

TM03, TE16  
TU77

TM03/TE16, TU77 DRT  
CZTEDDO

AH-A801D-MC  
FICHE 1 OF 1

OCT 1983  
COPYRIGHT © 77-83  
MADE IN USA



A large grid of approximately 15 columns and 15 rows of small, illegible text fragments, likely representing a data table or a series of microfilm frames. The text is too small to be read accurately.



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45

.REM %

IDENTIFICATION

PRODUCT CODE: AC-A800D-MC  
PRODUCT NAME: CZTEDDO TMO3-TE16/TU77 DATA RELIBILITY PROGRAM  
DATE CKEATED: 11 - JULY - 1983  
MAINTAINER: TAPE DIAGNOSTIC GROUP  
AUTHOR: J. G. ADAMS

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 (©) 1977,1983 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

	PARAGRAPH	SUBJECT	PAGE
47			
48			
49			
50			
51	1.	ABSTRACT	3
52	2.	REQUIREMENTS	3
53	3.	LOADING PROCEDURE	3
54	4.	STARTING PROCEDURE	4
55	5.	DATA PATTERNS	11
56	6.	RANDOMIZATION	12
57	7.	DYNAMIC PARAMETERS	13
58	8.	CONSOLE SWITCH	14
59	9.	ERROR PRINTOUTS	19
60	10.	STATISTICS PRINTOUT	27
61	11.	AUTO SEQUENCE	28
62	12.	TESTING PROCEDURES	30
63	13.	LISTING	32

65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109

1. ABSTRACT

-----  
THIS PROGRAM IS DESIGNED TO BE USE<sup>D</sup> BY AN EXPERIENCED ENGINEER /TECHNICIAN FOR EVALUATION AND DEBUGGING OF MAG TAPE DRIVES. THE PROGRAM IS CAPABLE OF EXERCISING THE TE16 MAGNETIC ON A MASSBUS THROUGH THE TM03 MAG TAPE CONTROLLER. ANY COMBINATION OF TM03'S & TE16'S UP TO A MAXIMUM OF EIGHT (8), MAY BE TESTED BY A SINGLE EXECUTION OF THE PROGRAM. THIS FLEXIBILITY IS POSSIBLE BECAUSE THE PROGRAM HAS NO FIXED PARAMETERS OR TESTING SEQUENCE. THE ENTIRE TEST PLAN, INCLUDING PARAMETERS AND OPERATING SEQUENCE, IS DETERMINED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS AND SETTING OF CONSOLE SWITCHES.

THE PROGRAM PROVIDES FOR TESTING OF ALL TAPE DRIVE FUNCTIONS SUCH AS WRITING, READING, REWINDING, TAPE POSITIONING, EOT - BOT SENSING AND ASSUMES A GOOD RH AND TM03.

HOWEVER; THE RH AND TM03 ARE TESTED SOMEWHAT INTRINSICALLY DURING THE TEST CYCLE IN ORDER TO PROVIDE FULL INFORMATION ABOUT ANY ERROR CONDITIONS DETECTED.

DURING A TEST CYCLE, CHECKS ARE MADE FOR STATUS ERRORS, DATA ERRORS, POSITION ERRORS, WORD COUNT AND CURRENT MEMORY ADDRESS ERRORS WHEREVER APPLICABLE AS DETECTED BY THE RH OR TM03.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR  
B. 8K OF CORE  
C. TELETYPE  
D. TM03 TAPE CONTROLLER  
E. 1 TO 8 MAG TAPE DRIVES  
F. MASSBUS CONTROLLER

3. LOADING PROCEDURE

-----  
USE STANDARD PROCEDURE FOR LOADING BINARY TAPES

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  
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

4. STARTING PROCEDURE

THERE ARE FOUR (4) STARTING ADDRESSES THAT MAY BE USED;  
200(8),204(8),210(8),AND 240(8):

- A. 200(8): THIS ADDRESS MUST BE USED ON INITIAL START FROM LOAD AS ALL PARAMETERS ARE ENTERED FROM HERE. REQUESTS ARE PRINTED ON THE TELETYPE FOR ENTRY OF RH STARTING ADDRESS, VECTOR ADDRESS, DRIVE NUMBER(TMO3 ADDRESS), SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN NUMBER, TAPE MARK AND STALL FOR READ, WRITE, AND TURNAROUND. ALL REPOSES SHOULD BE MADE IN OCTAL AND WITHIN THE LIMITS OF THE PARAMETER. A QUESTION MARK (?) WILL BE TYPED IF ANY CHARACTER ENTERED IS NOT BETWEEN 0 THRU 7 (OCTAL). THE CHARACTER MAY BE RETYPED FOLLOWING THE QUESTION MARK. IF THE RESPONSE IS NOT WITHIN ITS LIMITS. A QUESTION MARK (?) IS TYPED AND THE ENTIRE RESPONSE MAY BE RENTERED. SOME RESPONSES REQUIRE MORE THAN ONE (1) CHARACTER, BUT NONE REQUIRES MORE THAN SIX (6). RESPONSES OF MORE THAN ONE CHARACTER NEED NOT HAVE LEADING ZEROS AND SHOULD BE TERMINATED BY A CARRIAGE RETURN IF LESS THAN THE MAXIMUM NUMBER OF CHARACTERS IS INPUT.
- B. 204(8): THIS ADDRESS SHOULD BE USED ANYTIME A RESTART OF THE PROGRAM IS NECESSARY AND THE PARAMETERS ENTERED AT THE INITIAL START OF 200(8) NEED NOT BE CHANGED. ALSO NOTE THAT ANY DATA PATTERN WHICH HAD BEEN GENERATED BY SETTING THE RANDOM DATA SWITCH (CONSOLE SWITCH EIGHT) WILL NOT BE OVERWRITTEN AND THEREFORE IS HELD IN CORE FOR USE UNTIL CONSOLE SWITCH EIGHT(8) IS AGAIN SET AND THAT ALL STATISTICS WIL
- C. 210(8): THIS ADDRESS IS THE SAME AS USING 204(8) IN THAT THE PREVIOUSLY SET PARAMETERS ARE USED; HOWEVER, THE DATA PATTERN IS RETURNED TO THE FIXED PATTERN ORIGINALLY CALLED FOR AT THE 200(8) START AND ALL STATISTICS ARE CLEARED TO
- D. 240(8): THIS IS A SPECIAL ADDRESS WHICH WILL CAUSE THE PROGRAM TO EXECUTE A PREDETERMINED TEST PLAN ON ALL AVAILABLE DRIVES AND SLAVES. THE ONLY INPUT REQUIRED BY THE OPERATOR IS A RESPONSE TO REQUESTS FOR THE RH ADDRESS, VECTOR ADDRESS, CONTINUOUS OPERATION OF THE SEQUENCE, AND NRZ ONLY.
- E. 300(8): THIS ADDRESS IS TO BE USED AS A RESTART ONLY AND WILL PERFORM JUST AS IN 200(8) EXCEPT THAT THE PARAMETER INPUT LIST IS SHORTENED. THE SHORT PARAMETER LIST CONSISTS OF DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, FORMAT, RECORD COUNT, CHARACTER COUNT, PATTERN, TAPE MARK, AND

167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222

INTERCHANGE READ.  
\*\*NOTE SEE ALSO SECTION 8-CONSOLE SWITCH SETTINGS

THE FOLLOWING IS AN EXPLANATION OF THE INITIAL START (200 OCTAL) REQUESTS AND RESPONSES:

REGISTER START: THE RESPONSE REQUIRED FOR THIS REQUEST IS TO ENTER THE ADDRESS OF THE FIRST RH REGISTER (CS1) AS A SIX DIGIT UNIBUS ADDRESS.

VECTOR ADDRESS: THE RESPONSE FOR THIS REQUEST IS TO ENTER THE INTERRUPT VECTOR ADDRESS USED BY THE RH AS A THREE (3) DIGIT ADDRESS.

DRIVE NUMBER: THE DRIVE NUMBER (MASSBUS ADDRESS OF THE TM03) IS ENTERED AS ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THROUGH 7.

SLAVE NUMBER: THE SLAVE NUMBER IS ENTERED AS ONE (1) OCTAL CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THROUGH 7. WHEN THE SLAVE NUMBER HAS BEEN ENTERED AND IS LEGAL, THE PROGRAM TESTS FOR THE PRESENCE OF A SLAVE OF THAT NUMBER. IF THE SLAVE IS AVAILABLE A PRINTOUT OF 7 CHANNEL, IF APPLICABLE, AND ITS SERIAL NUMBER (IN BCD) WILL BE MADE TO ASSIST THE OPERATOR IN SETTING OF DENSITY, PARITY, AND FORMAT. A CHECK IS MADE FOR THE PROPER SETTING OF THE DRIVE TYPE REGISTER; IF WRONG, A MESSAGE IS PRINTED FOR INFORMATION ONLY. IF THE SLAVE IS NOT AVAILABLE, A MESSAGE STATING SO WILL BE PRINTED AND A NEW SLAVE NUMBER REQUEST WILL BE ISSUED. WHEN A GOOD SLAVE NUMBER HAS BEEN ENTERED, REQUESTS FOR OPERATING DENSITY PARITY AND FORMAT ARE MADE FOR THAT SLAVE AND SHOULD BE RESPONDED TO ACCORDING TO THAT PARTICULAR SLAVE'S NEEDS. AS MANY AS EIGHT (8) SLAVE NUMBER REQUESTS MAY BE USED, HOWEVER, AT LEAST ONE MUST BE USED. THE SLAVE NUMBERS AND THEIR RESPECTIVE DENSITY, PARITY AND FORMAT MAY BE ENTERED IN ANY ORDER. THE INFORMATION FOR EACH SLAVE ENTERED IS LOADED INTO A TABLE FOR REFERENCE IN TESTING. IF LESS THAN EIGHT(8) SLAVES ARE REQUIRED, THEN RESPONDING TO THE SLAVE NUMBER REQUEST WITH A CARRIAGE RETURN WILL TERMINATE THE SLAVE ENTRIES AND CONTINUE TO THE NEXT PARAMETER. IT SHOULD BE REMEMBERED

223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
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

THAT AT LEAST ONE SLAVE NUMBER REQUEST  
MUST BE ENTERED. IF THE FIRST  
REQUEST IS RESPONDED TO BY A CARRIAGE  
RETURN, THEN THE REQUEST WILL BE REPEATED.

DENSITY: THE DESITY REQUEST IS RESPONDED TO BY ONE (1) OCTAL  
CHARACTER AND MUST BE WITHIN THE LIMITS OF 0 THRU 4.  
AS EACH SLAVE NUMBER IS ENTERED, A REQUEST FOR THE  
OPERATING DENSITY FOR THAT SLAVE IS TYPED. THE  
RESPONSE MEANINGS ARE AS FOLLOWING:

- A. 3 = 800BPI, NRZI
- B. 4 = 1600BPI, PE (9 CHANNEL ONLY)

PARITY: THE PARITY REQUEST IS RESPONDED TO BY ONE (1)  
OCTAL CHARACTER AND MUST BE EITHER 0 OR 1.

- A. 1 = EVEN PARITY
- B. 0 = ODD PARITY

FORMAT: THE FORMAT REQUEST IS RESPONDED  
TO BY TWO (2) CHARACTERS  
AND SHOULD BE AS FOLLOWS

- A. 14 = 9 CHANNEL NORMAL (TWO FRAMES PER WORD)
- B. 15 = CORE DUMP (FOUR FRAMES PER WORD)
- C. 16 = PDP-15 OR IBM COMPATABLE (TWO FRAMES PER  
(DATA IS BYTE SWAPPED ON TAPE))

RECORD COUNT: THIS REQUEST IS RESPONDED TO BY A SIX (6) CHARACTER  
OCTAL NUMBER FROM 1 TO 177777. REMEMBER LEADING  
ZEROS ARE NOT REQUIRED AND IF LESS THAN SIX  
CHARACTERS ARE ENTERED, A CARRIAGE RETURN  
WILL TERMINATE THE RESPONSE. THE RECORD COUNT  
IS USED IN CONJUNCTION WITH THE CHARACTER COUNT  
TO ESTABLISH A BLOCKING FACTOR FOR USE IN READ OR  
WRITE CYCLES.

CHARACTER COUNT: THIS RESPONSE IS ENTERED AS FOUR (4) OCTAL  
CHARACTERS WITHIN THE LIMITS OF 20 THRU 4000. AGAIN  
LEADING ZEROS ARE NOT REQUIRED AND A CARRIAGE  
RETURN TERMINATES A LESS THAN FOUR (4) CHARACTER  
RESPONSE. THE CHARACTER COUNT IN CONJUNCTION  
WITH THE RECORD COUNT IS USED TO ESTABLISH  
THE BLOCK SIZE (CHARACTERS PER RECORD, AND  
RECORDS PER BLOCK) USED IN READ AND WRITE CYCLES.  
THE SAME BLOCKING IS USED ON ALL AVAILABLE UNITS.

275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328

**PATTERN NUMBER:** THIS RESPONSE IS A TWO (2) CHARACTER OCTAL NUMBER WITHIN THE LIMITS OF 0 THRU 15(8). THE NUMBER ENTERED WILL CAUSE A SPECIFIC DATA PATTERN TO BE USED FOR ALL READING AND WRITING. THIS DATA PATTERN IS NOT CHANGED UNLESS RANDOM DATA IS REQUESTED BY SETTING CONSOLE SWITCH EIGHT (8) TO A ONE. RESETTING OF THE RANDOM DATA SWITCH DOES NOT CAUSE REVERSION TO THE FIXED PATTERN, BUT WILL HOLD THE LAST GENERATED PATTERN UNTIL A RESTART IS DONE FROM LOCATION 200(8), 210(8), OR 300(8). WHEN 0' RATING IN NRZ MODE (DENSITY 0-5) THE PROGRAM CONSTRUCTS AND SAVES BOTH AN EXPECTED CRC CHARACTER AND AN LRC CHARACTER FOR COMPARISONS WITH THE HARDWARE GENERATED CHECK CHARACTER IN BOTH READ AND WRITE. THE SELECTION OF DATA PATTERN ZERO (0) HAS A SPECIAL USE. PATTERN NUMBER ZERO (0) WILL CAUSE TO BE READ IN AT THE HIGH SPEED PAPER TAPE READER ANY DATA PATTERN DESIRED. THE EXTERNAL INPUT DATA THROUGH THE READER IS DONE BY PREPARING A PAPER TAPE WITH A PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D) ANY CONFIGURATION OF BITS AND CHARACTERS MAY BE USED AND A LIMIT OF 377(8) CHARATERS IS IMPOSED. WHEN EXTERNAL DATA IS INPUT, THE ENTIRE WRITE BUFFER IN CORE IS FILLED WITH THE PATTERN SO THAT ANY SIZE RECORD MAY BE USED. DATA PATTERN PATTERN ZERO (0) EXTERNAL PAPER TAPE NEED ONLY BE READ ONCE AT INITIAL START OF 200(8), AND NEED NOT BE READ AGAIN UNLESS OVERWRITTEN BY RANDOM DATA. BE SURE TO LOAD THE READER BEFORE PRESSING START.

**TAPE MARK:** THE TAPE MARK REQUEST IS USED TO DETERMINE IF THE OPERATOR WISHES TO HAVE EACH DATA BLOCK SEPERATED BY A TAPE MARK. IF RESPONDED TO BY A ONE (1) THE TAPE MARK WILL BE WRITTEN AND WHEN READING WILL BE EXPECTED AT THE END OF DATA BLOCK. A ZERO (0) RESPONSE WILL DISALLOW TAPE MARK. PLEASE NOTE THAT THE TAPE MARK RECORD INCREASES THE BLOCK SIZE BY ONE (1) RECORD; IN OTHER WORDS, A BLOCK OF 100 RECORDS WILL HAVE THE TAPE MARK AS RECORD 101.

**INTERCHANGE PEAD:** THIS REQUEST IS RESPONDED TO BY A SINGLE CHARACTER INPUT OF EITHER ONE (1) OR ZERO (0). A RESPONSE OF ONE (1) WILL CAUSE ALL READING TO BE DONE IN THE INTERCHANGE MODE. A ZERO RESPONSE WILL CAUSE READING IN NORMAL MODE.



330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
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

SINGLE PASS: THIS REQUEST IS RESPONDED TO BY EITHER A ONE (1) OR A ZERO (0). RESPONSE OF 1, WILL CAUSE THE TEST TO BE STOPPED AFTER THE LAST AVAILABLE DRIVE REACHES END OF TAPE. A RESPONSE OF 0, WILL ALLOW CONTINUOUS RUNNING THROUGH MULTIPLE PASSES. TO RESTART AT END OF PASS, PRESS CONTINUE, OR RESTART AT THE CONSOLE.

STALLS: THE STALL REQUESTS ARE RESPONDED TO BY A SIX (6) CHARACTER OCTAL NUMBER WITHIN THE LIMITS OF 1 THRU 177777. LEADING ZEROS ARE NOT REQUIRED AND AN ENTRY OF LESS THAN SIX (6) CHARACTERS SHOULD BE TERMINATED BY A CARRIAGE RETURN. EACH INCREMENT OF THE VALUE ADDS ABOUT 2.6 MICSEC TO THE DELAY.

READ: THE TIME DELAY BETWEEN EACH RECORD READ

WRITE: THE TIME DELAY BETWEEN EACH RECORD WRITTEN

TURN AROUND: TIME DELAY BETWEEN CHANGES OF TAPE DIRECTION (FORWARD, TO REVERSE, ETC.) AND BETWEEN BLOCKS.

FIXED PARAMETERS: IT SHOULD BE NOTED THAT ALL PARAMTERS EXCEPT FOR THE SLAVE DESCRIPTION VALUES (SLAVE NUMBER, DENSITY, PARITY, AND FORMAT) HAVE NOMINAL VALUES ALREADY STORED IN THE PROGRAM. COUNT, CHARACTER COUNT, TAPE MARK AND STALLS) IS TYPED. ITS PRESENT STORED VALUE IS ALSO PRINTED. IF THESE VALUES NEED NOT BE CHANGED, SIMPLY TYPE A CARRIAGE RETURN AS RESPONSE AND NO CHANGE WILL BE MADE. EACH START OF THE PROGRAM AT 200(8) WILL SHOW THE CURRENT VALUES OF THESE PARAMETERS AS PER THE LAST ENTRY. WHEN A FRESH LOAD OF THE PAPER TAPE IS DONE, THE PARAMETERS WILL REFLECT THE FIXED VALUES STORED IN THE PROGRAM.

