 005367 175222 DEC TEMP1
794 003430 001373 BNE EXEC3
795 003432 032777 020000 175160 BIT #20000,ASWR ;SEE IF PRINT ERROR
796 003440 001013 BNE EXECX ;IF NOT: BR
797 003442 005767 175246 TST HDRFL ;SEE IF DONE HEADER
798 003446 001004 BNE EXEC4 ;IF SO: BR
799 003450 016704 175204 MOV EMADDR,R4
800 003454 004767 001150 JSR PC,TTOUT ;PRINT HEADER
801 003460 012704 006257 EXEC4: MOV #MSG11,R4
802 003464 004767 001140 JSR PC,TTOUT ;PRINT NO INTERRUPT MESSAGE
803 003470 005777 175124 EXECX: TST ASWR ;SEE IF HALT ON ERROR
804 003474 100001 BPL EXECXX ;IF NOT: BR
805 003476 000000 HALT
806 003500 004767 001736 EXECXX: JSR PC,CKSWR ;TEST FOR IG
807 003504 000240 NOP
808 003506 000207 RTS PC ;RETURN TO CALLER
809

```

```

;STATUS ERROR CHECK SUBROUTINE*****
810
811
812 003510 005777 175066 ERCHK: TST  @MTC ;SEE IF ANY ERROR BITS
813 003514 100002 BPL ERCHK1 ;IF NOT: BR
814 003516 000167 000044 JMP ERPT ;ELSE PRINT ERROR
815 003522 005777 175056 ERCHK1: TST @MTBC ;SEE IF BYTE COUNT IS ZERO
816 003526 001402 BEQ ERCHK2 ;IF SO: BR
817 003530 000167 000032 JMP ERPT ;ELSE PRINT ERROR
818 003534 016703 175140 ERCHK2: MOV BYTES,R3
819 003540 005403 NEG R3
820 003542 066703 175130 ADD BADR,R3 ;SET EXPT BUS ADDRESS
821 003546 005767 175116 TST SPFLG ;SEE IF SPACE OPERATION
822 003552 001401 BEQ ERCHK3 ;IF NOT: BR
823 003554 000207 RTS PC
824 003556 020377 175024 ERCHK3: CMP R3,@MTBA ;SEE IF EXPT=RCVD
825 003562 001001 BNE ERPT ;IF NOT: BR
826 003564 000207 RTS PC ;ELSE EXIT
827 003566 000240 ERPT: NOP
828 003570 032777 020000 175022 BIT #20000,@SWR ;SEE IF SHOULD PRINT
829 003576 001067 BNE ERPTX ;IF NOT: BR
830 003600 005767 175110 TST HDRFL ;SEE IF DONE HEADER
831 003604 001006 BNE ERPT1 ;IF SO: BR
832 003606 016704 175046 MOV EMADDR,R4
833 003612 004767 001012 JSR PC,TTOUT ;ELSE PRINT HEADER
834 003616 005267 175072 INC HDRFL ;SET FLAG
835 003622 016704 175034 ERPT1: MOV ERRAD,R4
836 003626 004767 000776 JSR PC,TTOUT ;PRINT ERROR CODE
837 003632 012704 006161 MOV #MSG5,R4
838 003636 004767 000766 JSR PC,TTOUT ;PRINT MTS TAG
839 003642 017703 174732 MOV @MTS,R3
840 003646 004767 001144 JSR PC,OCPE ;PRINT MTS
841 003652 012704 006170 MOV #MSG6,R4
842 003656 004767 000746 JSR PC,TTOUT ;PRINT MTC TAG
843 003662 017703 174714 MOV @MTC,R3
844 003666 004767 001124 JSR PC,OCPE ;PRINT MTC
845 003672 012704 006177 MOV #MSG7,R4
846 003676 004767 000726 JSR PC,TTOUT ;PRINT BYTE COUNT TAG
847 003702 017703 174676 MOV @MTBC,R3
848 003706 004767 001114 JSR PC,OCPE ;PRINT BYTE COUNT
849 003712 005767 174752 TST SPFLG ;SEE IF PRINT BA
850 003716 001017 BNE ERPTX ;IF NOT: BR
851 003720 012704 006207 MOV #MSG8,R4
852 003724 004767 000700 JSR PC,TTOUT ;PRINT BUS ADDRESS TAG
853 003730 017703 174652 MOV @MTBA,R3
854 003734 004767 001066 JSR PC,OCPE ;PRINT CURRENT ADDRESS
855 003740 016703 174734 MOV BYTES,R3
856 003744 005403 NEG R3
857 003746 066703 174724 ADD BADR,R3
858 003752 004767 001050 JSR PC,OCPE ;PRINT EXPT ADDRESS
859 003756 005777 174636 ERPTX: TST @SWR ;SEE IF HALT ON ERROR
860 003762 100001 BPL ERPTXX ;IF NOT: BR
861 003764 000000 HALT
862 003766 004767 001450 ERPTXX: JSR PC,CKSWR ;TEST FOR 1G
863 003772 000167 000240 JMP SCOPE ;GO SEE IF SCOPE ON ERROR

```

