

# TS03

## DRIVE FUNCTION TIMER CZTSEB0

AH-9436B-MC

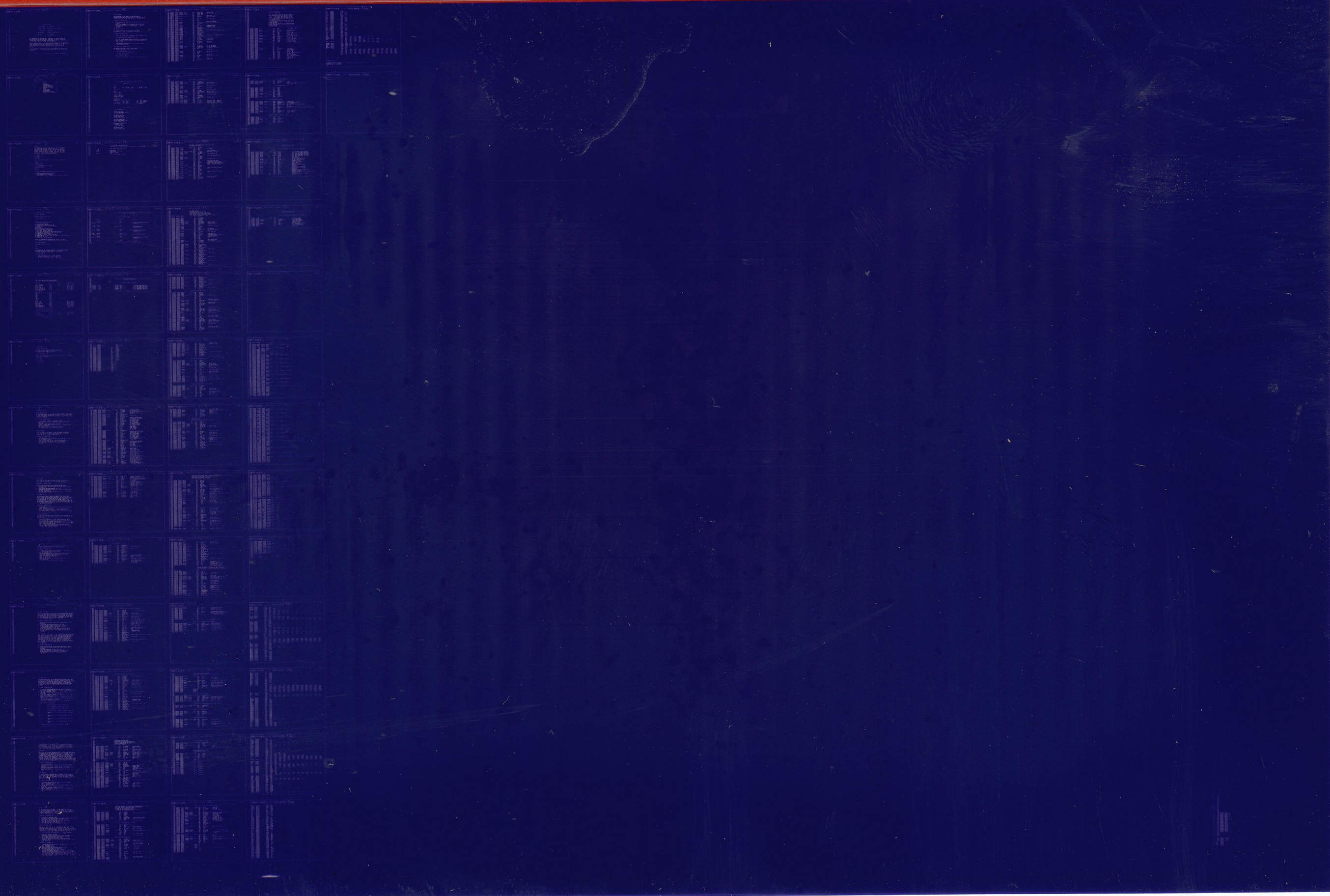
JUL 1978

COPYRIGHT © 75-78

**digital**

FICHE 1 OF 1

MADE IN USA



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42

.REM %

IDENTIFICATION

PRODUCT CODE: AC-9453B-MC  
PRODUCT NAME: CZTSEBO TS03 DR FCTN TMR  
DATE CREATED: JUNE 1978  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: R. B. BARNES

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) 1975,1978, BY DIGITAL EQUIPMENT CORPORATION

43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	3
3.	LOADING PROCEDURE	3
4.	STARTING PROCEDURE	4
5.	OPERATING PROCEDURE	4
6.	ERRORS	4
7.	RESTRICTIONS	6
8.	MISCELLANEOUS	6
9.	PROGRAM DESCRIPTION	7
10.	REGISTER BIT ASSIGNMENTS	15

57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87

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).

88 4. PRESS "START" (PROGRAM WILL LOAD).  
89  
90 4. STARTING PROCEDURE  
91  
92 4.1 CONTROL SWITCH SETTINGS: NONE  
93  
94 4.2 STARTING ADDRESS  
95  
96 200  
97  
98 4.3 PROGRAM AND/OR OPERATOR ACTION  
99  
100 LOAD PROGRAM INTO MEMORY.  
101 SET DESIRED TS03 TAPE UNITS ON-LINE.  
102 SET SWITCH REGISTER TO STARTING ADDRESS.  
103 LOAD ADDRESS.  
104 PRESS START.  
105 ENTER STARTING REGISTER ADDRESS.  
106 IF SPEED TEST ONLY, ENTER A ONE(1).  
107 IF ALL OTHERS, ENTER A ZERO(0).  
108 IF SPEED TEST, MOUNT 800 BPI SKEW TAPE AND TYPE CR.  
109 THE PROGRAM WILL AUTOMATICLY FIND THE AVAILABLE  
110 TS03 TAPE UNITS TO BE TESTED.  
111 THE PROGRAM WILL BEGIN TIMING FUNCTIONS.  
112 ON COMPLETION OF ALL TESTS "END OF TIMING" WILL BE PRINTED AND  
113 THE PROCESSOR WILL HALT.  
114 TO REPEAT TEST: PRESS CONTINUE.  
115  
116 \*NOTE\* THE PROGRAM HAS BEEN MODIFIED TO TEST 1-8 TS03  
117 UNITS UNDER ACT11M. IN THIS MODE SPEED TEST IS NOT SELECTED.  
118  
119 5. OPERATING PROCEDURE  
120  
121 5.1 OPERATIONAL SWITCH SETTINGS  
122  
123 NONE  
124  
125 6. ERRORS  
126  
127 THE PROGRAM HAS NO INTERNAL ERROR DETECTION FACILITIES AND,  
128 THEREFORE, NO ACTUAL ERROR TYPEOUTS. THE VALIDITY OF THE  
129 TIMES MEASURED MUST BE DETERMINED BY THE OPERATOR.  
130  
131 6.1 TIME RELATIONSHIPS  
132  
133 A. "READ SHUTDOWN" MUST BE < "WRITE SHUTDOWN".  
134 B. GAPS MUST = 8>7>6>5>4>3, 3=2=1 (+OR- 5.0).  
135 C. "WRITE EOF" SHOULD BE SLIGHTLY > "WRITE XIRG".

136  
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

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 - 13.6
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	50.2	"	
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	"	366.0 - 300.0
READ FROM BOT	235.0	"	258.0 - 212.0
WRITE EOF	385.0	"	420.0 - 340.0
EOR 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 \*\*\*\*\*

178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198

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

199  
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

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".



234  
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

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.

286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305

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.

306  
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

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"

353  
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

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:

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

407  
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

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".

460  
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

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".

513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559

9.14 SPACE SHUTDOWN

SPACE SHUTDOWN IS THE AMOUNT OF TIME NECESSARY TO CONTINUE MOVING TAPE AFTER A RECORD IS SPACED OVER IN THE FORWARD DIRECTION FOR THE SAME REASONS AS "READ SHUTDOWN" .

PROCEDURE TO MEASURE TIME:

- A. SPACE FORWARD FUNCTION USED TO TIME "EOR TO EOF SPACE TIME" IS USED.
- B. AFTER "EOF" BECOMES A 1, INDICATING THE END OF THE RECORD (EOF), MONITOR "SETTLEDOWN" UNTIL IT BECOMES A 1.
- C. THE TIME FROM "EOF" UNTIL "SETTLEDOWN" IS "SPACE SHUTDOWN".

9.15 ONE INCH DATA TIME

ONE INCH OF DATA, 800 BPI IS WRITTEN AND TIMED TO DETERMINE IF DATA IS TRANSFERRED AT PROPER RATE.

PROCEDURE TO MEASURE TIME:

- A. INITIALIZE BYTE RECORD COUNTER AND CURRENT MEMORY ADDRESS.
- B. ISSUE WRITE FUNCTION, 800 BPI, SET "GO".
- C. WAIT FOR CURRENT MEMORY ADDRESS REGISTER TO INDICATE 2ND BYTE IS OUTPUT AND THEN MONITOR BYTE RECORD COUNTER UNTIL EQUAL TO ZERO.
- D. TIME FROM 2ND BYTE OUTPUT UNTIL BYTE RECORD COUNTER = 0 IS "ONE INCH DATA TIME"

9.16 TAPE SPEED TEST (FORWARD ONLY)

ONE INCH OF TAPE (800 BYTES) IS READ FROM A SKEW TAPE IN THE FORWARD DIRECTION TO ASSURE TAPE SPEED.

- A. ASSURE OPERATOR MOUNTS SKEW TAPE
- B. ISSUE READ FUNCTION, 800 BPI, SET "GO".
- C. AWAIT 2ND BYTE TRANSFER.
- D. TIME FROM 2ND BYTE UNTIL LAST BYTE IS ONE INCH OF TAPE

560  
561  
562 10. STATUS AND COMMAND REGISTER BIT ASSIGNMENTS (TMA-11)  
563  
564 COMMAND REGISTER  
565  
566 15 ERROR  
567  
568 14 DEN 8 00 = 200 BPI 7 TRACK 10 = 800 BPI 7 TRACK  
569 13 DEN 5 01 = 556 BPI 7 TRACK 11 = 800 BPI 9 TRACK  
570 12 POWER CLEAR  
571  
572 11 PARITY 0 = ODD 1 = EVEN  
573 10 UNIT SEL. BIT 2  
574 9 UNIT SEL. BIT 1  
575  
576 8 UNIT SEL. BIT 0  
577 7 CONTROL UNIT READY  
578 6 INTERRUPT ENABLE  
579  
580 5 ADDRESS BIT 17  
581 4 ADDRESS BIT 16  
582 3 FUNCTION BIT 2 000 = OFF LINE 100 = SPACE FORWARD  
583 001 = READ 101 = SPACE REVERSE  
584 2 FUNCTION BIT 1 010 = WRITE 110 = WRITE XIRG  
585 1 FUNCTION BIT 0 011 = WRITE EOF 111 = REWIND  
586 0 GO  
587  
588 STATUS REGISTER  
589  
590  
591 15 ILLEGAL COMMAND (ILC)  
592  
593 14 END OF FILE (EOF)  
594 13 CYCLICAL REDUNDANCY ERROR (CRE)  
595 12 PARITY ERROR (PAE)  
596  
597 11 BUS GRANT LATE (BGL)  
598 10 END OF TAPE (EOT)  
599 9 RECORD LENGTH ERROR (RLE)  
600  
601 8 BAD TAPE ERROR (BTE)  
602 7 NON EXISTENT MEMORY (NXM)  
603 6 SELECT REMOTE (SELR)  
604  
605 5 BEGINNING OF TAPE (BOT)  
606 4 7 CHANNEL (7CH)  
607 3 SETTLE DOWN (SDWN)  
608  
609 2 WRITE LOCK (WRL)  
610 1 REWIND STATUS (RWS)  
611 0 TAPE UNIT READY (TUR)  
612  
613 X  
614