- A. RECORD COUNT = 100
- B. CHARACTER COUNT = 200
- C. PATTERN NUMBER = 1
- D. TM=0
- E. INTERCHANGE READ = 0
- F. SINGLE PASS = 0
- G. READ STALL = 1
- H. WRITE STALL = 1
- I. TURN AROUND STALL = 1

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  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431

SAMPLE START AT 200(8):

THE FOLLOWING IS A SAMPLE OF THE  
PRINTED REQUESTS AND THEIR RESPONSES.  
RESPONSES ARE ENCLOSED IN PARENS FOR  
CLARITY ONLY AND (CR) MEANS CARRIAGE RETURN

LOAD ADDRESS 200(8), SET CONSOLE SWITCHES, PRESS START SWITCH:

TE16 TAPE DRIVE TEST

REGISTER START=172440(172440)  
VECTOR ADDRESS=224(CR)  
DRIVE NUMBER (4)  
SLAVE NUMBER=(5) SN: 5009  
DENSITY=(3)  
PARITY=(0)  
FORMAT=(14)  
SLAVE NUMBER=(2) 9 CHAN SN: 0022  
DENSITY=(3)  
PARITY=(1)  
FORMAT=(15)  
SLAVE NUMBER=(CR)  
RECORD COUNT=100 (500)(CR)  
CHARACTER COUNT=200 (38)?(7)(CR)  
PATTERN NUMBER=1 (22)  
?  
(6)(CR)  
TM=(0)  
INTERCHANGE READ=(1)  
SINGLE PASS=(0)

ENTER STALLS  
READ=1 (CR)  
WRITE=1 (CR)  
TURN AROUND=1 (3000)(CR)

THE PROGRAM WILL NOW PERFORM THE TEST CYCLE SET IN  
THE CONSOLE SWITCHES ON SLAVE FIVE (5) THEN TWO (2),  
ONE BLOCK ON EACH UNIT PER CYCLE, USING DATA PATTERN  
NUMBER SIX (6) WITH A BLOCKING FACTOR OF 37 CHARACTERS  
PER RECORD AND 500 RECORDS PER BLOCK. THE DELAYS ARE SET  
FOR MINIMUM ON READ AND WRITE, AND APPROXIMATELY .75  
SECONDS ON TURN AROUND.

NO TAPE MARKS WILL BE WRITTEN AND ALL READING  
WILL BE DONE IN INTERCHANGE MODE (MAINT MODE 0001).

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  
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

4.1 AUTOMATIC MODE OPERATION  
-----

IF THE PROGRAM IS LOADED AND RUN IN AUTOMATIC (CHAIN) MODE THE AUTO ACCEPT SEQUENCE TEST PLAN IS RUN. SEE SEC 11. BELOW; THE SOFTWARE SWR IS INVOKED WITH A SWITCH SETTING OF 000000 IF LOADED VIA ACT11. NO OPERATOR INTERVENTION IS REQUIRED.

\*\*EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMDP CHAIN MODE THE PROGRAM WILL TEST ALL SLAVES ON THE FIRST AVAILABLE DRIVE EXCEPT SLAVE 0.

\*\*NOTE: IN ORDER TO CHANGE THE DEFAULT SETTING OF THE SOFTWARE SWR, CHANGE LOC: 176(SWREG:) TO THE DESIRED SETTING.

5. DATA PATTERNS  
-----

THERE ARE FIFTEEN DATA PATTERN GENERATORS STORED IN CORE AND ANY ONE OF THESE MAY BE SELECTED. THE ONE UNIQUE CASE IS PATTERN ZERO(0); SELECTION OF PATTERN ZERO(0) REQUIRES THAT A PREVIOUSLY PREPARED PAPER TAPE BE ENTERED AT THE HIGH SPEED READER. THIS TAPE CONTAINS A DATA PATTERN OF NO MORE THAN 377 OCTAL CHARACTERS. THE FIRST CHARACTER READ IN IS THE NUMBER OF ACTUAL DATA CHARACTERS THAT ARE CONTAINED ON THE TAPE. EACH DATA CHARACTER MAY BE ANY COMBINATION OF BITS AND WILL BE LOADED INTO CORE AS THEY APPEAR ON THE TAPE. NO MATTER HOW MANY CHARACTERS ARE ON TAPE, THE ENTIRE WRITE BUFFER (4000 CHARACTERS) WILL BE FILLED WITH THE PATTERN ENTERED SO THAT ANY SIZE RECORD CAN BE USED. (SEE DTC MAINDEC-11-DZTUF-A-D) THE PROGRAM GENERATES A CYLIC REDUNDENCY CHECK CHARACTER (CRC) AND A LONGITUDINAL REDUNDENCY CHECK CHARACTER (LRC) FOR COMPARISONS AGAINST THE CRC AND LRC GENERATED BY THE HARDWARE IN NRZI READS OR WRITES.

THE FOLLOWING IS A LIST OF THE DATA PATTERNS AVAILABLE:

- DATA0: EXTERNAL INPUT THRU HIGH SPEED READER (SEE DTC)
- DATA1: ALL ONE BITS IN ALL CHARACTERS
- DATA2: ALL ZERO BITS IN ALL CHARACTERS
- DATA3: A ONE BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ZEROS
- DATA4: A ZERO BIT WALKING FROM RIGHT TO LEFT IN A FIELD OF ONES.
- DATA5: ALTERNATING ONE AND ZERO BITS IN EACH CHARACTER
- DATA6: ALTERNATING ZERO AND ONE BITS IN EACH CHARACTER
- DATA7: SAME AS DATA5 BUT WITH EVERY OTHER CHARACTER COMPLEMENTED
- DATA10: WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS
- DATA11: INCREMENTING CHARACTERS (000-377)
- DATA12: DECREMENTING CHARACTERS (377-000)
- DATA13: ALTERNATING CHARACTERS OF ALL ZERO AND ALL ONE BITS
- DATA14: WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS
- DATA15: AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0

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  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537

6. RANDOMIZATION  
-----

THERE ARE THREE (3) VALUES THAT MAY BE GENERATED RANDOMLY;  
DATA, CHARACTER COUNT, AND RECORD COUNT. THESE ARE NORMALLY SET TO  
SOME FIXED VALUE BUT MAY BE RANDOMIZED BY SETTING THE APPROPRIATE  
CONSOLE SWITCHES.

- A. RANDOM DATA: (CONSOLE SWITCH 8)  
GENERATES AN ENTIRE BUFFER, CHARACTER BY CHARACTER, OF RANDOM DATA WHEN SWITCH 8 IS SET TO A ONE. ONCE SET, THE RESETTING OF SWITCH 8 CAUSES THE LAST GENERATED PATTERN TO BE RETAINED IN CORE. A RESTART AT LOCATION 200(8) OR 210(8) WILL CAUSE REVERSION OF THE DATA TO THE FIXED PATTERN REQUESTED INITIALLY. A RESTART AT LOCATION 204(8) WILL HOLD THE LAST GENERATED PATTERN IN CORE UNTIL SWITCH 8 IS AGAIN SET.  
ALTHOUGH THE DATA IS GENERATED AS RANDOM, THE PROGRESSION OF RANDOM CHARACTERS IS ALWAYS THE SAME FROM THE OUTSET OF RANDOMIZATION. THEREFORE IT IS POSSIBLE TO GENERATE ONE TAPE REEL OF RANDOM DATA ON ONE UNIT, RESTART THE PROGRAM TO RE-ESTABLISH THE OUTSET POINT, AND READ THE RANDOM TAPE REEL ON ANOTHER UNIT FOR COMPATABILITY TESTING. IN MULTIDRIVE SYSTEMS THE SAME BLOCK OF DATA, WHETHER RANDOM OR FIXED, IS WRITTEN OR READ ON EACH AVAILABLE UNIT IN THE ORDER THAT THEY WERE ENTERED, BEFORE BEING CHANGED.
- B. RANDOM CHARACTER COUNT: (CONSOLE SWITCH 7)  
GENERATES A DIFFERENT NUMBER OF CHARACTERS PER RECORD TO BE WRITTEN ON EACH BLOCK CYCLE. THE SAME NUMBER OF CHARACTERS PER RECORD IS WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE BEING CHANGED. RESETTING SWITCH 7 HOLDS THE LAST VALUE GENERATED.
- C. RANDOM RECORD COUNT: (CONSOLE SWITCH 6)  
GENERATES A DIFFERENT NUMBER OF RECORDS FOR EACH BLOCK OF DATA WRITTEN OR READ ON EACH BLOCK CYCLE. THE SAME NUMBER OF RECORDS IS WRITTEN OR READ ON EACH AVAILABLE UNIT BEFORE BEING CHANGED. RESETTING SWITCH 6 HOLDS LAST VALUE GENERATED.

539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558

7. DYNAMIC PARAMETERS:  
-----

THE THREE (3) STALL VALUES ARE CONSIDERED TO BE DYNAMIC PARAMETERS AS THEY MAY BE CHANGED WHILE THE PROGRAM IS RUNNING BY TYPING A CONTROL B CHARACTER AT THE TELETYPE. AS SOON AS THE BUS IS RELEASED BY THE MAG TAPE OPERATION IN PROGRESS, THE PROGRAM WILL RESPOND TO THE CONTROL C INPUT BY TYPING A REQUEST FOR NEW STALL PARAMETERS. THE LAST VALUES THAT WERE ENTERED WILL BE PRINTED AS THE STORED VALUES AND MAY BE CHANGED BY ENTERING NEW VALUES OR LEFT UNCHANGED BY TYPING A CARRIAGE RETURN. THE YOZZLE STALL IS ALSO DYNAMIC AND CAN BE CHANGED BY TYPING A CONTROL B WHILE DOING A YOZZLE. A YOZZLE STALL REQUEST WILL BE PRINTED AND SHOULD BE RESPONDED TO WITH THE DESIRED VALUE.

560  
561  
562  
563  
564  
565  
566  
567  
568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601  
602  
603  
604  
605  
606  
607  
608  
609  
610  
611  
612  
613  
614  
615

8. CONSOLE SWITCH SETTINGS

CONTROL:

- 1) CONTROL G <^G>:  
SELECTS SOFTWARE SWR AND ALLOWS USER TO SELECT NEW SWITCHES.  
THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW=  
WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWR.  
AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE  
OF THE FOLLOWING AT THE TTY:  
A) TYPE A NUMBER TO BE LOADED INTO THE SOFTWARE SWR  
B) IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWR  
CONTENTS WILL NOT BE CHANGED.
- 2) CONTROL A <^A>:  
ALTERNATES USAGE OF THE SWR BETWEEN THE HARDWARE SWR & SOFTWARE SWR.
- 3) CONTROL B <^B>:  
SEE SECTION 7 DYNAMIC PARAMETERS
- 4) CONTROL U <^U>:  
DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

THE CONSOLE SWITCHES ARE USED TO SET UP THE TEST CYCLE  
DESIRED, TO GENERATE RANDOM VALUES, AND TO CONTROL ERROR  
RESPONSES. THE SWITCHES SHOULD BE SET IN THE DESIRED  
MANNER BEFORE PRESSING THE START SWITCH BECAUSE THEY  
ARE ALL DYNAMIC AND WILL RUN THE PROGRAM IN ANY  
CONFIGURATION. ALL SWITCHES SET TO ZERO(0) IS NORMAL.

- SW15: 1=STOP ON ERROR  
0=CONTINUE ON ERROR
- SW14: 1=PRINT READ/WRITE STATISTICS  
0=DO NOT PRINT STATS
- SW13: 1=DO NOT CHECK DATA ERRORS  
0=CHECK DATA ERRORS
- SW12: 1=DO NOT CHECK WRITE STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)  
0=CHECK WRITE STATUS ERRORS
- SW11: 1=DO NOT CHECK READ STATUS ERRORS (NOR CLEAR THEM IF THEY DO OCCUR)  
0=CHECK READ STATUS ERRORS
- SW10: 1=DO NOT PRINT ANY ERRORS (EXCEPT CATASTROPHIC ERRORS)  
0=PRINT ALL ERRORS
- SW9: 1=REWIND ALL AVAILABLE TAPES  
0=DO NOT REWIND
- SW8: 1=GENERATE RANDOM DATA  
0=USED FIXED DATA

616  
617  
618  
619  
620  
621  
622  
623  
624  
625  
626  
627  
628  
629  
630  
631  
632  
633  
634  
635  
636  
637  
638  
639  
640

SW7: 1=GENERATE RANDOM CHARACTER COUNT  
0=USE FIXED CHARACTER COUNT

SW6: 1=GENERATE RANDOM RECORD COUNT  
0=USED FIXED RECORD COUNT

SW5: 1=YOZZLE ON CURRENT RECORD  
0=DO NOT YOZZLE ON RECORD

SW4: 1=DO WRITE/READ RETRIES  
0=DO NOT RETRY

SW3: 1=DO NOT READ FORWARD  
0=READ FORWARD

SW2: 1=DO NOT READ REVERSE  
0=READ REVERSE

SW1: 1=READ FORWARD FIRST  
0=READ REVERSE FIRST

SW0: 1=DO NOT WRITE  
0=WRITE

642  
643  
644  
645  
646  
647  
648  
649  
650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669  
670  
671  
672  
673  
674  
675  
676  
677  
678  
679  
680  
681  
682  
683  
684  
685  
686  
687

SWITCH EXPLANATION AND EXAMPLES:

SW0-3:

THESE SWITCHES ARE USED TO CONTROL THE SEQUENCE OF MAG TAPE OPERATIONS PREFORMED ON EACH AVAILABLE UNIT. THE BLOCK OF DATA DESCRIBED THROUGH THE RESPONSES TO TELETYPE REQUESTS AT INITIAL START WILL BE EITHER WRITTEN OR READ FROM EACH AVAILABLE UNIT IN THE ORDER THAT THEY WERE ENTERED. THE SEQUENCE OF OPERATIONS IS CALLED A CYCLE, AND WILL BE PERFORMED CONTINUOUSLY UNTIL STOPPED BY THE OPERATOR. WHEN END OF TAPE IS REACHED, THE UNIT WILL BE REWOUND AND FLAGGED AS UNAVAILABLE FOR TEST UNTIL ALL UNITS HAVE REACH EOT, AT WHICH TIME TESTING IS RESUMED ON ALL AVAILABLE UNITS.

EXAMPLES: 0-3

- A. SW0=0,SW1=0,SW2=1,SW3=1  
WRITE ONLY X RECORDS OF Y CHARACTERS
- B. SW0=0,SW1=0,SW2=1,SW3=0  
WRITE THEN BACKSPACE AND READ FORWARD X RECORDS
- C. SW0=0,SW1=0,SW2=0,SW3=1  
WRITE THEN READ REVERSE X RECORDS.
- D. SW0=0,SW1=0,SW2=0,SW3=0  
WRITE THEN READ REVERSE AND READ FORWARD X RECORDS
- E. SW0=0,SW1=1,SW2=0,SW3=0  
WRITE THEN BACKSPACE AND READ FORWARD THEN REVERSE
- F. SW0=1,SW1=0,SW2=1,SW3=0  
READ TAPE FORWARD X RECORDS
- G. SW0=1,SW1=0,SW2=0,SW3=1  
READ TAPE REVERSE X RECORDS
- H. SW0=1,SW1=0,SW2=0,SW3=0  
READ TAPE REVERSE THEN FORWARD
- I. SW0=1,SW1=1,SW2=0,SW3=0  
READ TAPE FORWARD THEN REVERSE



689  
690  
691  
692  
693  
694  
695  
696  
697  
698  
699  
700  
701  
702  
703  
704  
705  
706  
707  
708  
709  
710  
711  
712  
713  
714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736  
737  
738  
739  
740  
741  
742

SW4:

SWITCH FOUR (4), WHEN SET TO A ONE (1), WILL CAUSE ANY DATA RELATED ERROR TO BE RETRIED. THE WRITE RETRY SCHEME CONSISTS OF REWRITING THE RECORD IN THE SAME SPOT ON TAPE FOUR (4) TIMES. IF ALL FOUR (4) REPEATS ARE SUCCESSFUL, THE RECORD IS CONSIDERED AS RECOVERED, AND A TAPE WRITE ERROR IS LOGGED. IF ANY OF THE FOUR (4) REPEATS IS UNSUCCESSFUL, A SKIP ERASE IS DONE, A SUPECTED BAD TAPE SPOT IS LOGGED AT THIS BLOCK AND RECORD NUMBER, AND A SECOND RETRY OF FOUR REPEATS IS DONE. IF AFTER FOUR (4) RETRIES, THE RECORD CANNOT BE RECOVERED A NOTIFICATION IS PRINTED, AND TESTING IS RESUMED ON THE NEXT RECORD. IF 20(8) BAD TAPE SPOTS ARE FOUND, THE SLAVE WILL BE REWOUND AND REMOVED FROM TESTING WITH AN APPROPRIATE MESSAGE PRINTED. THE READ RETRY SCHEME CONSISTS OF REREADING THE RECORD UP TO EIGHT TIMES. IF ALL EIGHT REREADS ARE BAD, IT IS A HARD ERROR. IF ANY REREAD IS SUCCESSFUL, THIS IS A SOFT ERROR. IF THE ORIGINAL ERROR IS OF THE NON-RETRYABLE TYPE (IE: ILF,RMR,ILR,NEF,CBUSPE), THE RETRY SCHEME IS NOT ENTERED AND A MESSAGE IS PRINTED.

SW5:

SWITCH FIVE (5) WHEN SET DURING A READ FORWARD OR REVERSE WILL CAUSE THE TAPE TO CONTINUOUSLY READ THE CURRENT RECORD BY SPACING EITHER FORWARD OR REVERSE AND REREADING THAT RECORD. THIS TAPE MOVEMENT IS CALLED YOZZLING. THERE IS A SOFTWARE DELAY EXECUTED BETWEEN EACH SPACE/READ OF THE RECORD AND IT MAY BE VARIED BY TYPING CONTROL C ON THE TELETYPE DURING THE EXECUTION OF THE YOZZLE AND RESPONDING TO THE PRINTED REQUEST WITH A SIX (6) DIGIT VALUE. THE YOZZLE STALL IS PRESET TO A VALUE OF 3000 IN THE PROGRAM TO PREVENT EXCESSIVE TAPE WEAR, BUT MAY BE SET TO ANY VALUE THROUGH THE TELETYPE.

SW6-8:

THESE THREE (3) SWITCHES CONTROL THE RANDOMIZATION OF DATA AND BLOCK SIZE AND MAY BE SET AND RESET AT ANY TIME. THE ACTUAL CHANGE WILL TAKE PLACE BETWEEN BLOCK CYCLES.

SW9:

SWITCH NINE (9) WHEN SET WILL CAUSE ALL AVAILABLE TAPE UNITS TO BE REWOUND AT THE END OF THE CURRENT BLOCK CYCLE. TESTING WILL BE RESUMED AT A BLOCK COUNT OF ONE (1) WHEN ALL UNITS HAVE REACHED BOT.

744  
745  
746  
747  
748  
749  
750  
751  
752  
753  
754  
755  
756  
757  
758  
759  
760  
761  
762  
763  
764  
765  
766  
767  
768  
769  
770  
771  
772  
773  
774  
775  
776  
777  
778  
779  
780  
781  
782  
783  
784  
785  
786  
787  
788  
789  
790

SW10-13:

THESE SWITCHES ARE USED TO CONTROL THE ERROR HANDLING TO BE DONE ON THE TAPE OPERATION DESCRIBED BY SWITCHES 0-3.

- A. SWITCH TEN (10) WHEN SET TO A ONE WILL DISALLOW ANY ERROR PRINTOUTS MADE ON THE OPERATION IN PROGRESS. CATASTROPHIC FAILURES AND INFORMATION PRINTOUTS WILL STILL OCCUR. IE: UNIT NOT AVAILABLE, ILLEGAL BOT, DROP OR PICK OVERFLOW, AND EOT REWIND.
- B. SWITCH ELEVEN (11) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON READ (FORWARD OR REVERSE) OPERATIONS.
- C. SWITCH TWELVE (12) WHEN SET TO A ONE WILL DISALLOW THE CHECKING FOR STATUS ERRORS ON WRITE OPERATIONS.
- D. SWITCH THIRTEEN (13) WHEN SET TO A ONE WILL DISALLOW THE CHECKING OF READ DATA. THIS SWITCH HAS NO EFFECT ON STATUS CHECKING.