```

;DATA CHECK SUBROUTINE*****
864
865
866 003776 000240
867 004000 005067 174704
868 004004 001112
869 004006 001112
870 004010 005267 174674
871 004014 122122
872 004016 005300
873 004020 001371
874 004022 000167 000150
875 004026 000240
876 004030 012767 000001 174654
877 004036 032777 020000 174554
878 004044 001054
879 004046 005767 174642
880 004052 001007
881 004054 016704 174600
882 004060 004767 000544
883 004064 012767 000001 174622
884 004072 012704 006277
885 004076 005767 174614
886 004102 001004
887 004104 005267 174606
888 004110 004767 000514
889 004114 012704 006313
890 004120 004767 000504
891 004124 016703 174560
892 004130 004767 000672
893 004134 012704 006321
894 004140 004767 000464
895 004144 111103
896 004146 004767 001102
897 004152 012704 006326
898 004156 004767 000446
899 004162 111203
900 004164 004767 001064
901 004170 000240
902 004172 000167 177612
903 004176 000240
904 004200 005767 174506
905 004204 001404
906 004206 005777 174406
907 004212 100001
908 004214 000000
909 004216 004767 001220
910 004222 000240
911 004224 005067 174466
912 004230 005067 174456
913 004234 000207

DCHK:  NOP
      CLR  CRCNT      ;CLEAR COUNTER
DCHK0:  CMPB (R1),(R2) ;SEE IF EXPT DATA=RCVD DATA
      BNE  DCHK1      ;IF NOT: BR
DCHK1:  INC  CRCNT      ;BUMP CHARACTER COUNTER
      CMPB (R1)+,(R2)+
      DEC  RD          ;SEE IF DONE
      BNE  DCHK0      ;IF NOT: BR
      JMP  DCHKX      ;ELSE GO TO EXIT ROUTINE
DCHKE:  NOP
      MOV  #1,DERFL    ;SET ERROR FLAG
      BIT  #20000,DSWR ;SEE IF PRINT ERROR
      BNE  DCHKX      ;IF NOT: BR
      TST  HDRFL      ;SEE IF DONE HEADER
      BNE  DCHKE1     ;IF SO: BR
      MOV  EMADDR,R4
      JSR  PC,TTOUT    ;PRINT HEADER
DCHKE1: MOV  #1,HDRFL  ;SET HEADER FLAG
      MOV  #MSG12,R4
      TST  PFLG        ;SEE IF PRINTED DATA ERROR TAG
      BNE  DCHKE2     ;IF SO: BR
      INC  PFLG
DCHKE2: JSR  PC,TTOUT ;ELSE PRINT DATA ERROR TAG
      MOV  #MSG13,R4
      JSR  PC,TTOUT    ;PRINT CHAR NUMBER TAG
      MOV  CRCNT,R3
      JSR  PC,OCTP     ;PRINT CHAR NUMBER
      MOV  #MSG14,R4
      JSR  PC,TTOUT    ;PRINT GOOD TAG
      MOVB (R1),R3
      JSR  PC,DOUT     ;PRINT GOOD CHARACTER
      MOV  #MSG15,R4
      JSR  PC,TTOUT    ;PRINT BAD TAG
      MOVB (R2),R3
      JSR  PC,DOUT     ;PRINT BAD CHARACTER
DCHKX:  NOP
      JMP  DCHK1      ;CONTINUE FOR ALL BYTES
      TST  DERFL      ;SEE IF ANY ERROR
      BEQ  DCHKXX     ;IF NOT: BR
      TST  DSWR      ;SEE IF HALT ON ERROR
      BPL  DCHKXX     ;IF NOT: BR
DCHKXX: JSR  PC,CKSWR ;TEST FOR IG
      NOP
      CLR  PFLG        ;CLEAR PRINT FLAG
      CLR  DERFL      ;CLEAR DATA ERROR FLAG
      RTS  PC         ;RETURN

```