615  
616  
617  
618  
619  
620  
621  
622  
623  
624  
625

001000  
005670  
000000

.TITLE CZTSEBO TS03 DR FCTN TMR  
:COPYRIGHT: (C) 1975,1978 DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754  
:  
:LOAD ADDRESS 200, PRESS START  
:  
:STACK=1000  
:BLENGTH=3000.  
:ENABL ABS,AMA  
:=0  
:TRAP CATCHER FROM 0 TO 1000

```
626  
627 : *****  
628 :                               MODIFIED JAN 23 1978  
629 :  
630 : ++  
631 :                               ACT11 AND XXDP HOOKS  
632 : --  
633  
634          001000          $SVPC=.          ;SAVE PC  
635  
636          000040          .=40  
637 000040          000          DRIVE: .BYTE 0          ;DRIVE # FOR XXDP LOAD MEDIUM  
638                                     ;ASSEMBLE AS A 0  
639  
640          000041          .=41  
641 000041          000          MEDIUM: .BYTE 0          ;XXDP LOAD MEDIUM  
642                                     ;ASSEMBLE AS A 0  
643  
644          000042          .=42  
645 000042          000000          .WCRD 0          ;AUTO/MAN MODE INDICATOR  
646                                     ;ASSEMBLE AS A 0  
647  
648          000046          .=46  
649 000046          006232          .WORD $ENDAD          ;SET TO $ENDAD IN .SEOP  
650  
651          000052          .=52  
652 000052          000000          .WORD 0          ;CHARACTERISTICS OF PROGRAM  
653                                     ;ASSEMBLE AS A 0  
654  
655          001000          .=$SVPC          ;RESTORE PC  
656  
657 : *****  
658
```

```
659
660      : *****
661      :                               MODIFIED JAN 23 1978
662      :
663      : ++
664      :                               ACT11 AND XXDP MODE INDICATORS
665      : --
666
667 001000 000000      AUTOM: .WORD 0      :AUTOMATIC MODE INDICATOR
668 001002      000      ACT11M: .BYTE 0      :ACT11 AUTO MODE INDICATOR
669 001003      000      XXDPM: .BYTE 0      :XXDP AUTO MODE INDICATOR
670 001004      000      ADUMPM: .BYTE 0      :ACT11 DUMP MODE INDICATOR
671 001005      000      XDUMPM: .BYTE 0      :XXDP DUMP MODE INDICATOR
672
673      : *****
674
```

675					
676		000200		. =200	
677	000200	000137	001062	JMP	START
678		001000		. =1000	
679	001000	172520		MTS:	172520
680	001002	172522		MTC:	172522
681	001004	172524		BC:	172524
682	001006	172526		CA:	172526
683	001010	172530		MTD:	172530
684	001012	172532		MTRD:	172532
685	001014	177570		SR:	177570
686	001016	177560		TKS:	177560
687	001020	177562		TKB:	177562
688	001022	177564		TPS:	177564
689	001024	177566		TPB:	177566
690	001026	000224		MTV:	224
691	001030	172520		REGS:	172520
692	001032	177776		CC:	177776
693	001034	000000		R10:	0
694	001036	000000		R11:	0
695	001040	000000		R12:	0
696	001042	000000		R13:	0
697	001044	000000		T5DRV:	0
698	001046	000000		T11T:	0
699	001050	000000		TIB:	0
700	001052	000000		TEMP1:	0
701	001054	000000		SPTF:	0
702	001056	000000		UDES:	0
703	001060	000000		TEMPO:	0

```

704 001062 012706 001000          START:  MOV    #STACK,%6      ;INITIALIZE STACK
705 001066 012777 000340 177736    MOV    #340,@ACC      ;SET PRIORITY LEVEL 7
706 001074 004737 007706          JSR    PC,CKMODE      ;CHECK FOR MODE OF OPERATION ++ C.W
707 001100 105737 001002          TSTB   ACT11M         ;ACT MODE? ++ C.W
708 001104 001101          BNE    STO            ;BRANCH - IF YES ++ C.W
709 001106 012737 011072 010070    MOV    #MSG28,MESAGE
710 001114 004737 007566          JSR    %7, TOP        ;PRINT TITLE
711 001120 012737 011246 010070    MOV    #MSG31,MESAGE
712 001126 004737 007566          JSR    %7, TOP        ;REQUEST UNIBUS ADDRESS
713 001132 012705 001030          MOV    #REGS,%5      ;GET ADDRESS OF RESPONSE
714 001136 012701 000006          MOV    #6,%1         ;SET SIZE OF ENTRY
715 001142 012702 172700          MOV    #172700,%2    ;SET UPPER LIMIT
716 001146 012703 172300          MOV    #172300,%3    ;SET LOWER LIMIT
717 001152 004737 007326          JSR    %7,TTR        ;GO GET RESPONSE
718 001156 013700 001030          MOV    REGS,%0       ;GET UNIBUS ADDRESS
719 001162 012701 001000          MOV    #MTS,%1      ;SET START OF TABLE
720 001166 012702 000006          MOV    #6,%2        ;SET SIZE OF TABLE
721 001172 010021          STRS:  MOV    %0,(1)+   ;LOAD TABLE
722 001174 005720          TST    (0)+         ;BUMP ADDRESS
723 001176 005302          DEC    %2           ;SEE IF DONE
724 001200 001374          BNE    STRS         ;IF NOT: BR
725 001202 012737 011271 010070    MOV    #MSG32,MESAGE
726 001210 004737 007566          JSR    %7, TOP        ;REQUEST SPEED TESTS ONLY
727 001214 005037 001054          CLR    SPTF         ;CLEAN SPEED TEST FLAG
728 001220 012705 001054          MOV    #SPTF,%5     ;GET FLAG ADDRESS
729 001224 012701 000001          MOV    #1,%1        ;SET SIZE OF ENTRY
730 001230 012702 000001          MOV    #1,%2        ;SET UPPER LIMIT
731 001234 012703 000000          MOV    #0,%3        ;SET LOWER LIMIT
732 001240 004737 007326          JSR    %7,TTR        ;GO GET RESPONSE
733 001244 005737 001054          TST    SPTF         ;SEE IF SHOULD DO SPEED TEST
734 001250 001417          BEQ    STO          ;IF NOT: BR
735 001252 012737 011325 010070    MOV    #MSG33,MESAGE
736 001260 004737 007566          JSR    %7, TOP        ;REQUEST SKEW TAPE MOUNT
737 001264 012705 001052          MOV    #TEMP1,%5    ;SET RESPONSE ADDRESS
738 001270 012701 000001          MOV    #1,%1        ;SET SIZE
739 001274 012702 000001          MOV    #1,%2        ;SET UPPER
740 001300 012703 000000          MOV    #0,%3        ;SET LOWER
741 001304 004737 007326          JSR    PC,TTR        ;AWAIT SKEW TAPE MOUNT
742 001310 005037 006372          STO:  CLR    DRIVES
743 001314 012777 010000 177460    MOV    #10000,@MTC  ;POWER CLEAR
744 001322 012777 000000 177452    MOV    #0,@MTC      ;SELECT UNIT 0
745 001330 032777 000100 177442    BIT    #100,@MTS    ;SEE IF 0 IS THERE
746 001336 001403          BEQ    STOA         ;IF NOT: BR
747 001340 052737 000200 006372    BIS    #200,DRIVES  ;SET 0 IN AVAILABLE TABLE
748 001346 012777 000400 177426    STOA: MOV    #400,@MTC ;SELECT UNIT 1
749 001354 032777 000100 177416    BIT    #100,@MTS    ;SEE IF 1 IS THERE
750 001362 001436          BEQ    STOB         ;IF NOT: BR
751 001364 052737 000100 006372    BIS    #100,DRIVES  ;SET 1 IN AVAILABLE TABLE
752 001372 012737 000002 001056    MOV    #2,UDES      ;GET UNIT 2 ++ C.W
753 001400 022737 000010 001056    $ACT: CMP    #10,UDES   ;ALL UNITS DONE ++ C.W
754 001406 001002          BNE    1$          ;BRANCH - IF NO ++ C.W
755 001410 000137 006200          JMP    T13         ;GO DO END OF PASS ++ C.W
756 001414 113737 001056 001060    1$:  MOVB   UDES,TEMPO  ;SELECT UNIT ++ C.W
757 001422 000337 001060          SWAB  TEMPO        ;POSITION UNIT ++ C.W
758 001426 042777 003400 177346    BIC    #3400,@MTC   ;CLEAR OLD NUMBER ++ C.W
759 001434 053777 001060 177340    BIS    TEMPO,@MTC   ;SET NEW UNIT NUMBER ++ C.W

```

```
760 001442 032777 000100 177330 BIT #100,@MTS ;IS UNIT AVAILABLE ++ C.W
761 001450 001403 BEQ STOB ;BRANCH - IF NO ++ C.W
762 001452 053737 001060 006372 BIS TEMPO,DRIVES ;SET AVAILABLE UNIT IN TABLE ++ C.W
763 001460 005737 006372 STOB: TST DRIVES ;SEE IF ANY UNITS AVAILABLE
764 001464 001021 BNE STOC ;IF SO: BR
765 001466 105737 001002 TSTB ACT11M ;ACT MODE? ++ C.W
766 001472 001407 BEQ 1$ ;BRANCH - IF NO ++ C.W
767 001474 022737 000010 001056 CMP #10,UDES ;ALL UNITS CHECKED? ++ C.W
768 001502 001403 BEQ 1$ ;BRANCH - IF YES ++ C.W
769 001504 105237 001056 INCB UDES ;GET NEXT UNIT ++ C.W
770 001510 000733 BR $ACT ;CONTINUE TEST ++ C.W
771 001512 012737 011154 010070 1$: MOV #MSG30,MESAGE
772 001520 004737 007566 JSR %7,TOP ;PRINT NO UNITS
773 001524 000000 HALT
774 001526 000670 BR STO ;RETRY UNIT SELECTION
775 001530 000240 STOC: NOP
776 001532 004737 006250 JSR %7,RSFDRV ;RESET DRIVES
777 001536 004737 006474 ST1: JSR %7,STRREW ;START REWIND
778 001542 004737 006314 JSR %7,CHGDRV ;DONE ALL DRIVES?
779 001546 000773 BR ST1 ;NO
780 001550 004737 006534 ST2: JSR %7,WATREW ;WAIT FOR BOT
781 001554 004737 006314 JSR %7,CHGDRV ;DONE ALL DRIVES?
782 001560 000773 BR ST2 ;NO
```

```
783
784 ;PRINT HEADER
785
786 001562 012737 010115 010070      MOV      #MSG2,MESAGE
787 001570 004737 007566              JSR      %7, TOP      ;PRINT "FUNCTION"
788 001574 012737 010142 010070  ST3:  MOV      #MSG2A,MESAGE
789 001602 004737 007566              JSR      %7, TOP      ;PRINT "UNIT"
790 001606 013737 006374 007310      MOV      FDRIVE,DIGIT
791 001614 000337 007310              SWAB     DIGIT
792 001620 042737 177770 007310      BIC      #177770,DIGIT
793 001626 052737 000060 007310      BIS      #60,DIGIT
794 001634 105777 177162              TSTB    @TPS
795 001640 100375                      BPL     .-4
796 001642 013777 007310 177154      MOV      DIGIT,@TPB   ;PRINT DRIVE "NUMBER"
797 001650 004737 006314              JSR      %7, CHGDRV   ;DONE ALL DRIVES?
798 001654 000747                      BR      ST3           ;NO
799 001656 004737 006454              JSR      %7, ST1S     ;STORE ONES IN WRITE BUFFER
800 001662 012737 011126 010070      MOV      #MSG29,MESAGE
801 001670 004737 007566              JSR      %7, TOP      ;PRINT RANGE HEADER
802 001674 005737 001054              TST     SPTF          ;SEE IF SPEED TESTS ONLY
803 001700 001402                      BEQ     T1            ;IF NOT: BR
804 001702 000137 006032              JMP     T11          ;ELSE DO SPEED TEST
```

```

805 ;TIME WRITE FROM BOT DELAY, AND WRITE SHUTDOWN
806
807 001706 012700 006610 T1: MOV #TM1,%0 ;INITIALIZE TIME BUFFERS
808 001712 012701 006634 MOV #TM2,%1
809 001716 004737 006436 T1A: JSR %7,WRINT
810 001722 013777 006374 177052 MOV FDRIVE,@MTC ;SELECT DRIVE
811 001730 052777 040005 177044 BIS #40005,@MTC ;800 BPI, WRITE, GO
812 001736 005037 006606 CLR TIME
813 001742 022777 011774 177036 T1B: CMP #WBUF+2,@CA ;IS 2ND WORD OUTPUT?
814 001750 003403 BLE T1C ;YES
815 001752 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
816 001756 000771 BR T1B
817 001760 013720 006606 T1C: MOV TIME,(0)+ ;SAVE "WRITE FROM BOT DELAY" TIME
818 001764 005037 006606 CLR TIME
819 001770 005777 177010 TST @BC ;SEE IF WORD COUNT DONE
820 001774 001375 BNE -4 ;IF NOT: BR
821 001776 032777 000010 176774 T1D: BIT #10,@MTC ;HAS SETTLEDOWN SET?
822 002004 001003 BNE T1E ;YES
823 002006 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
824 002012 000771 BR T1D
825 002014 013721 006606 T1E: MOV TIME,(1)+ ;SAVE "WRITE SHUTDOWN" TIME
826 002020 004737 006314 JSR %7,CHGDRV ;DONE ALL DRIVES
827 002024 000734 BR T1A ;NO
828 002026 012720 177777 MOV #-1,(0)+ ;TERMINATE TIMES
829 002032 012721 177777 MOV #-1,(1)+ ;TERMINATE TIMES
830 002036 012737 010156 010070 MOV #MSG3,MESAGE
831 002044 012700 006610 MOV #TM1,%0
832 002050 004737 007050 JSR %7,TYPTIM ;PRINT "WRITE FROM BOT DELAY" TIMES
833 002054 012737 011427 010070 MOV #RMSG1,MESAGE
834 002062 004737 007566 JSR %7, TOP ;PRINT RANGE
835 002066 012737 010204 010070 MOV #MSG4,MESAGE
836 002074 012700 006634 MOV #TM2,%0
837 002100 004737 007050 JSR %7,TYPTIM ;PRINT "WRITE SHUTDOWN" TIMES
838 002104 012737 011447 010070 MOV #RMSG2,MESAGE
839 002112 004737 007566 JSR %7, TOP ;PRINT RANGE
  