\*\*NOTE THAT WHEN SW11 OR 12 ARE SET, NOT ONLY ARE ERRORS NOT CHECKED, BU  
\*\*\*THEREFOR USE CAUTION TO ASSURE THAT OPERATIONS ARE NOT UNEXECUTED DUE  
\*\*\*\*DO NOT SET SW 11 OR 12 TO A ONE (1), DURING A RETRY SEQUENCE.

SW14:

SWITCH FOURTEEN (14) WHEN SET TO A ONE (1) WILL PRINT THE ACCUMULATED READ/WRITE STATISTICS FOR THE SELECTED SLAVE UNDER TEST AT THE END OF THE CURRENT BLOCK CYCLE. THE STATISTICS PRINTED ARE THE NUMBER OF BITS DROPPED OR PICKED, THE NUMBER OF RETRIES, WRITE ERRORS, READ ERRORS, AND DATA ERRORS.

SW15:

SWITCH FIFTEEN (15) WHEN SET TO A ONE, WILL CAUSE THE PROGRAM TO HALT ON ANY ERROR DETECTED BY THE OPERATION IN PROGRESS. IF BOTH SWITCH TEN (10) AND FIFTEEN (15) ARE SET, THE ACTUAL ERROR DETECTED WILL NOT BE PRINTED BUT WILL CAUSE A HALT. IF SWITCH TEN (10) IS RESET BEFORE PRESSING CONTINUE, THE ERROR WHICH CAUSED THE HALT WILL BE PRINTED BEFORE TESTING IS RESUMED.

792  
793  
794  
795  
796  
797  
798  
799  
800  
801  
802  
803  
804  
805  
806  
807  
808  
809  
810  
811  
812  
813  
814  
815  
816  
817  
818  
819  
820  
821  
822  
823  
824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841

9. ERROR PRINTOUTS  
-----

THERE ARE THREE TYPES OF ERROR PRINTOUTS MADE BY THE PROGRAM; OPERATION ERRORS, DATA ERRORS, AND CONDITION ERRORS. EACH ERROR MESSAGE PRINTED IS PROCEEDED BY A TWO LINE HEADER WHICH CONTAINS THE DRIVE NUMBER, SLAVE NUMBER, DENSITY, PARITY, AND FORMAT ON THE FIRST LINE, AND THE BLOCK NUMBER, RECORD NUMBER, RECORD SIZE, AND ERROR TYPE ON THE SECOND.

A. OPERATION ERRORS:

THESE ARE ERRORS WHICH CAN OCCUR AS A DIRECT RESULT OF A TAPE OPERATION.

1. READ/WRITE STATUS ERRORS: THESE ARE DETECTED BY EITHER THE TMO3 ITSELF OR BY THE MASSBUS CONTROLLER. ALL STATUS ERRORS WILL BE REPORTED.
2. TAPE POSITION ERRORS: THESE ARE INDICATED BY AN INCORRECT SPACE OR REWIND OPERATION IN WHICH TAPE POSITION BECOMES UNRELIABLE.

B. DATA ERRORS:

DATA ERRORS WILL OCCUR WHEN TAPE IS BEING READ AND THE DATA FROM TAPE DOES NOT MATCH THE EXPECTED DATA. WHEN READING IN THE REVERSE DIRECTION, THE RECORD NUMBERS WILL BE COUNTED DOWN FROM LAST TO FIRST. THE CHARACTER NUMBERS IN REVERSE READS WILL ALSO BE COUNTED DOWN IN ORDER TO REFLECT TAPE POSITION RATHER THAN THE ORDER TRANSFERRED.

BECAUSE DATA RECORDS CAN BE UP TO FOUR THOUSAND CHARACTERS LONG, AN ERROR CONDITION WHICH WILL CAUSE THE ENTIRE RECORD TO READ INCORRECTLY COULD CAUSE A VERY LENGTHY PRINTOUT. THEREFORE, A COUNTER OF SUCCESSIVE BAD CHARACTERS IS EMPLOYED. IF TEN (10) CHARACTERS IN SUCCESSION ARE BAD, A NOTIFICATION IS PRINTED (BAD RECORD) AND THE NEXT TWENTY FIVE (25) CHARACTERS ARE SKIPPED BEFORE CHECKING IS RESUMED. IF THE BAD RECORD CONDITION OCCURS THREE (3) TIMES IN ONE RECORD, THE REST OF THE RECORD IS SKIPPED, DOWN TO THE LAST TEN (10) CHARACTERS WHICH WILL BE CHECKED. THE SKIPPING AND RESUMPTION OF CHECKING WILL ONLY BE DONE ON RECORDS WHICH ARE LONG ENOUGH TO ALLOW IT.

843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870  
871  
872  
873  
874  
875  
876  
877  
878  
879  
880  
881  
882  
883  
884  
885  
886  
887  
888  
889  
890  
891  
892  
893  
894  
895  
896  
897  
898

C. CONDITION ERRORS: (CATASTROPHIC)

THESE PRINTOUTS REFLECT THE STATE OF THE TAPE SYSTEM  
EITHER BEFORE OR AFTER AN OPERATION

1. EOT: WHEN EOT (END OF TAPE) IS ENCOUNTERED DURING  
EITHER A READ OR WRITE, THE CYCLE IS COMPLETED  
ON THE SHORTENED BLOCK AFTER WHICH THE SLAVE  
WILL BE REWOUND AND FLAGGED AS UNAVAILABLE  
FOR TESTING UNTIL ALL SLAVES HAVE REACHED EOT AND  
ARE REWOUND. WHEN THE LAST AVAILABLE SLAVE  
HAS REACHED EOT AND BEEN REWOUND TO BOT,  
TESTING WILL BE RESUMED ON ALL SLAVES.
2. ILLEGAL BOT: WHEN A SLAVE ENCOUNTERS BOT DURING  
A READ, WRITE, OR SPACE OPERATION, AN ERROR  
IS PRINTED AND THE PROGRAM HALTED. THIS IS  
A CATASTROPHIC ERROR. TESTING MAY BE RESUMED  
BY PRESSING CONTINUE; BUT A RESTART IS  
SUGGESTED.
3. NO INTERRUPT RETURNED: EACH TAPE OPERATION SHOULD BE  
TERMINATED BY THE SETTING OF AN INTERRUPT IN  
THE CPU. IF NO INTERRUPT IS RETURNED WITHIN  
THE APPROPRIATE TIME, AN ERROR IS PRINTED.
4. NO MEDIUM ON-LINE: BEFORE AN OPERATION IS ATTEMPTED,  
THE TM03 IS CHECKED FOR MOL. IF IT IS NOT  
SET, AN ERROR IS PRINTED, AND THE PROGRAM STOPPED.  
TESTING MAY BE RESUMED BY PRESSING CONTINUE.
5. NO BOT ON REWIND: AS EACH SLAVE IS REWOUND A CHECK  
IS MADE TO ASSURE THAT PROPER POSITION AT BOT  
IS ESTABLISHED. IF BOT IS NOT SET UPON COMPLETION OF  
A REWIND, AN ERROR IS PRINTED AND THE PROGRAM  
WILL HALT. PRESS CONTINUE TO RESUME TESTING.
6. POSITION ERROR: IF POSITION IS LOST DURING A RETRY,  
A MESSAGE IS PRINTED, THE TAPE REWOUND,  
AND REMOVED FROM TESTING UNTIL ALL ARE  
RESTARTED AT BLOCK ONE.
7. BAD TAPE OVERFLOW: IF 20(8) BAD TAPE SPOTS ARE FOUND,  
A MESSAGE IS PRINTED, THE TAPE REWOUND,  
AND REMOVED FROM TESTING UNTIL ALL ARE  
RESTARTED AT BLOCK ONE.
8. HARD READ ERROR: IF ANY HARD READ ERROR IS ENCOUNTERED  
DURING A RETRY, A MESSAGE IS PRINTED  
REGARDLESS OF THE SETTING OF SW10.
9. NON-RETRYABLE: IF ANY NON-RETRYABLE ERROR IS ENCOUNTERED, A  
MESSAGE IS PRINTED REGARDLESS OF THE SETTING OF SW10.

900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931

D. EXAMPLES:

GLOSSARY:

BN = CURRENT BLOCK NUMBER  
RN = CURRENT RECORD NUMBER  
RS = RECORD SIZE, IN FRAMES  
WE = WRITE STATUS ERROR  
RE = READ STATUS ERROR  
SE = SPACE ERROR  
TM = TAPE MARK  
F = FORWARD  
R = REVERSE  
CS1 = RH/TE16 CONTROL REGISTER  
WC = RH WORD COUNT  
BA = RH BUS ADDRESS  
FC = TE16 FRAME COUNT  
CS2 = RH CONTROLLER STATUS  
DS = TE16 DRIVE STATUS  
ER = TE16 ERROR REGISTER  
AS = ATTENTION SUMMARY  
CK = TE16 CHECK CHARACTER  
DB = RH DATA BUFFER  
MR = TE16 MAINTENANCE REGISTER  
DT = TE16 DRIVE TYPE  
SN = TE16 SERIAL NUMBER  
TC = TE16 TEST CONTROL  
\*F = DATA FORMAT  
\*P = PARITY  
\*D = DENSITY  
\*PATRN = DATA PATTERN NUMBER (R = RANDOM)

933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954  
955  
956  
957  
958  
959  
960  
961  
962  
963  
964  
965  
966  
967  
968  
969  
970  
971  
972  
973  
974  
975  
976  
977  
978

EXAMPLE 1: IN THIS EXAMPLE SLAVE 1 ON TMO3 0 WAS OPERATING AT 1600 BPI IN ODD PARITY USING THE NINE CHANNEL NORMAL DATA FORMAT. A WRITE STATUS ERROR WAS DETECTED. THE BAD STATUS INDICATES THAT AN UNCORRECTABLE DATA ERROR (BIT 6 OF ER) AND A PE FORMAT ERROR (BIT 7 OF ER) OCCURED DURING THE WRITE OPERATION OF THE SIXTH (6) RECORD OF THE FIFTY (50) RECORDS IN BLOCK (2). THE SIZE OF THE RECORD WAS TWO HUNDRED (200) FRAMES. THE CHECK CHARACTER REFLECTS THE BAD TRACK.

DRIVE NO. 0 \*SLAVE NO. 1 \*D 4 \*P C \*F 14 \*PATRN 1  
\*BN 2 \*RN 6-50 \*RS = 200 \*WE  
CS1 144260  
CS2 100  
DS 150640  
ER 300  
WC 0  
CK 4

EXAMPLE 2: IN THIS EXAMPLE SLAVE 3 ON TMO3 1 WAS OPERATING AT 800 BPI IN EVEN PARITY USING THE NINE CHANNEL NORMAL DATA FORMAT. A READ STATUS ERROR WAS DETECTED DURING THE REVERSE READ OF THE TENTH (10) RECORD OF THE 25 RECORDS IN THIS BLOCK (12). THE SIZE OF THE RECORD IS TWENTY (20) FRAMES. THE PRINTOUT INDICATES THE DETECTION OF A VERTICAL PARITY ERROR (VPE: BIT 6 OF ER) AND A CYCLIC REDUNDENCY ERROR (CRC: BIT 15 OF ER). THE CRC CHARACTER, AS RECEIVED, IS NOT AS EXPECTED AND IS PRINTED SHOWING BOTH THE ACTUAL (FIRST) AND THE EXPECTED (LAST).

DRIVE NO. 2 \*SLAVE NO. 3 \*D 3 \*P 1 \*F 14 \*PATRN 3  
\*BN 12 \*RN 10-25 \*RS 20 \*RE R  
CS1 144276  
CS2 100  
DS 150600  
ER 100100  
WC 0  
CRC 767-777

980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990  
991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025

EXAMPLE 3: IN THIS EXAMPLE, THE HEADER IS THE SAME AS  
IN EXAMPLE TWO (2) EXCEPT THAT THE ERROR TYPE  
REFLECTS A READ ERROR IN THE FORWARD  
DIRECTION. IT IS NORMAL FOR THE SYSTEM  
TO DETECT AN ERROR IN THE FORWARD AND  
REVERSE DIRECTION AT THE SAME RECORD.  
REMEMBER THAT IN REVERSE OPERATIONS THE  
RECORD NUMBER IS COUNTED DOWN SO THAT  
RECORD NUMBER TEN (10) WILL SHOWN IN  
THE PROPER POSITION IN BOTH FORWARD AND  
REVERSE.

DRIVE NO. 2 \*SLAVE NO. 3 \*D 3 \*P 1 \*F 14 \*PATRN 2  
\*BN 12 \*RN 10-25 \*RS 20 \*RE F  
CS1 144270  
CS2 100  
DS 150600  
ER 100100  
WC 0  
CRC 767-777

EXAMPLE 4: IN EXAMPLES 2 AND 3 THE READ OPERATION  
RESULTED IN BAD STATUS, HOWEVER THE  
DATA ASSOCIATED WITH THE OPERATION WAS  
NOT BAD (OR WAS NOT CHECKED: SW 13=1).  
THIS EXAMPLE (4) SHOWS A PRINTOUT REFLECTING  
A READ STATUS ERROR ACCOMPANIED BY BAD  
DATA IN CHARACTERS FOUR (4) AND SIX (6).

DRIVE NO. 2 \*SLAVE NO. 3 \*D 3 \*P 1 \*F 14 \*PATRN 2  
\*BN 12 \*RN 10-25 \*RS 20 \*RE F  
CS1 144270  
CS2 100  
DS 150600  
ER 100100  
WC 0  
CRC 767-777  
CN 4  
G 11111111  
B 10111111  
CN 6  
G 11111111  
B 10111111

1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044  
1045  
1046  
1047  
1048  
1049  
1050  
1051  
1052  
1053  
1054  
1055  
1056  
1057  
1058  
1059  
1060  
1061  
1062  
1063  
1064  
1065  
1066  
1067  
1068  
1069  
1070  
1071

EXAMPLE 5: THIS EXAMPLE SHOWS A READ DATA ERROR WHICH OCCURRED, WITHOUT AN ACCOMPANING STATUS ERROR, WHICH RESULTED IN A BAD RECORD.

DRIVE NO. 3 \*SLAVE NO. 1 \*D 4 \*P 0 \*F 14 \*PATRN R  
\*BN 100 \*RN 66-200 \*RS 2000 \*DE F

CN 0  
G 11111111  
B 00000000  
CN 1  
G 11111111  
B 00000000  
CN 2  
G 11111111  
B 00000000  
CN 3  
G 11111111  
B 00000000  
CN 4  
G 11111111  
B 00000000  
CN 5  
G 11111111  
B 00000000  
CN 6  
G 11111111  
B 00000000  
CN 7  
G 11111111  
B 00000000

BAD RECORD

EXAMPLE 6: THE FOLLOWING EXAMPLE SHOWS THE RESULT OF A SPACE OPERATION THAT SHOULD HAVE SPACED REVERSE OVER AN ENTIRE 100 RECORD BLOCK BUT WHICH TERMINATED AT THE END OF 40 RECORDS. LEAVING A POSITION ERROR OF 40

DRIVE NO. 2 \*SLAVE NO. 6 \*D 2 \*P 0 \*F 14  
\*BN 3 \*RN 100-100 \*RS 1000 \*SE R  
ERR AMT 40



1073  
1074  
1075  
1076  
1077  
1078  
1079  
1080  
1081  
1082  
1083  
1084  
1085  
1086  
1087  
1088  
1089  
1090  
1091  
1092  
1093  
1094  
1095  
1096  
1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120

EXAMPLE 7: THIS EXAMPLE REFLECTS AN ERROR DETECTED WHILE WRITING A TAPE MARK (TM) AT THE END OF THE CURRENT DATA BLOCK PER OPTION RESPONSE TM=1. NOTE THAT THE TM RECORD NUMBER IS ONE GREATER THAN THE TOTAL NUMBER OF DATA RECORDS IN THE CURRENT BLOCK.

DRIVE NO. 1 \*SLAVE NO. 1 \*D 2 \*P 0 \*F 14  
\*BN 67 \*RN 101-100 \*RS 36 \*WE TM  
CS1 144226  
CS2 300  
DS 150604  
ER 1000  
WC 0

EXAMPLE 8: THIS EXAMPLE SHOWS TWO (2) PRINTOUTS REFLECTING A WRITE RETRY WHICH WAS NOT SUCCESSFUL THE FIRST TIME, BUT WHICH DID RECOVER ON THE SECOND. THE UNSUCCESSFUL RETRY IS LOGGED AS A SUSPECTED BAD TAPE SPOT BY ITS BLOCK AND RECORD NUMBER.

DRIVE NO. 0 \*SLAVE NO. 2 \*D 4 \*P 0 \*F 14 \*PATRN 5  
\*BN 2 \*RN 12-20 \*RS 667 \*WE  
CS1 144260  
CS2 100  
DS 150640  
ER 100  
WC 0  
\*\*\*ORIGINAL ERROR\*\*\*

DRIVE NO. 0 SLAVE NO. 2 \*D 4 \*P 0 \*F 14 \*PATRN 6  
\*BN 2 \*RN 12-20 \*RS 667 \*WE  
CS1 144260  
CS2 100  
DS 150640  
ER 100  
WC 0  
SUSPECT BAD TAPE  
RETRY: 0  
REPT: 0  
RECOVERED  
RETRY: 1

1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152  
1153  
1154  
1155  
1156

EXAMPLE 9: IF , DURING A WRITE RETRY THE BACKSPACE OR THE ERASE OPERATION RESULT IN AN ERROR, THE ERROR WILL BE PRINTED AND THE PROGRAM HALTED. THIS EXAMPLE SHOWS THE ERROR PRINT FOR A SPACE AND AN ERASE (2 EXAMPLES)

DRIVE NO. 1 \*SLAVE NO. 1 \*D 3 \*P 0 \*F 14  
BN 12 \*RN 8-64 \*RS 500 \*SE RTRY  
ERR AMT 1

DRIVE NO. 1 \*SLAVE NO. 1 \*D 3 \*P 0 \*F 14  
\*BN 12 \*RN 8-64 \*RS 500 \*ERASE  
CS1 144224  
CS2 100  
DS 150600  
ER 400  
WC 0

EXAMPLE 10: THIS EXAMPLE SHOWS THE PRINTOUT FROM A REWIND OPERATION WHICH DOES NOT HAVE BOT SET AT THE END.

DRIVE NO. 2 \*SLAVE NO. 3 \*D 3 \*P 0 \*F 14  
\*BN 66 \*RN 15-20 \*RS 1000  
NOT BOT ON REWIND: HALT

EXAMPLE 11: THIS EXAMPLE SHOWS THE PRINTOUT MADE WHEN THERE IS NO INTERRUPT RETURNED AT THE END OF AN OPERATION.

