

TS03

DRIVE FUNCTION TIMER
MD-11-DZTSE-A

EP DZTSE A DL A

OCT 1976

COPYRIGHT © 1976

digital

FICHE 1 OF 1

Made in U.S.A.

The table on the left side of the page consists of 40 small, individual diagrams or data tables arranged in a 10x4 grid. Each cell contains a small schematic or data table related to the drive function timer. The diagrams are too small to read clearly but appear to be technical specifications or timing diagrams.

.REPT 0

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTSE-A-D
 PRODUCT NAME: TS03 DRIVE FUNCTION TIMER
 DATE CREATED: 15 APR, 1975
 MAINTAINER: DIAGNOSTIC ENGINEERING
 AUTHOR: R. B. BARNES
 COPYRIGHT: (C) 1975 DIGITAL EQUIPMENT CORPORATION
 MAYNARD, MASS 01754

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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. BY DIGITAL EQUIPMENT CORPORATION

MAINDEC-11-DZTSE-A-D
 TS03 DRIVE FUNCTION TIMER
 R. B. BARNES
 15 APR 1975
 DIAGNOSTIC ENGINEERING
 MAYNARD, MASS 01754

61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
00

1. ABSTRACT

THE TS03 DRIVE FUNCTION TIMER ASSISTS IN THE TESTING OF THE TMA-11 CONTROL UNIT AND TS03 TAPE UNIT. SELECTED OPERATIONS ARE EXECUTED, TIMED, AND THE TIMES ARE THEN PRINTED (IN MILLISECONDS). THERE IS NO LIMIT OR ERROR TESTING FACILITIES IN THE PROGRAM, THE DECISION ON THE VALIDITY OF TIMES MEASURED MUST BE MADE BY THE OPERATOR. EITHER 1 OR 2 TS03 UNITS MAY BE SELECTED.

2. REQUIREMENTS

2.1 EQUIPMENT

PDP-11 WITH TMA-11 CONTROL UNIT AND 1 OR 2 TS03 TAPE UNITS.

2.2 STORAGE

2.2.1 PROGRAM STORAGE

THE PROGRAM REQUIRES 4K OF MEMORY.

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL BINARY TAPES SHOULD BE FOLLOWED

1. ABSOLUTE LOADER MUST BE IN MEMORY.
2. PLACE BINARY TAPE IN READER.
3. LOAD ADDRESS *7500 (* DETERMINED BY LOCATION OF LOADER).

E01

MAINDEC-11-DTSE-A-D
DTSEA.P11

TS03 DRIVE FUNCTION TIMER

MACY11 27(732) 10-SEP-76 16:11 PAGE 4

92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136

- 4. PRESS "START" (PROGRAM WILL LOAD).
- 4. STARTING PROCEDURE
 - 4.1 CONTROL SWITCH SETTINGS: NONE
 - 4.2 STARTING ADDRESS
 - 200
 - 4.3 PROGRAM AND/OR OPERATOR ACTION
 - LOAD PROGRAM INTO MEMORY.
 - SET DESIRED TS03 TAPE UNITS ON-LINE.
 - SET SWITCH REGISTER TO STARTING ADDRESS.
 - LOAD ADDRESS.
 - PRESS START.
 - ENTER STARTING REGISTER ADDRESS.
 - IF SPEED TEST ONLY, ENTER A ONE(1).
 - IF ALL OTHERS, ENTER A ZERO(0).
 - IF SPEED TEST, MOUNT 800 BPI SKEW TAPE AND TYPE CR.
 - THE PROGRAM WILL AUTOMATICLY FIND THE AVAILABLE TS03 TAPE UNITS TO BE TESTED.
 - THE PROGRAM WILL BEGIN TIMING FUNCTIONS.
 - ON COMPLETION OF ALL TESTS "END OF TIMING" WILL BE PRINTED AND THE PROCESSOR WILL HALT.
 - TO REPEAT TEST: PRESS CONTINUE.
- 5. OPERATING PROCEDURE
 - 5.1 OPERATIONAL SWITCH SETTINGS
 - NONE
- 6. ERRORS
 - THE PROGRAM HAS NO INTERNAL ERROR DETECTION FACILITIES AND, THEREFORE, NO ACTUAL ERROR TYPEOUTS. THE VALIDITY OF THE TIMES MEASURED MUST BE DETERMINED BY THE OPERATOR.
- 6.1 TIME RELATIONSHIPS
 - A. "READ SHUTDOWN" MUST BE < "WRITE SHUTDOWN".
 - B. GAPS MUST = 8>7>6>5>4>3. 3=2=1 (+OR- 5.0).
 - C. "WRITE EOF" SHOULD BE SLIGHTLY > "WRITE XIRG".

137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178

6.2 TIME LIMITS AND PRINTOUT FORMAT EXAMPLE
***** (ALL TIMES ARE IN MILLISECONDS) *****

FUNCTION	UNIT 0	UNIT 1	RANGE MAX - MIN
WRITE FROM BOT	565.0	SAME	621.0 - 509.0
WRITE SHUTDOWN	15.6	"	17.6 - 33.6
WRITE START	35.8	"	39.0 - 32.0
SETTLE DOWN DELAY	33.0	"	36.0 - 30.0
WRITE TO ERASE HEAD	69.0	"	76.0 - 58.0
BACKSPACE SHUTDOWN	7.0	"	7.7 - 6.3
READ SHUTDOWN	7.0	"	7.7 - 6.3
GAPS SHOULD=8>7>6>5>4>3, 3=2=1 (+OR- 5.0)			
GAP 1	50.0	"	
GAP 2	50.0	"	
GAP 3	50.0	"	
GAP 4	53.6	"	
GAP 5	67.1	"	
GAP 6	90.4	"	
GAP 7	103.7	"	
GAP 8	117.4	"	
WRITE START	35.9	"	39.0 - 32.0
WRITE XIRG	333.1	"	356.0 - 300.0
READ FROM BOT	235.0	"	258.0 - 212.0
WRITE EOF	385.0	"	420.0 - 340.0
FOR TO EOF SPACE	360.0	"	400.0 - 320.0
SPACE SHUTDOWN	6.9	"	7.6 - 6.3
ONE INCH DATA TIME	81.0	"	89.0 - 73.0

***** END OF TIMING *****

6.3 SPEED TEST TIME PRINTOUT EXAMPLE

FUNCTION	UNIT 0	UNIT 1	RANGE MAX-MIN
TAPE SPEED FIND	81.0	81.0	89.0 - 73.0

***** END OF TIMING *****

179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

AT LEAST ONE TS03 TAPE UNIT MUST BE "ON-LINE".
ALSO MAKE CERTAIN THAT EACH TS03 THAT IS "ON-LINE"
HAS A UNIQUE UNIT NUMBER SELECTED.

7.2 OPERATING RESTRICTIONS

TMA-11 INSTRUCTION TEST MUST RUN WITHOUT ERRORS BEFORE ATTEMPTING
TO OPERATE THIS PROGRAM.

8. MISCELLANEOUS

8.1 EXECUTION TIME

NOT APPLICABLE

200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234

9.0 PROGRAM DESCRIPTION

9.1 WRITE FROM BOT DELAY

WRITE FROM BOT DELAY IS THE TIME NECESSARY TO MOVE THE BEGINNING OF TAPE (BOT) MARKER APPROXIMATELY 6 INCHES PAST THE WRITE HEAD. THE FIRST RECORD ON TAPE MUST BE WRITTEN AT LEAST 3 INCHES AWAY FROM THE BOT MARKER.

PROCEDURE TO MEASURE TIME:

- A. IF TS03 IS NOT AT BOT IT IS REWOUND TO BOT.
- B. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER.
- C. ISSUE WRITE FUNCTION, 800 BPI, SET "GO".
- D. MONITOR CURRENT MEMORY ADDRESS REGISTER TO DETERMINE WHEN 2ND BYTE IS OUTPUT.
- E. THE TIME FROM "GO" UNTIL 2ND BYTE IS OUTPUT IS APPROXIMATELY EQUAL TO "WRITE FROM BOT DELAY".

9.2 WRITE SHUTDOWN

WRITE SHUTDOWN IS THE AMOUNT OF TIME NECESSARY TO CONTINUE MOVING TAPE AFTER A RECORD IS WRITTEN SO THAT THE PROPER INTERRECORD GAP WILL EXIST BETWEEN RECORDS.

PROCEDURE TO MEASURE TIME:

- A. THE PROGRAM USES THE SAME RECORD THAT WAS WRITTEN TO TIME "WRITE FROM BOT DELAY".
- B. AFTER THE LAST BYTE (BC=0), INDICATING THE END OF THE RECORD, MONITOR "SETTLEDOWN" UNTIL IT BECOMES A 1.
- C. THE TIME FROM "BC=0" UNTIL "SETTLEDOWN" IS "WRITE SHUTDOWN".

235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286

9.3 WRITE START

WRITE START IS THE TIME NECESSARY FOR TAPE TO ACCELERATE TO FULL SPEED AND GUARANTEE A 1/2 INCH INTERRECORD GAP.

PROCEDURE TO MEASURE TIME:

SAME AS "WRITE FROM BOT" EXCEPT NOW WE ARE NOT AT BOT.

- A. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER.
- B. ISSUE WRITE FUNCTION, 800 BPI, SET "GO".
- C. MONITOR CURRENT MEMORY ADDRESS REGISTER TO DETERMINE WHEN 2ND BYTE IS OUTPUT.
- D. THE TIME FROM "GO" UNTIL 2ND BYTE IS OUTPUT IS APPROXIMATELY EQUAL TO "WRITE START".

9.4 SETTLEDOWN DELAY

TAPE DOES NOT ACTUALLY COME TO A COMPLETE STOP UNTIL SOME PERIOD OF TIME AFTER SHUTDOWN HAS ENDED. ALSO, AFTER TAPE HAS FULLY STOPPED, AN ADDITIONAL PERIOD OF TIME IS NECESSARY FOR THE TAPE AND HARDWARE TO "SETTLEDOWN" AND BECOME STABLE. THE "SETTLEDOWN DELAY" IS THE PERIOD OF TIME NECESSARY FOR THE TAPE AND MECHANICAL CHARACTERISTICS OF THE TS03 TO BECOME STABLE, SO THAT THE UNIT CANNOT BE OPERATED, START/STOP, AT A FREQUENCY WHERE IT IS MECHANICALLY RESONANT.

PROCEDURE TO MEASURE TIME:

- A. THE PROGRAM USES THE SAME RECORD THAT WAS WRITTEN TO TIME "WRITE START".
- B. AFTER "SETTLEDOWN" BECOMES A 1, INDICATING THE START OF SETTLEDOWN, MONITOR "TU READY" UNTIL IT BECOMES A 1.
- C. THE TIME FROM "SETTLEDOWN" UNTIL "TU READY" IS "SETTLEDOWN".

9.5 WRITE TO ERASE HEAD

THE PURPOSE OF THE ERASE HEAD IS TO INSURE THAT THE TAPE IS IN THE SAME FLUX STATE AS THE WRITE HEADS. THIS IS NECESSARY FOR SEVERAL REASONS.

1. START/STOP CHARACTERISTICS VARY AMONG TAPE UNITS AND IT WOULD BE POSSIBLE TO LEAVE OLD DATA IN THE INTERRECORD GAPS WHEN USING A TAPE ON MORE THAN ONE UNIT.
2. A TAPE PREVIOUSLY USED AT ONE RECORDING DENSITY COULD NOT BE USED LATER AT ANOTHER DENSITY.
3. TRACK ALIGNMENT AND HEAD WIDTH VARY FROM TAPE UNIT TO TAPE UNIT AND IT WOULD BE POSSIBLE FOR DATA TO BE LEFT ON THE TRACK EDGES FROM OLD RECORDS.

287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306

THE "WRITE TO ERASE HEAD" TEST INSURES THAT THE TAPE IN FRONT OF THE WRITE HEAD IS ERASED DURING A WRITE OPERATION.

PROCEDURE TO MEASURE TIME:

- A. A LONG RECORD HAS BEEN WRITTEN FROM BOT. SAME RECORD THAT WAS USED TO TIME "WRITE FROM BOT DELAY".
- B. TAPE IS REWOUND TO BOT.
- C. BYTE RECORD COUNTER IS INITIALIZED FOR A 3 BYTE RECORD AND CURRENT MEMORY ADDRESS REGISTER IS INITIALIZED.
- D. ISSUE WRITE FUNCTION, 800 BPI, SET "GO".
- E. AWAIT CUR AT END OF CURRENT WRITE.
- F. ISSUE A 2 BYTE READ.
- G. TIME FROM GO UNTIL CUR OF THE READ IS WRITE TO ERASE HEAD TIME.
- H. IF TIME IS TOO SHORT, ERASE HEAD IS INOPERATIVE.

307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353

9.6 BACKSPACE SHUTDOWN

"BACKSPACE SHUTDOWN" IS THE LENGTH OF TIME NECESSARY TO GUARANTEE THAT IF A WRITE OPERATION FOLLOWS A BACKSPACE THE TAPE WILL BE POSITIONED SUCH THAT ALL PREVIOUS DATA IS IN FRONT OF THE WRITE AND ERASE HEADS AND WILL BE ERASED. "BACKSPACE SHUTDOWN" MUST BE LESS THAN "WRITE START" SO THAT INTERRECORD GAPS WILL INCREASE IF A BACKSPACE/REWRITE OPERATION IS INITIATED.

PROCEDURE TO MEASURE TIME:

- A. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER.
- B. ISSUE WRITE EOF FUNCTION, 800 BPI, SET "GO"
- C. AFTER EOF RECORD IS WRITTEN WAIT FOR "TU READY".
- D. SET BYTE RECORD COUNTER TO BACKSPACE 1 RECORD.
- E. ISSUE BACKSPACE FUNCTION, SET "GO".
- F. AFTER "EOF" BECOMES A 1, INDICATING THE RECOGNITION OF THE "EOF" RECORD, MONITOR "SETTLEDOWN" UNTIL IT BECOMES A 1.
- G. THE TIME FROM "EOF" UNTIL "SETTLEDOWN" IS "BACKSPACE SHUTDOWN".