```



```

840 ;TIME WRITE START AND SETTLEDOWN DELAY
841
842 002116 004737 006250 T2: JSR %7,RSFDRV ;RESET DRIVE SELECTION
843 002122 012700 006610 MOV #TM1,%0
844 002126 012701 006634 MOV #TM2,%1
845 002132 004737 006436 T2A: JSR %7,WRINT
846 002136 013777 006374 176636 MOV FDRIVE,@MTC ;SELECT DRIVE
847 002144 052777 040005 176630 BIS #40005,@MTC ;800 BPI, WRITE, GO
848 002152 005037 006606 CLR TIME
849 002156 022777 011774 176622 T2B: CMP #WBUF+2,@CA ;IS 2ND WORD OUTPUT
850 002164 003403 BLE T2C ;YES
851 002166 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
852 002172 000771 BR T2B
853 002174 013720 006606 T2C: MOV TIME,(0)+ ;SAVE "WRITE START" TIME
854 002200 005037 006606 CLR TIME
855 002204 005777 176574 TST @BC
856 002210 001375 BNE .-4
857 002212 032777 000010 176560 BIT #10,@MTC
858 002220 001774 BEQ .-6 ;WAIT FOR SETTLEDOWN TO SET
859 002222 006077 176552 T2D: ROR @MTC
860 002226 103403 BCS T2E ;WAIT FOR TU READY
861 002230 004737 006564 JSR %7,TIMER
862 002234 000772 BR T2D
863 002236 013721 006606 T2E: MOV TIME,(1)+ ;SAVE "SETTLEDOWN" TIME
864 002242 004737 006314 JSR %7,CHGDRV
865 002246 000731 BR T2A
866 002250 012720 177777 MOV #-1,(0)+ ;TERMINATE TIMES
867 002254 012721 177777 MOV #-1,(1)+ ;TERMINATE TIMES
868 002260 012737 010232 010070 MOV #MSG5,MESSAGE
869 002266 012700 006610 MOV #TM1,%0
870 002272 004737 007050 JSR %7,TYPTIM ;PRINT "WRITE START" TIMES
871 002276 012737 011466 010070 MOV #RMSG3,MESSAGE
872 002304 004737 007566 JSR %7,TOP
873 002310 012737 010260 010070 MOV #MSG6,MESSAGE
874 002316 012700 006634 MOV #TM2,%0
875 002322 004737 007050 JSR %7,TYPTIM ;PRINT "SETTLEDOWN" TIMES
876 002326 012737 011505 010070 MOV #RMSG4,MESSAGE
877 002334 004737 007566 JSR %7,TOP
  
```

```
878 ;TIME WRITE TO ERASE HEAD
879 ;LONG RECORD WAS PREVIOUSLY WRITTEN
880 ;WRITE A 3 BYTE RECORD AND POWER CLEAR
881 ;DISTANCE FROM NEW DATA TO OLD IS
882 ;ERASE HEAD DISTANCE
883
884 002340 004737 006474 T3: JSR X7,STRREW ;START REWIND
885 002344 004737 006314 JSR X7,CHGDRV ;DONE ALL DRIVES?
886 002350 000773 BR T3 ;NO
887 002352 004737 006534 T3A: JSR X7,WATREW ;IS DRIVE AT BOT?
888 002356 004737 006314 JSR X7,CHGDRV ;DONE ALL DRIVES
889 002362 000773 BR T3A ;NO
890 002364 012777 177775 176412 T3B: MOV #-3,@BC ;3 BYTE RECORD
891 002372 012777 011772 176406 MOV #WBUF,@CA ;INITIALIZE CURRENT ADDRESS
892 002400 013777 006374 176374 MOV FDRIVE,@MTC ;SELECT DRIVE
893 002406 052777 040005 176366 BIS #40005,@MTC ;800BPI, WRITE, GO
894 002414 105777 176362 TSTB @MTC
895 002420 100375 BPL -4 ;AWAIT CUR
896 002422 004737 006314 JSR X7,CHGDRV ;DONE ALL DRIVES
897 002426 000756 BR T3B ;NO
898
899 ;NOW READ OVER PARTIAL RECORD
900
901 002430 012700 006610 MOV #TM1,%O
902 002434 012777 177776 176342 T3D: MOV #-2,@BC
903 002442 012777 011772 176336 MOV #WBUF,@CA
904 002450 013777 006374 176324 MOV FDRIVE,@MTC ;SELECT DRIVE
905 002456 052777 040003 176316 BIS #40003,@MTC ;800BPI, READ, GO
906 002464 005037 006606 CLR TIME ;CLEAR TIME
907 002470 005777 176310 T3E: TST @BC ;WAIT FOR NEXT WORD IN
908 002474 001403 BEQ T3F ;HAVE IT
909 002476 004737 006564 JSR X7,TIMER ;NO, COUNT TIME
910 002502 000772 BR T3E
911 002504 013720 006606 T3F: MOV TIME,(0)+ ;SAVE 'WRITE TO ERASE HEAD TIME'
912 002510 105777 176266 TSTB @MTC ;SEE IF CUR
913 002514 100375 BPL -4 ;AWAIT CUR
914 002516 004737 006314 JSR X7,CHGDRV ;DONE ALL DRIVES
915 002522 000744 BR T3D ;NO
916 002524 012720 177777 MOV #-1,(0)+ ;TERMINATE TIMES
917 002530 012737 010306 010070 MOV #MSG7,MESAGE
918 002536 012700 006610 MOV #TM1,%O
919 002542 004737 007050 JSR X7,TYPTIM ;PRINT 'WRITE TO ERASE HEAD TIMES'
920 002546 012737 011524 010070 MOV #RMSG5,MESAGE
921 002554 004737 007566 JSR X7,TOP
922 002560 004737 006474 T3G: JSR X7,STRREW ;START REWIND
923 002564 004737 006314 JSR X7,CHGDRV ;DONE ALL DRIVES
924 002570 000773 BR T3G ;NO
925 002572 004737 006534 T3H: JSR X7,WATREW ;DRIVE AT BOT
926 002576 004737 006314 JSR X7,CHGDRV ;DONE ALL DRIVES
927 002602 000773 BR T3H ;NO
```

```

928 ;TIME WRITE NONSTOP GAP, BACKSPACE SHUTDOWN AND READ SHUTDOWN
929 ;WRITE ONE RECORD, FOLLOW WITH ONE RECORD NONSTOP
930 ;FOLLOWED BY ONE RECORD START-STOP
931 ;FOLLOWED BY WRITE-BACKSPACE-READ-WRITE
932 ;FOLLOWED BY WRITE-BACKSPACE-WRITE
933
934 002604 004737 006454 T4: JSR X7,ST1S
935 002610 012700 006610 MOV #TM1,X0 ;INITIALIZE TIME BUFFERS
936 002614 012701 006634 MOV #TM2,X1
937 002620 012702 006660 MOV #TM3,X2
938 002624 005037 006606 T4AA: CLR TIME
939 002630 004737 006436 JSR X7,WRINT
940 002634 013777 006374 176140 MOV FDRIVE,@MTC ;TRACK AND DRIVE NUMBERS
941 002642 052777 040005 176132 BIS #40005,@MTC ;800 BPI, WRITE, GO
942 002650 105777 176126 TSTB @MTC
943 002654 100375 BPL .-4 ;WAIT FOR CU READY
944
945 ;HAVE FIRST RECORD WRITTEN, GO NONSTOP
946
947 002656 004737 006436 JSR X7,WRINT
948 002662 005277 176114 INC @MTC ;GO
949 002666 022777 011774 176112 T4A: CMP #WBUF+2,@CA ;IS 2ND WORD OUTPUT?
950 002674 003403 BLE T4B ;YES
951 002676 004737 006564 JSR X7,TIMER ;NO, COUNT TIME
952 002702 000771 BR T4A
953 002704 013720 006606 T4B: MOV TIME,(0)+ ;SAVE 'WRITE NONSTOP GAP' TIME
954 002710 005037 006606 CLR TIME
955 002714 105777 176062 TSTB @MTC
956 002720 100375 BPL .-4 ;WAIT FOR CU READY
957 002722 006077 176052 ROR @M5
958 002726 103375 BCC .-4 ;WAIT FOR TU READY
959
960 ;WRITE-BACKSPACE-READ-WRITE
961
962 002730 004737 006436 JSR X7,WRINT
963 002734 013777 006374 176040 MOV FDRIVE,@MTC ;DRIVE SELECT
964 002742 052777 040007 176032 BIS #40007,@MTC ;800 BPI, WRITE EOF, GO
965 002750 105777 176026 TSTB @MTC
966 002754 100375 BPL .-4 ;WAIT FOR CU READY
967 002756 012777 177777 176020 MOV #-1,@BC ;BACKSPACE 1 RECORD
968 002764 042777 000016 176010 BIC #16,@MTC
969 002772 052777 000013 176002 BIS #13,@MTC ;SPACE REVERSE, GO
970 003000 000240 NOP
971 003002 032777 040000 175770 T4BA: BIT #40000,@M5 ;SEE IF EOF
972 003010 001774 BEQ T4BA ;IF NOT: BR
973 003012 000240 NOP
974 003014 000240 NOP
975 003016 032777 000010 175754 T4C: BIT #10,@M5 ;HAS SETTLEDOWN SET?
976 003024 001003 BNE T4D ;YES
977 003026 004737 006564 JSR X7,TIMER ;NO, COUNT TIME
978 003032 000771 BR T4C
979 003034 006077 175740 T4D: ROR @M5
980 003040 103375 BCC .-4 ;WAIT FOR TU READY
981 003042 013721 006606 MOV TIME,(1)+ ;SAVE 'BACKSPACE SHUTDOWN' TIME
982 003046 004737 006436 JSR X7,WRINT
983 003052 005037 006606 CLR TIME