DRIVE NO. 7 \*SLAVE NO. 7 \*D 2 \*P 1 \*F 14  
\*BN 1 \*RN 25-26 \*RS 1200  
NO INTERRUPT

1158  
1159  
1160  
1161  
1162  
1163  
1164  
1165  
1166  
1167  
1168  
1169  
1170  
1171  
1172  
1173  
1174  
1175  
1176  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207

10. STATISTICS PRINTOUT  
-----

THE PROGRAM, THROUGH ITS ERROR CHECKING, IS ABLE TO GATHER CERTAIN STATISTICS ABOUT THE PERFORMANCE OF EACH UNIT UNDER TEST. THIS INFORMATION IS PRINTED OUT WHENEVER A UNIT IS REWOUND FROM END OF TAPE, OR BECAUSE IT IS TO BE REMOVED FROM TESTING DUE TO SOME CATASTROPHIC ERROR. (POSITION LOST, BAD TAPE OVERFLOW) THE STATISTICS MAY BE PRINTED AT ANY TIME BY SETTING SWITCH 14 TO A ONE (1). THIS PRESENTS A PICTURE OF PERFORMANCE UP TO THIS TIME. THE STATISTICS WILL BE CLEARED UPON REWIND OF THE UNIT; BUT NOT BY SETTING SW 14.

STATISTICS PRINT EXAMPLE (A HEADER WILL PRECEED THE STATS)

DROPS: 0 3 0 0 0 6 45 0  
PICKS: 1 0 0 0 0 0 0 2  
RETRY: 1  
WTERR: 2  
REFWD: 3  
SOFT: 2  
HARD: 1  
DEFWD: 0  
REREV: 4  
SOFT: 1  
HARD: 3  
DEREV: 0  
2 BAD TAPE SPOTS  
0 \*BN 1 \*RN 2  
1 \*BN 15 \*RN 100

\*\* NOTE \*\* DROPS AND PICKS REFLECT CORE BIT POSITIONS.  
THE FOLLOWING IS A TABLE OF CORE BITS TO TRACK NUMBER.

TRACK NO.	7	6	5	3	9	1	8	2
CORE BIT	7	6	5	4	3	2	1	0

DROPS: NUMBER OF DATA BITS DROPPED: PER CORE BIT(SEE NOTE ABOVE)  
PICKS: NUMBER OF DATA BITS PICKED UP: PER CORE BIT(SEE NOTE ABOVE)  
RETRY: NUMBER OF WRITE RETRIES  
WTERR: NUMBER OF WRITE ERRORS NOT ASSOCIATED WITH BAD TAPE  
REFWD: NUMBER OF READ FORWARD STATUS ERRORS  
REREV: NUMBER OF READ REVERSE STATUS ERRORS  
SOFT: NUMBER OF RECOVERED READ ERRORS  
HARD: NUMBER OF UNRECOVERED READ ERRORS  
DEFWD: NUMBER OF FORWARD DATA ERRORS WITH NO ASSOCIATED STATUS ERROR  
DEREV: NUMBER OF REVERSE DATA ERRORS WITH NO ASSOCIATED STATUS ERROR

1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250

11. AUTO SEQUENCE  
-----

THE AUTO SEQUENCE (START AT ADDRESS 240) WILL EXECUTE A  
PREDETERMINED TEST PLAN ON ALL AVAILABLE SLAVES ON EACH  
AVAILABLE TMO3. THE ONLY OPERATOR RESPONSE IS TO THE TYPED  
REQUESTS FOR THE RH ADDRESS, VECTOR, CONTINUOUS OR SINGLE  
CYCLE, AND NRZ ONLY. ALL SWITCHES REMAIN ACTIVE AND MAY BE  
USED NORMALLY; HOWEVER THE IDEA IS TO LEAVE ALL SWITCHES  
DOWN AND ALLOW FULL EXECUTION OF THE TEST PLAN FOR  
SYSTEM CHECKOUT.

SAMPLE START AT 240(8): AUTO SEQUENCE.

LOAD ADDRESS 240(8), SET SWITCHES TO ZERO, PRESS START:

TE16 AUTO SEQUENCE TEST  
ENTER CONDITIONS IN OCTAL

REGISTER START = 172400(172440)  
VECTOR ADDRESS = 224(CR)  
NRZ ONLY: (0)  
AUTO CONT: (1)

THIS EXAMPLE SHOWS AN AUTO SEQUENCE START WITH THE RH  
AT BUS ADDRESS 172440 AND A VECTOR OF 224. ALL AVAILABLE  
HARDWARE WILL BE TESTED CONTINUOUSLY IN BOTH NRZ AND PE MODE.

AS EACH TMO3 AND ITS SLAVES ARE FOUND, A DIVIDER LINE OF  
ASTERICKS WILL BE PRINTED FOLLOWED BY A PRINTOUT OF THE  
TMO3 AND ITS SLAVES BEING TESTED. AS EACH TMO3 AND  
ITS SLAVES ARE FINISHED, ANOTHER DIVIDER IS PRINTED  
BEFORE TESTING IS RESUMED ON THE NEXT AVAILABLE DRIVE.

WHEN ALL AVAILABLE HARDWARE HAS BEEN TESTED,  
A PRINTOUT OF END OF SEQUENCE WILL BE DONE AND THE  
PROGRAM WILL EITHER HALT (AUTO CONT = 0) OR RESTART WITH  
THE FIRST AVAILABLE UNIT (AUTO CONT = 1).

1252  
1253  
1254  
1255  
1256  
1257  
1258  
1259  
1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284

AUTO SEQUENCE TEST PLAN.

THE AUTO SEQUENCE WILL EXECUTE BOTH AN NRZ AND A PE CYCLE. EACH CYCLE WILL BE STARTED FROM BOT AND CONSIST OF VARIOUS DATA PATTERNS INTENDED TO BE WORST CASE FOR THAT PARTICULAR MODE.

1. NRZ CYCLE:

SIX (6) BLOCKS OF ONE HUNDRED (100) RECORDS OF FOUR THOUSAND (4000) CHARACTERS FOR EACH OF THE FOUR DATA PATTERNS.

PATTERN 1: ALL ONES DATA IN ALL BYTES  
PATTERN 10: WALKING ONE/ALL ONE  
PATTERN 14: WALKING ZERO/ALL ZERO  
RANDOM DATA: RANDOM

2. PE CYCLE: (IF NRZ ONLY = 0)

SIX BLOCKS OF ONE HUNDRED (100) RECORDS OF FOUR THOUSAND (4000) CHARACTERS EACH FOR EACH OF THREE DATA PATTERNS, THEN RANDOM DATA BLOCKS TO END OF TAPE.

PATTERN 10: WALKING ONE/ALL ONE  
PATTERN 14: WALKING ZERO/ALL ZERO  
PATTERN 15: THREE (3) 0 CHARACTERS, TWO (2) ALL CHARACTERS, THREE 0 THEN COMPLIMENT PATTERN. REPEATED FOR A FULL BUFFER  
RANDOM DATA: RANDOM

1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307  
1308  
1309  
1310  
1311  
1312  
1313  
1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333

12. TESTING PROCEDURES

-----  
AS PREVIOUSLY STATED THIS PROGRAM CONTAINS NO FIXED TESTS. THE ENTIRE TEST CYCLE TO BE EXECUTED IS DESCRIBED BY THE OPERATOR THROUGH RESPONSES TO TELETYPE REQUESTS FOR PARAMETERS AND CONSOLE SWITCH SETTINGS FOR OPERATION. THE OPERATION SELECTED WILL BE EXECUTED WITH THE PARAMETERS ENTERED CONTINUOUSLY ON EACH AVAILABLE UNIT, ONE BLOCK AT A TIME, UNTIL STOPPED BY THE OPERATOR. THE OPERATION MAY BE CHANGED DYNAMICALLY BY CHANGING THE CONSOLE SWITCHES AT ANY TIME. THE PROGRAM WILL ATTEMPT TO PERFORM ANY OPERATION SET AND THEREFORE CAUTION SHOULD BE TAKEN TO ASSURE THAT THE UNIT IS CAPABLE OF PERFORMING AS REQUESTED. FOR INSTANCE, ONE SHOULD NOT ATTEMPT TO PERFORM READ OPERATIONS ON A TAPE WHICH HAS NOT BEEN WRITTEN AS THE DATA, IF ANY, IS UNPREDICTABLE. HOWEVER, IF A TAPE HAS BEEN WRITTEN WITH THIS PROGRAM, IT CAN BE READ AS OFTEN AS DESIRED WITHOUT BEING REWRITTEN. THIS IS A GOOD PROCEDURE TO USE FOR TESTING TAPE COMPATABILITY. SCOPING OF TAPE UNITS BECOMES SIMPLE; BY SETTING THE DESIRED OPERATION AND ITS PARAMETER, A UNIT MAY BE CONTINUOUSLY EXERCISED IN ANY MANNER DESIRED. BY USING THE VARIOUS ERROR CONTROL SWITCHES AND ENTERING THE NEEDED STALL, ANY FUNCTION CAN BE SCOPED RATHER EASILY. RELIABILITY TESTING CAN BE PERFORMED BY USE OF THE RANDOMIZATION CAPABILITY. PERHAPS A CYCLE OF RANDOM TESTING MIGHT BE SET UP AND ALLOWED TO RUN FOR SOME PERIOD OF TIME, THE STATISTICAL COLLECTION OF DROPS AND PICKS IS THEN SIGNIFICANT. INTERMITTANT PROBLEMS CAN BE FOUND BY SETTING THE DESIRED OPERATION IN MOTION AND DISALLOWING ERROR PRINTOUTS WHILE ALLOWING A HALT ON ERROR. THE ERROR THAT CAUSED THE HALT CAN BE PRINTED BY RESETTING CONSOLE SWITCH TEN AND PRESSING CONTINUE. IF SOME PARTICULAR DATA PATTERN SHOULD BE CAUSING DATA ERROR, USE OF THE YOZZLE SWITCH AND ITS ASSOCIATED STALL WILL TO ALLOW SCOPING OF THIS PARTICULAR RECORD.

AS YOU SEE, THERE ARE MYRIAD TESTING PROCEDURES WHICH COULD BE PERFORMED. THE PARAMETERS, TAPE OPERATIONS, ERROR EXAMINATION AND REPORTING ARE ALL AT YOUR DISCRETION.

TRY IT, YOU'LL LIKE IT.

z

```
1335 .LIST BIN,LOC,SEQ
1336 .TITLE CZTEDDO TM03-TE16/TU77 DRT
1337 :DATA RELIABILITY TEST
1338 :AC-A800D-MC
1339 :21 FEB 1977
1340 :J.G.ADAMS
1341
1342 :REVISED (++) J.G.ADAMS MAY 1978
1343 :++B
1344 :++B
1345 :++B
1346 :++B
1347 :++B
1348
1349 :(++C) M.PAGE FEB 79
1350 :++C
1351 :
1352 :
1353
1354 :REVISED MAY 1983 BY B. LEBLANC
1355
1356 .MCALL .SACT11,.$EOP,$SAVE,$RESTORE,$CHAIN
1357 .NLIST MC
1358 .LIST ME
1359 .ENABLE ABS,AMA
1360
1361 :CONSOLE SWITCHES*****
1362
1363 :SW15: 1=STOP ON ERROR
1364 :      0=CONTINUE ON ERROR
1365 :SW14: 1=PRINT READ/WRITE STATS
1366 :      0=DO NOT PRINT STATS
1367 :SW13: 1=DO NOT CHECK DATA
1368 :      0=CHECK DATA
1369 :SW12: 1=DO NOT CHECK WRITE ERRORS
1370 :      0=CHECK WRITE ERRORS
1371 :SW11: 1=DO NOT CHECK READ ERRORS
1372 :      0=CHECK READ ERRORS
1373 :SW10: 1=DO NOT PRINT ERRORS
1374 :      0=PRINT ERRORS
1375 :SW9:  1=REWIND TAPE
1376 :      0=DO NOT REWIND
1377 :SW8:  1=USE RANDOM DATA
1378 :      0=USE FIXED DATA PATTERN
1379 :SW7:  1=USE RANDOM CHARACTER COUNT
1380 :      0=USE FIXED CHAR COUNT
1381 :SW6:  1=USE RANDOM RECORD COUNT
1382 :      0=USE FIXED RECORD COUNT
1383 :SW5:  1=YOZZLE ON CURRENT RECORD
1384 :      0=DO NOT YOZZLE
1385 :SW4:  1=DO BOTH READ AND WRITE RETRIES
1386 :      0=INHIBIT RETRIES
1387 :SW3:  1=DO NOT READ FORWARD
1388 :      0=READ FORWARD
1389 :SW2:  1=DO NOT READ REVERSE
1390 :      0=READ REVERSE
```

1)INCORRECT RECORD COUNT  
STORED WHEN EOT REACHED ON WRITE  
2)ADJUST STACK PTR ON BAD TAPE OVFLW  
3)ADDED TU77 TEST CAPABILITY  
4)DOES NOT GENERATE LRC/CRC ON FIRST  
RECORD IN AUTO ACCEPT MODE

RECORD NUMBERING SYSTEM NOT CONSISTENT  
BETWEEN FORWARD AND REVERSE TAPE MOVEMEN  
FORMAT ERROR (BIT 4) MADE RETRYABLE

FIXED AID #CC0001450

CZTEDDO TMO3-TE16/TU77 DRT  
CZTEDD.P11 06-JUL-83 14:42

MACY11 30(1046) 06-JUL-83 21:03 F 3  
PAGE 25-2

SEQ 0031

1391  
1392  
1393  
1394  
1395

:SW1: 1=READ FORWARD FIRST  
:SW1: 0=READ REVERSE FIRST  
:SW0: 1=DO NOT WRITE  
:SW0: 0=WRITE  
:IF SWR <15::00> = 177777 OR NOT AVAILABLE USE SOFTWARE SWITCH REGISTER





```
1457
1458
1459
1467 000020 000020
1468 000020 022774
1469 000022 000340
1470
1471 000004
1472 000034
1473 000034 023126
1474 000036 000340
1475 104400
1476
(1)
(1) 000040
(1) 000042
(1) 000042 000000
(1) 000046 000046
(1) 000046 004676
(1) 000052 000052
(1) 000052 000000
(1) 000040
1477
1478 000060
1479 000060 020734
1480 000062 000340
1481
1482
1483
1484 000176
1485 000176 000000
1486
1487
1488 000200
1489 000200 000137 003022
1490
1491 000204
1492 000204 000137 003136
1493
1494 000210
1495 000210 005037 014404
1496 000214 000137 003144
1497
1498
1499
1500 000224
1501 000224 021160
1502 000226 000340
1503
1504
1505
1506 000240
1507 000240 005237 000734
1508 000244 000137 003122
```