9.7 READ SHUTDOWN

READ SHUTDOWN IS THE AMOUNT OF TIME NECESSARY TO CONTINUE MOVING TAPE AFTER A RECORD IS READ SO THAT THERE IS ENOUGH GAP FOR TAPE TO BE FULLY ACCELERATED IF A READ IS FOLLOWED BY A BACKSPACE. "READ SHUTDOWN" MUST ALSO BE LESS THAN "WRITE SHUTDOWN" TO GUARANTEE THAT THE WRITE AND ERASE HEADS WILL BE POSITIONED SUCH THAT ALL PREVIOUS DATA IS IN FRONT OF THE HEADS AND WILL BE ERASED IF A WRITE FOLLOWS A READ. IN ADDITION, WHEN A WRITE FOLLOWS A READ THE INTERRECORD GAP MUST STILL BE AT LEAST 1/2 OF AN INCH.

PROCEDURE TO MEASURE TIME:

- A. RECORD PREVIOUSLY USED IN "BACKSPACE SHUTDOWN" IS READ.
- B. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER
- C. ISSUE READ FUNCTION, 800 BPI, SET "GO".
- D. AFTER "EOF" BECOMES A 1, INDICATING THE END OF THE RECORD, MONITOR "SETTLEDOWN" UNTIL IT BECOMES A 1.
- E. THE TIME FROM "EOF" UNTIL "SETTLEDOWN" IS "READ SHUTDOWN"

354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407

9.8 GAP CONSISTENCY

FOR PROPER OPERATION, THE INTERRECORD GAPS ON TAPE MUST ALWAYS BE AT LEAST 1/2 OF AN INCH. THIS WILL ALLOW DATA WRITTEN USING ONE TAPE UNIT TO BE READ ON ANOTHER TAPE UNIT WHEN THE START/STOP CHARACTERISTICS OF EACH UNIT ARE DIFFERENT. THE MINIMUM GAP SIZE OF 1/2 INCH IS GENERATED WHEN A WRITE FOLLOWS A READ. ALL OTHER GAPS SHOULD BE LARGER DEPENDING ON HOW THEY WERE WRITTEN.

PROCEDURE TO MEASURE TIME:

- A. A TOTAL OF NINE RECORDS ARE WRITTEN ON TAPE (FROM BOT) UTILIZING DIFFERENT SEQUENCES TO GENERATE THE INTERRECORD GAPS.
- B. THE TAPE IS REWOUND TO BOT.
- C. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER.
- D. ISSUE READ FUNCTION, 800 BPI, SET "GO".
- E. WAIT FOR "CU READY" TO BECOME A 1, THEN REPEAT STEP C AND RESET "GO" TO CONTINUE.
- F. MONITOR CURRENT MEMORY ADDRESS TO DETERMINE WHEN 2ND BYTE IS INPUT.
- G. THE TIME FROM WHEN "GO" IS RESET UNTIL THE 2ND BYTE IS INPUT WILL REFLECT THE SIZE OF THE GAP.
- H. STEPS E, F ARE REPEATED UNTIL ALL 8 GAPS ARE MEASURED.

PROGRAM SEQUENCE FOR EACH GAP:

- GAP 1 WRITE FOLLOWED BY A WRITE (START/STOP).
- GAP 2 WRITE FOLLOWED BY A WRITE (START/STOP).
- GAP 3 READ FOLLOWED BY A WRITE (START/STOP).
- GAP 4 WRITE-BACKSPACE FOLLOWED BY A WRITE (START/STOP).
- GAP 5 SAME AS GAP 4 EXCEPT WRITE-BACKSPACE REPEATED 2 TIMES.
- GAP 6 SAME AS GAP 4 EXCEPT WRITE-BACKSPACE REPEATED 3 TIMES.
- GAP 7 SAME AS GAP 4 EXCEPT WRITE-BACKSPACE REPEATED 4 TIMES.
- GAP 8 SAME AS GAP 4 EXCEPT WRITE-BACKSPACE REPEATED 5 TIMES.

GAP LENGTHS SHOULD REFLECT THE FOLLOWING RELATIONSHIP:

$$8 > 7 > 6 > 5 > 4 > 3, 3 = 2 = 1 (+OR- 5.0).$$

408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460

9.9 WRITE START

THIS IS A REPEAT OF THE "WRITE START" TEST PREVIOUSLY COMPLETED (REFERENCE 9.3). IT'S PURPOSE IS TO DETERMINE IF TAPE WILL DRIFT BACKWARDS TO BOT IF A "POWER CLEAR" IS ISSUED AS SOON AS BOT DISAPPEARS WHEN MOVING FORWARD FROM BOT. TIME SHOULD EQUAL "WRITE START" AS MEASURED IN 9.3.

9.10 WRITE XIRG

WRITE WITH AN EXTENDED INTERRECORD GAP IS A FUNCTION THAT CAUSES THE GENERATION OF AN INTERRECORD GAP THAT IS AT LEAST 3 INCHS LONG AS COMPARED WITH THE NORMAL 3/4 INCH GAP. THE PURPOSE IS TO ELIMINATE WRITE ERRORS THAT MAY BE CAUSED BY A DEFECTIVE AREA ON TAPE. NORMALLY ONE REWRITE WITH XIRG WOULD BE SUFFICIENT TO MOVE PAST THE BAD SPOT, HOWEVER IF IT ISN'T, THE PROCEDURE WOULD BE TO REPEAT THE "BACKSPACE-REWRITE WITH XIRG" SEQUENCE UNTIL A RECORD IS WRITTEN WITHOUT ERRORS. EACH SUCCESSIVE REWRITE WOULD ADD 3 INCHES TO THE INTERRECORD GAP UNTIL "GOOD" TAPE WAS REACHED.

PROCEDURE TO MEASURE TIME:

- A. TAPE IS NOT AT BOT
- B. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER.
- C. ISSUE WRITE WITH XIRG FUNCTION, 800 BPI, SET "GO".
- D. MONITOR CURRENT MEMORY ADDRESS REGISTER TO DETERMINE WHEN 2ND BYTE IS OUTPUT.
- E. THE TIME FROM "GO" UNIT 2ND BYTE IS OUTPUT IS "WRITE WITH XIRG".

9.11 READ FROM BOT

THE FIRST RECORD WRITTEN ON TAPE IS SUPPOSED TO BE AT LEAST 6 INCHES FROM THE BOT MARKER. IN THE EVENT THAT THIS CONDITION WASN'T MET IT IS STILL DESIREABLE TO READ THE RECORD. READ FROM BOT IS THE TIME FROM WHEN A READ FUNCTION IS ISSUED UNTIL THE 2ND BYTE IS INPUT.

PROCEDURE TO MEASURE TIME:

- A. THE RECORD THAT WAS WRITTEN JUST OFF BOT DURING "WRITE START" (REFERENCE 9.10) IS USED.
- B. TAPE IS REWOUND TO BOT
- C. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER.
- D. ISSUE READ FUNCTION, 800 BPI, SET "GO".
- E. MONITOR CURRENT MEMORY ADDRESS REGISTER TO DETERMINE WHEN 2ND BYTE IS INPUT.
- F. THE TIME FROM "GO" UNTIL 2ND BYTE IS INPUT IS "READ FROM BOT".

461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513

9.12 WRITE EOF.

TO WRITE AN END OF FILE MARK IT IS NECESSARY FOR TAPE TO MOVE 3 INCHES BEFORE WRITING. IN THAT RESPECT IT IS SIMILAR TO WRITING A RECORD WITH EXTENDED INTERRECORD GAP. HOWEVER, AN EOF MARK CORRESPONDS TO A 1 BYTE RECORD. THE TIME SHOULD BE SLIGHTLY LARGER THAN "WRITE XIRG".

PROCEDURE TO MEASURE TIME:

- A. TAPE UNIT IS REWOUND TO BOT.
- B. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS REGISTER.
- C. ISSUE WRITE FUNCTION, 800 BPI, SET "GO".
- D. WAIT FOR "CU READY" AND THEN "TU READY" TO BECOME A 1.
- E. ISSUE WRITE EOF FUNCTION, 800 BPI, SET "GO".
- F. WAIT FOR "CU READY" TO BECOME A 1.
- G. THE TIME FROM "GO" UNTIL "CU READY" IS "WRITE EOF".

9.13 EOR TO EOF SPACE TIME

EOB TO EOF SPACE TIME IS THE TIME NEEDED TO MOVE TAPE FROM THE END OF A RECORD TO AN END OF FILE MARK WRITTEN AFTER IT. THE PROCEDURE USED TURNS OUT TO BE A TEST OF THE WRITE AND ERASE HEAD POLARITIES. IF THE TIME PRINTED IS EQUAL TO ZERO IT IS AN INDICATION THAT THE EOF WAS NOT FOUND WHEN "CU READY" BECAME A 1.

THIS COULD INDICATE ONE OR MORE OF THE FOLLOWING PROBLEMS:

1. ERASE HEAD POLARITY REVERSED.
2. ERASE HEAD CURRENT NOT SUFFICIENT TO FULLY SATURATE TAPE.
3. ONE OR MORE OF WRITE HEAD TRACKS POLARITY REVERSED.
4. ONE OR MORE SENSITIVE READ AMPLIFIERS.
5. WRITE EOF FUNCTION DIDN'T REALLY WRITE AN EOF MARK. OTHERWISE "EOB TO EOF SPACE TIME" SHOULD BE SLIGHTLY LARGER THAN "WRITE EOF".

PROCEDURE TO MEASURE TIME:

- A. A RECORD AND EOF WAS PREVIOUSLY WRITTEN FROM BOT FOR "WRITE EOF" (REFERENCE 9.12).
- B. TAPE IS REWOUND TO BOT.
- C. REWRITE RECORD OVER PREVIOUSLY WRITTEN RECORD.
- D. BACKSPACE OVER RECORD JUST WRITTEN.
- E. SET BYTE RECORD COUNTER TO SPACE 2 RECORDS.
- F. ISSUE SPACE FORWARD FUNCTION, SET "GO".
- G. WAIT FOR BYTE RECORD COUNTER TO INDICATE THAT 1ST RECORD HAS BEEN SPACED OVER THEN MONITOR "CU READY" UNTIL IT BECOMES A 1. AFTER "CU READY" CHECK TO SEE IF "EOF" IS A 1 IN STATUS REGISTER. IF "EOF" NOT SET THEN ZERO TIME COUNTER.
- H. TIME FROM BYTE RECORD COUNTER =-1 UNTIL "CU READY" IS "EOB TO EOF SPACE TIME".

614			
615			
616			
617			
618			
619		001000	
620		005670	
621			
622		000000	
623			
624		000200	
625	000200	000167	000652
626		001000	
627		172520	
628	001000	172522	
629	001004	172524	
630	001006	172526	
631	001010	172530	
632	001012	172532	
633	001014	177570	
634	001016	177560	
635	001020	177562	
636	001022	177564	
637	001024	177566	
638	001026	000224	
639	001030	172520	
640	001032	177776	
641	001034	000000	
642	001036	000000	
643	001040	000000	
644	001042	000000	
645	001044	000000	
646	001046	000000	
647	001050	000000	
648	001052	000000	
649	001054	000000	

```

.TITLE MAINDEC-11-DZTSE-A-D TSD3 DRIVE FUNCTION TIMER
: COPYRIGHT: (C) 1975 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
: LOAD ADDRESS 200, PRESS START
:
: STACK=1000
: BLENTH=3000.
: .ENABL ABS
: =0
: TRAP CATCHER FROM 0 TO 1000
: =200
: JMP START
: =1000
MTS: 172520
MTC: 172522
BC: 172524
CA: 172526
MTC: 172530
MTRD: 172532
SR: 177570
TKS: 177560
TKB: 177562
TPS: 177564
TPB: 177566
MTV: 224
REGS: 172520
CC: 177776
R10: 0
R11: 0
R12: 0
R13: 0
TSDRV: 0
T11T: 0
T1B: 0
TEMP1: 0
SP.F: 0