```



```
1036
1037
1038
1039 003362 004737 006436
1040 003366 013777 006374 175406
1041 003374 052777 040005 175400
1042 003402 105777 175374
1043 003406 100375
1044 003410 004737 006436
1045 003414 005277 175362
1046 003420 105777 175356
1047 003424 100375
1048 003426 012777 177777 175350
1049 003434 013777 006374 175340
1050 003442 052777 040013 175332
1051 003450 105777 175326
1052 003454 100375
1053 003456 004737 006436
1054 003462 013777 006374 175312
1055 003470 052777 040005 175304
1056 003476 105777 175300
1057 003502 100375
1058 003504 012737 177777 001036
1059 003512 012737 177777 001040
1060 003520 012737 177777 001042
1061 003526 012737 177776 001034

;WRITE RECORDS TO BE USED IN GAP TEST
T4G: JSR %7,WRINT
      MOV FDRIVE,@MTC ;SELECT DRIVE
      BIS #40005,@MTC ;800 BPI, WRITE, GO
      TSTB @MTC
      BPL .-4 ;WAIT FOR CU READY
      JSR %7,WRINT
      INC @MTC ;WRITE NEXT
      TSTB @MTC
      BPL .-4 ;WAIT FOR CU READY
      MOV #-1,@BC
      MOV FDRIVE,@MTC ;SELECT DRIVE
      BIS #40013,@MTC ;800 BPI, BACKSPACE, GO
      TSTB @MTC
      BPL .-4 ;WAIT FOR CU READY
      JSR %7,WRINT
      MOV FDRIVE,@MTC
      BIS #40005,@MTC ;800 BPI, WRITE, GO
      TSTB @MTC
      BPL .-4
      MOV #-1,R11 ;INDICATES BACK 3 COMPLETE
      MOV #-1,R12 ;INDICATES BACK 4 COMPLETE
      MOV #-1,R13 ;INDICATES BACK 5 COMPLETE
      MOV #-2,R10 ;FIRST SEQUENCE BACK 2 TIMES
```

```

1062          ;NOW WRITE, BACKSPACE, WRITE, BACKSPACE, WRITE
1063          ;GAP SHOULD GET LARGER
1064
1065 003534 004737 006436      MULWRT: JSR    %7,WRINT
1066 003540 005277 175236      INC    @MTC          ;GO NONSTOP
1067 003544 105777 175232      TSTB   @MTC
1068 003550 100375              BPL    .-4          ;WAIT FOR DONE
1069 003552 012777 177777 175224 MULBAK: MOV    #-1,@BC  ;BACKSPACE 1 RECORD
1070 003560 042777 000016 175214 BIC    #16,@MTC
1071 003566 052777 000013 175206 BIS    #13,@MTC      ;SET BACKSPACE, GO
1072 003574 105777 175202      TSTB   @MTC
1073 003600 100375              BPL    .-4          ;WAIT FOR BACKSPACE DONE
1074 003602 004737 006436      JSR    %7,WRINT
1075 003606 042777 000016 175166 BIC    #16,@MTC
1076 003614 052777 000005 175160 BIS    #5,@MTC       ;SET WRITE, GO
1077 003622 105777 175154      TSTB   @MTC
1078 003626 100375              BPL    .-4          ;WAIT FOR WRITE DONE
1079 003630 005237 001034      INC    R10          ;BACKSPACED ENOUGH TIMES?
1080 003634 001346              BNE    MULBAK       ;NO BACKSPACE AND WRITE AGAIN
1081 003636 005237 001036      INC    R11          ;DONE 3 BACKSPACE SEQUENCES?
1082 003642 001004              BNE    MUL1         ;YES
1083 003644 012737 177775 001034 MOV    #-3,R10
1084 003652 000730              BR     MULWRT
1085 003654 005237 001040      MUL1:  INC    R12          ;DONE 4 BACKSPACE SEQUENCES?
1086 003660 001004              BNE    MUL2         ;YES
1087 003662 012737 177774 001034 MOV    #-4,R10
1088 003670 000721              BR     MULWRT
1089 003672 005237 001042      MUL2:  INC    R13          ;DONE 5 BACKSPACE SEQUENCES?
1090 003676 001004              BNE    MUL3         ;YES
1091 003700 012737 177773 001034 MOV    #-5,R10
1092 003706 000712              BR     MULWRT
1093 003710 006077 175064      MUL3:  ROR    @MTC
1094 003714 103375              BCC    .-4          ;WAIT FOR TU READY
1095 003716 004737 006474      JSR    %7,STRREW    ;START REWIND
1096 003722 004737 006314      JSR    %7,CHGDRV
1097 003726 000615              BR     T4G
  
```

```

1098 ;NOW READ NONSTOP
1099 ;ACCUMULATE GAP TIMES ON READ
1100 ;TYPE ACCUMULATED TIMES AT END OF READ
1101 ;GAP1 SHOULD = GAP2, GAP3 < GAP1 AND GAP2
1102 ;GAP4 THRU GAP8 SHOULD GET INCREASINGLY LONGER
1103 003730 005037 001044 CLR T5DRV
1104
1105 003734 004737 006534 T5: JSR %7,WATREW
1106 003740 004737 006436 JSR %7,WRINT
1107 003744 012700 006610 MOV #TM1,%0
1108 003750 063700 001044 ADD T5DRV,%0
1109 003754 013777 006374 175020 MOV FDRIVE,@MTC ;SELECT DRIVE
1110 003762 052777 040003 175012 BIS #40003,@MTC ;800 BPI, READ, GO
1111 003770 012737 177770 001034 MOV #-8.,R10 ;COUNT 8 GAPS
1112 003776 105777 175000 T5A: TSTB @MTC
1113 004002 100375 BPL -4 ;WAIT FOR CU READY
1114 004004 004737 006436 JSR %7,WRINT
1115 004010 005037 006606 CLR TIME
1116 004014 005277 174762 INC @MTC ;GO NONSTOP
1117 004020 022777 011774 174760 T5B: CMP #WBUF+2,@CA ;IS 2ND WORD OUTPUT
1118 004026 003403 BLE T5C ;YES
1119 004030 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
1120 004034 000771 BR T5B
1121 004036 013720 006606 T5C: MOV TIME,(0)+ ;SAVE GAP TIME
1122 004042 012710 177777 MOV #-1,(0) ;TERMINATE, JUST IN CASE AT END
1123 004046 062700 000022 ADD #22,%0 ;STEP GAP POINTER
1124 004052 005237 001034 INC R10 ;DONE ALL 8 GAPS?
1125 004056 001347 BNE T5A ;NO
1126 004060 006077 174714 ROR @MTS
1127 004064 103375 BCC -4 ;WAIT FOR TU READY
1128 004066 004737 006474 JSR %7,STRREW ;START REWIND
1129 004072 062737 000002 001044 ADD #2,T5DRV ;+2 TO DRIVE TIME POINTER
1130 004100 004737 006314 JSR %7,CHGDRV
1131 004104 000713 BR T5
1132 004106 112737 000061 010523 MOVB #'1,MSG11A+6
1133 004114 012737 010436 010070 MOV #MSG11,MESAGE
1134 004122 004737 007566 JSR %7,TOP
1135 004126 012737 010515 010070 MOV #MSG11A,MESAGE
1136 004134 012700 006610 MOV #TM1,%0
1137 004140 004737 007050 JSR %7,TYPTIM ;PRINT "GAP 1"
1138 004144 105237 010523 INCB MSG11A+6
1139 004150 012737 010515 010070 MOV #MSG11A,MESAGE
1140 004156 012700 006634 MOV #TM2,%0
1141 004162 004737 007050 JSR %7,TYPTIM ;PRINT "GAP 2"
1142 004166 105237 010523 INCB MSG11A+6
1143 004172 012737 010515 010070 MOV #MSG11A,MESAGE
1144 004200 012700 006660 MOV #TM3,%0
1145 004204 004737 007050 JSR %7,TYPTIM ;PRINT "GAP 3"
1146 004210 105237 010523 INCB MSG11A+6
1147 004214 012737 010515 010070 MOV #MSG11A,MESAGE
1148 004222 012700 006704 MOV #TM4,%0
1149 004226 004737 007050 JSR %7,TYPTIM ;PRINT "GAP 4"
1150 004232 105237 010523 INCB MSG11A+6
1151 004236 012737 010515 010070 MOV #MSG11A,MESAGE
1152 004244 012700 006730 MOV #TM5,%0
1153 004250 004737 007050 JSR %7,TYPTIM ;PRINT "GAP 5"

```

```

1154 004254 105237 010523          INCB  MSG11A+6
1155 004260 012737 010515 010070  MOV   #MSG11A,MESAGE
1156 004266 012700 006754          MOV   #TM6,X0
1157 004272 004737 007050          JSR   Z7,TYPTIM      ;PRINT "GAP 6"
1158 004276 105237 010523          INCB  MSG11A+6
1159 004302 012737 010515 010070  MOV   #MSG11A,MESAGE
1160 004310 012700 007000          MOV   #TM7,X0
1161 004314 004737 007050          JSR   Z7,TYPTIM      ;PRINT "GAP 7"
1162 004320 105237 010523          INCB  MSG11A+6
1163 004324 012737 010515 010070  MOV   #MSG11A,MESAGE
1164 004332 012700 007024          MOV   #TM8,X0
1165 004336 004737 007050          JSR   Z7,TYPTIM      ;PRINT "GAP 8"
1166 004342 004737 006454          JSR   Z7,ST1S
1167
1168                               ;TIME WRITE START NOT AT BOT
1169
1170 004346 012700 006610          T6:   MOV   #TM1,X0
1171 004352 012701 006634          MOV   #TM2,X1
1172 004356 004737 006436          T6A:  JSR   Z7,WRINT
1173 004362 013777 006374 174412  MOV   FDRIVE,@MTC      ;SELECT DRIVE
1174 004370 105777 174406          TSTB  @MTC
1175 004374 100375          BPL   .-4
1176 004376 006077 174376          ROR   @MTC
1177 004402 103375          BCC   .-4              ;WAIT FOR TU READY
1178 004404 052777 040005 174370  BIS   #40005,@MTC      ;800 BPI, WRITE, GO
1179 004412 032777 000040 174360  BIT   #40,@MTC
1180 004420 001374          BNE   .-6              ;WAIT FOR BOT TO CLEAR
1181 004422 052777 010000 174352  BIS   #10000,@MTC      ;POWER CLEAR
1182 004430 013777 006374 174344  MOV   FDRIVE,@MTC
1183 004436 004737 006436          JSR   Z7,WRINT
1184 004442 006077 174332          ROR   @MTC
1185 004446 103375          BCC   .-4              ;WAIT FOR TU READY
1186 004450 005037 006606          CLR   TIME
1187 004454 013777 006374 174320  MOV   FDRIVE,@MTC      ;SELECT DRIVE
1188 004462 052777 040005 174312  BIS   #40005,@MTC      ;800 BPI, WRITE, GO
1189 004470 022777 011774 174310  T6B:  CMP   #WBUF+2,@CA      ;IS 2ND WORD OUTPUT?
1190 004476 003403          BLE   T6C              ;YES
1191 004500 004737 006564          JSR   Z7,TIMER        ;NO, COUNT TIME
1192 004504 000771          BR    T6B
1193 004506 006077 174266          T6C:  ROR   @MTC
1194 004512 103375          BCC   .-4              ;WAIT FOR TU READY
1195 004514 013720 006606          MOV   TIME,(0)+       ;SAVE "WRITE START" TIME
1196 004520 005037 006606          CLR   TIME
1197 004524 004737 006436          JSR   Z7,WRINT
1198 004530 013777 006374 174244  MOV   FDRIVE,@MTC      ;SELECT DRIVE
1199 004536 052777 040015 174236  BIS   #40015,@MTC      ;800 BPI, WRITE XIRG, GO
1200 004544 022777 011774 174234  T6D:  CMP   #WBUF+2,@CA      ;IS 2ND WORD OUTPUT
1201 004552 003403          BLE   T6E              ;YES
1202 004554 004737 006564          JSR   Z7,TIMER        ;NO COUNT TIME
1203 004560 000771          BR    T6D
1204 004562 006077 174212          T6E:  ROR   @MTC
1205 004566 103375          BCC   .-4              ;WAIT FOR TU READY
1206 004570 013721 006606          MOV   TIME,(1)+       ;SAVE "WRITE XIRG" TIME
1207 004574 004737 006474          JSR   Z7,STRREW
1208 004600 004737 006314          JSR   Z7,CHGDRV
1209 004604 000664          BR    T6A