;  
:TRAP CATCHERS\*\*\*\*\*  
.=20  
.WORD TTOUT ;SET IOT TRAP TO TTOUT ROUTINE  
.WORD 340 ;PRIORITY LEVEL 7  
TYPE=IOT ;EQUATE TYPE TO AN IOT INSTRUCTION  
.=34  
.WORD OCTP ;SET TRAP TRAP TO OCTP ROUTINE  
.WORD 340  
TYPOCT=TRAP ;EQUATE TYPOCT TO TRAP INSTRUCTION  
:ACT11 HOOK \*\*\*\*\*  
\$SVPC= ;SAVE CURRENT LOCATION CTR  
.=42  
.WORD 0  
.=46  
.WORD \$ENDAD ;SET LOCATION 46  
.=52  
.WORD 0 ;SET LOCATION 52 = 0  
.\$SVPC ;RESTORE LOCATION CTR  
:TTY INTERRUPT VECTOR\*\*\*\*\*  
.=60  
.WORD TTINT ;TTY INTERRUPT HANDLER ADDRESS  
.WORD 340 ;PRIORITY LEVEL 7  
:SOFTWARE SWITCH REGISTER\*\*\*\*\*  
:INVOKED IF SWR <15::00> = 177777 OR NOT AVAILABLE  
.=176  
SWREG: .WORD 0  
:START ADDRESS\*\*\*\*\*  
.=200  
JMP START ;ENTER PARAMETERS VIA TTY  
.=204  
JMP STARTC ;USE FIXED PARAMETERS; HOLD DATA  
.=210  
CLR RDFL  
JMP STARTA ;USE FIXED PARAMETERS; NEW DATA  
:MAG TAPE INTERRUPT VECTOR\*\*\*\*\*  
.=224  
MTINT ;MAG TAPE INTERRUPT HANDLER ADDRESS  
340  
:AUTO SEQUENCE START\*\*\*\*\*  
.=240  
INC ASEQF ;SET AUTO SEQUENCE FLAG  
JMP STAUT ;GO TO START OF AUTO SEQUENCE

```
1510 ;SHORT CONVERSATION RESTART*****
1511
1512 ;=300
1513 000300 000300 013444 INC SCVFL ;SET SHORT CONVERSATION FLAG
1514 000304 000137 003022 JMP START ;ENTER SHORT PARAMETER LIST
1515
1516 ;=510
1517 ;TU16 REGISTER EQUIVS*****
1518
1519 000510 172440 C1: 172440
1520 000512 172442 WC: 172442
1521 000514 172444 BA: 172444
1522 000516 172446 FC: 172446
1523 000520 172450 CS: 172450
1524 000522 172452 DS: 172452
1525 000524 172454 ER: 172454
1526 000526 172456 AS: 172456
1527 000530 172460 CC: 172460
1528 000532 172462 DB: 172462
1529 000534 172464 MR: 172464
1530 000536 172466 DT: 172466
1531 000540 172470 SN: 172470
1532 000542 172472 TC: 172472
1533
1534 ;CONSTANTS*****
1535
1536 000544 172440 REGS: 172440 ;STARTING REGISTER ADDRESS (CS1)
1537 000546 000224 VECT: 224 ;VECTOR ADDRESS (RH INTERRUPT)
1538 000550 000000 DVN: 0 ;DRIVE NUMBER
1539 000552 000000 UDES: 0 ;UNIT DESCRIPTION (PARITY,DENSITY,UNIT,FORMAT)
1540 000554 000100 RCNT: 100 ;RECORD COUNTER
1541 000556 176000 FMCNT: 176000 ;NUMBER OF CHAR (4 - 4000) OCTAL IN TWOS COMPLEMENT
1542 000560 000001 PATRN: 1 ;DATA PATTERN SELECTOR (0 - 15) OCTAL
1543 000562 000000 RDCMD: 0 ;READ COMMAND
1544 000564 000001 TMEX: 1 ;TAPE MARK FLAG: 1=TM 0=NO TM
1545 000566 000000 CRCC: 0 ;CRC CORRECTION FLAG (YES=1,NO=0)
1546 000570 000000 INTRF: 0 ;INTERCHANGE READ 1=YES 0=NO
1547 000572 000000 SPFLG: 0 ;SINGLE PASS 1=YES 0=NO
1548 000574 000001 RSTAL: 1 ;READ STALL
1549 000576 000001 WSTAL: 1 ;WRITE STALL
1550 000600 000001 TSTAL: 1 ;TURN AROUND STAL
1551 000602 002000 YSTAL: 2000 ;YOZZLE STAL
1552 000604 000010 RETRY: 10 ;READ RETRY NUMBER
1553 000606 177776 PSW: 177776 ;PROCESSOR STATUS
1554 000610 177570 SWR: 177570 ;CONSOLE SWITCHES
1555 000612 177560 TKS: 177560 ;TTY READ STATUS REGISTER
1556 000614 177562 TKB: 177562 ;TTY READ BUFFER
1557 000616 177564 TPS: 177564 ;TTY PUNCH STATUS REGISTER
1558 000620 177566 TPB: 177566 ;TTY PUNCH OUTPUT REGISTER
1559 000622 177550 PRS: 177550 ;H/S READER STATUS REGISTER
1560 000624 177552 PRB: 177552 ;H/S READER BUFFER
1561 000626 153624 RANBAS: 153624 ;RANDOM NUMBER GENERATOR BASE
1562 000630 032561 RANSAV: 032561 ;RANDOM NUMBER BUFFER
1563 000632 000100 RCSAV: 100 ;RECORD COUNT SAVE
1564 000634 176000 FCSAV: 176000 ;FRAME COUNT SAVE
1565
```

```
1566                                     :FLAGS AND COUNTERS*****
1567
1568 000636 000000      TINF: 0          :TTY ENTRY FLAG
1569 000640            STFLG:           :
1570 000640 000000      TOB: 0          :TTY OUTPUT BUFFER
1571 000642 000000      TIB: 0          :TTY INPUT BUFFER
1572 000644 000000      TEMP1: 0         :TEMP STORAGE
1573 000646 000000      TEMP2: 0         :TEMP STORAGE
1574 000650 000000      TEMP3: 0         :TEMP STORAGE
1575 000652 000000      EMADDR: 0        :ERROR MSG ADDRESS STORAGE
1576 000654 000000      BLCNTR: 0       :BLOCK COUNTER
1577 000656 000000      BBC: 0          :BAD RECORD COUNTER
1578 000660 000000      EOTREC: 0       :EOT FLAG
1579 000662 000000      RTRN: 0         :INTERRUPT RETURN STORAGE
1580 000664 000000      HDRFL: 0        :HEADER FLAG
1581 000666 000000      STAL: 0         :DELAY STORAGE
1582 000670 000000      PFLG: 0         :PRINT FLAG
1583 000672 000000      MTC1: 0         :MAG TAPE CONT REGISTER BUFFER
1584 000674 000000      UNP: 0          :UNIT TABLE POINTER
1585 000676 000000      TMFLG: 0        :TAPE MARK FLAG
1586 000700 000000      RPCNT: 0        :REPEAT COUNTER
1587 000702 000000      RTCNT: 0        :RETRY COUNTER
1588 000704 000000      DERFL: 0        :DATA ERROR FLAG
1589 000706 000000      SERFL: 0        :STATUS ERROR FLAG
1590 000710 000000      BCNT: 0         :BIT COUNTER
1591 000712 000000      RTYFL: 0        :RETRY FLAG
1592 000714 000000      UPS: 0          :UNIT POINTER SAVE
1593 000716 000000      BDPP: 0         :BITS DROPPED POINTER
1594 000720 000000      BPKP: 0         :BITS PICKED POINTER
1595 000722 000000      ERSV: 0         :ERROR SAVE LOC
1596 000724 000000      BTFLG: 0        :BAD TAPE FLAG
1597 000726 000000      BTSTF: 0        :STATISTIC PRINT FLAG
1598 000730 000000      BTPT: 0         :BAD TAPE POINTER
1599 000732 000000      ERTFL: 0        :ERASE FLAG
1600 000734            ENDFLG:           :
1601 000734 000000      ASEQF: 0        :AUTO SEQ FLAG
1602 000736 000000      ABLCNT: 0       :AUTO BLOCK COUNTER
1603 000740 000000      ASEQCF: 0      :AUTO SEQ CONTINUOUS FLAG
```

```
1605
1606 ;UNIT ORDER AND DESCRIPTION TABLE *****
1607
1608 000742 000000 UN1: 0 ;THIS TABLE IS LOADED
1609 000744 000000 UN2: 0 ;WITH UNIT NUMBERS AND
1610 000746 000000 UN3: 0 ;THEIR DESCRIPTIONS IN
1611 000750 000000 UN4: 0 ;THE ORDER THAT THEY
1612 000752 000000 UN5: 0 ;WILL BE TESTED
1613 000754 000000 UN6: 0
1614 000756 000000 UN7: 0
1615 000760 000000 UN8: 0
1616 000762 177777 UNX: -1
1617
1618 ;UNIT DROPS AND PICKS POINTERS*****
1619
1620 000764 001204 PIK1: BP00
1621 000766 001224 PIK2: BP10
1622 000770 001244 PIK3: BP20
1623 000772 001264 PIK4: BP30
1624 000774 001304 PIK5: BP40
1625 000776 001324 PIK6: BP50
1626 001000 001344 PIK7: BP60
1627 001002 001364 PIK8: BP70
1628 001004 001404 DRP1: BD00
1629 001006 001424 DRP2: BD10
1630 001010 001444 DRP3: BD20
1631 001012 001464 DRP4: BD30
1632 001014 001504 DRP5: BD40
1633 001016 001524 DRP6: BD50
1634 001020 001544 DRP7: BD60
1635 001022 001564 DRP8: BD70
1636
1637 ;UNIT BAD TAPE POINTERS*****
1638
1639 001024 001604 BTADDR: BT00
1640 001026 001710 BT01
1641 001030 002014 BT02
1642 001032 002120 BT03
1643 001034 002224 BT04
1644 001036 002330 BT05
1645 001040 002434 BT06
1646 001042 002540 BT07
1647
1648 ;UNIT WRITE RETRY COUNTER*****
1649
1650 ;SET START OF STATISTICS TABLE
1651 001044 STTBL:
1652 001044 000000 RTY1: 0
1653 001046 000000 RTY2: 0
1654 001050 000000 RTY3: 0
1655 001052 000000 RTY4: 0
1656 001054 000000 RTY5: 0
1657 001056 000000 RTY6: 0
1658 001060 000000 RTY7: 0
1659 001062 000000 RTY8: 0
1660
```

```
1661                                     :UNIT WRITE ERRORS*****
1662
1663 001064 000000      WTER1: 0
1664 001066 000000      WTER2: 0
1665 001070 000000      WTER3: 0
1666 001072 000000      WTER4: 0
1667 001074 000000      WTER5: 0
1668 001076 000000      WTER6: 0
1669 001100 000000      WTER7: 0
1670 001102 000000      WTER8: 0
1671
1672                                     :UNIT READ FORWARD ERRORS*****
1673
1674 001104 000000      RDER1: 0
1675 001106 000000      RDER2: 0
1676 001110 000000      RDER3: 0
1677 001112 000000      RDER4: 0
1678 001114 000000      RDER5: 0
1679 001116 000000      RDER6: 0
1680 001120 000000      RDER7: 0
1681 001122 000000      RDER8: 0
1682
1683                                     :UNIT DATA ERRORS FORWARD*****
1684
1685 001124 000000      DATER1: 0
1686 001126 000000      0
1687 001130 000000      0
1688 001132 000000      0
1689 001134 000000      0
1690 001136 000000      0
1691 001140 000000      0
1692 001142 000000      0
1693
1694                                     :UNIT READ REVERSE ERRORS*****
1695
1696 001144 000000      RDERR1: 0
1697 001146 000000      0
1698 001150 000000      0
1699 001152 000000      0
1700 001154 000000      0
1701 001156 000000      0
1702 001160 000000      0
1703 001162 000000      0
1704
1705                                     :UNIT DATA ERRORS REVERSE*****
1706
1707 001164 000000      DEREV1: 0
1708 001166 000000      0
1709 001170 000000      0
1710 001172 000000      0
1711 001174 000000      0
1712 001176 000000      0
1713 001200 000000      0
1714 001202 000000      0
```

```
1716 ;DRUPS + PICKS PER CHANNEL PER UNIT*****
1717
1718 001204 000000 BP00: 0
1719 001224 001224 .=.+16
1720 001224 000000 BP10: 0
1721 001244 001244 .=.+16
1722 001244 000000 BP20: 0
1723 001264 001264 .=.+16
1724 001264 000000 BP30: 0
1725 001304 001304 .=.+16
1726 001304 000000 BP40: 0
1727 001324 001324 .=.+16
1728 001324 000000 BP50: 0
1729 001344 001344 .=.+16
1730 001344 000000 BP60: 0
1731 001364 001364 .=.+16
1732 001364 000000 BP70: 0
1733 001404 001404 .=.+16
1734 001404 000000 BD00: 0
1735 001424 001424 .=.+16
1736 001424 000000 BD10: 0
1737 001444 001444 .=.+16
1738 001444 000000 BD20: 0
1739 001464 001464 .=.+16
1740 001464 000000 BD30: 0
1741 001504 001504 .=.+16
1742 001504 000000 BD40: 0
1743 001524 001524 .=.+16
1744 001524 000000 BD50: 0
1745 001544 001544 .=.+16
1746 001544 000000 BD60: 0
1747 001564 001564 .=.+16
1748 001564 000000 BD70: 0
1749 001604 .=.+16
1750
1751
```

1753				
1754				:UNIT BAD TAPE COUNTER:16 PER SLAVE*****
1755				
1756	001604	000000	BT00:	0
1757		001710		.=.+102
1758	001710	000000	BT01:	0
1759		002014		.=.+102
1760	002014	000000	BT02:	0
1761		002120		.=.+102
1762	002120	000000	BT03:	0
1763		002224		.=.+102
1764	002224	000000	BT04:	0
1765		002330		.=.+102
1766	002330	000000	BT05:	0
1767		002434		.=.+102
1768	002434	000000	BT06:	0
1769		002540		.=.+102
1770	002440	000000	BT07:	0
1771		002644		.=.+102
1772				
1773				:UNIT END OF TAPE COUNTERS 1 PER SLAVE*****
1774				
1775	002644	000000	EOTCO:	0
1776	002646	000000		0
1777	002650	000000		0
1778	002652	000000		0
1779	002654	000000		0
1780	002656	000000		0
1781	002660	000000		0
1782	002662	000000		0
1783				
1784				:UNIT READ FORWARD SOFT ERROR*****
1785				
1786	002664	000000	RFSOFT:	0
1787	002666	000000		0
1788	002670	000000		0
1789	002672	000000		0
1790	002674	000000		0
1791	002676	000000		0
1792	002700	000000		0
1793	002702	000000		0
1794				
1795				:UNIT READ REVERSE SOFT ERROR*****
1796				
1797	002704	000000	RRSOFT:	0
1798	002706	000000		0
1799	002710	000000		0
1800	002712	000000		0
1801	002714	000000		0
1802	002716	000000		0
1803	002720	000000		0
1804	002722	000000		0
1805				



1807  
1808  
1809  
1810 002724 000000  
1811 002726 000000  
1812 002730 000000  
1813 002732 000000  
1814 002734 000000  
1815 002736 000000  
1816 002740 000000  
1817 002742 000000

:UNIT READ FORWARD HARD ERROR\*\*\*\*\*  
RFHARD: 0  
0  
0  
0  
0  
0  
0  
0

1818  
1819  
1820  
1821 002744 000000  
1822 002746 000000  
1823 002750 000000  
1824 002752 000000  
1825 002754 000000  
1826 002756 000000  
1827 002760 000000  
1828 002762 000000

:UNIT READ REVERSE HARD ERROR\*\*\*\*\*  
RRHARD: 0  
0  
0  
0  
0  
0  
0  
0

1829  
1830 002764

:SET END OF STATISTICS TABLE  
ENDTBL:

1831  
1832  
1833  
1834 002764 002764  
1835 002766 013656  
1836 002770 014016  
1837 002772 014036  
1838 002774 014042  
1839 002776 014066  
1840 003000 014076  
1841 003002 014104  
1842 003004 014112  
1843 003006 014140  
1844 003010 014170  
1845 003012 014210  
1846 003014 014232  
1847 003016 014242  
1848 003020 014272  
1849

:DATA PATTERN GENERATORS\*\*\*\*\*  
DATBL: .  
DATA0: DAT0  
DATA1: DAT1  
DATA2: DAT2  
DATA3: DAT3  
DATA4: DAT4  
DATA5: DAT5  
DATA6: DAT6  
DATA7: DAT7  
DATA10: DAT10  
DATA11: DAT11  
DATA12: DAT12  
DATA13: DAT13  
DATA14: DAT14  
DATA15: DAT15

:ENTRY TABLE  
:EXTERNAL INPUT FROM H/S READER(SEE MAINDEC-11-DZTUF)  
:ALL ONES  
:ALL ZEROS  
:WALKING ONE  
:WALKING ZERO  
:ALTERNATING ONE/ZERO  
:ALTERNATING ZERO/ONE  
:ALTERNATING ONE/ZERO IN ALTERNATING CHARACTERS  
:WALKING ONE/ALL ONE IN ALTERNATING CHARACTERS  
:ALL BITS 0-377  
:ALL BITS 377-0  
:ALTERNATING CHARACTERS 0 AND 377  
:WALKING ZERO/ALL ZERO IN ALTERNATING CHARACTERS  
:AUTO SEQUENCE PATTERN 0,0,-1,-1,-1,0,0

1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
(1)  
1872  
1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898

003022 012706 000500  
003026 005037 000734  
003032 005027  
(1) 003034 000000  
(1) 003036 005737 000042  
(1) 003042 001407  
(1) 003044 012737 000176 000610  
(1) 003052 005237 003034  
(1) 003056 000137 003062  
(1) 003062  
003062 122737 000006 000041  
003070 001003  
003072 000004 026114  
003076 000000  
003100 005737 003034  
003104 001406  
003106 005237 000734  
003112 000004 024157  
003116 000137 021226  
  
003122 012737 000001 000636  
003130 005037 014404  
003134 000405  
  
003136 005037 000636  
003142 000442  
  
003144 005037 000636  
003150 012700 000640  
003154 012701 000074  
003160 105020  
003162 005301  
003164 001375  
003166 012706 000500

```
.EVEN
*****
:PROGRAM START AND SEQUENCE FORMATTER:
:
:THIS ROUTINE IS USED TO PERFORM ALL HOUSEKEEPING,
:DECIDE WHICH TRANSPORT TO TEST AND ITS AVAILABILITY,
:LOAD THE WRITE BUFFER WITH THE SELECTED DATA PATTERN,
:GENERATE ANY RANDOM NUMBER AND THEN EXECUTE
:THE TEST CYCLE REQUESTED BY THE SWITCH SETTING.
:AT THE END OF THE TEST CYCLE THE NEXT UNIT IS SELECTED
:AND CHECKED FOR AVAILABILITY AND THE TEST CYCLE IS
:EXECUTED ON IT.
:THE READ WRITE STATS MAY BE PRINTED AT THE END OF
:EACH TEST CYCLE VIA CONSOLE SWITCH FOURTEEN (14).
*****

:START 200, & 300*****
START:  MOV    #500,SP      ;SET STACK PTR
      CLR    ASEQF        ;CLEAR AUTO SEQUENCE FLAG
      CLR    (PC)+        ;CLEAR CHAIN INDICATOR
      CHNFLG: .WORD 0      ;CHAIN MODE INDICATOR
                               ;1/0 = CHAIN/NOT CHAIN MODE
                               ;BRANCH IF IN DUMP MODE
      TST    @#42
      BEQ    50$
      MOV    #SWREG,SWR    ;INVOKE SOFTWARE SWR
      INC    CHNFLG        ;SET CHNFLG = CHAIN MODE
      JMP    3$           ;GO TO CHAIN ADDRESS
      50$:
      3$:  CMPB   #6,@#41    ;BRANCH IF LOADED VIA TMDP
      BNE    4$
      TYPE,MSG120        ;ADVISE USER TO REMOVE TMDP FROM SLAVE
      HALT
      4$:  TST    CHNFLG    ;SEE IF IN CHAIN MODE
      BEQ    STAUT
      INC    ASEQF        ;SET AUTO SEQUENCE FLAG
      TYPE,MSG30         ;TYPE TITLE
      JMP    ASEQ0       ;GO TO AUTO SEQUENCER

:START 240*****
STAUT: MOV    #1,TINF     ;SET TTY ENTRY FLAG
      CLR    RDFL        ;CLEAR RANDOM DATA FLAG
      BR    STARTB

:START 204*****
STARTC: CLR    TINF      ;CLEAR TTY INPUT FLAG
      BR    STARTD

:START 210*****
STARTA: CLR    TINF      ;CLEAR TTY ENTRY FLAG
STARTB: MOV    #STFLG,R0  ;GET STARTING ADDRESS OF FLAGS
      MOV    #ENDFLG-STFLG,R1
      1$:  CLRB   (R0)+    ;CLEAR FLAGS AND COUNTERS
      DEC    R1
      BNE    1$
      MOV    #500,SP     ;SET STACK POINTER
```