```

914                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
915
916 004236 004767 001200 SCOPE: JSR PC,CKSWR ;TEST FOR ↑G
917 004242 000240 NOP
918 004244 032777 040000 174346 BIT #40000,↯SWR ;SEE IF LOOP ON ERROR
919 004252 001001 BNE SCOPE1 ;IF SO: BR
920 004254 000207 RTS PC ;ELSE EXIT
921 004256 000240 SCOPE1: NOP
922 004260 005726 TST (SP)+ ;RESET STACK
923 004262 000240 NOP
924 004264 017703 174374 MOV ↯LTADD,R3
925 004270 000113 JMP (R3) ;LOOP ON ERROR
926
927                                     ;TEST ITERATION SUBROUTINE*****
928
929 004272 000240 ITER: NOP
930 004274 004767 001142 JSR PC,CKSWR ;TEST FOR ↑G
931 004300 032777 010000 174312 BIT #10000,↯SWR ;SEE IF ITERATIONS
932 004306 001403 BEQ ITER1 ;IF SO: BR
933 004310 005067 174372 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
934 004314 000207 RTS PC ;ELSE EXIT
935 004316 005267 174364 ITER1: INC ITCNT ;BUMP COUNTER
936 004322 026767 174360 174304 CMP ITCNT,ITAMT ;SEE IF DONE ALL
937 004330 001767 BEQ ITER0 ;IF SO: BR
938 004332 005726 TST (SP)+ ;RESET STACK
939 004334 017700 174326 MOV ↯ITRLP,RO ;SET ITERATION POINTER
940 004340 000110 JMP (RO) ;GO ITERATE
941
942                                     ;MAG TAPE INTERRUPT HANDLER*****
943
944 004342 000240 MTINT: NOP
945 004344 022626 CMP (SP)+,(SP)+ ;RESET STACK POINTER
946 004346 042777 000100 174226 BIC #100,↯MTC ;CLEAR INTERRUPT ENABLE
947 004354 000240 NOP
948 004356 000240 NOP
949 004360 000207 RTS PC ;RETURN TO CALLER
950
951                                     ;TTY INTERRUPT HANDLER*****
952
953 004362 000240 TTINT: NOP
954 004364 000240 NOP
955 004366 000240 NOP
956 004370 000002 RTI
957

```



```

958 ;*****
959 ;TTY ENTRY SUBROUTINE:
960 ;
961 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
962 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
963 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
964 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
965 ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
966 ;THE CALLING ROUTINE.
967 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
968 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
969 ;MAY BE REENTERED.
970 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
971 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
972 ;CARRIAGE RETURN
973 ;*****
974
975 004372 005067 174254 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
976 004376 005000 CLR RO
977 004400 004767 000152 TTR0: JSR PC,TIN ;GO READ CHARACTER
978 004404 122767 000215 174236 CMPB #215,TIB ;SEE IF CR
979 004412 001005 BNE TTR1 ;IF NOT: BR
980 004414 005767 174232 TST TEMP1 ;SEE IF FIRST CHARACTER
981 004420 001446 BEQ TTR5 ;IF SO: BR
982 004422 000167 000066 JMP TTR2 ;ELSE GO LOAD VALUE
983 004426 122767 000260 174214 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0
984 004434 101402 BLOS TTR1A ;IF NOT: BR
985 004436 000167 000076 JMP TINER ;ELSE GO TO ERROR
986 004442 122767 000270 174200 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7
987 004450 101002 BHI TTR1B ;IF NOT: BR
988 004452 000167 000062 JMP TINER ;ELSE GO TO ERROR
989 004456 005267 174170 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
990 004462 000241 CLC
991 004464 006100 ROL RO
992 004466 000241 CLC
993 004470 006100 ROL RO ;SHIFT 3 LEFT
994 004472 000241 CLC
995 004474 006100 ROL RO
996 004476 042767 177770 174144 BIC #177770,TIB ;STRIP ASCII
997 004504 056700 174140 BIS TIB,RO ;LOAD CHARACTER
998 004510 005301 DEC R1 ;SEE IF DONE
999 004512 001332 BNE TTR0 ;IF NOT: BR
1000 004514 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
1001 004516 101402 BLOS TTR3 ;IF OT: BR
1002 004520 000167 000014 JMP TINER ;ELSE GO TO ERROR
1003 004524 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
1004 004526 101402 BLOS TTR4 ;IF NOT: BR
1005 004530 000167 000004 JMP TINER ;ELSE GO TO ERROR
1006 004534 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
1007 004536 000207 TTR5: RTS PC ;EXIT
1008

```