```



```

1210 004606 012720 177777      MOV      #-1,(0)+      ;TERMINATE TIMES
1211 004612 012721 177777      MOV      #-1,(1)+      ;TERMINATE TIMES
1212 004616 012737 010232 010070  MOV      #MSG5,MESAGE
1213 004624 012700 006610      MOV      #TM1,%0
1214 004630 004737 007050      JSR      %7,TYPTIM      ;TYPE 'WRITE START' TIME
1215 004634 012737 011577 010070  MOV      #RMSG8,MESAGE
1216 004642 004737 007566      JSR      %7,TOP
1217 004646 012737 010543 010070  MOV      #MSG12,MESAGE
1218 004654 012700 006634      MOV      #TM2,%0
1219 004660 004737 007050      JSR      %7,TYPTIM      ;TYPE 'WRITE XIRG' TIME
1220 004664 012737 011616 010070  MOV      #RMSG9,MESAGE
1221 004672 004737 007566      JSR      %7,TOP
1222 004676 004737 006534      T6F:    JSR      %7,WATREW
1223 004702 004737 006314      JSR      %7,CHGDRV
1224 004706 000773      BR       T6F           ;WAIT FOR ALL DRIVES AT BOT.
1225
1226      ;NOW TIME 'READ FROM BOT DELAY
1227
1228 004710 012700 006610      T7:    MOV      #TM1,%0
1229 004714 005037 006606      T7A:    CLR      TIME
1230 004720 004737 006436      JSR      %7,WRINT
1231 004724 013777 006374 174050  MOV      FDRIVE,@MTC      ;SELECT DRIVE
1232 004732 052777 040003 174042  BIS      #40003,@MTC      ;800 BPI, READ GO
1233 004740 022777 011774 174040  T7B:    CMP      #WBUF+2,@CA      ;IS 2ND WORD INPUT?
1234 004746 003403      BLE      T7C           ;YES
1235 004750 004737 006564      JSR      %7,TIMER      ;NO COUNT TIME
1236 004754 000771      BR       T7B
1237 004756 013720 006606      T7C:    MOV      TIME,(0)+      ;SAVE 'READ FROM BOT' TIME
1238 004762 105777 174014      TSTB    @MTC
1239 004766 100375      BPL      #-4           ;WAIT FOR CU READY.
1240 004770 004737 006314      JSR      %7,CHGDRV      ;DONE ALL DRIVES?
1241 004774 000747      BR       T7A           ;NO
1242 004776 006077 173776      ROR      @MTC
1243 005002 103375      BCC      #-4
1244 005004 012720 177777      MOV      #-1,(0)+      ;TERMINATE TIMES
1245 005010 012737 010571 010070  MOV      #MSG13,MESAGE
1246 005016 012700 006610      MOV      #TM1,%0
1247 005022 004737 007050      JSR      %7,TYPTIM      ;PRINT 'READ FROM BOT' TIME
1248 005026 012737 011636 010070  MOV      #RMSG10,MESAGE
1249 005034 004737 007566      JSR      %7,TOP
1250 005040 004737 006454      JSR      %7,ST1S
1251
1252      ;TIME 'LAST CHARACTER INPUT TO CU READY'
1253
1254 005044 012700 006610      T8:    MOV      #TM1,%0
1255 005050 004737 006436      T8A:    JSR      %7,WRINT
1256 005054 005037 006606      CLR      TIME
1257 005060 013777 006374 173714  MOV      FDRIVE,@MTC      ;SELECT DRIVE
1258 005066 052777 040003 173706  BIS      #40003,@MTC      ;800 BPI, READ, GO
1259 005074 005777 173704      TST      @BC
1260 005100 001375      BNE      #-4           ;WAIT FOR LAST WORD IN
1261 005102 105777 173674      T8B:    TSTB    @MTC      ;IS CU READY?
1262 005106 100403      BMI      T8C           ;YES
1263 005110 004737 006564      JSR      %7,TIMER      ;NO, COUNT TIME
1264 005114 000772      BR       T8B
1265 005116 006077 173656      T8C:    ROR      @MTC

```

```

1266 005122 103375          BCC      .-4          ;WAIT FOR TU READY
1267 005124 013720 006606   MOV      TIME,(0)+    ;SAVE "LAST CHAR TO CU READY" TIME
1268 005130 004737 006474   JSR      %7,STRREW    ;REWIND
1269 005134 004737 006314   JSR      %7,CHGDRV    ;ANYMORE DRIVES?
1270 005140 000743          BR       T8A          ;YES
1271 005142 012720 177777   MOV      #-1,(0)+    ;TERMINATE TIMES
1272 005146 004737 006534   T8D:    JSR      %7,WATREW
1273 005152 004737 006314   JSR      %7,CHGDRV
1274 005156 000773          BR       T8D
1275
1276          ;TIME "WRITE EOF"
1277          ;WRITE A 3 BYTE RECORD FROM BOT FOLLOWED BY AN EOF.
1278
1279 005160 012700 006610   T9:     MOV      #TM1,%0
1280 005164 005037 006606   T9A:    CLR      TIME
1281 005170 012777 177775 173606   MOV      #-3,%ABC    ;WRITE 3 BYTES
1282 005176 012777 011772 173602   MOV      #WBUF,%ACA
1283 005204 013777 006374 173570   MOV      FDRIVE,%MTC ;SELECT DRIVE
1284 005212 052777 040005 173562   BIS      #40005,%MTC ;800 BPI, WRITE, GO
1285 005220 105777 173556   TSTB    %MTC
1286 005224 100375          BPL     .-4
1287 005226 006077 173546   ROR     %MTC
1288 005232 103375          BCC     .-4          ;WAIT FOR TU READY
1289 005234 042777 000016 173540   BIC     #16,%MTC
1290 005242 052777 000007 173532   BIS     #7,%MTC
1291 005250 105777 173526   T9B:    TSTB    %MTC    ;WRITE EOF, GO
1292 005254 100403          BMI     T9C          ;IS CU READY SET?
1293 005256 004737 006564   JSR     %7,TIMER    ;YES
1294 005262 000772          BR      T9B          ;NO, COUNT TIME
1295 005264 013720 006606   T9C:    MOV      TIME,(0)+ ;SAVE "WRITE EOF" TIME
1296 005270 004737 006474   JSR     %7,STRREW  ;REWIND
1297 005274 004737 006314   JSR     %7,CHGDRV  ;ANYMORE DRIVES?
1298 005300 000731          BR      T9A          ;YES
1299 005302 012720 177777   MOV     #-1,(0)+   ;TERMINATE TIMES
1300 005306 012737 010645 010070   MOV     #MSG15,MESAGE
1301 005314 012700 006610   MOV     #TM1,%0
1302 005320 004737 007050   JSR     %7,TYPTIM  ;PRINT "WRITE EOF" TIMES
1303 005324 012737 011656 010070   MOV     #RMSG11,MESAGE
1304 005332 004737 007566   JSR     %7,TOP
1305 005336 004737 006534   T9D:    JSR     %7,WATREW
1306 005342 004737 006314   JSR     %7,CHGDRV
1307 005346 000773          BR      T9D
  
```

```

1308 ;TIME 'EOR TO EOF SPACE TIME', 'SPACE SHUTDOWN' AND 'ONE INCH DATA TIME'.
1309 ;WRITE A 3 BYTE RECORD OVER ONE PREVIOUSLY WRITTEN
1310 ;AND THEN SPACE FORWARD 1 RECORD
1311 ;TIME FROM GO UNTIL EOF IS REACHED
1312
1313 005350 012700 006610 T10: MOV #TM1,%0
1314 005354 012701 006634 MOV #TM2,%1
1315 005360 012702 006660 MOV #TM3,%2
1316 005364 005037 006606 T10A: CLR TIME
1317 005370 012777 177775 173406 MOV #-3,@BC ;3 BYTE RECORD
1318 005376 012777 011772 173402 MOV #WBUF,@CA
1319 005404 013777 006374 173370 MOV FDRIVE,@MTC ;SELECT DRIVE
1320 005412 105777 173364 TSTB @MTC
1321 005416 100375 BPL .-4 ;WAIT FOR CU READY
1322 005420 012777 177776 173356 MOV #-2,@BC ;SPACE FORWARD 2 RECORDS
1323 005426 042777 000016 173346 BIC #16,@MTC
1324 005434 052777 000011 173340 BIS #11,@MTC ;SPACE FORWARD, GO
1325 005442 022777 177777 173334 T10B: CMP #-1,@BC
1326 005450 001374 BNE T10B ;WAIT FOR 1ST RECORD TO BE SPACED OVER
1327 005452 032777 040000 173320 T10C: BIT #40000,@MTC ;IS EOF SET?
1328 005460 001014 BNE T10D ;YES
1329 005462 105777 173314 TSTB @MTC ;IS CU READY
1330 005466 100403 BMI T10CC ;YES
1331 005470 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
1332 005474 000766 BR T10C
1333 005476 032777 040000 173274 T10CC: BIT #40000,@MTC ;HAVE EOF
1334 005504 001002 BNE T10D ;IS EOF SET?
1335 005506 005037 006606 CLR TIME ;NO, SET ERROR
1336 005512 013720 006606 T10D: MOV TIME,(0)+ ;SAVE 'EOR TO EOF SPACE TIME'
1337 005516 005037 006606 CLR TIME
1338 005522 000240 NOP
1339 005524 000240 NOP
1340 005526 000240 NOP
1341 005530 032777 000010 173242 T10E: BIT #10,@MTC ;IS SETTLEDOWN SET?
1342 005536 001003 BNE T10F ;YES
1343 005540 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
1344 005544 000771 BR T10E
1345 005546 013721 006606 T10F: MOV TIME,(1)+ ;SAVE 'SPACE SHUTDOWN' TIME
1346 005552 105777 173224 TSTB @MTC ;SEE IF CUR
1347 005556 100375 BPL .-4 ;IF NOT: BR
1348 005560 012777 176340 173216 MOV #-800.,@BC ;1 INCH OF DATA
1349 005566 012777 011772 173212 MOV #WBUF,@CA
1350 005574 005037 006606 CLR TIME
1351 005600 013777 006374 173174 MOV FDRIVE,@MTC ;SELECT DRIVE
1352 005606 105777 173170 TSTB @MTC ;WAIT FOR CU READY
1353 005612 100375 BPL .-4
1354 005614 052777 040005 173160 BIS #40005,@MTC ;800 BPI, WRITE, GO
1355 005622 022777 011774 173156 CMP #WBUF+2,@CA ;IS 2ND BYTE OUTPUT
1356 005630 003374 BGT .-6 ;NO
1357 005632 005777 173146 T10G: TST @BC ;YES IS LAST BYTE OUT
1358 005636 001403 BEQ T10H ;YES
1359 005640 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
1360 005644 000772 BR T10G
1361
1362 005646 013722 006606 T10H: MOV TIME,(2)+ ;SAVE 'ONE INCH DATA TIME'
1363 005652 004737 006474 JSR %7,STRREW ;REWIND