1899	003172	004737	004122		JSR	PC,RANSET	:GO RESET RANDOM BASE	
1900	003176	012700	001044		MOV	#STTBL,R0	:GET STARTING ADDRESS OF STAT TABLE	
1901	003202	012701	001720		MOV	#ENDTBL-STTBL,R1	:AND # OF BYTES IN TABLE	
1902	003206	105020		2\$:	CLRB	(R0)+	:CLEAR STATISTIC COUNTERS	
1903	003210	005301			DEC	R1		
1904	003212	001375			BNE	2\$		
1905	003214	012700	000742		MOV	#UN1,R0	:SET ALL SLAVES ON-LINE	
1906	003220	022710	177777	3\$:	CMP	#-1,(R0)	:BRANCH IF AT END OF TABLE	
1907	003224	001403			BEQ	4\$		
1908	003226	042720	040000		BIC	#40000,(R0)+	:MARK SLAVE ON-LINE	
1909	003232	000772			BR	3\$		
1910	003234	012737	177777	013652	4\$:	MOV	#-1,PATS	:PRESET PATTERN
1911	003242	012737	000001	000654	STARTE:	MOV	#1,BLCNTR	:PRESET BLOCK COUNTER
1912	003250	013746	000004		STARTD:	MOV	@#4,-(SP)	:SAVE ERROR TRAP VECTOR
1913	003254	013746	000006		MOV	@#6,-(SP)		
1914	003260	022737	000176	000610	CMP	#SWREG,SWR	:BRANCH IF SOFTWARE SWR	
1915	003266	001413			BEQ	2\$	:ALREADY SELECTED	
1916	003270	012737	003314	000004	MOV	#1\$,@#4	:SET TIMEOUT TRAP TO 1\$ BELOW	
1917	003276	005037	000006		CLR	@#6		
1918	003302	022777	177777	175300	CMP	#177777,@SWR	:BRANCH IF SWR = 177777 TRAP	
1919	003310	001402			BEQ	2\$	:IF NOT AVAIL (1\$) OTHERWISE	
1920	003312	000404			BR	3\$	:GO TO 3\$	
1921	003314	022626			1\$:	CMP	(SP)+,(SP)+	:RESET STACK
1922	003316	012737	000176	000610	2\$:	MOV	#SWREG,SWR	:SET SWR = SOFTWARE SWR
1923	003324	012637	000006		3\$:	MOV	(SP)+,@#6	:RESTORE ERROR TRAP
1924	003330	012637	000004		MOV	(SP)+,@#4		
1925	003334	012706	000500		MOV	#500,SP		
1926	003340	004737	011750		JSR	PC,TINP	:GO GET PARAMETERS FROM TTY	
1927	003344	012777	000040	175146	MOV	#40,@CS	:INITIALIZE	
1928	003352	005000			STAUTO:	CLR	R0	:POINT TO FIRST ENTRY
1929	003354	022760	177777	000742	1\$:	CMP	#-1,UN1(R0)	:BRANCH IF LAST ENTRY
1930	003362	001406			BEQ	2\$		
1931	003364	042760	100000	000742	BIC	#100000,UN1(R0)	:CLEAR EOT FLAG	
1932	003372	062700	000002		ADD	#2,R0	:POINT TO NEXT UNIT ENTRY	
1933	003376	000766			BR	1\$	:CONTINUE CLEARING	
1934	003400	113737	004731	004730	2\$:	MOVB	REOTC+1,REOTC	:RESTORE EOT COUNTER
1935	003406	012777	000100	175176	START1:	MOV	#100,@TKS	:SET KEYBOARD IE BIT
1936	003414	013700	000674		MOV	UNP,R0	:R0 = UNIT TABLE POINTER	
1937	003420	022760	177777	000742	STAR1A:	CMP	#-1,UN1(R0)	:BRANCH IF LAST ENTRY
1938	003426	001404			BEQ	STAR1B		
1939	003430	016037	000742	000552	MOV	UN1(R0),UDES	:LOAD NEXT UNIT DESCRIPTION	
1940	003436	000445			BR	START4		
1941	003440	005237	000654		STAR1B:	INC	BLCNTR	:BUMP BLOCK COUNTER
1942	003444	005737	000734		TST	ASEQF	:SEE IF AUTO SEQ	
1943	003450	001411			BEQ	STAR1C	:IF NOT: BR	
1944	003452	023737	000654	000736	CMP	BLCNTR,ABL CNT	:SEE IF DONE SEQ	
1945	003460	001005			BNE	STAR1C	:IF NOT: BR	
1946	003462	005037	000654		CLR	BLCNTR	:RESET BLOCK CNTR	
1947	003466	005037	000674		CLR	UNP	:RESET UNIT POINTER	
1948	003472	000207			RTS	PC	:RETURN TO AUTO SEQ	
1949	003474	005037	000674		STAR1C:	CLR	UNP	
1950	003500	005000			CLR	R0		
1951	003502	016037	000742	000552	MOV	UN1(R0),UDES	:LOAD FIRST UNIT DESCRIPTION	
1952	003510	105777	175074		TSTB	@SWR	:SEE IF RANDOM RECORD SIZE	
1953	003514	100002			BPL	START2	:IF NOT: BR	
1954	003516	004737	011664		JSR	PC,CCNTR	:GO GENERATE RANDOM RECORD SIZE	

```

1955 003522 032777 000400 175060 START2: BIT #400,@SWR ;SEE IF RANDOM DATA
1956 003530 001402 BEQ START3 ;IF NOT: BR
1957 003532 004737 014342 JSR PC,DATR ;GO GENERATE RANDOM DATA
1958 003536 032777 000100 175044 START3: BIT #100,@SWR ;SEE IF RANDOM RECORD COUNT
1959 003544 001402 BEQ START4 ;IF NOT: BR
1960 003546 004737 011724 JSR PC,RCNTR ;GO GENERATE RANDOM RECORD COUNT
1961 003552 032760 140000 000742 START4: BIT #140000,UN1(R0) ;BRANCH IF UNIT AT EOT
1962 003560 001065 BNE START7 ;OR MARKED OFF-LINE
1963 003562 012777 000040 174730 MOV #40,@CS ;DO A MASSBUS CLEAR
1964 003570 013777 000550 174722 MOV DVN,@CS ;SET DRIVE NUMBER
1965 003576 013777 000552 174736 MOV UDES,@TC ;SET SLAVE NUMBER
1966 003604 105777 174712 1$: TSTB @DS ;SEE IF SLAVE AVAIL
1967 003610 100405 BMI 2$ ;IF SO: BR
1968 003612 005337 000666 DEC STAL
1969 003616 001372 BNE 1$ ;AWAIT TUR
1970 003620 000137 020312 JMP OFFLINE ;GO MARK DRIVE OFF-LINE
1971 003624 004737 013472 2$: JSR PC,DSUP ;GO SET UP WRITE DATA
1972 003630 004737 005236 JSR PC,INIT ;INIT SLAVE
1973 003634 004737 004732 JSR PC,RWND ;REWIND
1974 003640 004737 005352 JSR PC,WRITE ;WRITE
1975 003644 013737 000600 000666 MOV TSTAL,STAL ;SET TURN AROUND DELAY
1976 003652 004737 011654 JSR PC,STALL ;DELAY
1977 003656 004737 007210 JSR PC,RSEQ ;GO TO READ SEQUENCER
1978 003662 013737 000600 000666 MOV TSTAL,STAL ;SET TURN AROUND DELAY
1979 003670 004737 011654 JSR PC,STALL ;DELAY
1980 003674 032777 040000 174706 BIT #40000,@SWR ;SEE IF SHOULD PRINT STATISTICS
1981 003702 001414 BEQ START7 ;IF NOT: BR
1982 003704 012700 000001 MOV #1,R0 ;SET RECORD COUNTER TO 1
1983 003710 004737 022012 JSR PC,PAPRT ;PRINT CYCLE NUMBER
1984 003714 004737 003744 JSR PC,STP ;GO PRINT STATS
1985 003720 005237 000726 INC BTSTF ;SET STAT ONLY PRINT
1986 003724 004737 007126 JSR PC,BTPRT ;PRINT BAD TAPE STATS
1987 003730 005037 000726 CLR BTSTF ;CLEAR FLAG
1988 003734 062737 000002 000674 START7: ADD #2,UNP ;POINT TO NEXT UNIT
1989 003742 000621 START8: BR START1 ;CONTINUE

```

1991  
1992  
1993 003744 004737 016370  
1994 003750 000004 025123  
1995 003754 013700 000674  
1996 003760 016003 001044  
1997 003764 104400  
1998 003766 000004 025234  
1999 003772 016003 001064  
2000 003776 104400  
2001 004000 000004 025223  
2002 004004 016003 001104  
2003 004010 104400  
2004 004012 000004 026001  
2005 004016 016003 002664  
2006 004022 104400  
2007 004024 000004 026012  
2008 004030 016003 002724  
2009 004034 104400  
2010 004036 000004 025320  
2011 004042 016003 001124  
2012 004046 104400  
2013 004050 000004 025155  
2014 004054 016003 001144  
2015 004060 104400  
2016 004062 000004 026001  
2017 004066 016003 002704  
2018 004072 104400  
2019 004074 000004 026012  
2020 004100 016003 002744  
2021 004104 104400  
2022 004106 000004 025307  
2023 004112 016003 001164  
2024 004116 104400  
2025 004120 000207  
2026  
2027  
2028  
2029 004122 012737 153624 000626  
2030 004130 012737 032561 000630  
2031 004136 013737 000632 000554  
2032 004144 013737 000634 000556  
2033 004152 000207  
2034

```
***** SUBROUTINE TO PRINT STATISTICS *****  
STP: JSR PC,DPPRT ;PRINT DROPS AND PICKS  
TYPE,MSG65 ;TYPE MSG  
MOV UNP,R0  
MOV RTY1(R0),R3  
TYPOCT ;PRINT RETRIES  
TYPE,MSG73 ;TYPE MSG  
MOV WTER1(R0),R3  
TYPOCT ;PRINT WRITE ERRORS  
TYPE,MSG72 ;TYPE MSG  
MOV RDER1(R0),R3  
TYPOCT ;PRINT READ FORWARD ERRORS  
TYPE,MSG113 ;TYPE MSG  
MOV RFSOFT(R0),R3  
TYPOCT ;PRINT FORWARD SOFT ERRORS  
TYPE,MSG114 ;TYPE MSG  
MOV RFHARD(R0),R3  
TYPOCT ;PRINT HARD FORWARE ERRORS  
TYPE,MSG77 ;TYPE MSG  
MOV DATER1(R0),R3  
TYPOCT ;PRINT DATA ERROR FORWARD NUMBER  
TYPE,MSG68 ;TYPE MSG  
MOV RDERR1(R0),R3  
TYPOCT ;PRINT REVESE ERROR NUMBER  
TYPE,MSG113 ;TYPE MSG  
MOV RRSOFT(R0),R3  
TYPOCT ;PRINT REVERSE SOFT ERROR  
TYPE,MSG114 ;TYPE MSG  
MOV RRHARD(R0),R3  
TYPOCT ;TYPE MSG  
TYPE,MSG76 ;TYPE MSG  
MOV DEREV1(R0),R3  
TYPOCT ;PRINT DATA REVERSE ERROR NUMBER  
RTS PC ;RETURN  
  
;RANDOM BASE RESET*****  
RANSET: MOV #153624,RANBAS ;RESET BASE  
MOV #32561,RANSAV ;RESET BUFFER  
MOV RCSAV,RCNT ;RESET RECORD COUNT  
MOV FCSAV,FMCNT ;RESET FRAME COUNT  
RTS PC
```

```
2036 :*****
2037 :REWIND FROM EOT:
2038 :
2039 :WHEN ANY TRANSPORT BEING TESTED REACHES END OF TAPE
2040 :DURING A READ OR WRITE OPERATION, IT WILL BE REWOUND
2041 :AND FLAGGED AS UNAVAILABLE UNTIL ALL AVAILABLE UNITS
2042 :HAVE REACHED EOT AT WHICH TIME ALL TESTING WILL BE RESUMED
2043 :AT A BLOCK COUNT OF ONE (1). A MESSAGE WILL BE
2044 :PRINTED ON THE SUPERVISORS CONSOLE AS EACH UNIT REACHES
2045 :EOT AND IS REWOUND.
2046 :*****
2047
2048 004154 013777 000552 174360 REOT: MOV UDES,@TC ;LOAD TAPE CONTROL REGISTER
2049 004162 013700 000674 MOV UNP,R0 ;GET UNIT POINTER
2050 004166 032760 040000 000742 BIT #40000,UN1(R0) ;BRANCH IF UNIT MARKED OFF-LINE
2051 004174 001014 BNE 2$
2052 004176 012777 000011 174304 MOV #11,@C1 ;DRIVE CLEAR
2053 004204 105777 174312 1$: TSTB @DS ;WAIT FOR DRY
2054 004210 100375 BPL 1$
2055 004212 012777 000007 174270 MOV #7,@C1 ;START REWIND
2056 004220 005737 000724 TST BTFLG ;SEE IF BAD TAPE OVERFLOW REWIND
2057 004224 001004 BNE 3$ ;IF SO: BR
2058 004226 013700 000660 2$: MOV EOTREC,R0
2059 004232 042700 100000 BIC #100000,R0 ;SET RECORD NUMBER OF EOT
2060 004236 005037 000660 3$: CLR EOTREC ;CLEAR EOT INDICATOR & REC COUNT
2061 004242 004737 022012 JSR PC,PAPRT ;PRINT HEADER
2062 004246 022737 000002 000724 CMP #2,BTFLG ;SEE IF POSITION ERROR
2063 004254 001004 BNE 4$ ;IF NOT: BR
2064 004256 012737 025674 004306 MOV #MSG109.6$ ;SET POSITION ERROR MSG
2065 004264 000407 BR 5$
2066 004266 022737 000001 000724 4$: CMP #1,BTFLG ;SEE IF BAD TAPE OVERFLOW
2067 004274 001006 BNE REOT1C ;IF NOT: BR
2068 004276 012737 025527 004306 MOV #MSG106.6$ ;SET BAD TAPE OVERFLOW MSG
2069 004304 000004 5$: TYPE ;TYPE MSG
2070 004306 000000 6$: .WORD 0 ;WILL CONTAIN MESSAGE ADDRESS
2071 004310 000411 BR REOT1E
2072 004312 000004 023660 REOT1C: TYPE,MSG20 ;TYPE EOT MSG
2073 004316 013704 000674 MOV UNP,R4
2074 004322 005264 002644 INC EOTC(R4) ;BUMP CNTR
2075 004326 016403 002644 MOV EOTC(R4),R3
2076 004332 104400 TYPOCT ;PRINT EOT CNTR
2077 004334 000004 025552 REOT1E: TYPE,MSG16A ;TYPE MSG
2078 004340 005037 000724 CLR BTFLG ;CLEAR BAD TAPE FLAG
2079 004344 004737 003744 JSR PC,STP ;PRINT STATS
2080 004350 004737 007126 JSR PC,BTPRT ;PRINT BAD TAPE STATS
2081 004354 013700 000674 REOT2: MOV UNP,R0 ;GET UNIT POINTER
2082 004360 032760 040000 000742 BIT #40000,UN1(R0) ;BRANCH IF UNIT MARKED OFF-LINE
2083 004366 001010 BNE REOT2A
2084 004370 105777 174126 TSTB @DS ;BRANCH IF DRY SET
2085 004374 100405 BMI REOT2A
2086 004376 005337 000666 DEC STAL
2087 004402 001364 BNE REOT2 ;WAIT DRY
2088 004404 000137 020312 JMP OFFLINE ;GO MARK SLAVE OFFLINE
2089
2090 004410 105337 004730 REOT2A: DECB REOTC ;SEE IF LAST UNIT TO REACH EOT
2091 004414 001410 BEQ REOT3 ;IF SO: BR
```

2092	004416	013700	000674			MOV	UNP,R0	
2093	004422	052760	100000	000742		BIS	#100000,UN1(R0)	:SET EOT FLAG
2094	004430	005726				TST	(SP)+	:RESET STACK POINTER
2095	004432	000137	003734			JMP	START7	:GO TO NEXT UNIT
2096	004436	113737	004731	004730	REOT3:	MOV	REOTC+1,REOTC	:RESTORE UNITS EOT COUNTER
2097	004444	005037	000674			CLR	UNP	
2098	004450	005000				CLR	R0	:POINT TO FIRST UNIT
2099	004452	016037	003742	000552	REOT4:	MOV	UN1(R0),UDES	:LOAD UNIT DESCRIPTION
2100	004460	013777	000552	174054		MOV	UDES,@TC	:SELECT SLAVE
2101	004466	032760	040000	000742		BIT	#40000,UN1(R0)	:BRANCH IF UNIT NOT MARKED OFF-LINE
2102	004474	001412				BEQ	1\$	
2103	004476	032777	010000	174016		BIT	#10000,@DS	:BRANCH IF MEDIUM NOT ON LINE
2104	004504	001427				BEQ	10\$	
2105	004506	062737	000401	004730		ADD	#401,REOTC	:INCREMENT # OF UNITS UNDER TEST
2106	004514	042760	140000	000742		BIC	#140000,UN1(R0)	:MARK UNIT BACK ON-LINE
2107	004522	012777	000011	173760	1\$:	MOV	#11,@C1	:DRIVE CLEAR
2108	004530	105777	173766		2\$:	TSTB	@DS	:WAIT FOR DRIVE READY
2109	004534	100375				BPL	2\$	
2110	004536	012777	000007	173744		MOV	#7,@C1	:REWIND UNIT
2111	004544	032777	000002	173750	3\$:	JIT	#2,@DS	:WAIT FOR BOT TO SET
2112	004552	001774				BEQ	3\$	
2113	004554	032777	020000	173740	4\$:	BIT	#20000,@DS	:WAIT FOR PIP TO CLEAR
2114	004562	001374				BNE	4\$	:AWAIT PIP RESET
2115								
2116	004564	042760	100000	000742	10\$:	BIC	#100000,UN1(R0)	:CLEAR EOT FLAG
2117	004572	062737	000002	000674		ADD	#2,UNP	
2118	004600	013700	000674			MOV	UNP,R0	:POINT TO NEXT UNIT
2119	004604	022760	177777	000742		CMP	#-1,UN1(R0)	:BRANCH IF NOT LAST UNIT
2120	004612	001317				BNE	REOT4	
2121	004614	005037	000674		REOT7:	CLR	UNP	:CLEAR UNIT POINTER
2122	004620	005037	000636			CLR	TINF	:CLEAR TTY INPUT FLAG
2123	004624	005737	000734			TST	ASEQF	:SEE IF AUTO SEQ
2124	004630	001402				BEQ	REOTX	:IF NOT: BR
2125	004632	005726				TST	(SP)+	:RESET STACK POINTER
2126	004634	000207				RTS	PC	:RETURN TO AUTO SEQ
2127	004636	004737	004122		REOTX:	JSR	PC,RANSET	:GO RESET RANDOM BASE
2128	004642	012737	177777	013652		MOV	#-1,PATS	:PRESET PATTERN
2129	004650	005037	014404			CLR	RDFL	:CLEAR RANDOM FLAG
2130	004654	005737	000572			TST	SPFLG	:SEE IF SINGLE PASS
2131	004660	001421				BEQ	REOTXX	:IF NOT: BR
2132	004662	000004	025430		TEND:	TYPE,MSG100		:TYPE MSG
2133	004666	013700	000042			MOV	@#42,R0	:GET ACT11 RETURN ADDRESS
(1)	004672	001405				BEQ	HERE	:BRANCH IF NOT ACT11
(1)	004674	000005				RESET		
(1)	004676	004710			\$ENDAD:	JSR	PC,(R0)	
(1)	004700	000240				NOP		
(1)	004702	000240				NOP		
(1)	004704	000240				NOP		
(1)	004706	000240			HERE:	NOP		
2134	004710	005737	003034			TST	CHNFLG	:BRANCH IF NOT CHAIN MODE
2135	004714	001402				BEQ	1\$	
2136	004716	000137	021226			JMP	ASEQ0	:RETURN TO AUTO SEQUENCER
2137	004722	000000			1\$:	HALT		
2138	004724	000137	003242		REOTXX:	JMP	STARTE	:RESTART AT BLOCK NUMBER ONE
2139	004730	000000			REOTC:	0		:EOT UNIT COUNTER

```
2141 :*****
2142 :REWIND ALL AVAIL TAPES:
2143 :
2144 :THIS ROUTINE; ENTERED VIA CONSOLE SWITCH NINE (9),
2145 :WILL REWIND ALL AVAILABLE TAPES TO BOT NO MATTER
2146 :WHERE THEY ARE CURRENTLY POSITIONED AND RESUME TESTING
2147 :ON THE CURRENTLY SELECTED UNIT.
2148 :*****
2149
2150 004732 032777 001000 173650 RWND: BIT #1000,@SWR ;SEE IF SHOULD REWIND
2151 004740 001001 BNE RWNDA ;IF SO: BR
2152 004742 000207 RTS PC ;ELSE EXIT
2153 004744 013737 000674 000714 RWNDA: MOV UNP,UPS ;SAVE UNIT POINTER
2154 004752 005037 000674 CLR UNP ;CLEAR POINTER
2155 004756 005037 000660 CLR EOTREC ;CLEAR EOT FLAG
2156 004762 113737 004731 004730 MOV#B REOTC+1,REOTC ;++B RESTORE UNIT CTR
2157 004770 013700 000674 RWND0: MOV UNP,RO ;POINT TO UNIT ENTRY
2158 004774 022760 177777 000742 CMP #-1,UN1(RO) ;BRANCH IF LAST ENTRY
2159 005002 001437 BEQ RWND2
2160 005004 032760 140000 000742 BIT #140000,UN1(RO) ;BRANCH IF ALREADY REWINDING
2161 005012 001024 BNE RWND1A ;OR MARKED OFF LINE
2162 005014 016037 000742 000552 MOV UN1(RO),UDES ;SET UNIT DESCRIPTION
2163 005022 013777 000552 173512 MOV UDES,@TC ;LOAD COMMAND REGISTER
2164 005030 012777 000011 173452 MOV #11,@C1 ;DRIVE CLEAR
2165 005036 012777 000007 173444 MOV #7,@C1 ;START REWIND
2166 005044 105777 173452 1$: TSTB @DS
2167 005050 100405 BMI RWND1A ;IF DRY: BR
2168 005052 005337 000666 DEC STAL
2169 005056 001372 BNE 1$ ;AWAIT DRY
2170 005060 000137 020312 JMP OFFLINE ;GO MARK UNIT OFF LINE
2171 005064 042760 100000 000742 RWND1A: BIC #100000,UN1(RO) ;CLEAR EOT FLAG
2172 005072 062737 000002 000674 ADD #2,UNP ;BUMP POINTER
2173 005100 000733 BR RWND0 ;DO NEXT UNIT
2174 005102 005037 000674 RWND2: CLR UNP ;CLEAR POINTER
2175 005106 013700 000674 RWND3: MOV UNP,RO ;POINT TO UNIT ENTRY
2176 005112 022760 177777 000742 CMP #-1,UN1(RO) ;BRANCH IF LAST ENTRY
2177 005120 001433 BEQ RWNDX
2178 005122 016037 000742 000552 MOV UN1(RO),UDES ;SET UNIT DESCRIPTION
2179 005130 032760 040000 000742 BIT #40000,UN1(RO) ;BRANCH IF UNIT MARKED OFF LINE
2180 005136 001015 BNE RWND5
2181 005140 013777 000552 173374 MOV UDES,@TC ;LOAD UNIT DESCRIPTION
2182 005146 032777 020000 173346 1$: BIT #20000,@DS
2183 005154 001374 BNE 1$ ;AWAIT PIP RESET
2184 005156 032777 000002 173336 BIT #2,@DS ;BRANCH IF SLAVE AT BOT
2185 005164 001002 BNE RWND5
2186 005166 000137 020312 JMP OFFLINE ;PRINT OFFLINE MESSAGE
2187 005172 062737 000002 000674 RWND5: ADD #2,UNP ;BUMP POINTER
2188 005200 012777 000011 173302 MOV #11,@C1 ;DRIVE CLEAR
2189 005206 000737 BR RWND3 ;DO NEXT UNIT
2190
2191 005210 013700 000714 RWNDX: MOV UPS,RO ;RESTORE UNIT POINTER
2192 005214 010037 000674 MOV RO,UNP
2193 005220 016037 000742 000552 MOV UN1(RO),UDES ;RESET UNIT DESCRIPTION
2194 005226 013777 000552 173306 MOV UDES,@TC
2195 005234 000207 RTS PC ;RETURN TO TEST
2196
```