```

```

650 001056 012706 001000 START: MOV #STACK,%6 ;INITIALIZE STACK
651 001062 012777 000340 177742 MOV #340,%0C ;SET PRIORITY LEVEL 7
652 001070 012767 010467 006412 MOV #MSG28,MESAGE
653 001076 004767 006266 JSR %7, TOP ;PRINT TITLE
654 001102 012767 010656 006400 MOV #MSG31,MESAGE
655 001110 004767 006254 JSR %7, TOP ;REQUEST UNIBUS ADDRESS
656 001114 012705 001030 MOV #REGS,%5 ;GET ADDRESS OF RESPONSE
657 001120 012701 000006 MOV #6,%1 ;SET SIZE OF ENTRY
658 001124 012702 172700 MOV #172700,%2 ;SET UPPER LIMIT
659 001130 012703 172300 MOV #172300,%3 ;SET LOWER LIMIT
660 001134 004767 005770 JSR %7, TTR ;GO GET RESPONSE
661 001140 016700 177664 MOV REGS,%0 ;GET UNIBUS ADDRESS
662 001144 012701 001000 MOV #MTS,%1 ;SET START OF TABLE
663 001150 012702 000006 MOV #6,%2 ;SET SIZE OF TABLE
664 001154 010021 STRS: MOV %0,(1)+ ;LOAD TABLE
665 001156 005720 TST (0)+ ;BUMP ADDRESS
666 001160 005302 DEC %2 ;SEE IF DONE
667 001162 001374 BNE STRS ;IF NOT: BR
668 001164 012767 010701 006316 MOV #MSG32,MESAGE
669 001172 004767 006172 JSR %7, TOP ;REQUEST SPEED TESTS ONLY
670 001176 005067 177652 CLR SPTF ;CLEAN SPEED TEST FLAG
671 001202 012705 001054 MOV #SPTF,%5 ;GET FLAG ADDRESS
672 001206 012701 000001 MOV #1,%1 ;SET SIZE OF ENTRY
673 001212 012702 000001 MOV #1,%2 ;SET UPPER LIMIT
674 001216 012703 000000 MOV #0,%3 ;SET LOWER LIMIT
675 001222 004767 005702 JSR %7, TTR ;GO GET RESPONSE
676 001226 005767 177622 TST SPTF ;SEE IF SHOULD DO SPEED TEST
677 001232 001417 BEQ STO ;IF NOT: BR
678 001234 012767 010735 006246 MOV #MSG33,MESAGE
679 001242 004767 006122 JSR %7, TOP ;REQUEST SKEW TAPE MOUNT
680 001246 012705 001052 MOV #TEMP1,%5 ;SET RESPONSE ADDRESS
681 001252 012701 000001 MOV #1,%1 ;SET SIZE
682 001256 012702 000001 MOV #1,%2 ;SET UPPER
683 001262 012703 000000 MOV #0,%3 ;SET LOWER
684 001266 004767 005636 JSR PC,TTR ;AWAIT SKEW TAPE MOUNT
685 001272 005067 004676 STO: CLR DRIVES
686 001276 012777 010000 177476 MOV #10000,%MTC ;POWER CLEAR
687 001304 012777 000000 177470 MOV #0,%MTC ;SELECT UNIT 0
688 001312 032777 000100 177460 BIT #100,%MTS ;SEE IF 0 IS THERE
689 001320 001403 BEQ STOA ;IF NOT: BR
690 001322 052767 000200 004644 BIS #200,DRIVES ;SET 0 IN AVAILABLE TABLE
691 001330 012777 000400 177444 STOA: MOV #400,%MTC ;SELECT UNIT 1
692 001336 032777 000100 177434 BIT #100,%MTS ;SEE IF 1 IS THERE
693 001344 001403 BEQ STOB ;IF NOT: BR
694 001346 052767 000100 004620 BIS #100,DRIVES ;SET 1 IN AVAILABLE TABLE
695 001354 005767 004614 STOB: TST DRIVES ;SEE IF ANY UNITS AVAILABLE
696 001360 001007 BNE STOC ;IF SO: BR
697 001362 012767 010564 006120 MOV #MSG30,MESAGE
698 001370 004767 005774 JSR %7, TOP ;PRINT NO UNITS
699 001374 000000 HALT
700 001376 000735 BR STO ;RETRY UNIT SELECTION
701 001400 000240 STOC: NOP
702 001402 004767 004444 JSR %7,RSFDRV ;RESET DRIVES
703 001406 004767 004664 ST1: JSR %7,STRREW ;START REWIND
704 001412 004767 004500 JSR %7,CHGDRV ;DONE ALL DRIVES?
705 001416 000773 BR ST1 ;NO

```

706	001420	004767	004712
707	001424	004767	004466
709	001430	000773	

ST2: JSR
JSR
BR

%7, WATREW
%7, CHGDRV
ST2

:WAIT FOR BOT
:DONE ALL DRIVES?
:NO

```

709
710           ;PRINT HEADER
711
712 001432 012767 007512 006050      MOV      #MSG2,MESAGE
713 001440 004767 005724              JSR      %7, TOP           ;PRINT "FUNCTION"
714 001444 012767 007537 006036  ST3:  MOV      #MSG2A,MESAGE
715 001452 004767 005712              JSR      %7, TOP           ;PRINT "UNIT"
716 001456 016767 004514 005426      MOV      FDRIVE,DIGIT
717 001464 000367 005422              SWAB     DIGIT
718 001470 042767 177770 005414      BIC      #177770,DIGIT
719 001476 052767 000060 005406      BIS      #60,DIGIT
720 001504 105777 177312              TSTB    @TPS
721 001510 100375                      BPL      -4
722 001512 016777 005374 177304      MOV      DIGIT,@TPB      ;PRINT DRIVE "NUMBER"
723 001520 004767 004372              JSR      %7, CHGDRV      ;DONE ALL DRIVES?
724 001524 000747                      BR       ST3             ;NO
725 001526 004767 004524              JSR      %7, ST15        ;STORE ONES IN WRITE BUFFER
726 001532 012767 010536 005750      MOV      #MSG29,MESAGE
727 001540 004767 005624              JSR      %7, TOP           ;PRINT RANGE HEADER
728 001544 005767 177304              TST     SP1F            ;SEE IF SPEED TESTS ONLY
729 001550 001402                      BEQ     T1               ;IF NOT: BR
730 001552 000167 004106              JMP     T11             ;ELSE DO SPEED TEST

```

```

731                                     :TIME WRITE FROM BOT DELAY, AND WRITE SHUTDOWN
732
733 001556 012700 006412 T1:      MOV      #TM1,%0      ;INITIALIZE TIME BUFFERS
734 001562 012701 006436      MOV      #TM2,%1
735 001566 004767 004446 T1A:     JSR      %7,WRINT
736 001572 016777 004400      MOV      FDRIVE,%MTC      ;SELECT DRIVE
737 001600 052777 040005 177202      BIS      #40005,%MTC      ;800 BPI, WRITE, GO
738 001606 005067 004576      CLR
739 001612 022777 011404 177166 T1B:     CMP      #WBUF+2,%CA      ;IS 2ND WORD OUTPUT?
740 001620 003403      BLE      TIC
741 001622 004767 004540      JSR      %7,TIMER      ;YES
742 001626 000771      BR       T1B           ;NO, COUNT TIME
743 001630 016720 004554 T1C:     MOV      TIME,(0)+      ;SAVE "WRITE FROM BOT DELAY" TIME
744 001634 005067 004550      CLR      TIME
745 001640 005777 177140      TST      @BC
746 001644 001375      BNE      #-4           ;SEE IF WORD COUNT DONE
747 001646 032777 000010 177124 T1D:     BIT      #10,%MTC      ;IF NOT: BR
748 001654 001003      BNE      T1E           ;HAS SETTLEDOWN SET?
749 001656 004767 004504      JSR      %7,TIMER      ;YES
750 001662 000771      BR       T1D           ;NO, COUNT TIME
751 001664 016721 004520 T1E:     MOV      TIME,(1)+      ;SAVE "WRITE SHUTDOWN" TIME
752 001670 004767 004222      JSR      %7,CHGDRV      ;DONE ALL DRIVES
753 001674 000734      BR       T1A           ;NO
754 001676 012720 177777      MOV      #-1,(0)+      ;TERMINATE TIMES
755 001702 012721 177777      MOV      #-1,(1)+      ;TERMINATE TIMES
756 001706 012767 007553 005574      MOV      #MSG3,MESSAGE
757 001714 012700 006412      MOV      #TM1,%0
758 001720 004767 004726      JSR      %7,TYPTIM      ;PRINT "WRITE FROM BOT DELAY" TIMES
759 001724 012767 011037 005556      MOV      #RMSG1,MESSAGE
760 001732 004767 005432      JSR      %7,TOP
761 001736 012767 007601 005544      MOV      #MSG4,MESSAGE
762 001744 012700 006436      MOV      #TM2,%0
763 001750 004767 004676      JSR      %7,TYPTIM      ;PRINT "WRITE SHUTDOWN" TIMES
764 001754 012767 011057 005526      MOV      #RMSG2,MESSAGE
765 001762 004767 005402      JSR      %7,TOP        ;PRINT RANGE

```

```

766                                     ;TIME WRITE START AND SETTLEDOWN DELAY
767
768 001766 004767 004060                T2:   JSR   %7,RSFDRV      ;RESET DRIVE SELECTION
769 001772 012700 006412                MOV   #TM1,%0
770 001776 012701 006436                MOV   #TM2,%1
771 002002 004767 004232                T2A:  JSR   %7,WRINT
772 002006 016777 004164 176766        MOV   FDRIVE,%MTC      ;SELECT DRIVE
773 002014 052777 040005 176760        BIS   #40005,%MTC     ;800 BPI, WRITE, GO
774 002022 005067 004362                CLR   TIME
775 002026 022777 011404 176752        T2B:  CMP   #WBUF+2,%CA ;IS 2ND WORD OUTPUT
776 002034 003403                        BLE   T2C              ;YES
777 002036 004767 004324                JSR   %7,TIMER        ;NO, COUNT TIME
778 002042 000771                        BR    T2B
779 002044 016720 004340                T2C:  MOV   TIME,(0)+   ;SAVE "WRITE START" TIME
780 002050 005067 004334                CLR   TIME
781 002054 005777 176724                TST   %B0
782 002060 001375                        BNE   .-4
783 002062 032777 000010 176710        BIT   #10,%MTC
784 002070 001774                        BEQ   .-6              ;WAIT FOR SETTLEDOWN TO SET
785 002072 006077 176702                T2D:  ROR   %MTC
786 002076 103403                        BCS   T2E              ;WAIT FOR TU READY
787 002100 004767 004262                JSR   %7,TIMER
788 002104 000772                        BR    T2D
789 002106 016721 004276                T2E:  MOV   TIME,(1)+   ;SAVE "SETTLEDOWN" TIME
790 002112 004767 004000                JSR   %7,CHGDRV
791 002116 000731                        BR    T2A
792 002120 012720 177777                MOV   #-1,(0)+       ;TERMINATE TIMES
793 002124 012721 177777                MOV   #-1,(1)+       ;TERMINATE TIMES
794 002130 012767 007627 005352        MOV   #MSG5,MESSAGE
795 002136 012700 006412                MOV   #TM1,%0
796 002142 004767 004504                JSR   %7,TYPTIM      ;PRINT "WRITE START" TIMES
797 002146 012767 011076 005334        MOV   #RMSG3,MESSAGE
798 002154 004767 005210                JSR   %7,TOP
799 002160 012767 007655 005322        MOV   #MSG6,MESSAGE
800 002166 012700 006436                MOV   #TM2,%0
801 002172 004767 004454                JSR   %7,TYPTIM      ;PRINT "SETTLEDOWN" TIMES
802 002176 012767 011115 005304        MOV   #RMSG4,MESSAGE
803 002204 004767 005160                JSR   %7,TOP
    
```

```

804                                     : TIME WRITE TO ERASE HEAD
805                                     : LONG RECORD WAS PREVIOUSLY WRITTEN
806                                     : WRITE A 3 BYTE RECORD AND POWER CLEAR
807                                     : DISTANCE FROM NEW DATA TO OLD IS
808                                     : ERASE HEAD DISTANCE
809
810 002210 004767 004062 T3: JSR %7, STREW ; START REWIND
811 002214 004767 003676 JSR %7, CHGDRV ; DONE ALL DRIVES?
812 002220 000773 BR T3 ; NO
813 002222 004767 004110 T3A: JSR %7, WATREW ; IS DRIVE AT BOT?
814 002226 004767 003664 JSR %7, CHGDRV ; DONE ALL DRIVES
815 002232 000773 BR T3A ; NO
816 002234 012777 177775 176542 T3B: MOV #-3, @BC ; 3 BYTE RECORD
817 002242 012777 011402 176536 MOV #WBUF, @CA ; INITIALIZE CURRENT ADDRESS
818 002250 016777 003722 176524 MOV FDRIVE, @MTC ; SELECT DRIVE
819 002256 052777 040005 176516 BIS #40005, @MTC ; 800BPI, WRITE, GO
820 002264 105777 176512 TSTB @MTC
821 002270 100375 BPL .-4 ; AWAIT CUR
822 002272 004767 003620 JSR %7, CHGDRV ; DONE ALL DRIVES
823 002276 000756 BR T3B ; NO
824
825                                     : NOW READ OVER PARTIAL RECORD
826
827 002300 012700 006412 MOV #TM1, %0
828 002304 012777 177776 176472 T3D: MOV #-2, @BC
829 002312 012777 011402 176466 MOV #WBUF, @CA
830 002320 016777 003652 176454 MOV FDRIVE, @MTC ; SELECT DRIVE
831 002326 052777 040003 176446 BIS #40003, @MTC ; 800BPI, READ, GO
832 002334 005067 004050 CLR TIME ; CLEAR TIME
833 002340 005777 176440 T3E: TST @BC ; WAIT FOR NEXT WORD IN
834 002344 001403 BEQ T3F ; HAVE IT
835 002346 004767 004014 JSR %7, TIMER ; NO, COUNT TIME
836 002352 000772 BR T3E
837 002354 016720 004030 T3F: MOV TIME, (0)+ ; SAVE "WRITE TO ERASE HEAD TIME"
838 002360 105777 176416 TSTB @MTC ; SEE IF CUR
839 002364 100375 BPL .-4 ; AWAIT CUR
840 002366 004767 003524 JSR %7, CHGDRV ; DONE ALL DRIVES
841 002372 000744 BR T3D ; NO
842 002374 012720 177777 MOV #-1, (0)+ ; TERMINATE TIMES
843 002400 012767 007703 005102 MOV #MSG7, MESSAGE
844 002406 012700 006412 MOV #TM1, %0
845 002412 004767 004234 JSR %7, TYPTIM ; PRINT "WRITE TO ERASE HEAD TIMES"
846 002416 012767 011134 005064 MOV #RMSG5, MESSAGE
847 002424 004767 004740 JSR %7, TOP
848 002430 004767 003642 T3G: JSR %7, STREW ; START REWIND
849 002434 004767 003456 JSR %7, CHGDRV ; DONE ALL DRIVES
850 002440 000773 BR T3G ; NO
851 002442 004767 003670 T3H: JSR %7, WATREW ; DRIVE AT BOT
852 002446 004767 003444 JSR %7, CHGDRV ; DONE ALL DRIVES
853 002452 000773 BR T3H ; NO