```

```

1364 005656 004737 006314 JSR %7,CHGDRV ;ANYMORE DRIVES?
1365 005662 000640 BR T10A ;YES
1366 005664 012720 177777 MOV #-1,(0)+ ;TERMINATE TIMES
1367 005670 012721 177777 MOV #-1,(1)+
1368 005674 012722 177777 MOV #-1,(2)+
1369 005700 012737 010673 010070 MOV #MSG16,MESAGE
1370 005706 012700 006610 MOV #TM1,%0
1371 005712 004737 007050 JSR %7,TYPTIM ;PRINT "EOR TO EOF SPACE TIME"
1372 005716 012737 011676 010070 MOV #RMSG12,MESAGE
1373 005724 004737 007566 JSR %7, TOP
1374 005730 012737 010721 010070 MOV #MSG18,MESAGE
1375 005736 012700 006634 MOV #TM2,%0
1376 005742 004737 007050 JSR %7,TYPTIM ;PRINT "SPACE SHUTDOWN" TIME
1377 005746 012737 011716 010070 MOV #RMSG13,MESAGE
1378 005754 004737 007566 JSR %7, TOP
1379 005760 012737 010747 010070 MOV #MSG20,MESAGE
1380 005766 012700 006660 MOV #TM3,%0
1381 005772 004737 007050 JSR %7,TYPTIM ;PRINT "ONE INCH DATA TIME"
1382 005776 012737 011734 010070 MOV #RMSG14,MESAGE
1383 006004 004737 007566 JSR %7, TOP
1384 006010 105737 001002 TSTB ACT11M ;ACT MODE? ++ C.W
1385 006014 001404 BEQ 1$ ;BRANCH - IF NO ++ C.W
1386 006016 105237 001056 INCB UDES ;GET NEXT UNIT ++ C.W
1387 006022 000137 001400 JMP $ACT ;CONTINUE TEST ++ C.W
1388 006026 000137 006200 1$: JMP T13 ;GO TO END OF TIMING
1389
1390 ;SPECIAL TAPE SPEED TEST USING 800 BPI SKEW TAPE
1391 ;READ FOWARD FROM BOT, MONITOR CA UNTIL IT EQUALS WBUF+2.
1392 ;TIME FROM WBUF+2 UNTIL BC=0 IS ONE INCH OF TAPE.
1393
1394 006032 000240 T11: NOP
1395 006034 012700 006610 MOV #TM1,%0
1396 006040 005037 006606 T11A: CLR TIME ;INITIALIZE TIME
1397 006044 013777 006374 172730 MOV FDRIVE,@MTC ;SELECT DRIVE
1398 006052 105777 172724 TSTB @MTC
1399 006056 100375 BPL -.4 ;AWAIT CUR
1400 006060 012777 176340 172716 MOV #-800.,@BC ;800 BYTES =1 INCH OF DATA
1401 006066 012777 011772 172712 MOV #WBUF,@CA ;SET BUS ADDRESS
1402 006074 052777 040003 172700 BIS #40003,@MTC ;LOAD 800 BPI READ AND GO
1403 006102 022777 011774 172676 CMP #WBUF+2,@CA
1404 006110 003374 BGT -.6 ;AWAIT 2ND BYTE
1405 006112 005777 172666 T11B: TST @BC ;SEE IF LAST BYTE
1406 006116 001403 BEQ T11C ;IF SO: BR
1407 006120 004737 006564 JSR %7,TIMER ;NO, COUNT TIME
1408 006124 000772 BR T11B
1409 006126 013720 006606 T11C: MOV TIME,(0)+ ;SAVE TAPE SPEED TIME
1410 006132 004737 006502 JSR %7,STRW ;REWIND
1411 006136 004737 006314 JSR %7,CHGDRV ;ANYMORE DRIVER?
1412 006142 000736 BR T11A ;YES
1413 006144 012720 177777 MOV #-1,(0)+ ;TERMINALS TIMER
1414 006150 012737 010775 010070 MOV #MSG21,MESAGE
1415 006156 012700 006610 MOV #TM1,%0 ;PRINT "TAPE SPEED FWD"
1416 006162 004737 007050 JSR %7,TYPTIM
1417 006166 012737 011753 010070 MOV #RMSG15,MESAGE
1418 006174 004737 007566 JSR %7, TOP ;PRINT RANGE
1419 006200 012737 011023 010070 T13: MOV #MSG27,MESAGE