```

1009 ;TTY ENTRY ERROR SUBROUTINE*****
1010
1011 004540 012704 006333 TINER: MOV #MSG16,R4
1012 004544 004767 000060 JSR PC,TTOUT ;PRINT?
1013 004550 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
1014 004554 000207 RTS PC ;REDO VALUE ENTRY
1015
1016 ;TTY READ SUBROUTINE*****
1017
1018 004556 005077 174042 TTIN: CLR @TKS
1019 004562 005077 174040 CLR @TKB
1020 004566 005067 174056 CLR TIB
1021 004572 005277 174026 INC @TKS
1022 004576 105777 174022 TTIN1: TSTB @TKS
1023 004602 100375 BPL TTIN1
1024 004604 017767 174016 174036 MOV @TKB,TIB
1025 004612 105777 174012 TTIN2: TSTB @TPS
1026 004616 100375 BPL TTIN2
1027 004620 116777 174024 174004 MOVB TIB,@TPB
1028 004626 000207 RTS PC
1029
1030 ;TTY OUTPUT SUBROUTINE*****
1031
1032 004630 112467 174012 TTOUT: MOVB (R4)+,TOB
1033 004634 122767 000043 174004 CMPB #43,TOB
1034 004642 001452 BEQ TEX
1035 004644 122767 000045 173774 CMPB #45,TOB
1036 004652 001407 BEQ TCRLF
1037 004654 122767 000041 173764 CMPB #41,TOB
1038 004662 001443 BEQ TBELL
1039 004664 004767 000064 JSR PC,TOG
1040 004670 000757 BR TTOUT
1041 004672 112767 000015 173746 TCRLF: MOVB #15,TOB
1042 004700 004767 000050 JSR PC,TOG
1043 004704 012703 000004 MOV #4,R3
1044 004710 005067 173732 TCRLFA: CLR TOB
1045 004714 004767 000034 JSR PC,TOG
1046 004720 005303 DEC R3
1047 004722 001372 BNE TCRLFA ;DO FILLERS
1048 004724 112767 000012 173714 MOVB #12,TOB
1049 004732 004767 000016 JSR PC,TOG
1050 004736 105767 173766 TSTB RDSW
1051 004742 100401 BMI 1$
1052 004744 000731 BR TTOUT
1053 004746 005067 173756 1$: CLR RDSW
1054 004752 000406 BR TEX
1055 004754 105777 173650 TOG: TSTB @TPS
1056 004760 100375 BPL TOG
1057 004762 116777 173660 173642 MOVB TOB,@TPB
1058 004770 000207 RTS PC
1059 004772 012703 000002 TBELL: MOV #2,R3
1060 004776 012767 000007 173642 TBELA: MOV #7,TOB
1061 005004 004767 177744 JSR PC,TOG
1062 005010 005303 DEC R3
1063 005012 001371 BNE TBELA
1064 005014 000705 BR TTOUT

```

NO2

TS03 SUPPLEMENTAL INSTRUCTION TEST
DZTSFC.SRC

MACY11 27(732) 03-NOV-76 08:48 PAGE 27

1065

```

;OCTAL OUTPUT SUBROUTINE*****
1066
1067
1068 005016 012767 000001 000226 OCTPE: MOV #1,OFL
1069 005024 000402 BR OCTPE1
1070 005026 005067 000220 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
1071 005032 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
1072 005034 001007 SNE OCTPD ;IF NOT ZERO: BR
1073 005036 005767 000210 TST OFL ;SEE IF PRINT ALL 0
1074 005042 001004 BNE OCTPD ;IF SO: BR
1075 005044 004767 000162 JSR PC,OCTPG1 ;ELSE PRINT ZERO
1076 005050 000167 000120 JMP OCTP3 ;SPACE AND EXIT
1077 005054 032704 100000 OCTPD: BIT #100000,R4 ;SEE IF MSD = 1
1078 005060 001406 BEQ OCTP1 ;IF NOT: BR
1079 005062 012704 000001 MOV #1,R4
1080 005066 004767 000116 JSR PC,OCTPG ;PRINT 1
1081 005072 000167 000006 JMP OCTP2
1082 005076 005004 OCTP1: CLR R4
1083 005100 004767 000104 JSR PC,OCTPG ;PRINT 0
1084 005104 010304 OCTP2: MOV R3,R4
1085 005106 006004 ROR R4
1086 005110 006004 ROR R4
1087 005112 006004 ROR R4 ;POSITION DIGIT
1088 005114 006004 ROR R4
1089 005116 000304 SWAB R4
1090 005120 004767 000064 JSR PC,OCTPG ;PRINT DIGIT 2
1091 005124 010304 MOV R3,R4
1092 005126 006004 ROR R4
1093 005130 000304 SWAB R4
1094 005132 004767 000052 JSR PC,OCTPG ;PRINT DIGIT 3
1095 005136 010304 MOV R3,R4
1096 005140 006104 ROL R4
1097 005142 006104 ROL R4
1098 005144 000304 SWAB R4
1099 005146 004767 000036 JSR PC,OCTPG ;PRINT DIGIT 4
1100 005152 010304 MOV R3,R4
1101 005154 006004 ROR R4
1102 005156 006004 ROR R4
1103 005160 006004 ROR R4
1104 005162 004767 000022 JSR PC,OCTPG
1105 005166 010304 MOV R3,R4
1106 005170 004767 000014 JSR PC,OCTPG ;PRINT DIGIT 5
1107 005174 012767 000240 173444 OCTP3: MOV #240,TOB
1108 005202 004767 177546 JSR PC,TOG ;PRINT SPACE
1109 005206 000207 RTS PC ;EXIT

```