2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223

005236 013746 000552  
005242 012777 000040 173250  
005250 013777 000550 173242  
005256 011677 173260  
005262 042716 174377  
005266 022726 001400  
005272 001005  
005274 032777 000040 173220  
005302 001422  
005304 000404  
005306 032777 000040 173206 1\$:  
005314 001015  
005316 012777 000007 173164 2\$:  
005324 105777 173172 20\$:  
005330 100375  
005332 032777 020000 173162 3\$:  
005340 001374  
005342 012777 000011 173140  
005350 000207 4\$:

```
*****  
:INITIALIZE SELECTED SALVE  
:THIS ROUTINE REWINDS AND SETS THE PROPER DENSITY IF  
:THE DENSITY REQUIRED FOR THE TEST IS DIFFERENT FROM  
:THE DENSITY AT WHICH THE SLAVE IS SELECTED.  
*****  
INIT:  MOV      UDES,-(SP)      ;GET UNIT DESCRIPTION  
      MOV      #40,@CS        ;DO A MASSBUS CLEAR  
      MOV      DVN,@CS        ;LOAD DRIVE #  
      MOV      (SP),@TC        ;LOAD SLAVE # & SLAVE DESCRIPTION  
      BIC      #174377,(SP)    ;CLEAR ALL BUT DENSITY BITS  
      CMP      #1400,(SP)+     ;BRANCH IF NOT NRZ  
      BNE     1$  
      BIT      #40,@DS        ;BRANCH IF SLAVE IS IN PE MODE  
      BEQ     4$              ;PES = 0  
      BR      2$  
1$:    BIT      #40,@DS        ;BRANCH IF SLAVE IS IN PE MODE  
      BNE     4$              ;PES = 1  
2$:    MOV      #7,@C1         ;LOAD REWIND COMMAND  
20$:   TSTB    @DS             ;WAIT FOR READY  
      BPL     20$  
3$:    BIT      #20000,@DS     ;WAIT FOR PIP = 0  
      BNE     3$  
4$:    MOV      #11,@C1        ;CLEAR DRIVE  
      RTS     PC
```

2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258 005352 032777 000001 173230  
2259 005360 001402  
2260 005362 000137 006132  
2261 005366 013700 000554  
2262 005372 012737 023546 000652  
2263 005400 013777 000556 173110  
2264 005406 012777 026344 173100  
2265 005414 112737 000060 000672  
2266 005422 012737 005434 000662  
2267 005430 000137 020372  
2268 005434 032777 002000 173060  
2269 005442 001412  
2270 005444 005737 000660  
2271 005450 100407  
2272 005452 005300  
2273 005454 052700 100000  
2274 005460 010037 000660  
2275 005464 012700 000002  
2276 005470 032777 010000 173112  
2277 005476 001002  
2278 005500 004737 016522  
2279 005504 013737 000576 000666  
2280 005512 004737 011654

```

:*****
:WRITE ROUTINE:
:
:THIS ROUTINE IS USED TO WRITE ONTO TAPE THE BLOCK
:OF DATA DESCRIBED BY THE OPERATOR AND SET UP
:IN THE SEQUENCE FORMATTER. THE TAPE UNIT TO BE USED
:HAS BEEN ASSIGNED BY THE SEQUENCE FORMATTER AND
:ITS PARAMETERS SET IN A UNIT DESCRIPTION WORD.
:AS EACH RECORD OF THE BLOCK IS WRITTEN, IT IS CHECKED
:FOR STATUS ERRORS, WORD COUNT ZERO, AND CORRECT CURRENT
:MEMORY ADDRESS. IF THE WRITE OPERATION RESULTS IN
:ANY ERROR CONDITION, A WRITE RETRY OF THAT OPERATION
:MAY BE DONE BY SETTING SWITCH FOUR (4) TO A ONE (1).
:THE RETRY CONSISTS OF A BACKSPACE, ERASE FORWARD, AND
:REWRITE OF THE RECORD. (SEE WRITE RETRY SUBROUTINE)
:AFTER ALL DATA RECORDS IN THE BLOCK HAVE BEEN
:WRITTEN, THE WRITE ROUTINE WILL EXECUTE A WRITE
:TAPE MARK COMMAND IF THE TTY RESPONSE TM=1 WAS
:MADE AT INITIAL START. THE TM IS COUNTED AS TOTAL
:DATA RECORDS PLUS ONE (IE: IF 100 DATA RECORDS; TM=RECORD 101)
:IF THE WRITE OPERATION (DATA OR TM) CAUSES THE SELECTED SLAVE
:TO REACH END OF TAPE (EOT) AND THERE IS TO BE NO READING DONE,
:(SW2 AND SW3 SET TO A 1) THEN THE SLAVE IS REWOUND AND
:FLAGGED AS UNAVAILABLE FOR TESTING UNTIL ALL SLAVES HAVE
:REACHED EOT AND BEEN REWOUND AT WHICH TIME TESTING IS
:RESUMED ON ALL AVAILABLE SLAVES.
:WRITE RETRY MAY BE ALLOWED VIA CONSOLE SWITCH FOUR (4).
:ERROR CHECKING MAY BE DISALLOWED VIA CONSOLE SWITCH
:TWELVE (12).
:WRITING TO TAPE MAY BE DISALLOWED VIA CONSOLE SWITCH
:ZERO (0).
:*****

```

```

WRITE: BIT #1,@SWR ;SEE IF SHOULD WRITE
      BEQ WRITE
      JMP WEX ;IF NOT: BR
      MOV RCNT,RO ;RO=RECORD COUNT
      MOV #MSG5,EMADDR ;SET ERROR MSG ADDRESS
      MOV FMCNT,@FC ;LOAD CHAR COUNT
      MOV #WDATA,@BA ;SET DATA ADDR
      MOV #60,MTC1 ;SET WRITE OP COMMAND
      MOV #W1,RTRN ;SET RETURN ADDRESS
      JMP TAPG ;GO EXECUTE COMMAND
w1: BIT #2000,@DS ;SEE IF EOT
     BEQ 1$ ;IF NOT AT EOT: BR
     TST EOTREC ;BRANCH IF WRITTEN PAST EOT
     BMI 1$
     DEC RO ;ADJUST # OF RECORDS WRITTEN
     BIS #100000,RO ;SET EOT INDICATOR
     MOV RO,EOTREC ;SAVE RECORD COUNT
     MOV #2,RO ;SET TO WRITE 1 LAST RECORD
1$: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERRORS
     BNE 2$ ;IF NOT: BR
     JSR PC,ERCHK ;GO CHECK ERRORS
2$: MOV WSTAL,STAL ;SET DELAY
     JSR PC,STALL ;DELAY

```

2281	005516	005737	000712		TST	RTYFL		:SEE IF RETRY TIME
2282	005522	001401			BEQ	3\$		:IF NOT: BR
2283	005524	000207			RTS	PC		:ELSE RETURN
2284	005526	005737	000706	3\$:	TST	SERFL		:SEE IF WRITE ERROR
2285	005532	001446			BEQ	W5		:IF NOT: BR
2286	005534	013704	000674		MOV	UNP,R4		
2287	005540	005264	001064		INC	WTER1(R4)		:BUMP WRITE ERROR
2288	005544	005037	000706		CLR	SERFL		:CLEAR STATUS ERROR FLAG
2289	005550	032777	000020	173032	BIT	#20,@SWR		:SEE IF RETRY
2290	005556	001434			BEQ	W5		:IF NOT: BR
2291	005560	013703	000722		MOV	ERSAV,R3		
2292	005564	042703	102720		BIC	#102720,R3		:MASK UNRECOVERABLE ERROR
2293	005570	001407			BEQ	W4		:IF SO: BR
2294	005572	004737	022012		JSR	PC,PAPRT		:PRINT HEADER
2295	005576	000004	025331		TYPE,MSG78			:TYPE MSG
2296	005602	004737	010754		JSR	PC,NRTP		:PRINT ER FOR NON-RETRYABLE
2297	005606	000420			BR	W5		
2298	005610	013704	000674	W4:	MOV	UNP,R4		
2299	005614	005264	001044		INC	RTY1(R4)		:BUMP RETRY CNTR
2300	005620	032777	002000	172762	BIT	#2000,@SWR		:SEE IF PRINT ERRORS
2301	005626	001002			BNE	W4A		:IF NOT: BR
2302	005630	000004	025101		TYPE,MSG64			:TYPE MSG
2303	005634	005037	000702	W4A:	CLR	RTCNT		:CLEAR RETRY NUMBER
2304	005640	005037	000700		CLR	RPCNT		:CLEAR REPEAT COUNTER
2305	005644	004737	006166		JSR	PC,WRTY		:GO RETRY WRITE ERROR
2306	005650	005037	000712	W5:	CLR	RTYFL		:CLEAR RETRY COUNTER
2307	005654	005300			DEC	RO		:SEE IF DONE ALL
2308	005656	001245			BNE	W0		:IF NOT: BR
2309	005660	005737	000564	W6:	TST	TMEX		:SEE IF TM
2310	005664	001522			BEQ	WEX		:IF NOT: BR
2311	005666	005237	000676		INC	TMFLG		:SET TM FLAG
2312	005672	012737	025012	000652	WTM:	MOV	#MSG54,EMADDR	:POINT TO TM ERROR MSG
2313	005700	012737	000026	000672	MOV	#26,MTC1		:SET TM OP CODE
2314	005706	005077	172604		CLR	@FC		:LOAD FRAME COUNTER
2315	005712	012777	026344	172574	MOV	#WDATA,@BA		:LOAD BUS ADDRESS
2316	005720	012737	005732	000662	MOV	#WTMO,RTRN		:SAVE RETURN ADDRESS
2317	005726	000137	020372		JMP	TAPG		:WRITE TM
2318	005732	032777	010000	172650	WTMO:	BIT	#10000,@SWR	:SEE IF SHOULD CHECK ERRORS
2319	005740	001074			BNE	WEX		
2320	005742	032777	000004	172552	BIT	#4,@DS		:SEE IF TM STATUS
2321	005750	001011			BNE	WTM1		:IF SO: BR
2322	005752	012737	026344	020224	MOV	#WDATA,CADER		:SET EXPT BUS ADDRESS
2323	005760	012737	000001	020232	MOV	#1,DRVER		:INDICATE ERROR
2324	005766	004737	017352		JSR	PC,ERPT		:PRINT TM ERROR
2325	005772	000404			BR	WTM2		
2326	005774	012703	026344	WTM1:	MOV	#WDATA,R3		:SET EXPT ADDRESS
2327	005780	004737	016614		JSR	PC,ER2		:GO CHECK FOR OTHER ERRORS
2328	005784	005737	000712	WTM2:	TST	RTYFL		:SEE IF RETRY
2329	006010	001401			BEQ	WTM3		:IF NOT: BR
2330	006012	000207			RTS	PC		:ELSE RETURN TO RETRY ROUTINE
2331	006014	005737	000706	WTM3:	TST	SERFL		:SEE IF WRITE ERROR
2332	006020	001444			BEQ	WEX		:IF NOT: BR
2333	006022	013704	000674		MOV	UNP,R4		
2334	006026	005264	001064		INC	WTER1(R4)		:BUMP WRITE ERROR
2335	006032	032777	000020	172550	BIT	#20,@SWR		:SEE IF SHOULD RETRY
2336	006040	001434			BEQ	WEX		:IF NOT: BR