```

```
1420 006206 004737 007566      JSR    %7, TOP      ;PRINT 'END OF TIMING'
1421 006212 105737 001002      TSTB   ACT11M      ;ACT MODE? ++ C.W
1422 006216 001001                BNE    1$          ;BRANCH - IF YES ++ C.W
1423 006220 000000                HALT
1424 006222 013700 000042      1$:   MOV    @#42,R0 ;GET MONITOR'S ADDRESS ++ C.W
1425 006226 001405                BEQ    HERE        ;BRANCH - IF NOT AUTO MODE ++ C.W
1426 006230 000005                RESET             ;CLEAR THE WORLD ++ C.W
1427 006232 004710      $ENDAD: JSR    PC,(R0) ;GO TO MONITOR'S ADDRESS ++ C.W
1428 006234 000240                NOP
1429 006236 000240                NOP
1430 006240 000240                NOP
1431 006242 000240      HERE:   NOP
1432 006244 000137 001310      JMP    STO          ;RESTART ++ C.W
1433
1434      ;RESET DRIVE SELECTION TO LOWEST NUMBER
1435
1436 006250 005037 006366      RSFDRV: CLR    CDRIVE ;START WITH DRIVE 0
1437 006254 012737 100000 006370  MOV    #100000,CDRVBT ;INITIALIZE FOR 0
1438 006262 033737 006370 006372  RSF1:  BIT    CDRVBT,DRIVES ;MASK WITH SELCTED DRIVES
1439 006270 001006                BNE    RSF2
1440 006272 005237 006366                INC    CDRIVE      ;+1 TO DRIVE NUMBER
1441 006276 000241                CLC
1442 006300 006037 006370                ROR    CDRVBT      ;MOVE MASK BIT TO NEXT DRIVE
1443 006304 000766                BR     RSF1
1444 006306 004737 006376      RSF2:  JSR    %7,GTNINE ;CHECK 9 TRACK
1445 006312 000207                RTS
1446
```

```
1447 ;SELECT NEXT DRIVE IN SEQUENCE
1448 ;SKIP FIRST EXIT ADDRESS IF LAST DRIVE SELECTED
1449
1450 006314 105777 172462 CHGDRV: TSTB @MTC
1451 006320 100375 BPL .-4 ;AWAIT CUR
1452 006322 005237 006366 INC CDRIVE ;+1 TO DRIVE
1453 006326 000241 CLC
1454 006330 006037 006370 ROR CDRVBT ;MOVE MASK BIT TO NEXT DRIVE
1455 006334 001005 BNE CHG1
1456 006336 004737 006250 JSR %7,RSFDRV ;RESET TO LOWEST DRIVE
1457 006342 062716 000002 ADD #2,(6) ;+2 TO SKIP FIRST EXIT
1458 006346 000207 RTS %7 ;EXIT
1459 006350 033737 006370 006372 CHG1: BIT CDRVBT,DRIVES ;MASK WITH SELECTED DRIVES
1460 006356 001756 BEQ CHGDRV ;CHECK FOR NEXT DRIVE
1461 006360 004737 006376 JSR %7,GTNINE ;CHECK 9 TRACK
1462 006364 000207 RTS %7
1463 006366 000000 CDRIVE: 0
1464 006370 000000 CDRVBT: 0
1465 006372 000000 DRIVES: 0
1466 006374 000000 FDRIVE: 0
1467
1468 ;CHECK FOR NINE TRACK DRIVES
1469
1470 006376 013737 006366 006374 GTNINE: MOV CDRIVE,FDRIVE
1471 006404 000337 006374 SWAB FDRIVE ;POSITION UNIT SELECT BITS
1472 006410 042737 174377 006374 BIC #174377,FDRIVE ;CLEAR ALL OTHER BITS
1473 006416 032737 000010 006366 BIT #10,CDRIVE ;TEST FOR 9 TRACK
1474 006424 001403 BEQ GNT1 ;NO
1475 006426 052737 020000 006374 BIS #20000,FDRIVE ;YES SET 9 TRACK BIT
1476 006434 000207 GNT1: RTS %7
1477
1478 ;INITIALIZE BYTE COUNT AND CURRENT ADDRESS FOR WRITE
1479
1480 006436 012777 172110 172340 WRINT: MOV #-BLENGTH,@BC
1481 006444 012777 011772 172334 MOV #WBUF,@CA
1482 006452 000207 RTS %7
1483
1484 ;STORE 1'S IN WRITE BUFFER
1485
1486 006454 012700 011772 ST1S: MOV #WBUF,%0
1487 006460 012720 177777 ST1SA: MOV #-1,(0)+
1488 006464 022700 017664 CMP #WBUF+BLENGTH+2,%0
1489 006470 001373 BNE ST1SA
1490 006472 000207 RTS %7
1491
1492 ;START REWIND OPERATIONS
1493
1494 006474 013777 006374 172300 STRREW: MOV FDRIVE,@MTC ;SELECT DRIVE
1495 006502 105777 172274 STRW: TSTB @MTC
1496 006506 100375 BPL .-4 ;WAIT FOR CU READY
1497 006510 006077 172264 ROR @MTC
1498 006514 103375 BCC .-4 ;WAIT FOR TAPE UNIT READY
1499 006516 052777 000017 172256 BIS #17,@MTC ;GO REWIND
1500 006524 105777 172252 TSTB @MTC
1501 006530 100375 BPL .-4 ;WAIT FOR CONTROL UNIT READY
1502 006532 000207 RTS %7
```

```
1503
1504           ;WAIT FOR REWIND TO FINISH
1505
1506 006534 013777 006374 172240 WATREW: MOV    FDRIVE,@MTC
1507 006542 006077 172232          ROR    @MTS
1508 006546 103375          BCC    .-4
1509 006550 032777 000040 172222          BIT    #40,@MTS          ;IS BOT SET?
1510 006556 001001          BNE    .+4          ;YES
1511 006560 000000          HALT
1512 006562 000207          RTS    %7          ;ERROR, NOT AT BOT AFTER REWIND
1513
1514           ;KEEP COUNT OF ELAPSED TIME
1515           ;EXIT EVERY 100 USEC
1516
1517 006564 005777 172222          TIMER: TST    @MTRD
1518 006570 100375          BPL    .-4
1519 006572 005777 172214          TST    @MTRD
1520 006576 100775          BMI    .-4
1521 006600 005237 006606          INC    TIME          ;+1 TO 100 USEC COUNT
1522 006604 000207          RTS    %7
1523 006606 000000          TIME:  0
1524 006610 000000          TM1:   0
1525           006634          . = TM1+20.
1526 006634 000000          TM2:   0
1527           006660          . = TM2+20.
1528 006660 000000          TM3:   0
1529           006704          . = TM3+20.
1530 006704 000000          TM4:   0
1531           006730          . = TM4+20.
1532 006730 000000          TM5:   0
1533           006754          . = TM5+20.
1534 006754 000000          TM6:   0
1535           007000          . = TM6+20.
1536 007000 000000          TM7:   0
1537           007024          . = TM7+20.
1538 007024 000000          TM8:   0
1539           007050          . = TM8+20.
```

```
1540 ;PRINT TITLE OF TEST EXECUTED AND THE DRIVE TIMES
1541
1542 007050 004737 007566 TYPTIM: JSR %7, TOP ;PRINT TITLE
1543 007054 012037 007304 TYPT0: MOV (0)+, VALUE ;GET TIME
1544 007060 022737 177777 007304 CMP #-1, VALUE ;FINISHED TIME BUFFER
1545 007066 001001 BNE .+4
1546 007070 000207 RTS %7
1547 007072 012737 007316 007314 MOV #DECPNT+2, DECPNT ;INITIALIZE DECIMAL VALUE POINTER
1548 007100 012737 000040 007312 MOV #40, ZERO ;INITIALIZE SPACE
1549 007106 012737 177774 007306 MOV #-4, DIGCNT ;DIGIT COUNT
1550 007114 012737 177777 007310 TYPT1: MOV #-1, DIGIT ;INITIAL VALUE
1551 007122 005237 007310 TYPT2: INC DIGIT ;+1 TO VALUE
1552 007126 167737 000162 007304 SUB @DECPNT, VALUE ;SUBTRACT CONSTANT
1553 007134 100372 BPL TYPT2 ;NOT NEGATIVE YET
1554 007136 067737 000152 007304 ADD @DECPNT, VALUE ;RESTORE LAST POSITIVE VALUE
1555 007144 004737 007234 JSR %7, DECOUT ;PRINT DECIMAL DIGIT
1556 007150 005237 007306 INC DIGCNT ;+1 TO DIGIT COUNT
1557 007154 001006 BNE TYP2A
1558 007156 012737 010152 010070 MOV #MSG2B, MESSAGE
1559 007164 004737 007566 JSR %7, TOP
1560 007170 000731 BR TYPT0
1561 007172 022737 177777 007306 TYP2A: CMP #-1, DIGCNT ;CHECK FOR DECIMAL PLACE
1562 007200 001011 BNE TYPT3 ;NO
1563 007202 105777 171614 TSTB @TPS
1564 007206 100375 BPL .-4
1565 007210 012777 000056 171606 MOV #'., @TPB ;PRINT DECIMAL POINT
1566 007216 012737 000060 007312 MOV #60, ZERO
1567 007224 062737 000002 007314 TYPT3: ADD #2, DECPNT ;+2 TO DECIMAL VALUE POINTER
1568 007232 000730 BR TYPT1 ;DO AGAIN
1569
1570 007234 005737 007310 DECOUT: TST DIGIT ;IS DIGIT 0
1571 007240 001004 BNE DEC1 ;NO
1572 007242 013737 007312 007310 MOV ZERO, DIGIT ;SUPPRESS LEADING ZEROS
1573 007250 000406 BR DEC2
1574 007252 012737 000060 007312 DEC1: MOV #60, ZERO ;INITIALIZE ZERO AFTER SOME VALUE FOUND
1575 007260 052737 000060 007310 BIS #60, DIGIT ;CONVERT TO ANSCII
1576 007266 105777 171530 DEC2: TSTB @TPS
1577 007272 100375 BPL .-4
1578 007274 013777 007310 171522 MOV DIGIT, @TPB ;PRINT
1579 007302 000207 RTS %7
1580 007304 000000 VALUE: 0
1581 007306 000000 DIGCNT: 0
1582 007310 000000 DIGIT: 0
1583 007312 000040 ZERO: 40 ;CONTAINS ZERO OR SPACE
1584 007314 007316 DECPNT: .+2
1585 007316 001750 1000.
1586 007320 000144 100.
1587 007322 000012 10.
1588 007324 000001 1.
```



```
1589  
1590  
1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606 007326 005037 001052 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
1607 007332 005000 CLR %0  
1608 007334 004737 007514 TTR0: JSR %7,TIN ;GO READ CHARACTER  
1609 007340 122737 000215 001050 CMPB #215,TIB ;SEE IF CR  
1610 007346 001005 BNE TTR1 ;IF NOT: BR  
1611 007350 005737 001052 TST TEMP1 ;SEE IF FIRST CHARACTER  
1612 007354 001446 BEQ TTR5 ;IF SO: BR  
1613 007356 000137 007450 JMP TTR2 ;ELSE GO LOAD VALUE  
1614 007362 122737 000260 001050 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0  
1615 007370 101402 BLOS TTR1A ;IF NOT: BR  
1616 007372 000137 007474 JMP TINER ;ELSE GO TO ERROR  
1617 007376 122737 000270 001050 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THEN 7  
1618 007404 101002 BHI TTR1B ;IF NOT: BR  
1619 007406 000137 007474 JMP TINER ;ELSE GO TO ERROR  
1620 007412 005237 001052 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG  
1621 007416 000241 CLC  
1622 007420 006100 ROL %0  
1623 007422 000241 CLC  
1624 007424 006100 ROL %0 ;SHIFT 3 LEFT  
1625 007426 000241 CLC  
1626 007430 006100 ROL %0  
1627 007432 042737 177770 001050 BIC #177770,TIB ;STRIP ASCII  
1628 007440 053700 001050 BIS TIB,%0 ;LOAD CHARACTER  
1629 007444 005301 DEC %1 ;SEE IF DONE  
1630 007446 001332 BNE TTR0 ;IF NOT: BR  
1631 007450 020002 TTR2: CMP %0,%2 ;SEE IF EXCEEDED MAXIMUM LIMIT  
1632 007452 101402 BLOS TTR3 ;IF NOT: BR  
1633 007454 000137 007474 JMP TINER ;ELSE GO TO ERROR  
1634 007460 020300 TTR3: CMP %3,%0 ;SEE IF BELOW MINIMUM LIMIT  
1635 007462 101402 BLOS TTR4 ;IF NOT: BR  
1636 007464 000137 007474 JMP TINER ;ELSE GO TO ERROR  
1637 007470 010015 TTR4: MOV %0,(5) ;LOAD VALUE  
1638 007472 000207 TTR5: RTS %7 ;EXIT
```

```
1639
1640 ;TTY ENTRY ERROR SUBROUTINE*****
1641
1642 007474 012737 011422 010070 T1NER: MOV #MSG34,MESAGE
1643 007502 004737 007566 JSR %7, TOP ;PRINT?
1644 007506 162716 000020 SUB #20,(6) ;RESET SP TO START OF VALUE ROUTINE
1645 007512 000207 RTS %7 ;READO VALUE ENTRY
1646
1647 ;TTY READ SUBROUTINE*****
1648
1649 007514 005077 171276 TTIN: CLR @TKS
1650 007520 005077 171274 CLR @TKB
1651 007524 005037 001050 CLR TIB
1652 007530 005277 171262 INC @TKS
1653 007534 105777 171256 TTIN1: TSTB @TKS
1654 007540 100375 BPL TTIN1
1655 007542 017737 171252 001050 MOV @TKB,TIB
1656 007550 105777 171246 TTIN2: TSTB @TPS
1657 007554 100375 BPL TTIN2
1658 007556 113777 001050 171240 MOVB TIB,@TPB
1659 007564 000207 RTS %7
1660
1661 ;TELETYPE OUTPUT PACKAGE
1662
1663 007566 142777 000177 171226 TOP: BICB #177,@TPS ;CLEAR FLAGS
1664 007574 117737 000270 007704 MOVB @MESSAGE,EOMK ;SAVE MESSAGE DELIMETER
1665 007602 005237 010070 INC MESSAGE ;+2 TO POINTER
1666 007606 127737 000256 007704 TOP1: CMPB @MESSAGE,EOMK ;IS CHARACTER THE 2ND DELIMETER
1667 007614 001001 BNE .+4 ;NO
1668 007616 000207 RTS %7 ;YES END
1669 007620 127727 000244 000100 CMPB @MESSAGE,#'a ;IS CHARACTER AN a INDICATING A CARRIAGE RETURN
1670 007626 001411 BEQ TOP3 ;YES
1671 007630 105777 171166 TSTB @TPS
1672 007634 100375 BPL .-4
1673 007636 117777 000226 171160 MOVB @MESSAGE,@TPB ;PRINT CHARACTER
1674 007644 005237 010070 TOP2: INC MESSAGE ;+2 TO POINTER
1675 007650 000756 BR TOP1 ;LOOP
1676
1677 ;CARRIAGE RETURN, LINE FEED
1678
1679 007652 105777 171144 TOP3: TSTB @TPS
1680 007656 100375 BPL .-4
1681 007660 112777 000215 171136 MOVB #215,@TPB
1682 007666 105777 171130 TSTB @TPS
1683 007672 100375 BPL .-4
1684 007674 112777 000212 171122 MOVB #212,@TPB
1685 007702 000760 BR TOP2
1686 007704 000000 EOMK: 0
1687
```

```
1688
1689
1690
1691
1692
1693
1694
1695
1696 007706 005037 001000      CKMODE: CLR      AUTOM      ;INIT AUTOMATIC MODE INDICATOR
1697 007712 105037 001002      CLRB     ACT11M     ;INIT ACT11 AUTO MODE INDICATOR
1698 007716 105037 001003      CLRB     XXDPM      ;INIT XXDP AUTO MODE INDICATOR
1699 007722 105037 001004      CLRB     ADUMPM     ;INIT ACT11 DUMP MODE INDICATOR
1700 007726 105037 001005      CLRB     XDUMPM     ;INIT XXDP DUMP MODE INDICATOR
1701 007732 005737 000042      TST      @#42       ;AUTO MODE?
1702 007736 001424      BEQ      2$         ;BRANCH - IF NO
1703 007740 005237 001000      INC      AUTOM      ;SET AUTO MODE INDICATOR
1704 007744 032737 020000 000052  BIT      #20000,@#52 ;MANUAL INTERVENTION?
1705 007752 001402      BEQ      6$         ;BRANCH - IF NO
1706 007754 000137 010032      JMP      ABORT      ;ABORT THE PROGRAM
1707 007760 023737 000042 000046 6$:  CMP      @#42,@#46  ;ACT11 MODE?
1708 007766 001403      BEQ      1$         ;BRANCH - IF YES
1709 007770 105237 001003      INCB    XXDPM      ;INDICATE XXDP AUTO MODE
1710 007774 000415      BR      5$         ;AND EXIT
1711 007776 105237 001002      1$:  INCB    ACT11M   ;INDICATE ACT11 AUTO MODE
1712 010002 005037 001054      CLR      SPTF      ;DO NOT SELECT SPEED TEST
1713 010006 000410      BR      5$         ;AND EXIT
1714 010010 105737 000041      2$:  TSTB   @#41     ;MAN/MODE VIA ACT11/PAPER TAPE?
1715 010014 001003      BNE     3$         ;BRANCH - IF NOT
1716 010016 105237 001004      INCB    ADUMPM     ;INDICATE MAN/MODE VIA ACT11/PAPER TAPE
1717 010022 000402      BR      5$         ;AND EXIT
1718 010024 105237 001005      3$:  INCB    XDUMPM   ;INDICATE MANUAL MODE VIA XXDP
1719 010030 000207      5$:  RTS      PC      ;RETURN
1720
1721
1722
```

```
1723  
1724 :  
1725 : *****  
1726 :  
1727 :  
1728 : ++  
1729 : CHECK FOR DUMP MODE OR AUTOMATIC/ACT11-XXDP MODE  
1730 :  
1731 010032 000005 ABORT: RESET ;CLEAR THE WORLD  
1732 010034 012737 010072 010070 MOV #MSG0,MESAGE ;GET ABORT MESSAGE  
1733 010042 004737 007566 JSR %7, TOP ;PRINT ABORT MESSAGE  
1734 010046 105737 001003 TSTB XXDPM ;XXDP AUTO MODE?  
1735 010052 001405 BEQ 1$ ;BRANCH - IF NO  
1736 010054 013700 000042 MOV @#42,R0 ;GET MONITOR EXIT ADDRESS  
1737 010060 005037 000042 CLR @#42 ;USE AS ABORT FLAG  
1738 010064 004700 JSR PC,R0 ;EXIT TO XXDP MONITOR  
1739 010066 000777 1$: BR . ;AND HANG  
1740  
1741 :  
1742 : *****
```