```

1110
1111
1112
1113 005210 042704 177770
1114 005214 001004
1115 005216 005767 000030
1116 005222 001001
1117 005224 000207
1118 005226 005267 000020
1119 005232 052704 000260
1120 005236 010467 173404
1121 005242 004767 177506
1122 005246 010304
1123 005250 000207
1124 005252 000000
1125
1126
1127
1128 005254 005067 173366
1129 005260 012704 000010
1130 005264 110367 173356
1131 005270 105777 173334
1132 005274 100375
1133 005276 132767 000200 173342
1134 005304 001404
1135 005306 012777 000061 173316
1136 005314 000403
1137 005316 012777 000060 173306
1138 005324 006167 173316
1139 005330 005304
1140 005332 001356
1141 005334 000207
1142 005336 016703 173314
1143 005342 000303
1144 005344 004767 177704
1145 005350 016703 173302
1146 005354 004767 177674
1147 005360 000207
1148
1149
1150
1151
1152 005362 013746 000006
1153 005366 013746 000004
1154 005372 012737 005412 000004
1155 005400 022777 177777 173212
1156 005406 001402
1157 005410 000407
1158 005412 022626
1159 005414 012767 000176 173176
1160 005422 012767 000174 173172
1161 005430 012637 000004
1162 005434 012637 000006
1163 005440 000207
1164
1165

;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
      JSR PC,DOUT
      RTS PC

SUSWR: MOV #6,-(SP) ;SAVE VECTORS
      MOV #4,-(SP)
      MOV #1$,#4 ;SET UP FOR TIMEOUT
      CMP #-1,#SWR ;REFERENCE HARDWARE SWITCH REGISTER
      BEQ 2$
      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

```

```

1166 005442 022767 000176 173150 CKSWR:  CMP      #SWREG,SWR      ;SOFTWARE SWITCH REG PRESENT
1167 005450 001041                BNE      OUT        ;NO GET OUT
1168 005452 105777 173146          TSTB     @TKS        ;YES WAIT FOR
1169 005456 100036                BPL      OUT        ;READY GET CHARACTER
1170 005460 017767 173142 173162      MOV      @TKB,TIB   ;AND STRIP OFF
1171 005466 042767 177600 173154      BIC      #177600,TIB ;THE GARBAGE
1172 005474 022767 000007 173146      CMP      #7,TIB     ;IS IT A <TG>
1173 005502 001024                BNE      OUT
1174 005504 012704 006665          MOV      #SCNTG,R4
1175 005510 004767 177114          JSR      PC,TTOUT
1176 005514 012704 006671          CNTLU:  MOV      #MSWR,R4
1177 005520 004767 177104          JSR      PC,TTOUT
1178 005524 017703 173070          MOV      @SWR,R3
1179 005530 004767 177262          JSR      PC,OCTPE
1180 005534 012704 006703          MOV      #SMNEW,R4
1181 005540 004767 177064          JSR      PC,TTOUT
1182 005544 005037 000726          CLR      @TEMPST
1183 005550 004767 000002          JSR      PC,$READ
1184 005554 000207          OUT:    RTS        ;GO READ A LINE
1185                                     ;RETURN TO MAIN BODY OF PROGRAM
1186 005556 005067 173144          $READ:  CLR      TEMPST
1187 005562 012767 000007 173134      MOV      #7,COUNT
1188 005570 004767 176762          1$:    JSR      PC,TTIN ;GO READ A CHARACTER
1189 005574 042767 177600 173046      BIC      #177600,TIB ;STRIP OFF GARBAGE
1190 005602 122767 000025 173040      CMPB     #25,TIB    ;IS IT A TU?
1191 005610 001002                BNE      11$
1192 005612 005726          3$:    TST      (SP)+    ;BRANCH IF NOT
1193 005614 000737                BR       CNTLU     ;POP THE STACK
1194 005616 122767 000012 173024      11$:   CMPB     #12,TIB   ;START OVER
1195 005624 001016                BNE      2$
1196 005626 005767 172214          TST      46        ;IS IT A <LF>?
1197 005632 001406                BEQ      10$
1198 005634 005726          TST      (SP)+    ;BRANCH IF NOT
1199 005636 016716 172204          MOV      46,(SP)  ;TEST FOR ACT-11 HOOKS
1200 005642 062716 000010          ADD      #10,(SP) ;NO HOOKS-<LF> WAS MISTAKE
1201 005646 000742                BR       OUT
1202 005650 012704 006713          10$:   MOV      #NOACT,R4 ;GO TO THAT ADDRESS
1203 005654 004767 176750          JSR      PC,TTOUT ;LET OPR KNOW IT
1204 005660 000754                BR       3$
1205 005662 122767 000015 172760      2$:   CMPB     #15,TIB   ;AND RETURN
1206 005670 001013                BNE      4$
1207 005672 012767 000200 173030      MOV      #200,RDSW ;FOR ANOTHER TRY
1208 005700 004767 176766          JSR      PC,TCRLF ;IS IT A <CR>?
1209 005704 022767 000007 173012      CMP      #7,COUNT ;BRANCH IF NOT
1210 005712 001037                BNE      7$
1211 005714 005726          8$:   TST      (SP)+    ;ECHO IT WITH <LF>
1212 005716 000716                BR       OUT
1213 005720 122767 000060 172722      4$:   CMPB     #60,TIB   ;WAS IT FIRST CHARACTER
1214 005726 003004                BGT      5$
1215 005730 122767 000067 172712      CMPB     #67,TIB   ;CHANGE SWR IF NOT FIRST ONE
1216 005736 003005                BGT      6$
1217 005740 012704 006765          5$:   MOV      #SQUEST,R4 ;POP THE STACK
1218 005744 004767 176660          JSR      PC,TTOUT ;GET OUT
1219 005750 000720                BR       3$
1220 005752 006367 172750          6$:   ASL      TEMPST
1221 005756 006367 172744          ASL      TEMPST

```

1222	005762	006367	172740		ASL	TEMPST	
1223	005766	142767	000060	172654	BICB	#60,TIB	;GET NITTY-GRITTY
1224	005774	156767	172650	172724	BISB	TIB,TEMPST	
1225	006002	005367	172716		DEC	COUNT	;ONLY WANT 6 DIGITS
1226	006006	001754			BEQ	SS	
1227	006010	000667			BR	IS	
1228	006012	016777	172710	172600	MOV	TEMPST,DSWR	;CHANGE SWITCH REGISTER CONTENTS
1229	006020	000735			BR	BS	