```

```

854      ; TIME WRITE NONSTOP GAP, BACKSPACE SHUTDOWN AND READ SHUTDOWN
855      ; WRITE ONE RECORD, FOLLOW WITH ONE RECORD NONSTOP
856      ; FOLLOWED BY ONE RECORD START-STOP
857      ; FOLLOWED BY WRITE-BACKSPACE-READ-WRITE
858      ; FOLLOWED BY WRITE-BACKSPACE-WRITE
859
860 002454 004767 003576      T4:      JSR      %7,ST1S
861 002460 012700 006412      MOV      #TM1,%0      ; INITIALIZE TIME BUFFERS
862 002464 012701 006436      MOV      #TM2,%1
863 002470 012702 006462      MOV      #TM3,%2
864 002474 005067 003710      T4AA:    CLR      TIME
865 002500 004767 003534      JSR      %7,WRINT
866 002504 016777 003466      MOV      FDRIVE,%MTC      ; TRACK AND DRIVE NUMBERS
867 002512 052777 040005      BIS      #40005,%MTC      ; 800 BPI, WRITE, GO
868 002520 105777 176256      TSTB    %MTC
869 002524 100375      BPL      -.4      ; WAIT FOR CU READY
870
871      ; HAVE FIRST RECORD WRITTEN, GO NONSTOP
872
873 002526 004767 003506      JSR      %7,WRINT
874 002532 005277 176244      INC      %MTC      ; GO
875 002536 022777 011404      T4A:    CMP      #WBUF+2,%CA      ; IS 2ND WORD OUTPUT?
876 002544 003403      BLE      T4B      ; YES
877 002546 004767 003614      JSR      %7,TIMER      ; NO, COUNT TIME
878 002552 000771      BR      T4A
879 002554 016720 003630      T4B:    MOV      TIME,(0)+      ; SAVE "WRITE NONSTOP GAP" TIME
880 002560 005067 003624      CLR      TIME
881 002564 105777 176212      TSTB    %MTC
882 002570 100375      BPL      -.4      ; WAIT FOR CU READY
883 002572 006077 176202      ROR      %MTC
884 002576 103375      BCC      -.4      ; WAIT FOR TU READY
885
886      ; WRITE-BACKSPACE-READ-WRITE
887
888 002600 004767 003434      JSR      %7,WRINT
889 002604 016777 003366      MOV      FDRIVE,%MTC      ; DRIVE SELECT
890 002612 052777 040007      BIS      #40007,%MTC      ; 800 BPI, WRITE EOF, GO
891 002620 105777 176156      TSTB    %MTC
892 002624 100375      BPL      -.4      ; WAIT FOR CU READY
893 002626 012777 177777      MOV      #-1,%BC      ; BACKSPACE 1 RECORD
894 002634 042777 000016      BIC      #16,%MTC
895 002642 052777 000013      BIS      #13,%MTC      ; SPACE REVERSE, GO
896 002650 000240      NOP
897 002652 032777 040000      T4BA:   BIT      #40000,%MTC      ; SEE IF EOF
898 002660 001774      BEQ     T4BA      ; IF NOT: BR
899 002662 000240      NOP
900 002664 000240      NOP
901 002666 032777 000010      T4C:    BIT      #10,%MTC      ; HAS SETTLEDOWN SET?
902 002674 001003      BNE     T4D      ; YES
903 002676 004767 003464      JSR      %7,TIMER      ; NO, COUNT TIME
904 002702 000771      BR      T4C
905 002704 006077 176070      T4D:    ROR      %MTC
906 002710 103375      BCC      -.4      ; WAIT FOR TU READY
907 002712 016721 003472      MOV      TIME,(1)+      ; SAVE "BACKSPACE SHUTDOWN" TIME
908 002716 004767 003316      JSR      %7,WRINT
909 002722 005067 003462      CLR      TIME

```


910	002726	016777	003244	176046		MOV	FDRIVE, @MTC	;SELECT DRIVE
911	002734	052777	040003	176040		BIS	#40003, @MTC	;800 BPI, READ, GO
912	002742	032777	040000	176030	T4DA:	BIT	#40000, @MTC	
913	002750	001774				BEQ	T4DA	;AWAIT EOF
914	002752	032777	000010	176020	T4E:	BIT	#10, @MTC	;HAS SETTLEDOWN SET?
915	002760	001003				T4F		;YES
916	002762	004767	003400			JSR	%7, TIMER	;NO, COUNT TIME
917	002766	000771				BR	T4E	
918	002770	006077	176004		T4F:	ROR	@MTC	
919	002774	103375				BCC	.-4	;WAIT FOR TU READY
920	002776	016722	003406			MOV	TIME, (2)+	;SAVE "READ SHUTDOWN" TIME
921	003002	004767	003110			JSR	%7, CHGDRV	
922	003006	000632				BR	T4AA	
923	003010	012720	177777			MOV	#-1, (0)+	;TERMINATE TIMES
924	003014	012721	177777			MOV	#-1, (1)+	;TERMINATE TIMES
925	003020	012722	177777			MOV	#-1, (2)+	;TERMINATE TIMES
926	003024	012767	007757	004456		MOV	#MSG9, MESSAGE	
927	003032	012700	006436			MOV	#TM2, %0	
928	003036	004767	003610			JSR	%7, TYPTIM	;PRINT "BACKSPACE SHUTDOWN" TIMES
929	003042	012767	011153	004440		MOV	#RMSG6, MESSAGE	
930	003050	004767	004314			JSR	%7, TOP	
931	003054	012767	010005	004426		MOV	#MSG10, MESSAGE	
932	003062	012700	006462			MOV	#TM3, %0	
933	003066	004767	003560			JSR	%7, TYPTIM	;PRINT "READ SHUTDOWN" TIMES
934	003072	012767	011171	004410		MOV	#RMSG7, MESSAGE	
935	003100	004767	004264			JSR	%7, TOP	
936	003104	004767	003146			JSR	%7, ST1S	
937	003110	004767	003162		T4FA:	JSR	%7, STREW	;START REWIND
938	003114	004767	002776			JSR	%7, CHGDRV	;CHANGE DRIVE
939	003120	000773				BR	T4FA	
940	003122	004767	003210		T4FB:	JSR	%7, WATREW	;AWAIT BOT
941	003126	004767	002764			JSR	%7, CHGDRV	;CHANGE DRIVE
942	003132	000773				BR	T4FB	
943	003134	000240				NOP		
944	003136	004767	003076		T4FC:	JSR	%7, WRINT	;SET UP FOR WRITE
945	003142	016777	003030	175632		MOV	FDRIVE, @MTC	;SET DRIVE NUMBER
946	003150	052777	040005	175624		BIS	#40005, @MTC	;SET 800 BPI. WRITE. GO
947	003156	105777	175620			TSTB	@MTC	
948	003162	100375				BPL	.-4	;AWAIT CUR
949	003164	000240				NOP		
950	003166	004767	003046			JSR	%7, WRINT	;SET UP FOR NEXT WRITE
951	003172	005277	175604			INC	@MTC	;SET GO
952	003176	105777	175600			TSTB	@MTC	
953	003202	100375				BPL	.-4	;AWAIT CUR
954	003204	004767	003030			JSR	%7, WRINT	;SET UP FOR NEXT WRITE
955	003210	005277	175566			INC	@MTC	;SET GO
956	003214	105777	175562			TSTB	@MTC	
957	003220	100375				BPL	.-4	;AWAIT CUR
958	003222	004767	002670			JSR	%7, CHGDRV	;CHANGE DRIVE
959	003226	000743				BR	T4FC	
960	003230	000240				NOP		
961								

```

962
963
964 ;WRITE RECORDS TO BE USED IN GAP TEST
965 003232 004767 003002 T4G: JSR %7,WRINT
966 003236 016777 002734 175536 MOV FDRIVE,@MTC ;SELECT DRIVE
967 003244 052777 040005 175530 BIS #40005,@MTC ;800 BPI, WRITE, GO
968 003252 105777 175524 TSTB @MTC
969 003256 100375 BPL .-4 ;WAIT FOR CU READY
970 003260 004767 002754 JSR %7,WRINT
971 003264 005277 175512 INC @MTC ;WRITE NEXT
972 003270 105777 175506 TSTB @MTC
973 003274 100375 BPL .-4 ;WAIT FOR CU READY
974 003276 012777 177777 175500 MOV #-1,@BC
975 003304 016777 002666 175470 MOV FDRIVE,@MTC ;SELECT DRIVE
976 003312 052777 040013 175462 BIS #40013,@MTC ;800 BPI, BACKSPACE, GO
977 003320 105777 175456 TSTB @MTC
978 003324 100375 BPL .-4 ;WAIT FOR CU READY
979 003326 004767 002706 JSR %7,WRINT
980 003332 016777 002640 175442 MOV FDRIVE,@MTC
981 003340 052777 040005 175434 BIS #40005,@MTC ;800 BPI, WRITE, GO
982 003346 105777 175430 TSTB @MTC
983 003352 100375 BPL .-4
984 003354 012767 177777 175454 MOV #-1,R11 ;INDICATES BACK 3 COMPLETE
985 003362 012767 177777 175450 MOV #-1,R12 ;INDICATES BACK 4 COMPLETE
986 003370 012767 177777 175444 MOV #-1,R13 ;INDICATES BACK 5 COMPLETE
987 003376 012767 177776 175430 MOV #-2,R10 ;FIRST SEQUENCE BACK 2 TIMES

```

```

988                                     :NOW WRITE, BACKSPACE, WRITE, BACKSPACE, WRITE
989                                     ;GAP SHOULD GET LARGER
990
991 003404 004767 002630      MULWRT: JSR      %7,WRINT
992 003410 005277 175366      INC      @MTC          ;GO NONSTOP
993 003414 105777 175362      TSTB    @MTC
994 003420 100375              SPL      .-4          ;WAIT FOR DONE
995 003422 012777 177777 175354 MULBAK: MC'    #-1,@BC  ;BACKSPACE 1 RECORD
996 003430 042777 000016 175344 BIC      #16,@MTC
997 003436 052777 000013 175336 BIS      #13,@MTC      ;SET BACKSPACE, GO
998 003444 105777 175332      TSTB    @MTC
999 003450 100375              SPL      .-4          ;WAIT FOR BACKSPACE DONE
1000 003452 004767 002562      JSR     %7,WRINT
1001 003456 042777 000016 175316 BIC      #16,@MTC
1002 003464 052777 000005 175310 BIS      #5,@MTC      ;SET WRITE, GO
1003 003472 105777 175304      TSTB    @MTC
1004 003476 100375              SPL      .-4          ;WAIT FOR WRITE DONE
1005 003500 005267 175330      INC     R10          ;BACKSPACED ENOUGH TIMES?
1006 003504 001346              BNE     MULBAK       ;NO BACKSPACE AND WRITE AGAIN
1007 003506 005267 175324      INC     R11          ;DONE 3 BACKSPACE SEQUENCES?
1008 003512 001004              BNE     MUL1         ;YES
1009 003514 012767 177775 175312 MOV      #-3,R10
1010 003522 000730              BR      MULWRT
1011 003524 005267 175310      MUL1: INC     R12          ;DONE 4 BACKSPACE SEQUENCES?
1012 003530 001004              BNE     MUL2         ;YES
1013 003532 012767 177774 175274 MOV      #-4,R10
1014 003540 000721              BR      MULWRT
1015 003542 005267 175274      MUL2: INC     R13          ;DONE 5 BACKSPACE SEQUENCES?
1016 003546 001004              BNE     MUL3         ;YES
1017 003550 012767 177773 175256 MOV      #-5,R10
1018 003556 000712              BR      MULWRT
1019 003560 006077 175214      MUL3: ROR     @MTC
1020 003564 103375              BCC     .-4          ;WAIT FOR TU READY
1021 003566 004767 002504      JSR     %7,STRREW    ;START REWIND
1022 003572 004767 002320      JSR     %7,CHGDRV
1023 003576 000615              BR      T4G