CZTSEBO TS03 DR FCTN TMR  
CZTSEB.P11 16-FEB-78 11:08

MACY11 30A(1052) 16-FEB-78 11:10 F 4 PAGE 45

SEQ 0044

1743

Line	Time	Code	Time	Code	Message
1744	010070	000000			MESSAGE: 0
1745					
1746	010072	040057	051120	043517	MSG0: .ASCII ;/@PROGRAM ABORTED /;
1747	010100	040522	020115	041101	
1748	010106	051117	042524	004504	
1749	010114	057			
1750	010115	057	040100	052506	MSG2: .ASCII ;/@FUNCTION /;
1751	010122	041516	044524	047117	
1752	010130	020040	020040	020040	
1753	010136	020040	027440		
1754	010142	020057	047125	052111	MSG2A: .ASCII ;/ UNIT /;
1755	010150	027440			
1756	010152	020057	027440		MSG2B: .ASCII ;/ /;
1757	010156	040057	051127	052111	MSG3: .ASCII ;/@WRITE FROM BOT /;
1758	010164	020105	051106	046517	
1759	010172	041040	052117	020040	
1760	010200	020040	027440		
1761	010204	040057	051127	052111	MSG4: .ASCII ;/@WRITE SHUTDOWN /;
1762	010212	020105	044123	052125	
1763	010220	047504	047127	020040	
1764	010226	020040	027440		
1765	010232	040057	051127	052111	MSG5: .ASCII ;/@WRITE START /;
1766	010240	020105	052123	051101	
1767	010246	020124	020040	020040	
1768	010254	020040	027440		
1769	010260	040057	042523	052124	MSG6: .ASCII ;/@SETTLE DOWN DELAY /;
1770	010266	042514	042040	053517	
1771	010274	020116	042504	040514	
1772	010302	020131	027440		
1773	010306	040057	051127	052111	MSG7: .ASCII ;/@WRITE TO ERASE HEAD/;
1774	010314	020105	047524	042440	
1775	010322	040522	042523	044040	
1776	010330	040505	027504		
1777	010334	040057	051127	052111	MSG8: .ASCII ;/@WRITE NONSTOP GAP /;
1778	010342	020105	047516	051516	
1779	010350	047524	020120	040507	
1780	010356	020120	027440		
1781	010362	040057	040502	045503	MSG9: .ASCII ;/@BACKSPACE SHUTDOWN /;
1782	010370	050123	041501	020105	
1783	010376	044123	052125	047504	
1784	010404	047127	027440		
1785	010410	040057	042522	042101	MSG10: .ASCII ;/@READ SHUTDOWN /;
1786	010416	051440	052510	042124	
1787	010424	053517	020116	020040	
1788	010432	020040	027440		
1789	010436	040057	043500	050101	MSG11: .ASCII ;/@GAPS SHOULD = 8>7>6>5>4>3, 3=2=1 (+OR- 5.0)/;
1790	010444	020123	044123	052517	
1791	010452	042114	036440	034040	
1792	010460	033476	033076	032476	
1793	010466	032076	031476	020054	
1794	010474	036463	036462	020061	
1795	010502	025450	051117	020055	
1796	010510	027065	024460	057	
1797	010515	057	043500	050101	MSG11A: .ASCII ;/@GAP 1 /;
1798	010522	030440	020040	020040	
1799	010530	020040	020040	020040	

Line	Code	Address	Address	Address	Text
1800	010536	020040	020040	057	
1801	010543	057	053500	044522	MSG12: .ASCII ;/@WRITE XIRG /;
1802	010550	042524	054040	051111	
1803	010556	020107	020040	020040	
1804	010564	020040	020040	057	
1805	010571	057	051100	040505	MSG13: .ASCII ;/@READ FROM BOT /;
1806	010576	020104	051106	046517	
1807	010604	041040	052117	020040	
1808	010612	020040	020040	057	
1809	010617	057	046100	051501	MSG14: .ASCII ;/@LAST CHAR TO CU RDY/;
1810	010624	020124	044103	051101	
1811	010632	052040	020117	052503	
1812	010640	051040	054504	057	
1813	010645	057	053500	044522	MSG15: .ASCII ;/@WRITE EOF /;
1814	010652	042524	042440	043117	
1815	010660	020040	020040	020040	
1816	010666	020040	020040	057	
1817	010673	057	042500	051117	MSG16: .ASCII ;/@EOR TO EOF SPACE /;
1818	010700	052040	020117	047505	
1819	010706	020106	050123	041501	
1820	010714	020105	020040	057	
1821	010721	057	051500	040520	MSG18: .ASCII ;/@SPACE SHUTDOWN /;
1822	010726	042503	051440	052510	
1823	010734	042124	053517	020116	
1824	010742	020040	020040	057	
1825	010747	057	047500	042516	MSG20: .ASCII ;/@ONE INCH DATA TIME /;
1826	010754	044440	041516	020110	
1827	010762	040504	040524	052040	
1828	010770	046511	020105	057	
1829	010775	057	052100	050101	MSG21: .ASCII ;/@TAPE SPEED FWD /;
1830	011002	020105	050123	042505	
1831	011010	020104	053506	020104	
1832	011016	020040	020040	057	
1833	011023	057	025100	025052	MSG27: .ASCII ;/@*****END OF TIMING*****@/;
1834	011030	025052	025052	025052	
1835	011036	025052	047105	020104	
1836	011044	043117	052040	046511	
1837	011052	047111	025107	025052	
1838	011060	025052	025052	025052	
1839	011066	025052	027500		
1840	011072	040057	041500	052132	MSG28: .ASCII ;/@@CZTSEBO TS03 DR FCTN TMR/;
1841	011100	042523	030102	052040	
1842	011106	030123	020063	051104	
1843	011114	043040	052103	020116	
1844	011122	046524	027522		
1845	011126	020057	020040	040522	MSG29: .ASCII ;/ RANGE MAX - MIN@ /;
1846	011134	043516	020105	040515	
1847	011142	020130	020055	044515	
1848	011150	040116	027440		
1849	011154	040057	047516	052440	MSG30: .ASCII ;/@NO UNITS AVAILABLE FOR TEST @PRESS CONTINUE WHEN READY@/;
1850	011162	044516	051524	040440	
1851	011170	040526	046111	041101	
1852	011176	042514	043040	051117	
1853	011204	052040	051505	020124	
1854	011212	050100	042522	051523	
1855	011220	041440	047117	044524	

1856	011226	052516	020105	044127	
1857	011234	047105	051040	040505	
1858	011242	054504	027500		
1859	011246	040057	042522	044507	MSG31: .ASCII ;/@REGISTER START: /;
1860	011254	052123	051105	051440	
1861	011262	040524	052122	020072	
1862	011270	057			
1863	011271	057	051500	042520	MSG32: .ASCII ;/@SPEED TEST (1=YES-0=NO): /;
1864	011276	042105	052040	051505	
1865	011304	020124	030450	054475	
1866	011312	051505	030055	047075	
1867	011320	024517	020072	057	
1868	011325	057	046500	052517	MSG33: .ASCII ;/@AMOUNT SKEW TAPE ON ALL AVAILABLE UNITS: TYPE CR WHEN READY/;
1869	011332	052116	051440	042513	
1870	011340	020127	040524	042520	
1871	011346	047440	020116	046101	
1872	011354	020114	053101	044501	
1873	011362	040514	046102	020105	
1874	011370	047125	052111	035123	
1875	011376	052040	050131	020105	
1876	011404	051103	053440	042510	
1877	011412	020116	042522	042101	
1878	011420	027531			
1879	011422	020057	020077	057	MSG34: .ASCII ;/ ? /;
1880					
1881	011427	057	033040	030462	RMSG1: .ASCII ;/ 621.0 - 509.0/;
1882	011434	030056	026440	032440	
1883	011442	034460	030056	057	
1884	011447	057	020040	033461	RMSG2: .ASCII ;/ 17.6 - 13.6/;
1885	011454	033056	026440	030440	
1886	011462	027063	027466		
1887	011466	020057	031440	027071	RMSG3: .ASCII ;/ 39.0 - 32.0/;
1888	011474	020060	020055	031063	
1889	011502	030056	057		
1890	011505	057	020040	033063	RMSG4: .ASCII ;/ 36.0 - 30.0/;
1891	011512	030056	026440	031440	
1892	011520	027060	027460		
1893	011524	020057	033440	027066	RMSG5: .ASCII ;/ 76.0 - 58.0/;
1894	011532	020060	020055	034065	
1895	011540	030056	057		
1896	011543	057	020040	033440	RMSG6: .ASCII ;/ 7.7 - 6.3/;
1897	011550	033456	026440	033040	
1898	011556	031456	057		
1899	011561	057	020040	033440	RMSG7: .ASCII ;/ 7.7 - 6.3/;
1900	011566	033456	026440	033040	
1901	011574	031456	057		
1902	011577	057	020040	034463	RMSG8: .ASCII ;/ 39.0 - 32.0/;
1903	011604	030056	026440	031440	
1904	011612	027062	027460		
1905	011616	020057	033063	027066	RMSG9: .ASCII ;/ 366.0 - 300.0/;
1906	011624	020060	020055	030063	
1907	011632	027060	027460		
1908	011636	020057	032462	027070	RMSG10: .ASCII ;/ 258.0 - 212.0/;
1909	011644	020060	020055	030462	
1910	011652	027062	027460		
1911	011656	020057	031064	027060	RMSG11: .ASCII ;/ 420.0 - 340.0/;



CZTSEBO TS03 DR FCTN TMR  
CZTSEB.P11 16-FEB-78 11:08

J 4  
MACY11 30A(1052) 16-FEB-78 11:10 PAGE 49

SEQ 0048

1912	011664	020060	020055	032063	
1913	011672	027060	027460		
1914	011676	020057	030064	027060	RMSG12: .ASCII ;/ 400.0 - 320.0/;
1915	011704	020060	020055	031063	
1916	011712	027060	027460		
1917	011716	020057	020040	027067	RMSG13: .ASCII ;/ 7.7 - 6.3/;
1918	011724	020067	020055	027066	
1919	011732	027463			
1920	011734	020057	034040	027071	RMSG14: .ASCII ;/ 89.0 - 73.0/;
1921	011742	020060	020055	031467	
1922	011750	030056	057		
1923	011753	057	020040	034470	RMSG15: .ASCII ;/ 89.0 - 73.0/;
1924	011760	030056	026440	033440	
1925	011766	027063	027460		
1926					
1927					
1928	011772	000000			WBUF: .EVEN
1929		000001			0
					.END









T5	003734	1105#	1131															
T5A	003776	1112#	1125															
T5B	004020	1117#	1120															
T5C	004036	1118	1121#															
T5DRV	001044	697#	1103*	1108	1129*													
T6	004346	1170#																
T6A	004356	1172#	1209															
T6B	004470	1189#	1192															
T6C	004506	1190	1193#															
T6D	004544	1200#	1203															
T6E	004562	1201	1204#															
T6F	004676	1222#	1224															
T7	004710	1228#																
T7A	004714	1229#	1241															
T7B	004740	1233#	1236															
T7C	004756	1234	1237#															
T8	005044	1254#																
T8A	005050	1255#	1270															
T8B	005102	1261#	1264															
T8C	005116	1262	1265#															
T8D	005146	1272#	1274															
T9	005160	1279#																
T9A	005164	1280#	1298															
T9B	005250	1291#	1294															
T9C	005264	1292	1295#															
T9D	005336	1305#	1307															
UDES	001056	702#	752*	753	756	767	769*	1386*										
VALUE	007304	1543*	1544	1552*	1554*	1580#												
WATREW	006534	780	887	925	1014	1105	1222	1272	1305	1506#								
WBUF	011772	813	849	891	903	949	1117	1189	1200	1233	1282	1318	1349	1355				
		1401	1403	1481	1486	1488	1928#											
WRINT	006436	809	845	939	947	962	982	1018	1024	1028	1039	1044	1053	1065				
		1074	1106	1114	1172	1183	1197	1230	1255	1480#								
XDUMPM	001005	671#	1700*	1718*														
XXDPM	001003	669#	1698*	1709*	1734													
ZERO	007312	1548*	1566*	1572	1574*	1583#												
\$ACT	001400	753#	770	1387														
\$ENDAD	006232	649	1427#															
\$SVPC =	001000	634#	655															
.	= 011774	623#	625	634	636#	640#	644#	648#	651#	655#	676#	678#	795	820				
		856	858	895	913	943	956	958	966	980	993	1022	1027	1031				
		1043	1047	1052	1057	1068	1073	1078	1094	1113	1127	1175	1177	1180				
		1185	1194	1205	1239	1243	1260	1266	1286	1288	1321	1347	1353	1356				
		1399	1404	1451	1496	1498	1501	1508	1510	1518	1520	1525#	1527#	1529#				
		1531#	1533#	1535#	1537#	1539#	1545	1564	1577	1584	1667	1672	1680	1683				
		1739																

. ABS. 011774 000

ERRORS DETECTED: 0

CZTSEB,CZTSEB.SEQ/CRF/SOL/NL:TOC=SYSMAC.SML 400,1066 ,CZTSEB.P11 400,1424  
 RUN-TIME: 9 11 .6 SECONDS  
 RUN-TIME RATIO: 494/21=22.8

CZTSEBO TS03 DR FCTN TMR  
CZTSEB.P11 16-FEB-78 11:08

MACY11 30A(1052) 16-FEB-78 11:10 PAGE 56  
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0054

CORE USED: 31K (61 PAGES)

DOCUMENT PAGES: 54