```

1230                                     ;MESSAGE TABLE*****
1231
1232 006022 022445 051524 031460 MSG1: .ASCII /%TS03 INSTRUCTION TEST (DZTSF-B)#/
1233 006030 044440 051516 051124
1234 006036 041525 044524 047117
1235 006044 052040 051505 020124
1236 006052 042050 052132 043123
1237 006060 041055 021451
1238 006064 042445 052116 051105 MSG2: .ASCII /%ENTER UNIT NUMBER: #/
1239 006072 052440 044516 020124
1240 006100 052516 041115 051105
1241 006106 021472
1242 006110 022445 042441 042116 MSG3: .ASCII /%!END OF PASS: #/
1243 006116 047440 020106 040520
1244 006124 051523 020072 043
1245 006131 045 020441 042522 MSG4: .ASCII /%!!REWIND ERROR: NO BOT#/
1246 006136 044527 042116 042440
1247 006144 051122 051117 020072
1248 006152 047516 041040 052117
1249 006160 043
1250 006161 045 052115 035123 MSG5: .ASCII /%MTS: #/
1251 006166 021440
1252 006170 046445 041524 020072 MSG6: .ASCII /%MTC: #/
1253 006176 043
1254 006177 045 052115 041502 MSG7: .ASCII /%MTBC: #/
1255 006204 020072 043
1256 006207 045 052115 040503 MSG8: .ASCII /%MTCA: #/
1257 006214 020072 043
1258 006217 045 020441 040502 MSG9: .ASCII /%!!BACKSPACE ERROR#/
1259 006224 045503 050123 041501
1260 006232 020105 051105 047522
1261 006240 021522
1262 006242 020445 047041 052117 MSG10: .ASCII /%!!NOT READY#/
1263 006250 051040 040505 054504
1264 006256 043
1265 006257 045 020441 047516 MSG11: .ASCII /%!!NO INTERRUPT#/
1266 006264 044440 052116 051105
1267 006272 052522 052120 043
1268 006277 045 040504 040524 MSG12: .ASCII /%DATA ERROR#/
1269 006304 042440 051122 051117
1270 006312 043
1271 006313 045 047103 020072 MSG13: .ASCII /%CN: #/
1272 006320 043
1273 006321 045 035107 021440 MSG14: .ASCII /%G: #/
1274 006326 041045 020072 043 MSG15: .ASCII /%B: #/
1275 006333 077 043 MSG16: .ASCII /?#/
1276 006335 045 051127 052111 MSG17: .ASCII /%WRITE ERROR#/
1277 006342 020105 051105 047522
1278 006350 021522
1279 006352 051045 040505 020104 MSG18: .ASCII /%READ ERROR#/
1280 006360 051105 047522 021522
1281 006366 047045 020117 050117 MSG19: .ASCII /%NO OPI IN 10.5 FEET#/
1282 006374 020111 047111 030440
1283 006402 027060 020065 042506
1284 006410 052105 043
1285 006413 045 050117 020111 MSG20: .ASCII /%OPI WITHIN 4.0 FEET#/

```


1286	006420	044527	044124	047111	
1287	006426	032040	030056	043040	
1288	006434	042505	021524		
1289	006440	020440	047041	052117	MSG21: .ASCII / !!NOT AVAILABLE#/ /
1290	006446	040440	040526	046111	
1291	006454	041101	042514	043	
1292	006461	045	042522	044507	MSG22: .ASCII /%REGISTER START: #/ /
1293	006466	052123	051105	051440	
1294	006474	040524	052122	020072	
1295	006502	043			
1296	006503	045	042526	052103	MSG23: .ASCII /%VECTOR: #/ /
1297	006510	051117	020072	043	
1298					
1299					;TEST HEADER***** /
1300					
1301	006515	045	052045	051505	LT1MSG: .ASCII /%TEST 1: WRITE FROM ODD BYTE#/ /
1302	006522	020124	035061	053440	
1303	006530	044522	042524	043040	
1304	006536	047522	020115	042117	
1305	006544	020104	054502	042524	
1306	006552	043			
1307	006553	045	052045	051505	LT2MSG: .ASCII /%TEST 2: READ TO ODD BYTE#/ /
1308	006560	020124	035062	051040	
1309	006566	040505	020104	047524	
1310	006574	047440	042104	041040	
1311	006602	052131	021505		
1312	006606	022445	042524	052123	LT3MSG: .ASCII /%TEST 3: OPI TOO LONG#/ /