```

```

1024 :NOW READ NONSTOP
1025 :ACCUMULATE GAP TIMES ON READ
1026 :TYPE ACCUMULATED TIMES AT END OF READ
1027 :GAP1 SHOULD = GAP2, GAP3 < GAP1 AND GAP2
1028 :GAP4 THRU GAP8 SHOULD GET INCREASINGLY LONGER
1029 003600 C05067 175240 CLR T5DRV
1030
1031 003604 004767 002526 T5: JSR %7,WATREW
1032 003610 004767 002424 JSR %7,WRINT
1033 003614 012700 006412 MOV #TM1,%0
1034 003620 066700 175220 ADD T5DRV,%0
1035 003624 016777 002346 175150 MOV FORIVE,%MTC ;SELECT DRIVE
1036 003632 052777 040003 175142 SIS #40003,%MTC ;800 BPI. READ, GO
1037 003640 012767 177770 175166 MOV #8,%R10 ;COUNT 8 GAPS
1038 003646 105777 175130 T5A: TSTB %MTC
1039 003652 100375 BPL .-4 ;WAIT FOR CU READY
1040 003654 004767 002360 JSR %7,WRINT
1041 003660 005067 002524 CLR TIME
1042 003664 005277 175112 INC %MTC ;GO NONSTOP
1043 003670 022777 011404 175110 T5B: CMP #WBUF+2,%CA ;IS 2ND WORD OUTPUT
1044 003676 003403 BLE T5C ;YES
1045 003700 004767 002462 JSR %7,TIMER ;NO, COUNT TIME
1046 003704 000771 BR T5B
1047 003706 016720 002476 T5C: MOV TIME,(0)+ ;SAVE GAP TIME
1048 003712 012710 177777 MOV #-1,(0) ;TERMINATE, JUST IN CASE AT END
1049 003716 062700 000022 ADD #22,%0 ;STEP GAP POINTER
1050 003722 005267 175106 INC R10 ;DONE ALL 8 GAPS?
1051 003726 001347 BNE T5A ;NO
1052 003730 006077 175044 ROR %MTC
1053 003734 103375 BCC .-4 ;WAIT FOR TU READY
1054 003736 004767 002334 JSR %7,STRREW ;START REWIND
1055 003742 062767 000002 175074 ADD #2,T5DRV ;+2 TO DRIVE TIME POINTER
1056 003750 004767 002142 JSR %7,CHGDRV
1057 003754 000713 BR T5
1058 003756 112767 000061 004134 MOVB #1,MSG11A+6
1059 003764 012767 010033 003516 MOV #MSG11,MESAGE
1060 003772 004767 003372 JSR %7,TOP
1061 003776 012767 010112 003504 MOV #MSG11A,MESAGE
1062 004004 012700 006412 MOV #TM1,%0
1063 004010 004767 002636 JSR %7,TYPTIM ;PRINT "GAP 1"
1064 004014 105267 004100 INCB MSG11A+6
1065 004020 012767 010112 003462 MOV #MSG11A,MESAGE
1066 004026 012700 006436 MOV #TM2,%0
1067 004032 004767 002614 JSR %7,TYPTIM ;PRINT "GAP 2"
1068 004036 105267 004056 INCB MSG11A+6
1069 004042 012767 010112 003440 MOV #MSG11A,MESAGE
1070 004050 012700 006462 MOV #TM3,%0
1071 004054 004767 002572 JSR %7,TYPTIM ;PRINT "GAP 3"
1072 004060 105267 004034 INCB MSG11A+6
1073 004064 012767 010112 003416 MOV #MSG11A,MESAGE
1074 004072 012700 006506 MOV #TM4,%0
1075 004076 004767 002550 JSR %7,TYPTIM ;PRINT "GAP 4"
1076 004102 105267 004012 INCB MSG11A+6
1077 004106 012767 010112 003374 MOV #MSG11A,MESAGE
1078 004114 012700 006532 MOV #TM5,%0
1079 004120 004767 002526 JSR %7,TYPTIM ;PRINT "GAP 5"

```

```

1080 004124 105267 003770          INCB  MSG11A+6
1081 004130 012767 010112 003352      MOV   #MSG11A,MESAGE
1082 004136 012700 006556          MOV   #TM6,%0
1083 004142 004767 002504          JSR   %7,TYPTIM      ;PRINT "GAP 6"
1084 004146 105267 003746          INCB  MSG11A+6
1085 004152 012767 010112 003330      MOV   #MSG11A,MESAGE
1086 004160 012700 006502          MOV   #TM7,%0
1087 004164 004767 002462          JSR   %7,TYPTIM      ;PRINT "GAP 7"
1088 004170 105267 003724          INCB  MSG11A+6
1089 004174 012767 010112 003306      MOV   #MSG11A,MESAGE
1090 004202 012700 006626          MOV   #TM8,%0
1091 004206 004767 002440          JSR   %7,TYPTIM      ;PRINT "GAP 8"
1092 004212 004767 002040          JSR   %7,STIS
1093
1094                                     ;TIME WRITE START NOT AT BOT
1095
1096 004216 012700 006412      T6:   MOV   #TM1,%0
1097 004222 012701 006436          MOV   #TM2,%1
1098 004226 004767 002006      T6A:  JSR   %7,WRINT
1099 004232 016777 001740 174542      MOV   FDRIVE,%MTC      ;SELECT DRIVE
1100 004240 105777 174536          TSTB  %MTC
1101 004244 100375          BPL   -4
1102 004246 006077 174526          ROR   %MTC
1103 004252 103375          BCC   -4      ;WAIT FOR TU READY
1104 004254 052777 040005 174520      BIS   #40005,%MTC      ;800 BPI, WRITE, GO
1105 004262 032777 000040 174510          BIT   #40,%MTC
1106 004270 001374          BNE   -6      ;WAIT FOR BOT TO CLEAR
1107 004272 052777 010000 174502      BIS   #10000,%MTC      ;POWER CLEAR
1108 004300 016777 001672 174474          MOV   FDRIVE,%MTC
1109 004306 004767 001726          JSR   %7,WRINT
1110 004312 006077 174462          ROR   %MTC
1111 004316 103375          BCC   -4      ;WAIT FOR TU READY
1112 004320 005067 002064          CLR   TIME
1113 004324 016777 001646 174450      MOV   FDRIVE,%MTC      ;SELECT DRIVE
1114 004332 052777 040005 174442      BIS   #40005,%MTC      ;800 BPI, WRITE, GO
1115 004340 022777 011404 174440      T6B:  CMP   #WBUF+2,%CA      ;IS 2ND WORD OUTPUT?
1116 004346 003403          BLE   T6C          ;YES
1117 004350 004767 002012          JSR   %7,TIMER        ;NO, COUNT TIME
1118 004354 000771          BR    T6B
1119 004356 006077 174416      T6C:  ROR   %MTC
1120 004362 103375          BCC   -4      ;WAIT FOR TU READY
1121 004364 016720 002020          MOV   TIME,(0)+      ;SAVE "WRITE START" TIME
1122 004370 005067 002014          CLR   TIME
1123 004374 004767 001640          JSR   %7,WRINT
1124 004400 016777 001572 174374      MOV   FDRIVE,%MTC      ;SELECT DRIVE
1125 004406 052777 040015 174366      BIS   #40015,%MTC      ;800 BPI, WRITE XIRG. GO
1126 004414 022777 011404 174364      T6D:  CMP   #WBUF+2,%CA      ;IS 2ND WORD OUTPUT?
1127 004422 003403          BLE   T6E          ;YES
1128 004424 004767 001736          JSR   %7,TIMER        ;NO COUNT TIME
1129 004430 000771          BR    T6D
1130 004432 006077 174342      T6E:  ROR   %MTC
1131 004436 103375          BCC   -4      ;WAIT FOR TU READY
1132 004440 016721 001744          MOV   TIME,(1)+      ;SAVE "WRITE XIRG" TIME
1133 004444 004767 001626          JSR   %7,STARREW
1134 004450 004767 001442          JSR   %7,CHGDRV
1135 004454 000664          BR    T6A

```

```

1136 004456 012720 177777 MOV # -1,(0)+ ;TERMINATE TIMES
1137 004462 012721 177777 MOV # -1,(1)+ ;TERMINATE TIMES
1138 004466 012767 007627 003014 MOV #MSG5,MESSAGE
1139 004474 012700 006412 MOV #TM1,%0
1140 004500 004767 002146 JSR %7,TYPTIM ;TYPE "WRITE START" TIME
1141 004504 012767 011207 002776 MOV #RMSG8,MESSAGE
1142 004512 004767 002652 JSR %7,TOP
1143 004516 012767 010140 002764 MOV #MSG12,MESSAGE
1144 004524 012700 006436 MOV #TM2,%0
1145 004530 004767 002116 JSR %7,TYPTIM ;TYPE "WRITE XIRG" TIME
1146 004534 012767 011226 002746 MOV #RMSG9,MESSAGE
1147 004542 004767 002622 JSR %7,TOP
1148 004546 004767 001564 T6F: JSR %7,WATREW
1149 004552 004767 001340 JSR %7,CHGDRV
1150 004556 000773 BR T6F ;WAIT FOR ALL DRIVES AT BOT.
1151
1152 ;NOW TIME "READ FROM BOT DELAY"
1153
1154 004560 012700 006412 T7: MOV #TM1,%0
1155 004564 005067 001620 T7A: CLR TIME
1156 004570 004767 001444 JSR %7,WRINT
1157 004574 016777 001376 174200 MOV FDRIVE,%MTC ;SELECT DRIVE
1158 004602 052777 040003 174172 BIS #40003,%MTC ;800 BPI, READ GO
1159 004610 022777 011404 174170 T7B: CMP #WBUF+2,%CA ;IS 2ND WORD INPUT?
1160 004616 003403 BLE T7C ;YES
1161 004620 004767 001542 JSR %7,TIMER ;NO COUNT TIME
1162 004624 000771 BR T7B
1163 004626 016720 001556 T7C: MOV TIME,(0)+ ;SAVE "READ FROM BOT" TIME
1164 004632 105777 174144 TSTB %MTC
1165 004636 100375 BPL #-4 ;WAIT FOR CU READY.
1166 004640 004767 001252 JSR %7,CHGDRV ;DONE ALL DRIVES?
1167 004644 000747 BR T7A ;NO
1168 004646 006077 174126 ROR %MTC
1169 004652 103375 BCC #-4
1170 004654 012720 177777 MOV # -1,(0)+ ;TERMINATE TIMES
1171 004660 012767 010166 002622 MOV #MSG13,MESSAGE
1172 004666 012700 006412 MOV #TM1,%0
1173 004672 004767 001754 JSR %7,TYPTIM ;PRINT "READ FROM BOT" TIME
1174 004676 012767 011246 002604 MOV #RMSG10,MESSAGE
1175 004704 004767 002460 JSR %7,TOP
1176 004710 004767 001342 JSR %7,ST15
1177
1178 ;TIME "LAST CHARACTER INPUT TO CU READY"
1179
1180 004714 012700 006412 T8: MOV #TM1,%0
1181 004720 004767 001314 T8A: JSR %7,WRINT
1182 004724 005067 001460 CLR TIME
1183 004730 016777 001242 174044 MOV FDRIVE,%MTC ;SELECT DRIVE
1184 004736 052777 040003 174036 BIS #40003,%MTC ;800 BPI, READ. GO
1185 004744 005777 174034 TST %BC
1186 004750 001375 BNE #-4 ;WAIT FOR LAST WORD IN
1187 004752 105777 174024 T8B: TSTB %MTC ;IS CU READY?
1188 004756 100403 BMI T8C ;YES
1189 004760 004767 001402 JSR %7,TIMER ;NO, COUNT TIME
1190 004764 000772 BR T8B
1191 004766 006077 174006 T8C: ROR %MTC

```

E03

MAINDEC-11-DZTSE-A-D
DZTSEA.P11

TS03 DRIVE FUNCTION TIMER

MACY11 27(732) 10-SEP-76 16:11 PAGE 30

```

1192 004772 103375          BCC      .-4          ;WAIT FOR TU READY
1193 004774 016720 001410  MOV      TIME,(0)+   ;SAVE "LAST CHAR TO CU READY" TIME
1194 005000 004767 001272  JSR      %7,STRREW   ;REWIND
1195 005004 004767 001106  JSR      %7,CHGDRV   ;ANYMORE DRIVES?
1196 005010 000743          BR       T8A        ;YES
1197 005012 012720 177777  MOV      #-1,(0)+   ;TERMINATE TIMES
1198 005016 004767 001314  T8D:    JSR      %7,WATREW
1199 005022 004767 001070  JSR      %7,CHGDRV
1200 005026 000773          BR       T8D

;TIME "WRITE EOF"
;WRITE A 3 BYTE RECORD FROM BOT FOLLOWED BY AN EOF.
1205 005030 012700 006412  T9:     MOV      #TMI,%0
1206 005034 005067 001350  T9A:    CLR      TIME
1207 005040 012777 177775 173736  MOV      #-3,%B0C   ;WRITE 3 BYTES
1208 005046 012777 011402 173732  MOV      #WBUF,%CA
1209 005054 016777 001116 173720  MOV      %DRIVE,%MTC ;SELECT DRIVE
1210 005062 052777 040005 173712  BIS      #40005,%MTC ;800 BPI, WRITE, GO
1211 005070 105777 173706          TSTB     %MTC
1212 005074 100375          BPL      .-4
1213 005076 006077 173676  ROR      %MTC
1214 005102 103375          BCC      .-4          ;WAIT FOR TU READY
1215 005104 042777 000016 173670  BIC      #16,%MTC
1216 005112 052777 000007 173662  BIS      #7,%MTC
1217 005120 105777 173656  T9B:    TSTB     %MTC
1218 005124 100403          BMI      T9C
1219 005126 004767 001234  JSR      %7,TIMER
1220 005132 000772          BR       T9B
1221 005134 016720 001250  T9C:    MOV      TIME,(0)+   ;SAVE "WRITE EOF" TIME
1222 005140 004767 001132  JSR      %7,STRREW   ;REWIND
1223 005144 004767 000746  JSR      %7,CHGDRV   ;ANYMORE DRIVES?
1224 005150 000731          BR       T9A        ;YES
1225 005152 012720 177777  MOV      #-1,(0)+   ;TERMINATE TIMES
1226 005156 012767 010242 002324  MOV      #MSG15,MESAGE
1227 005164 012700 006412          MOV      #TMI,%0
1228 005170 004767 001456          JSR      %7,TYPTIM   ;PRINT "WRITE EOF" TIMES
1229 005174 012767 011266 002306  MOV      #RMSG11,MESAGE
1230 005202 004767 002162          JSR      %7,TOP
1231 005206 004767 001124  T9D:    JSR      %7,WATREW
1232 005212 004767 000700  JSR      %7,CHGDRV
1233 005216 000773          BR       T9D