1313	006614	031440	020072	050117	
1314	006622	020111	047524	020117	
1315	006630	047514	043516	043	
1316	006635	045	052045	051505	LT4MSG: .ASCII /%TEST 4: OPI TOO SHORT#/ /
1317	006642	020124	035064	047440	
1318	006650	044520	052040	047517	
1319	006656	051440	047510	052122	
1320	006664	043			
1321					
1322	006665	045	043536	043	\$CNTG: .ASCII /%↑G#/ /
1323	006671	045	020445	051441	\$MSWR: .ASCII /%!!SWR= #/ /
1324	006676	051127	020075	043	
1325	006703	040	047040	053505	\$MNEW: .ASCII / NEW= #/ /
1326	006710	020075	043		
1327	006713	045	047045	020117	\$NOACT: .ASCII /%NO ACT-11 HOOKS <LF> =INVALID --RETRY%#/ /
1328	006720	041501	026524	030461	
1329	006726	044040	047517	051513	
1330	006734	036040	043114	020076	
1331	006742	044475	053116	046101	
1332	006750	042111	026440	051055	
1333	006756	052105	054522	022445	
1334	006764	043			
1335	006765	045	022477	021445	\$QUEST: .ASCII /%?%#/ /
1336					.EVEN
1337					
1338	006772	177777			WDATA: -1
1339		007074			.=.+100
1340	007074	000000			RDATA: 0
1341		000001			.END

COMMEN	1#
ENDCOM	1#
ESCAPE	1#
GETPRI	1#
GETSWR	1#
MULT	1#
NEWTST	1#
POP	1#
PUSH	1#
REPORT	1#
SETPRI	1#
SETUP	1#
SKIP	1#
SLASH	1#
STARS	1#
SWRSU	1#
TYPBIN	1#
TYPDEC	1#
TYPNAM	1#
TYPNUM	1#
TYPOCS	1#
TYPOCT	1#
TYPTXT	1#
\$\$ESCA	1#
\$\$NEWT	1#
\$\$SKIP	1#
.EQUAT	1#
.HEADE	1#
.KT11	1#
.SETUP	1#
.SWRHI	1#
.\$ACT1	1#
.\$APT8	1#
.\$APTH	1#
.\$APTY	1#
.\$ASTA	1#
.\$CATC	1#
.\$CMTA	1#
.\$DB20	1#
.\$DB20	1#
.\$DIV	1#
.\$EOP	1#
.\$ERRO	1#
.\$ERRT	1#
.\$MULT	1#
.\$POWE	1#
.\$RAND	1#
.\$RDDE	1#
.\$RDOC	1#
.\$READ	1#
.\$R2AZ	1#
.\$SAVE	1#
.\$SB20	1#
.\$SB20	1#
.\$SCOP	1#
.\$SIZE	1#

N03

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

Instruction	Symbol 1	Symbol 2	Symbol 3	Symbol 4	Symbol 5	Symbol 6	Symbol 7	Symbol 8	Symbol 9	Symbol 10	Symbol 11	Symbol 12	Symbol 13	Symbol 14	Symbol 15
ADD	44	495	497	521	523	820	857	1200							
AND	110	509	513	723	754	816	822	905	932	937	981	1034	1036	1038	1078
BIC	110	1156	1197	1226											
BITB	111														
BLOS	111														
BNE	417	447	467	470	472	483	493	499	530	553	568	598	613	652	664
BR	721	741	777	804	813	860	907	1023	1026	1056	1132	1169			
BR	478	524	1040	1052	1054	1064	1069	1136	1157	1193	1201	1204	1212	1219	1227
CJC	327	451	465	481	484	489	501	549	594	667	758	765	788	790	867
CMP	911	912	933	975	976	1018	1019	1020	1044	1053	1070	1082	1128	1182	1186
CMPB	466	515	552	597	824	936	945	1000	1003	1155	1158	1166	1172	1209	
DEC	868	871	978	983	986	1033	1035	1037	1190	1194	1205	1213	1215		
HALT	122	471	482	567	612	651	696	768	791	793	872	998	1046	1062	1139
	57	59	61	63	65	67	69	71	73	75	77	79	81	83	
	87	89	91	93	95	97	99	101	103	105	107	109	111	113	
	117	119	121	123	125	127	129	131	133	135	137	139	141	143	
	147	149	151	153	155	157	159	161	163	165	167	169	171	173	
	177	179	181	183	185	187	189	191	193	195	197	199	201	203	
	207	209	211	213	215	217	219	221	223	225	227	229	231	233	
	237	239	241	243	245	247	249	251	253	255	257	259	261	263	
	267	269	271	273	275	277	279	281	283	285	287	289	291	293	
	297	299	301	303	305	307	309	311	313	315	317	319	321	323	
INC	327	329	331	333	335	337	339	341	343	345	347	349	351	353	
JMP	357	359	361	363	365	367	369	371	373	375	377	379	381	383	
JSR	415	419	421	423	428	430	432	437	450	455	460	468	474	507	526