```

F03

MAINDEC-11-DZTSE-A-D
DZTSEA.P11

TS03 DRIVE FUNCTION TIMER

MACY11 27(732) 10-SEP-76 16:11 PAGE 31

```

1234 :TIME "EOR TO EOF SPACE TIME", "SPACE SHUTDOWN" AND "ONE INCH DATA TIME".
1235 :WRITE A 3 BYTE RECORD OVER ONE PREVIOUSLY WRITTEN
1236 :AND THEN SPACE FORWARD 1 RECORD
1237 :TIME FROM GO UNTIL EOF IS REACHED
1238
1239 005220 012700 006412 T10: MOV #TM1,%0
1240 005224 012701 006436 MOV #TM2,%1
1241 005230 012702 006462 MOV #TM3,%2
1242 005234 005067 001150 T10A: CLR TIME
1243 005240 012777 173536 MOV #-3,%BC ;3 BYTE RECORD
1244 005246 012777 011402 173532 MOV #WBUF,%CA
1245 005254 016777 000716 173520 MOV FDRIVE,%MTC ;SELECT DRIVE
1246 005252 105777 173514 TSTB %MTC
1247 005266 100375 BPL -4 ;WAIT FOR CU READY
1248 005270 012777 177776 173506 MOV #-2,%BC ;SPACE FORWARD 2 RECORDS
1249 005276 042777 000016 173476 BIC #16,%MTC
1250 005304 052777 000011 173470 BIS #11,%MTC ;SPACE FORWARD, GO
1251 005312 022777 177777 173464 T10B: CMP #-1,%BC
1252 005320 001374 BNE T10B ;WAIT FOR 1ST RECORD TO BE SPACED OVER
1253 005322 032777 040000 173450 T10C: BIT #40000,%MTC ;IS EOF SET?
1254 005330 001014 BNE T10D ;YES
1255 005332 105777 173444 TSTB %MTC ;IS CU READY
1256 005336 100403 BMI T10CC ;YES
1257 005340 004767 001022 JSR %7,TIMER ;NO, COUNT TIME
1258 005344 000766 BR T10C
1259 005346 032777 040000 173424 T10CC: BIT #40000,%MTC ;HAVE EOF
1260 005354 001002 BNE T10D ;IS EOF SET?
1261 005356 005067 001026 CLR TIME ;NO, SET ERROR
1262 005362 016720 001022 T10D: MOV TIME,(0)+ ;SAVE "EOR TO EOF SPACE TIME"
1263 005366 005067 001016 CLR TIME
1264 005372 000240 NOP
1265 005374 000240 NOP
1266 005376 000240 NOP
1267 005400 032777 000010 173372 T10E: BIT #10,%MTC ;IS SETTLEDOWN SET?
1268 005406 001003 BNE T10F ;YES
1269 005410 004767 000752 JSR %7,TIMER ;NO, COUNT TIME
1270 005414 000771 BR T10E
1271 005416 016721 000766 T10F: MOV TIME,(1)+ ;SAVE "SPACE SHUTDOWN" TIME
1272 005422 105777 173354 TSTB %MTC ;SEE IF CUR
1273 005426 100375 BPL -4 ;IF NOT: BR
1274 005430 012777 176340 173346 MOV #-800,%BC ;1 INCH OF DATA
1275 005436 012777 011402 173342 MOV #WBUF,%CA
1276 005444 005067 000740 CLR TIME
1277 005450 016777 000522 173324 MOV FDRIVE,%MTC ;SELECT DRIVE
1278 005456 105777 173320 TSTB %MTC ;WAIT FOR CU READY
1279 005462 100375 BPL -4
1280 005464 052777 040005 173310 BIS #40005,%MTC ;800 BPI, WRITE, GO
1281 005472 022777 011404 173306 CMP #WBUF+2,%CA ;IS 2ND BYTE OUTPUT
1282 005500 003374 BGT -6 ;NO
1283 005502 005777 173276 T10G: TST %BC ;YES IS LAST BYTE OUT
1284 005506 001403 BEQ T10H ;YES
1285 005510 004767 000652 JSR %7,TIMER ;NO, COUNT TIME
1286 005514 000772 BR T10G
1287
1288 005516 016722 000666 T10H: MOV TIME,(2)+ ;SAVE "ONE INCH DATA TIME"
1289 005522 004767 000550 JSR %7,STRREW ;REWIND

```



```

1290 005526 004767 000364 JSR %7,CHGDRV ;ANYMORE DRIVES?
1291 005532 000640 BR T10A ;YES
1292 005534 012720 177777 MOV #-1,(0)+ ;TERMINATE TIMES
1293 005540 012721 177777 MOV #-1,(1)+
1294 005544 012722 177777 MOV #-1,(2)+
1295 005550 012767 010270 001732 MOV #MSG16,MESAGE
1296 005556 012700 006412 MOV #TM1,%0
1297 005562 004767 001064 JSR %7,TYPTIM ;PRINT "EOR TO EOF SPACE TIME"
1298 005566 012767 011306 001714 MOV #RMSG12,MESAGE
1299 005574 004767 001570 JSR %7,TOP
1300 005600 012767 010316 001702 MOV #MSG18,MESAGE
1301 005606 012700 006436 MOV #TM2,%0
1302 005612 004767 001034 JSR %7,TYPTIM ;PRINT "SPACE SHUTDOWN" TIME
1303 005616 012767 011326 001664 MOV #RMSG13,MESAGE
1304 005624 004767 001540 JSR %7,TOP
1305 005630 012767 010344 001652 MOV #MSG20,MESAGE
1306 005636 012700 006462 MOV #TM3,%0
1307 005642 004767 001004 JSR %7,TYPTIM ;PRINT "ONE INCH DATA TIME"
1308 005646 012767 011344 001634 MOV #RMSG14,MESAGE
1309 005654 004767 001510 JSR %7,TOP
1310 005660 000167 000146 JMP T13 ;GO TO END OF TIMING
1311
1312 ;SPECIAL TAPE SPEED TEST USING 800 BPI SKEW TAPE
1313 ;READ FOWARD FROM BOT. MONITOR CA UNTIL IT EQUALS WBUF+2.
1314 ;TIME FROM WBUF+2 UNTIL BC=0 IS ONE INCH OF TAPE.
1315
1316 005664 000240 T11: NOP
1317 005666 012700 006412 MOV #TM1,%0
1318 005672 005067 000512 T11A: CLR TIME ;INITIALIZE TIME
1319 005676 016777 000274 173076 MOV FDRIVE,%MTC ;SELECT DRIVE
1320 005704 105777 173072 TSTB %MTC
1321 005710 100375 BPL -.4 ;AWAIT CUR
1322 005712 012777 176340 173064 MOV #-800,%BC ;800 BYTES =1 INCH OF DATA
1323 005720 012777 011402 173060 MOV #WBUF,%CA ;SET BUS ADDRESS
1324 005726 052777 040003 173046 BIS #40003,%MTC ;LOAD 800 BPI READ AND GO
1325 005734 022777 011404 173044 CMP #WBUF+2,%CA
1326 005742 003374 BGT -.6 ;AWAIT 2ND BYTE
1327 005744 005777 173034 T11B: TST %BC ;SEE IF LAST BYTE
1328 005750 001403 BEQ T11C ;IF SO: BR
1329 005752 004767 000410 JSR %7,TIMER ;NO, COUNT TIME
1330 005756 000772 BR T11B
1331 005760 016720 000424 T11C: MOV TIME,(0)+ ;SAVE TAPE SPEED TIME
1332 005764 004767 000314 JSR %7,STRW ;REWIND
1333 005770 004767 000122 JSR %7,CHGDRV ;ANYMORE DRIVER?
1334 005774 000736 BR T11A ;YES
1335 005776 012720 177777 MOV #-1,(0)+ ;TERMINALS TIMER
1336 006002 012767 010372 001500 MOV #MSG21,MESAGE
1337 006010 012700 006412 MOV #TM1,%0 ;PRINT "TAPE SPEED FWD"
1338 006014 004767 000632 JSR %7,TYPTIM
1339 006020 012767 011363 001462 MOV #RMSG15,MESAGE
1340 006026 004767 001336 JSR %7,TOP ;PRINT RANGE
1341 006032 012767 010420 001450 T13: MOV #MSG27,MESAGE
1342 006040 004767 001324 JSR %7,TOP ;PRINT "END OF TIMING"
1343 006044 000000 HALT
1344 006046 000167 173220 JMP STO
1345

```

:RESET DRIVE SELECTION TO LOWEST NUMBER

```

1346
1347
1348 006052 005067 000112 RSFDRV: CLR CDRIVE ;START WITH DRIVE 0
1349 006056 012767 100000 000106 MOV #100000,CDRVBT ;INITIALIZE FOR 0
1350 006064 036767 000102 000102 RSF1: BIT CDRVBT,DRIVES ;MASK WITH SELCTED DRIVES .
1351 006072 001006 BNE RSF2
1352 006074 005267 000070 INC CDRIVE ;+1 TO DRIVE NUMBER
1353 006100 000241 CLC
1354 006102 006067 000064 ROR CDRVBT ;MOVE MASK BIT TO NEXT DRIVE
1355 006106 000766 BR RSF1
1356 006110 004767 000064 RSF2: JSR %7,GTNINE ;CHECK 3 TRACK
1357 006114 000207 RTS
1358
1359

```

```

1359                                     ;SELECT NEXT DRIVE IN SEQUENCE
1360                                     ;SKIP FIRST EXIT ADDRESS IF LAST DRIVE SELECTED
1361
1352 006116 105777 172660          CHGDRV: TSTB   @MTC
1363 006122 100375                BPL       -4           ;AWAIT CUR
1364 006124 005267 000040          INC       CDRIVE      ;+1 TO DRIVE
1365 006130 000241                CLC
1366 006132 006067 000034          ROR       CDRVBT      ;MOVE MASK BIT TO NEXT DRIVE
1367 006136 001005                BNE      CHG1
1368 006140 004767 177706          JSR      %7,RSFDRV    ;RESET TO LOWEST DRIVE
1369 006144 062716 000002          ADD      #2,(6)      ;+2 TO SKIP FIRST EXIT
1370 006150 000207                RTS
1371 006152 036767 000014 000014  CHG1: BIT     CDRVBT,DRIVES ;MASK WITH SELECTED DRIVES
1372 006160 001756                BEQ      CHGDRV       ;CHECK FOR NEXT DRIVE
1373 006162 004767 000012          JSR      %7,GTNINE    ;CHECK 9 TRACK
1374 006166 000207                RTS
1375 006170 000000          CDRIVE: 0
1376 006172 000000          CDRVBT: 0
1377 006174 000000          DRIVES: 0
1378 006176 000000          FDRIVE: 0
1379
1380                                     ;CHECK FOR NINE TRACK DRIVES
1381
1382 006200 016767 177764 177770  GTNINE: MOV    CDRIVE,FDRIVE
1383 006206 000367 177764          SWAB    FDRIVE      ;POSITION UNIT SELECT BITS
1384 006212 042767 174377 177756  BIC     #174377,FDRIVE ;CLEAR ALL OTHER BITS
1385 006220 032767 000010 177742  BIT     #10,CDRIVE    ;TEST FOR 9 TRACK
1386 006226 001403                BEQ     GNT1         ;NO
1387 006230 052767 020000 177740  BIS     #20000,FDRIVE ;YES SET 9 TRACK BIT
1388 006236 000207          GNT1:  RTS     %7
1389
1390                                     ;INITIALIZE BYTE COUNT AND CURRENT ADDRESS FOR WRITE
1391
1392 006240 012777 172110 172536  WRINT:  MOV    #-BLENGTH,@BC
1393 006246 012777 011402 172532  MOV    #WBUF,@CA
1394 006254 000207          RTS     %7
1395
1396                                     ;STORE 1'S IN WRITE BUFFER
1397
1398 006256 012700 011402          ST1S:  MOV    #WBUF,%0
1399 006262 012720 177777          ST1SA: MOV    #-1,(0)+
1400 006266 022700 017274          CMP    #WBUF+BLENGTH+2,%0
1401 006272 001373                BNE    ST1SA
1402 006274 000207          RTS     %7
1403
1404                                     ;START REWIND OPERATIONS
1405
1406 006276 016777 177674 172476  STRREW: MOV    FDRIVE,@MTC ;SELECT DRIVE
1407 006304 105777 172472          STRW:  TSTB   @MTC
1408 006310 100375                BPL     -4           ;WAIT FOR CU READY
1409 006312 006077 172462          ROR     @MTC
1410 006316 103375                BCC     -4           ;WAIT FOR TAPE UNIT READY
1411 006320 052777 000017 172454  BIS     #17,@MTC     ;GO REWIND
1412 006326 105777 172450          TSTB   @MTC
1413 006332 100375                BPL     -4           ;WAIT FOR CONTROL UNIT READY
1414 006334 000207          RTS     %7

```

```

1415
1416           ;WAIT FOR REWIND TO FINISH
1417
1418 006336 016777 177634 172436 WATREW: MOV     FDRIVE,@MTC
1419 006344 006077 172430         ROR     @MTC
1420 006350 103375         BCC     .-4
1421 006352 032777 000040 172420 BIT     #40,@MTC       ;IS BOT SET?
1422 006360 001001         BNE     .+4             ;YES
1423 006362 000000         HALT
1424 006364 000207         RTS     %?             ;ERROR, NOT AT BOT AFTER REWIND
1425
1426           ;KEEP COUNT OF ELAPSED TIME
1427           ;EXIT EVERY 100 USEC
1428
1429 006366 005777 172420 TIMER: TST     @MTRD
1430 006372 100375         BPL     .-4
1431 006374 005777 172412         TST     @MTRD
1432 006400 100775         BMI     .-4
1433 006402 005267 000002         INC     TIME       ;+1 TO 100 USEC COUNT
1434 006406 000207         RTS     %?
1435 006410 000000 TIME: 0
1436 006412 000000 TM1: 0
1437         006436         . = TM1+20.
1438 006436 000000 TM2: 0
1439         006462         . = TM2+20.
1440 006462 000000 TM3: 0
1441         006506         . = TM3+20.
1442 006506 000000 TM4: 0
1443         006532         . = TM4+20.
1444 006532 000000 TMS: 0
1445         006556         . = TMS+20.
1446 006556 000000 TMS: 0
1447         006602         . = TM6+20.
1448 006602 000000 TM7: 0
1449         006626         . = TM7+20.
1450 006626 000000 TMS: 0
1451         006652         . = TMS+20.