	528	532	554	559	561	563	573	575	579	580	599	603	606	608	617
	620	624	625	642	646	648	650	654	660	666	669	686	692	693	695
	699	705	706	710	729	731	733	735	737	739	743	757	773	775	779
	800	802	806	833	836	838	840	842	844	846	848	852	854	858	862
	882	888	890	892	894	896	898	900	909	916	930	977	1012	1039	1042
	1045	1049	1061	1075	1080	1083	1090	1094	1099	1104	1106	1108	1121	1144	1146
MOV	1175	1177	1179	1181	1183	1188	1203	1208	1218						
	331	413	414	418	420	422	424	425	426	427	429	431	433	434	435
	436	438	439	440	441	442	443	444	448	449	452	456	457	458	459
	464	473	478	479	480	490	494	496	502	510	514	520	522	525	527
	546	547	555	556	557	558	562	564	565	566	569	570	571	572	576
	577	578	592	593	600	601	602	605	607	609	610	611	614	615	616
	619	621	622	623	640	641	643	644	645	647	649	655	656	657	658

	659	662	665	668	670	685	687	688	689	690	691	694	700	701	702
	703	704	708	709	711	716	728	730	732	734	736	738	750	751	755
	756	772	774	781	782	783	784	789	799	801	818	832	835	837	839
	841	843	845	847	851	853	855	876	881	883	884	889	891	893	897
	924	939	1006	1011	1024	1043	1059	1060	1068	1071	1079	1084	1091	1095	1100
	1105	1107	1120	1122	1129	1135	1137	1142	1145	1152	1153	1154	1159	1160	1161
	1163	1170	1174	1176	1178	1180	1187	1199	1202	1207	1217	1228			
MOV8	548	550	595	895	899	1027	1032	1041	1048	1057	1130				
NEG	815	856													
NOP	412	477	488	545	560	574	591	604	618	639	653	661	684	698	707
	715	744	749	759	764	780	787	807	827	866	875	901	903	910	917
	921	923	929	944	947	948	953	954	955						
ROL	518	519	991	993	995	1096	1097	1138							
ROR	1085	1086	1087	1088	1092	1101	1102	1103							
RTI	956														
RTS	745	760	808	823	826	913	920	934	949	1007	1014	1028	1058	1109	1117
	1123	1141	1147	1163	1184										
SUB	1013														
SWAB	453	461	1089	1093	1099	1143									
TST	416	492	498	508	512	740	770	776	797	803	812	815	821	830	849
	859	879	885	904	906	922	938	980	1073	1115	1192	1196	1198	1211	
TSTB	720	1022	1025	1050	1055	1131	1168								
.ABS	17														
.ASCII	1232	1238	1242	1245	1250	1252	1254	1256	1258	1262	1265	1268	1271	1273	1274
	1275	1276	1279	1281	1285	1289	1292	1296	1301	1307	1312	1316	1322	1323	1325
	1327	1335													
.ENABL	1														
.END	1341														
.EVEN	1336														
.LIST	1														
.MACRO	1														
.NLIST	1														
.REPT	11	54													
.TITLE	13														

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

*.DZTSFC.SEG/SOL/CRF/PAGNUM/NL:TOC=SYSMAC.CO,DZTSFC.SRC
 RUN-TIME: 23 28 2 SECONDS
 RUN-TIME RATIO: 146/55=2.6
 CORE USED: 35K (69 PAGES)

D04

Spooler runtime 5 Seconds, 28 KCS, 134 disk reads, 4 disk writes, 41 pages

Job: [blacked out] Date: 07-Nov-76 16:14:02 Host: IPC-0 0078 (100) status

```

000000000000000000000000000000000000000000000000000000000000000011111111111111111111110
000000011111111111222222222233333333344444444444555555555566666666666777777777778888888888999999999900000000001111111111111112222222222333312
[blacked out]
000000000000000000000000000000000000000000000000000000000000000011111111111111111111110
900000001111111111222222222233333333334444444444455555555556666666666677777777777888888888899999999990000000000111111111112222222222333312
  
```