```

```

1452                                     :PRINT TITLE OF TEST EXECUTED AND THE DRIVE TIMES
1453
1454 006652 004767 000512                TYPTIM: JSR    %7, TOP                ;PRINT TITLE
1455 006656 012067 000224                TYPT0:  MOV    (0)+, VALUE            ;GET TIME
1456 006662 022767 177777 000216        CMP     #-1, VALUE                ;FINISHED TIME BUFFER
1457 006670 001001                        BNE     .+4
1458 006672 000207                        RTS     %7
1459 006674 012767 007120 000214        MOV     #DECPNT+2, DECPNT; INITIALIZE DECIMAL VALUE POINTER
1460 006702 012767 000040 000204        MOV     #40, ZERO                ;INITIALIZE SPACE
1461 006710 012767 177774 000172        MOV     #-4, DIGCNT              ;DIGIT COUNT
1462 006716 012767 177777 000166        TYPT1: MOV     #-1, DIGIT          ;INITIAL VALUE
1463 006724 005267 000162                TYPT2: INC     DIGIT              ;+1 TO VALUE
1464 006730 167767 000162 000150        SUB     @DECPNT, VALUE           ;SUBTRACT CONSTANT
1465 006736 100372                        BPL     TYPT2                    ;NOT NEGATIVE YET
1466 006740 067767 000152 000140        ADD     @DECPNT, VALUE           ;RESTORE LAST POSITIVE VALUE
1467 006746 004767 000064                JSR     %7, DECOUT              ;PRINT DECIMAL DIGIT
1468 006752 005267 000132                INC     DIGCNT                  ;+1 TO DIGIT COUNT
1469 006756 001006                        BNE     TYP2A
1470 006760 012767 007547 000522        MOV     #MSG2B, MESSAGE
1471 006766 004767 000376                JSR     %7, TOP
1472 006772 000731                        BR      TYPT0
1473 006774 022767 177777 000106        TYP2A: CMP     #-1, DIGCNT        ;CHECK FOR DECIMAL PLACE
1474 007002 001011                        BNE     TYPT3                    ;NO
1475 007004 105777 172012                TSTB   @TPS
1476 007010 100375                        BPL     .-4
1477 007012 012777 000056 172004        MOV     #'., @TPB                ;PRINT DECIMAL POINT
1478 007020 012767 000060 000066        MOV     #60, ZERO
1479 007026 062767 000002 000062        TYPT3: ADD     #2, DECPNT        ;+2 TO DECIMAL VALUE POINTER
1480 007034 000730                        BR      TYPT1                    ;DO AGAIN
1481
1482 007036 005767 000050                DECOUT: TST     DIGIT            ;IS DIGIT 0
1483 007042 001004                        BNE     DEC1                    ;NO
1484 007044 016767 000044 000040        MOV     ZERO, DIGIT            ;SUPPRESS LEADING ZEROS
1485 007052 000406                        BR      DEC2
1486 007054 012767 000060 000032        DEC1:  MOV     #60, ZERO        ;INITIALIZE ZERO AFTER SOME VALUE FOUND
1487 007062 052767 000060 000022        BIS     #60, DIGIT            ;CONVERT TO ANSCII
1488 007070 105777 171726                DEC2:  TSTB   @TPS
1489 007074 100375                        BPL     .-4
1490 007076 016777 000010 171720        MOV     DIGIT, @TPB            ;PRINT
1491 007104 000207                        RTS     %7
1492 007106 000000                VALUE: 0
1493 007110 000000                DIGCNT: 0
1494 007112 000000                DIGIT: 0
1495 007114 000040                ZERO: 40                        ;CONTAINS ZERO OR SPACE
1496 007116 007120                DECPNT: .+2
1497 007120 001750                1000.
1498 007122 000144                100.
1499 007124 000012                10.
1500 007126 000001                1.

```

```

1501 ;*****
1502 ;TTY ENTRY SUBROUTINE:
1503 ;
1504 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
1505 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
1506 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
1507 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
1508 ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
1509 ;THE CALLING ROUTINE.
1510 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
1511 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
1512 ;MAY BE REENTERED.
1513 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
1514 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
1515 ;CARRIAGE RETURN
1516 ;*****
1517
1518 007130 005067 171716 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
1519 007134 005000 CLR %0
1520 007136 004767 000154 TTRO: JSR %7,TTIN ;GO READ CHARACTER
1521 007142 122767 000215 171700 CMPB #215,TIB ;SEE IF CR
1522 007150 001005 BNE TTR1 ;IF NOT: BR
1523 007152 005767 171674 TST TEMP1 ;SEE IF FIRST CHARACTER
1524 007156 001446 BEQ TTR5 ;IF SO: BR
1525 007160 000167 000066 JMP TTR2 ;ELSE GO LOAD VALUE
1526 007164 122767 000260 171656 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN C
1527 007172 101402 BLOS TTR1A ;IF NOT: BR
1528 007174 000167 000076 JMP TTR1 ;ELSE GO TO ERROR
1529 007200 122767 000270 171642 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THEN 7
1530 007206 101002 BHI TTR1B ;IF NOT: BR
1531 007210 000167 000062 JMP TTR1 ;ELSE GO TO ERROR
1532 007214 005267 171632 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
1533 007220 000241 CLC
1534 007222 006100 ROL %0
1535 007224 000241 CLC
1536 007226 006100 ROL %0 ;SHIFT 3 LEFT
1537 007230 000241 CLC
1538 007232 006100 ROL %0
1539 007234 042767 177770 171606 BIC #177770,TIB ;STRIP ASCII
1540 007242 056700 171602 BIS TIB,%0 ;LOAD CHARACTER
1541 007246 005301 DEC %1 ;SEE IF DONE
1542 007250 001332 BNE TTRO ;IF NOT: BR
1543 007252 020002 TTR2: CMP %0,%2 ;SEE IF EXCEEDED MAXIMUM LIMIT
1544 007254 101402 BLOS TTR3 ;IF NOT: BR
1545 007256 000167 000014 JMP TTR1 ;ELSE GO TO ERROR
1546 007262 020300 TTR3: CMP %3,%0 ;SEE IF BELOW MINIMUM LIMIT
1547 007264 101402 BLOS TTR4 ;IF NOT: BR
1548 007266 000167 000004 JMP TTR1 ;ELSE GO TO ERROR
1549 007272 010015 TTR4: MOV %0,(5) ;LOAD VALUE
1550 007274 000207 TTR5: RTS %7 ;EXIT

```

```

1551
1552                                     ;TTY ENTRY ERROR SUBROUTINE*****
1553
1554 007276 012767 011032 000204 T1NER: MOV      #MSG34,MESAGE
1555 007304 004767 000060          JSR      %7, TOP          ;PRINT?
1556 007310 162716 000020          SUB      #20,(6)         ;RESET SP TO START OF VALUE ROUTINE
1557 007314 000207          RTS      %7          ;READO VALUE ENTRY
1558
1559                                     ;TTY READ SUBROUTINE*****
1560
1561 007316 005077 171474          TTIN:  CLR      @TKS
1562 007322 005077 171472          CLR      @TK9
1563 007326 005067 171516          CLR      TIB
1564 007332 005277 171460          INC      @TKS
1565 007336 105777 171454          TTIN1: TSTB     @TKS
1566 007342 100375          BPL      TTIN1
1567 007344 017767 171450 171476  MOV      @TKB,TIB
1568 007352 105777 171444          TTIN2: TSTB     @TPS
1569 007356 100375          BPL      TTIN2
1570 007360 116777 171464 171436  MOVB     TIB,@TPB
1571 007366 000207          RTS      %7
1572
1573                                     ;TELETYPE OUTPUT PACKAGE
1574
1575 007370 142777 000177 171424  TOP:   BICB     #177,@TPS          ;CLEAR FLAGS
1576 007376 117767 000106 000102  MOVB     @MESSAGE,EOMK      ;SAVE MESSAGE DELIMETER
1577 007404 005267 000100          INC      MESSAGE          ;+2 TO POINTER
1578 007410 127767 000074 000070  TOP1:  CMPB     @MESSAGE,EOMK    ;IS CHARACTER THE 2ND DELIMETER
1579 007416 001001          BNE      .+4              ;NO
1580 007420 000207          RTS      %7              ;YES END
1581 007422 127727 000062 000100  CMPB     @MESSAGE,#'a      ;IS CHARACTER AN a INDICATING A CARRIAGE RETURN
1582 007430 001411          BEQ      TOP3              ;YES
1583 007432 105777 171364          TSTB     @TPS
1584 007436 100375          BPL      .-4
1585 007440 117777 000044 171356  TOP2:  MOVB     @MESSAGE,@TPB ;PRINT CHARACTER
1586 007446 005267 000036          INC      MESSAGE          ;+2 TO POINTER
1587 007452 000756          BR       TOP1              ;LOOP
1588
1589                                     ;CARRIAGE RETURN, LINE FEED
1590
1591 007454 105777 171342          TOP3:  TSTB     @TPS
1592 007460 100375          BPL      .-4
1593 007462 112777 000215 171334  MOVB     #215,@TPB
1594 007470 105777 171326          TSTB     @TPS
1595 007474 100375          BPL      .-4
1596 007476 112777 000212 171320  MOVB     #212,@TPB
1597 007504 000760          BR       TOP2
1598 007506 000000          EOMK:  0

```

1599	007510	000000			MESSAGE: 0
1600	007512	040057	043100	047125	MSG2: .ASCII ;/QFUNCTION /;
1601	007520	052103	047511	020116	
1602	007526	020040	020040	020040	
1603	007534	020040	057		
1604	007537	057	052440	044516	MSG2A: .ASCII ;/ UNIT /;
1605	007544	020124	057		
1606	007547	057	020040	057	MSG2B: .ASCII ;/ /;
1607	007553	057	053500	044522	MSG3: .ASCII ;/QWRITE FROM BOT /;
1608	007560	042524	043040	047522	
1609	007566	020115	047502	020124	
1610	007574	020040	020040	057	
1611	007601	057	053500	044522	MSG4: .ASCII ;/QWRITE SHUTDOWN /;
1612	007606	042524	051440	052510	
1613	007614	042124	053517	020116	
1614	007622	020040	020040	057	
1615	007627	057	053500	044522	MSG5: .ASCII ;/QWRITE START /;
1616	007634	042524	051440	040524	
1617	007642	052122	020040	020040	
1618	007650	020040	020040	057	
1619	007655	057	051500	052105	MSG6: .ASCII ;/QSETTLE DOWN DELAY /;
1620	007662	046124	020105	047504	
1621	007670	047127	042040	046105	
1622	007676	054501	020040	057	
1623	007703	057	053500	044522	MSG7: .ASCII ;/QWRITE TO ERASE HEAD/;
1624	007710	042524	052040	020117	
1625	007716	051105	051501	020105	
1626	007724	042510	042101	057	
1627	007731	057	053500	044522	MSG8: .ASCII ;/QWRITE NONSTOP GAP /;
1628	007736	042524	047040	047117	
1629	007744	052123	050117	043440	
1630	007752	050101	020040	057	
1631	007757	057	041100	041501	MSG9: .ASCII ;/QBACKSPACE SHUTDOWN /;
1632	007764	051513	040520	042503	
1633	007772	051440	052510	042124	
1634	010000	053517	020116	057	
1635	010005	057	051100	040505	MSG10: .ASCII ;/QREAD SHUTDOWN /;
1636	010012	020104	044123	052125	
1637	010020	047504	047127	020040	
1638	010026	020040	020040	057	
1639	010033	057	040100	040507	MSG11: .ASCII ;/QGAPS SHOULD = 8>7>6>5>4>3, 3=2=1 (+OR- 5.0)/;
1640	010040	051520	051440	047510	
1641	010046	046125	020104	020075	
1642	010054	037070	037067	037066	
1643	010062	037065	037064	026063	
1644	010070	031440	031075	030475	
1645	010076	024040	047453	026522	
1646	010104	032440	030056	027451	
1647	010112	040057	040507	020120	MSG11A: .ASCII ;/QGAP 1 /;
1648	010120	020061	020040	020040	
1649	010126	020040	020040	020040	
1650	010134	020040	027440		
1651	010140	040057	051127	052111	MSG12: .ASCII ;/QWRITE XIRG /;
1652	010146	020105	044530	043522	
1653	010154	020040	020040	020040	
1654	010162	020040	027440		

1655	010166	040057	042522	042101	MSG13: .ASCII ;/READ FROM BOT /;
1656	010174	043040	047522	020115	
1657	010202	047502	020124	020040	
1658	010210	020040	027440		
1659	010214	040057	040514	052123	MSG14: .ASCII ;/LAST CHAR TO CU RDY/;
1660	010222	041440	040510	020122	
1661	010230	047524	041440	020125	
1662	010236	042122	027531		
1663	010242	040057	051127	052111	MSG15: .ASCII ;/WRITE EOF /;
1664	010250	020105	047505	020106	
1665	010256	020040	020040	020040	
1666	010264	020040	027440		
1667	010270	040057	047505	020122	MSG16: .ASCII ;/EOR TO EOF SPACE /;
1668	010276	047524	042440	043117	
1669	010304	051440	040520	042503	
1670	010312	020040	027440		
1671	010316	040057	050123	041501	MSG19: .ASCII ;/SPACE SHUTDOWN /;
1672	010324	020105	044123	052125	
1673	010332	047504	047127	020040	
1674	010340	020040	027440		
1675	010344	040057	047117	020105	MSG20: .ASCII ;/ONE INCH DATA TIME /;
1676	010352	047111	044103	042040	
1677	010360	052101	020101	044524	
1678	010366	042515	027440		
1679	010372	040057	040524	042520	MSG21: .ASCII ;/TAPE SPEED FWD /;
1680	010400	051440	042520	042105	
1681	010406	043040	042127	020040	
1682	010414	020040	027440		
1683	010420	040057	025052	025052	MSG27: .ASCII ;/*****END OF TIMING*****/;
1684	010426	025052	025052	025052	
1685	010434	042452	042116	047440	
1686	010442	020106	044524	044515	
1687	010450	043516	025052	025052	
1688	010456	025052	025052	025052	
1689	010464	040052	057		
1690	010467	057	040100	051524	MSG28: .ASCII ;/TS03 DRIVE FUNCTION TIMER (DZTSE-A)/;
1691	010474	031460	042040	044522	
1692	010502	042526	043040	047125	
1693	010510	052103	047511	020116	
1694	010516	044524	042515	020122	
1695	010524	042050	052132	042523	
1696	010532	040455	027451		
1697	010536	020057	020040	040522	MSG29: .ASCII ;/ RANGE MAX - MIN /;
1698	010544	043516	020105	040515	
1699	010552	020130	020055	044515	
1700	010560	040116	027440		
1701	010564	040057	047516	052440	MSG30: .ASCII ;/NO UNITS AVAILABLE FOR TEST PRESS CONTINUE WHEN READY/;
1702	010572	044516	051524	040440	
1703	010600	040526	046111	041101	
1704	010606	042514	043040	051117	
1705	010614	052040	051505	020124	
1706	010622	050100	042522	051523	
1707	010630	041440	047117	044524	
1708	010636	052516	020105	044127	
1709	010644	047105	051040	040505	
1710	010652	054504	027500		

1711	010656	040057	042522	044507	MSG31: .ASCII ;/ REGISTER START: /;
1712	010664	052123	051105	051440	
1713	010672	040524	052122	020072	
1714	010700	057			
1715	010701	057	051500	042520	MSG32: .ASCII ;/ SPEED TEST (1=YES-0=NO): /;
1716	010706	042105	052040	051505	
1717	010714	020124	030450	054475	
1718	010722	051505	030055	047075	
1719	010730	024517	020072	057	
1720	010735	057	046500	052517	MSG33: .ASCII ;/ AMOUNT SKEW TAPE ON ALL AVAILABLE UNITS: TYPE CR WHEN READY/;
1721	010742	052116	051440	042513	
1722	010750	020127	040524	042520	
1723	010756	047440	020116	046101	
1724	010764	020114	053101	044501	
1725	010772	040514	046102	020105	
1726	011000	047125	052111	035123	
1727	011006	052040	050131	020105	
1728	011014	051103	053440	042510	
1729	011022	020116	042522	042101	
1730	011030	027531			
1731	011032	020057	020077	057	MSG34: .ASCII ;/ ? /;
1732					
1733	011037	057	033040	030462	RMSG1: .ASCII ;/ 621.0 - 509.0/;
1734	011044	030056	026440	032440	
1735	011052	034460	030056	057	
1736	011057	057	020040	033461	RMSG2: .ASCII ;/ 17.6 - 13.6/;
1737	011064	033056	026440	030440	
1738	011072	027063	027466		
1739	011076	020057	031440	027071	RMSG3: .ASCII ;/ 39.0 - 32.0/;
1740	011104	020060	020055	031063	
1741	011112	030056	057		
1742	011115	057	020040	033063	RMSG4: .ASCII ;/ 36.0 - 30.0/;
1743	011122	030056	026440	031440	
1744	011130	027060	027460		
1745	011134	020057	033440	027066	RMSG5: .ASCII ;/ 76.0 - 58.0/;
1746	011142	020060	020055	034065	
1747	011150	030056	057		
1748	011153	057	020040	033440	RMSG6: .ASCII ;/ 7.7 - 6.3/;
1749	011160	033456	026440	033040	
1750	011166	031456	057		
1751	011171	057	020040	033440	RMSG7: .ASCII ;/ 7.7 - 6.3/;
1752	011176	033456	026440	033040	
1753	011204	031456	057		
1754	011207	057	020040	034463	RMSG8: .ASCII ;/ 39.0 - 32.0/;
1755	011214	030056	026440	031440	
1756	011222	027062	027460		
1757	011226	020057	033063	027066	RMSG9: .ASCII ;/ 366.0 - 300.0/;
1758	011234	020060	020055	030063	
1759	011242	027060	027460		
1760	011246	020057	032462	027070	RMSG10: .ASCII ;/ 258.0 - 212.0/;
1761	011254	020060	020055	030462	
1762	011262	027062	027460		
1763	011266	020057	031064	027060	RMSG11: .ASCII ;/ 420.0 - 340.0/;
1764	011274	020060	020055	032063	
1765	011302	027060	027460		
1766	011306	020057	030064	027060	RMSG12: .ASCII ;/ 400.0 - 320.0/;

1767	011314	020060	020055	031063	.
1768	011322	027060	027460		
1769	011326	020057	020040	027067	RMSG13: .ASCII ;/ 7.7 - 6.3/;
1770	011334	020067	020055	027066	
1771	011342	027463			
1772	011344	020057	034040	027071	RMSG14: .ASCII ;/ 89.0 - 73.0/;
1773	011352	020060	020055	031467	
1774	011360	030056	057		
1775	011363	057	020040	034470	RMSG15: .ASCII ;/ 89.0 - 73.0/;
1776	011370	030056	026440	033440	
1777	011376	027063	027460		
1778					
1779					
1780	011402	000000			WBUF: 0 .EVEN
1791		000001			.END

T1C	001630	740	743#		
T1D	001646	747#	750		
T1E	001664	748	751#		
T1G	005220	1239#			
T1GA	005234	1242#	1291		
T1GB	005312	1251#	1252		
T1GC	005322	1253#	1258		
T1GCC	005346	1256	1259#		
T1GD	005362	1254	1260	1262#	
T1GE	005400	1267#	1270		
T1GF	005416	1268	1271#		
T1GG	005502	1283#	1286		
T1GH	005516	1284	1288#		
T1I	005664	730	1316#		
T1IA	005672	1318#	1334		
T1IB	005744	1327#	1330		
T1IC	005760	1328	1331#		
T1IT	001046	646#			
T1J	006032	1310	1341#		
T2	001766	768#			
T2A	002002	771#	791		
T2B	002026	775#	778		
T2C	002044	776	779#		
T2D	002072	785#	788		
T2E	002106	786	789#		
T3	002210	810#	812		
T3A	002222	813#	815		
T3B	002234	816#	823		
T3C	002304	828#	841		
T3E	002340	833#	836		
T3F	002354	834	837#		
T3G	002430	848#	850		
T3H	002442	851#	853		
T4	002454	860#			
T4A	002536	875#	878		
T4AA	002474	864#	922		
T4B	002554	876	879#		
T4BA	002652	897#	898		
T4C	002666	901#	904		
T4D	002704	902	905#		
T4DA	002742	912#	913		
T4E	002752	914#	917		
T4F	002770	915	918#		
T4FA	003110	937#	939		
T4FB	003122	940#	942		
T4FC	003136	944#	959		
T4G	003232	965#	1023		
T5	003604	1031#	1057		
T5A	003646	1038#	1051		
T5B	003670	1043#	1046		
T5C	003706	1044	1047#		
T5DRV	001044	645#	1029#	1034	1055#
T6	004216	1096#			
T6A	004226	1098#	1135		
T6B	004340	1115#	1118		
T6C	004356	1116	1119#		

ADD	1034	1049	1055	1369	1466	1479									
BCC	884	906	919	1020	1053	1103	1111	1120	1131	1169	1192	1214	1410	1420	
BCS	786														
BEQ	677	689	693	729	784	834	898	913	1284	1328	1372	1386	1524	1582	
BGT	1292	1326													
BHI	1530														
BIC	718	894	996	1001	1215	1249	1384	1539							
BICB	1575														
BIS	690	694	719	737	773	819	831	867	890	895	911	946	967	976	991
	997	1002	1036	1104	1107	1114	1125	1158	1184	1210	1216	1250	1280	1324	1387
	1411	1497	1540												
BIT	689	692	747	783	897	901	912	914	1105	1253	1259	1267	1350	1371	1385
	1421														
BLE	740	776	876	1044	1116	1127	1160								
BLOS	1527	1544	1547												
BMI	1188	1218	1256	1432											
BNE	667	696	746	748	782	902	915	1006	1008	1012	1016	1051	1106	1186	1252
	1254	1260	1268	1351	1367	1401	1422	1457	1469	1474	1483	1522	1542	1579	
BPL	721	821	839	869	882	892	948	953	957	969	973	979	983	994	999
	1004	1039	1101	1165	1212	1247	1273	1279	1321	1363	1408	1413	1430	1455	1476
	1489	1566	1569	1584	1592	1595									
BR	700	705	708	724	742	750	753	778	788	791	812	815	823	836	841
	850	853	878	904	917	922	939	942	959	1010	1014	1018	1023	1046	1057
	1118	1129	1135	1150	1162	1167	1190	1196	1200	1220	1224	1233	1258	1270	1286
	1291	1330	1334	1355	1472	1480	1485	1587	1597						
CLC	1353	1365	1533	1535	1537										
CLR	670	685	738	744	774	780	832	864	890	909	1029	1041	1112	1122	1155
	1182	1206	1242	1261	1263	1276	1319	1348	1518	1519	1561	1562	1563		
CMP	739	775	875	1043	1115	1126	1159	1251	1281	1325	1400	1456	1473	1543	1546
CMPB	1521	1526	1529	1578	1581										
DEC	666	1541													
HALT	624	699	1343	1423											
INC	874	951	955	971	992	1005	1007	1011	1015	1042	1050	1352	1364	1433	1463
	1468	1532	1564	1577	1596										
INCB	1064	1068	1072	1076	1080	1084	1088								
JMP	625	730	1310	1344	1525	1528	1531	1545	1548						
JSR	653	655	660	669	675	679	684	698	702	703	704	706	707	713	715
	723	725	727	735	741	749	752	758	760	763	765	768	771	777	787
	790	796	798	801	803	810	811	813	814	822	835	840	845	847	848
	849	851	852	860	865	873	877	888	903	908	916	921	928	930	933
	935	936	937	938	940	941	944	950	954	958	965	970	979	991	1000
	1021	1022	1031	1032	1040	1045	1054	1056	1060	1063	1067	1071	1075	1079	1093
	1087	1091	1092	1098	1109	1117	1123	1128	1133	1134	1140	1142	1145	1147	1149
	1149	1156	1161	1166	1173	1175	1176	1181	1189	1194	1195	1198	1199	1219	1222
	1223	1228	1230	1231	1232	1257	1269	1285	1289	1290	1297	1299	1302	1304	1307
	1309	1329	1332	1333	1338	1340	1342	1356	1368	1373	1454	1467	1471	1520	1555
MOV	650	651	652	654	656	657	658	659	661	662	663	664	668	671	672
	673	674	678	680	681	682	683	686	687	691	697	712	714	716	722
	726	733	734	736	743	751	754	755	756	757	759	761	762	764	769
	770	772	779	789	792	793	794	795	797	799	800	802	816	817	818
	827	828	829	830	837	842	843	844	846	861	862	863	866	879	899
	893	907	910	920	923	924	925	926	927	929	931	932	934	945	966
	974	975	980	984	985	986	987	995	1009	1013	1017	1033	1035	1037	1047
	1048	1059	1061	1062	1065	1066	1069	1070	1073	1074	1077	1079	1081	1082	1085
	1086	1089	1090	1096	1097	1099	1108	1113	1121	1124	1132	1136	1137	1138	1139
	1141	1143	1144	1146	1154	1157	1163	1170	1171	1172	1174	1180	1183	1193	1197

K04

	1205	1207	1208	1209	1221	1225	1226	1227	1229	1239	1240	1241	1243	1244	1245
	1248	1262	1271	1274	1275	1277	1288	1292	1293	1294	1295	1296	1298	1300	1301
	1303	1305	1306	1308	1317	1319	1322	1323	1331	1335	1336	1337	1339	1341	1349
	1382	1392	1393	1398	1399	1406	1418	1455	1459	1460	1461	1462	1470	1477	1478
	1484	1486	1490	1549	1554	1567									
MOV8	1058	1570	1576	1585	1593	1596									
NOP	701	896	899	900	943	949	960	1264	1265	1266	1316				
ROL	1534	1536	1538												
ROR	795	883	905	918	1019	1052	1102	1110	1119	1130	1168	1191	1213	1354	1366
	1409	1419													
RTS	1357	1370	1374	1388	1394	1402	1414	1424	1434	1458	1491	1550	1557	1571	1580
SUB	1464	1556													
SWAB	717	1383													
TST	665	676	695	728	745	781	833	1185	1283	1327	1429	1431	1482	1523	
TSTB	720	820	838	868	881	891	947	952	956	968	972	977	982	993	998
	1003	1038	1100	1164	1187	1211	1217	1246	1255	1272	1278	1320	1362	1407	1412
	1475	1488	1565	1568	1583	1591	1594								
.ASCII	1600	1604	1606	1607	1611	1615	1619	1623	1627	1631	1635	1639	1647	1651	1655
	1659	1663	1667	1671	1675	1679	1683	1690	1697	1701	1711	1715	1720	1731	1733
	1736	1739	1742	1745	1748	1751	1754	1757	1760	1763	1766	1769	1772	1775	
.ENABL	621														
.END	1781														
.EVEN	1779														
.LIST	624														
.NLIST	524														
.REPT	2	624													
.TITLE	614														

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

*DZTSEA.DZTSEA.SEG/SOL/CRF/DS:ERFZ/EN:ABS=DSKM:DZTSEA.P11
RUN-TIME: 5 11 2 SECONDS
RUN-TIME RATIO: 42/21=2.0
CORE USED: 12K (23 PAGES)