2337	006042	013703	000722		MOV	ERSAV,R3		
2338	006046	042703	102720		BIC	#102720,R3	;MASK UNRECOVERABLE ERROR	
2339	006052	001407			BEQ	WTM4	;IF SO: BR	
2340	006054	004737	022012		JSR	PC,PAPRT	;PRINT HEADER	
2341	006060	000004	025331		TYPE,MSG78		;TYPE MSG	
2342	006064	004737	010754		JSR	PC,NRTP	;PRINT ER FOR NON-RETRYABLE	
2343	006070	000420			BR	WEX		
2344	006072	005037	000700	WTM4:	CLR	RPCNT	;CLEAR REPEAT CNTR	
2345	006076	013704	000674		MOV	UNP,R4		
2346	006102	005264	001044		INC	RTY1(R4)	;BUMP RETRY CNTR	
2347	006106	005037	000702		CLR	RTCNT	;CLEAR RETRY CNTR	
2348	006112	032777	002000	172470	BIT	#2000,@SWR	;SEE IF PRINT ERRORS	
2349	006120	001002			BNE	WTM4A	;IF NOT: BR	
2350	006122	000004	025101		TYPE,MSG64		;TYPE MSG	
2351	006126	004737	006166	WTM4A:	JSR	PC,WRTY	;GO DO RETRY	
2352	006132	005037	000712	WEX:	CLR	RTYFL	;CLEAR RETRY FLAG	
2353	006136	005037	000676		CLR	TMFLG	;CLEAR TAPE MARK FLAG	
2354	006142	005737	000660		TST	EOTREC	;BRANCH IF NOT AT EOT	
2355	006146	100006			BPL	WRWX		
2356	006150	032777	000014	172432	WRW:	BIT	#14,@SWR	;BRANCH IF EITHER READ ENABLED
2357	006156	001002			BNE	WRWX		
2358	006160	000137	004154		JMP	REOT	;ELSE REWIND	
2359	006164	000207		WRWX:	RTS	PC	;EXIT	

```
2361 :*****
2362 :WRITE ERROR RETRY
2363 :*****
2364 :*****
2365 :*****
2366 006166 012737 000001 000712 WRTY: MOV #1,RTYFL ;SET RETRY FLAG
2367 006174 004737 006554 WRTY0: JSR PC,WRTSB ;GO SPACE REVERSE FOR REPEAT
2368 006200 005737 000676 TST TMFLG ;SEE IF TAPE MARK TIME
2369 006204 001003 BNE WRTYTM ;IF SO: BR
2370 006206 004737 005372 JSR PC,W0 ;REWRITE RECORD
2371 006212 000402 BR WRTYR ;GO ON
2372 006214 004737 005672 WRTYTM: JSR PC,WTM ;GO WRITE TAPE MARK AGAIN
2373 006220 005737 000706 WRTYR: TST SERFL ;REWRITE GOOD
2374 006224 001022 BNE WRTY2 ;IF NOT: BR
2375 006226 005237 000700 INC RPCNT ;BUMP REPEAT COUNTER
2376 006232 022737 000004 000700 CMP #4,RPCNT ;SEE IF FOUR GOOD REPEATS
2377 006240 001355 BNE WRTY0 ;IF NOT: REPEAT
2378 006242 032777 002000 172340 BIT #2000,@SWR ;SEE IF PRINT
2379 006250 001007 BNE WRTY1 ;IF NOT: BR
2380 006252 000004 025514 TYPE,MSG105 ;TYPE MSG
2381 006256 000004 025123 TYPE,MSG65 ;TYPE MSG
2382 006262 013703 000702 MOV RTCNT,R3
2383 006266 104400 TYPOCT ;PRINT RETRY NUMBER
2384 006270 000207 WRTY1: RTS PC ;RESUME TESTING
2385 006272 013703 000722 WRTY2: MOV ERSAV,R3 ;GET ER
2386 006276 005037 000650 CLR TEMP3 ;CLEAR RECOVERABLE ERROR INDICATOR
2387 006302 042703 102720 BIC #102720,R3 ;MASK RECOVERABLE BITS
2388 006306 001412 BEQ WRTY2A ;IF RECOVERABLE: BR
2389 006310 004737 022012 JSR PC,PAPRT ;PRINT HEADER
2390 006314 000004 025331 TYPE,MSG78 ;TYPE MSG
2391 006320 004737 010754 JSR PC,NRTP ;PRINT ER
2392 006324 012737 000001 000650 MOV #1,TEMP3 ;SET FLAG
2393 006332 000406 BR WRTY2B
2394 006334 032777 002000 172246 WRTY2A: BIT #2000,@SWR ;SEE I. PRINT
2395 006342 001022 BNE WRTY3 ;IF NOT: BR
2396 006344 000004 025724 TYPE,MSG110 ;TYPE MSG
2397 006350 000004 025123 WRTY2B: TYPE,MSG65 ;TYPE MSG
2398 006354 013703 000702 MOV RTCNT,R3
2399 006360 104400 TYPOCT ;PRINT RETRY NUMBER
2400 006362 000004 025746 TYPE,MSG111 ;TYPE MSG
2401 006366 013703 000700 MOV RPCNT,R3
2402 006372 104400 TYPOCT ;PRINT REPEAT NUMBER
2403 006374 005737 000650 TST TEMP3 ;SEE IF DID NON-RECOVERABLE
2404 006400 001403 BEQ WRTY3 ;IF NOT: BR
2405 006402 005037 000650 CLR TEMP3 ;CLEAR FLAG
2406 006406 000207 RTS PC ;EXIT
2407 006410 005737 000702 WRTY3: TST RTCNT ;SEE IF FIRST RETRY
2408 006414 001004 BNE WRTY3A ;IF NOT: BR
2409 006416 013704 000674 MOV UNP,R4
2410 006422 005364 001064 DEC WTER1(R4) ;DECREMENT WRITE ERROR CNTR
2411 006426 013704 000674 WRTY3A: MOV UNP,R4 ;GET UNIT NUMBER
2412 006432 016437 001024 000730 MOV BTADDR(R4),BTPT ;GET ADDRESS OF UNIT BAD TAPE CNTR
2413 006440 017704 172264 MOV @BTPT,R4 ;GET COUNTER
2414 006444 005724 TST (R4)+ ;SET POINTER OFFSET
2415 006446 010477 172256 MOV R4,@BTPT
2416 006452 013703 000730 MOV BTPT,R3
```

```
2417 006456 060304          ADD      R3,R4          ;SET ABSOLUTE POINTER
2418 006460 013714 000654    MOV      BLCNTR,(R4)    ;SET BLOCK NUMBER
2419 006464 062704 000040    ADD      #40,R4         ;ADD RCNT OFFSET
2420 006470 013714 000554    MOV      RCNT,(R4)
2421 006474 160014          SUB      R0,(R4)        ;SET RECORD NUMBER
2422 006476 005214          INC      (R4)           ;CORRECT RECORD NUMBER
2423 006500 022777 000040 172222  CMP      #40,@BTPT     ;SEE IF TOO MANY BAD SPOTS
2424 006506 001002          BNE     WRTY4          ;IF NOT: BR
2425 006510 000137 006772    JMP      BTOV          ;ELSE GO TO BAD TAPE OVERFLOW
2426 006514 005237 000702    WRTY4:  INC      RTCNT     ;BUMP RETRY COUNTER
2427 006520 022737 000004 000702  CMP      #4,RTCNT     ;SEE IF DONE 4 RETRIES
2428 006526 001410          BEQ     WRTY5          ;IF SO: BR
2429 006530 013704 000674    MOV      UNP,R4
2430 006534 005264 001044    INC      RTY1(R4)      ;BUMP RETRY COUNTER
2431 006540 005237 000732    INC      ERTFL         ;SET ERASE FLAG
2432 006544 000137 006174    JMP      WRTY0         ;DO NEXT RETRY
2433 006550 000137 007176    WRTY5:  JMP      BTUR          ;ELSE GO TO BAD TAPE UNRECOVERABLE
2434
2435          ;WRITE RETRY BACKSPACE-ERASE SUBROUTINE*****
2436
2437 006554 005037 000706    WRTSB:  CLR      SERFL        ;CLEAR FLAG
2438 006560 013737 000600 000666  MOV      TSTAL,STAL
2439 006566 004737 011654    JSR     PC,STALL      ;DO TURN AROUND DELAY
2440 006572 012737 025134 000652  MOV      #MSG66,EMADDR ;SET ERROR CODE
2441 006600 012777 177777 171710  MOV      #-1,@FC      ;SET TO BACKSPACE 1 RECORD
2442 006606 012703 032352    MOV      #RDATA,R3   ;SET EXPECTED BA
2443 006612 010377 171676    MOV      R3,@BA
2444 006616 012737 000032 000672  MOV      #32,MTC1     ;SET BACK SPACE OP CODE
2445 006624 012737 006636 000662  MOV      #1$,PTRN     ;SET RETURN PC
2446 006632 000137 020372    JMP     TAPG          ;EXECUTE BACKSPACE COMMAND
2447 006636 004737 016614    1$:    JSR     PC,ER2       ;CHECK ERRORS
2448 006642 004737 011654    JSR     PC,STALL     ;STALL
2449 006646 005737 000706    TST     SERFL        ;SEE IF ERROR
2450 006652 001406          BEQ     WRTSB1        ;IF NOT: BR
2451 006654 012737 000002 000724  WRTSB0: MOV      #2,BTFLG  ;SET FLAG
2452 006662 022626          CMP     (SP)+,(SP)+  ;RESET STACK
2453 006664 000137 004154    JMP     REOT          ;GO REWIND AND REMOVE FROM TESTING
2454 006670 005737 000732    WRTSB1: TST     ERTFL     ;SEE IF SHOULD ERASE
2455 006674 001001          BNE     WRTSB2        ;IF SO: BR
2456 006676 000207          RTS     PC            ;RETURN
2457 006700 005037 000732    WRTSB2: CLR     ERTFL     ;CLEAR ERASE FLAG
2458 006704 005037 000700    CLR     RPCNT        ;CLEAR REPEAT CNTR
2459 006710 005037 000706    CLR     SERFL        ;CLEAR FLAG
2460 006714 012737 025146 000652  MOV      #MSG67,EMADDR ;SET ERROR CODE
2461 006722 005077 171570    CLR     @FC          ;CLEAR FRAME COUNT
2462 006726 012737 000024 000672  MOV      #24,MTC1     ;SET ERASE OP-CODE
2463 006734 012703 026344    MOV      #WDATA,R3   ;SET EXPECTED BA
2464 006740 010377 171550    MOV      R3,@BA
2465 006744 012737 006756 000662  MOV      #1$,PTRN     ;SET RETURN ADDRESS
2466 006752 000137 020372    JMP     TAPG          ;GO ERASE
2467 006756 004737 016614    1$:    JSR     PC,ER2       ;GO CHECK ERRORS
2468 006762 005737 000706    TST     SERFL        ;SEE IF ERROR
2469 006766 001740          BEQ     WRTSB1        ;IF NOT: BR
2470 006770 000731          BR      WRTSB0
2471
2472          ;BAD TAPE OVERFLOW SUBROUTINE*****
```

```
2473
2474 006772 005037 000712
2475 006776 012737 000001 000724
2476 007004 005726
2477 007006 000137 004154
2478 007012 013701 000730
2479 007016 005721
2480 007020 005000
2481 007022 010003
2482 007024 000241
2483 007026 006003
2484 007030 104400
2485 007032 000004 023610
2486 007036 011103
2487 007040 104400
2488 007042 000004 023615
2489 007046 062701 000040
2490 007052 012103
2491 007054 104400
2492 007056 162701 000040
2493 007062 005720
2494 007064 020077 171640
2495 007070 001403
2496 007072 000004 024155
2497 007076 000751
2498 007100 005737 000726
2499 007104 001007
2500 007106 012703 000041
2501 007112 013704 000730
2502 007116 005024
2503 007120 005303
2504 007122 001375
2505 007124 000207
2506

BTOV: CLR RTYFL ;CLEAR RETRY FLAG
      MOV #1,BTFLG ;SET BAD TAPE OVERFLOW FLAG
      TST (SP)+ ;++B ADJUST STACK PTR
      JMP REOT ;GO REWIND AND REMOVE FROM TESTING
BTOV0: MOV BTPT,R1 ;SET TABLE POINTER
      TST (R1)+
      CLR R0
BTOV1: MOV R0,R3
      CLC
      ROR R3 ;R3=R3/2 FOR CORRECT NUMBER
      TYPOCT ;PRINT ENTRY NUMBER
      TYPE,MSG13+1 ;TYPE MSG
      MOV (R1),R3
      TYPGCT ;PRINT BLOCK NUMBER
      TYPE,MSG14 ;TYPE MSG
      ADD #40,R1 ;SET POINTER OFFSET FOR RECOED NUMBER
      MOV (R1)+,R3
      TYPOCT ;PRINT RECORD NUMBER
      SUB #40,R1 ;RESET POINTER FOR BLOCK NUMBER
      TST (R0)+
      CMP R0,BTPT ;SEE IF DONE
      BEQ BTOV2 ;IF SO: BR
      TYPE '<CR><LF>' ;TYPE '<CR><LF>'
      BR BTOV1 ;CONTINUE
BTOV2: TST BTSTF ;SEE IF STAT ONLY PRINT
      BNE BTOVX ;IF SO: BR
      MOV #41,R3 ;SET SIZE OF TABLE
      MOV BTPT,R4 ;SET POINTER
      CLR (R4)+ ;CLEAR TABLE
      DEC R3 ;SEE IF DONE
      BNE BTOV3 ;IF NOT: BR
BTOVX: RTS PC ;RETURN
```

```
2508
2509 ;BAD TAPE STATISTIC PRINT*****
2510
2511 007126 000004 024155 BTPRT: TYPE,MSG28 ;TYPE '<CR><LF>'
2512 007132 013704 000674 MOV UNP,R4
2513 007136 016437 001024 000730 MOV BTADDR(R4),BTPT ;SET TABLE POINTER
2514 007144 017703 171560 MOV @BTPT,R3
2515 007150 000241 CLC
2516 007152 006003 ROR R3 ;CORRECT NUMBER
2517 007154 104400 TYPOCT ;PRINT NUMBER OF BAD SPOTS
2518 007156 000004 025760 TYPE,MSG112 ;TYPE MSG
2519 007162 005777 171542 TST @BTPT ;SEE IF ANY BAD SPOTS
2520 007166 001001 BNE BTPRT1 ;IF SO: BR
2521 007170 000207 RTS PC ;ELSE RETURN
2522 007172 000137 007012 BTPRT1: JMP BTOV0 ;PRINT STATS
2523
2524 ;BAD TAPE UNRECOVERABLE SUBROUTINE*****
2525
2526 007176 004737 022012 BTUR: JSR PC,PAPRT ;PRINT HEADER
2527 007202 000004 025613 TYPE,MSG107 ;TYPE MSG
2528 007206 000207 RTS PC ;RESUME TESTING
2529
```



```
2531 :*****
2532 :READ SEQUENCER:
2533 :
2534 :THIS ROUTINE IS USED TO DETERMINE THE SEQUENCE
2535 :IN WHICH READ TAPE OPERATIONS ARE TO BE PERFORMED.
2536 :THIS IS NECESSARY WHEN THE UNIT BEING TESTED IS
2537 :CAPABLE OF READING DATA IN BOTH THE FORWARD AND
2538 :REVERSE DIRECTIONS. CONSOLE SWITCHES ONE (1), TWO (2),
2539 :AND THREE (3) ARE USED TO DETERMINE THE READ SEQUENCE.
2540 :CONSOLE SWITCH ONE (1) DETERMINES WHETHER TO READ
2541 :THE BLOCK OF DATA FORWARD FIRST OR REVERSE FIRST.
2542 :SWITCH TWO (2) DISALLOWS READING IN THE REVERSE
2543 :DIRECTION AND SWITCH THREE (3) DISALLOWS READING IN
2544 :THE FORWARD DIRECTION.
2545 :*****
2546
2547 007210 005037 000562 RSEQ: CLR RDCMD
2548 007214 017704 171370 MOV @SWR,R4 ;READ SWITCHES
2549 007220 042704 177763 BIC #177763,R4 ;MASK READ BITS & SEE IF BOTH READS
2550 007224 001004 BNE RSR ;IF NOT: BR
2551 007226 032777 000002 171354 BIT #2,@SWR ;SEE IF READ REVERSE FIRST
2552 007234 001041 BNE RSFR ;IF NOT: BR
2553 007236 032777 000004 171344 RSR: BIT #4,@SWR ;SEE IF SHOULD READ REVERSE
2554 007244 001005 BNE RSF ;IF NOT: BR
2555 007246 012737 000001 000562 MOV #1,RDCMD ;LOAD READ REVERSE COMMAND
2556 007254 004737 007464 JSR PC,READ ;GO READ REVERSE
2557 007260 032777 000010 171322 RSF: BIT #10,@SWR ;SEE IF SHOULD READ FORWARD
2558 007266 001066 BNE RSEX ;IF NOT: BR
2559 007270 005737 000562 TST RDCMD ;SEE IF HAVE READ REVERSE
2560 007274 001406 BEQ RSFO ;IF NOT: BR
2561 007276 013737 000600 000666 MOV TSTAL,STAL
2562 007304 004737 011654 JSR PC,STALL ;DO READ STALL
2563 007310 000406 BR RSF1
2564 007312 032777 000001 171270 RSFO: BIT #1,@SWR ;SEE IF WRITE
2565 007320 001002 BNE RSF1 ;IF NOT: BR
2566 007322 004737 011402 JSR PC,BKSP ;GO BACKSPACE
2567 007326 005037 000562 RS^1: CLR RDCMD ;LOAD READ FORWARD COMMAND
2568 007332 004737 007464 JSR PC,READ ;GO READ
2569 007336 000442 BR RSEX ;GO TO EXIT
2570
2571 007340 012737 000001 000562 RSFR: MOV #1,RDCMD
2572 007346 032777 000010 171234 BIT #10,@SWR ;SEE IF SHOULD READ FORWARD
2573 007354 001012 BNE RSFR1 ;IF NOT: BR
2574 007356 032777 000001 171224 BIT #1,@SWR ;SEE IF WRITE
2575 007364 001002 BNE RSFR0 ;IF NOT: BR
2576 007366 004737 011402 JSR PC,BKSP ;GO BACKSPACE TO START
2577 007372 005037 000562 RSFR0: CLR RDCMD ;LOAD READ FORWARD COMMAND
2578 007376 004737 007464 JSR PC,READ ;GO READ FORWARD
2579 007402 032777 000004 171200 RSFR1: BIT #4,@SWR ;SEE IF SHOULD READ REVERSE
2580 007410 001015 BNE RSEX ;IF NOT: BR
2581 007412 005737 000562 TST RDCMD
2582 007416 001005 BNE RSFR2 ;IF READ REVERSE: BR
2583 007420 013737 000600 000666 MOV TSTAL,STAL ;DO READ STALL
2584 007426 004737 011654 JSR PC,STALL
2585 007432 012737 000001 000562 RSFR2: MOV #1,RDCMD ;LOAD READ REVERSE
2586 007440 004737 007464 JSR PC,READ ;GO READ REVERSE
```

CZTEDDO TMO3-TE16/TU77 DRT  
CZTEDD.P11 06-JUL-83 14:42

MACY11 30(1046) 06-JUL-83 21:03 F 5  
PAGE 39-1

SEQ 0057

2587 007444 005037 000562  
2588 007450 005737 000660  
2589 007454 001402  
2590 007456 000137 004154  
2591 007462 000207  
2592

RSEX: CLR RDCMD  
TST EOTREC  
BEQ RSFRX  
JMP REOT  
RSFRX: RTS PC

;BRANCH IF EOT NOT REACHED  
;REWIND AND REPORT STATS  
;EXIT

2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645  
2646  
2647  
2648  
2649

```
*****  
:READ ROUTINE:  
:  
:THIS ROUTINE PERFORMS THE READ OPERATION DETERMINED  
:BY THE READ SEQUENCE ROUTINE ONE RECORD AT A TIME.  
:AT THE END OF EACH READ OPERATION THE STATUS REGISTER  
:IS SCANNED FOR EITHER END OF TAPE OR BEGINNING OF TAPE.  
:IF EOT WAS REACHED, CONTROL WILL BE PASSED TO  
:THE EOT SUBROUTINE TO REWIND THE UNIT AND FLAG IT  
:UNAVAILABLE UNTIL ALL UNITS HAVE REACHED EOT.  
:IF BOT WAS REACHED AN ERROR IS PRINTED AND THE  
:PROGRAM WILL HALT. TESTING MAY BE RESUMED BY PRESSING  
:THE CONTINUE SWITCH.  
:IF A TAPE MARK IS EXPECTED (TM=1) THEN THE  
:READ ROUTINE EXPECTS THE FIRST RECORD OF A  
:READ REVERSE TO BE A TM, AND THE LAST RECORD  
:OF A READ FORWARD TO BE A TM. REMEMBER  
:THAT THE TM ADDS ONE (1) TO THE TOTAL NUMBER  
:OF RECORDS IN A BLOCK.  
:CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13) DETERMINE WHETHER  
:OR NOT TO CHECK FOR STATUS ERRORS (11) OR DATA ERRORS (13).  
:CONSOLE SWITCH FIVE (5) IS USED TO CAUSE A CONTINUOUS  
:READ AND SPACE (FORWARD OR REVERSE) OF THE CURRENT  
:RECORD ON TAPE (YOZZLE).  
*****
```

```
007464 013700 000554 READ: MOV RCNT,R0 ;LOAD REC CNTR  
007470 005737 000660 TST EOTREC ;SEE IF EOT  
007474 100012 BPL RDA ;IF NOT: BR  
007476 005737 000562 TST RDCMD ;SEE IF READ FORWARD  
007502 001407 BEQ RDA ;IF SO: BR  
007504 042737 100000 000660 BIC #100000,EOTREC ;CLEAR FLAG  
007512 013703 000660 MOV EOTREC,R3 ;GET MODIFIED RECORD COUNT  
007516 160300 SUB R3,R0 ;SET RECORD AT  
007520 005200 INC R0 ;SET TO PROPER NUMBER OF RECORDS  
007522 012737 023553 000652 RDA: MOV #MSG6,EMADDR ;SET ERROR MSG ADDRESS  
007530 005037 000676 CLR TMFLG  
007534 005737 000562 TST RDCMD  
007540 001406 BEQ RDO ;IF READ FORWARD: BR  
007542 005737 000564 TST TMEX ;SEE IF TM  
007546 001403 BEQ RDO ;IF NOT: BR  
007550 005237 000676 INC TMFLG ;SET TM FLAG  
007554 005200 INC R0  
007556 013777 000556 170732 RDO: MOV FMCNT,@FC ;LOAD CHAR CNTR  
007564 012777 032352 170722 MOV #RDATA,@BA ;LOAD DATA ADDR  
007572 005737 000562 TST RDCMD ;SEE IF READ REVERSE  
007576 001417 BEQ RD1A ;IF NOT: BR  
007600 013703 000556 MOV FMCNT,R3  
007604 005103 COM R3  
007606 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP  
007614 001402 BEQ RD1 ;IF NOT: BR  
007616 000241 CLC  
007620 006003 ROR R3 ;R3 = FC/2  
007622 060377 170666 RD1: ADD R3,@BA ;SET REVERSE BUS ADDRESS  
007626 012737 000076 000672 MOV #76,MTC1 ;SET READ REVERSE  
007634 000403 BR RD1B
```

2650	007636	012737	000070	000672	RD1A:	MOV	#70,MTC1	:SET READ FORWARD
2651	007644	012737	007656	000662	RD1B:	MOV	#RD2,RTM	:SET INTERRUPT RETURN ADDRESS
2652	007652	000137	020372			JMP	TAPG	:GO EXECUTE TAPE COMMAND
2653	007656	005737	000562		RD2:	TST	RDCMD	:IGNORE EOT IF READ REVERSE
2654	007662	001014				BNE	RD3	
2655	007664	032777	002000	170630		BIT	#2000,ADS	:SEE IF EOT
2656	007672	001410				BEQ	RD3	:IF NOT: BR
2657	007674	005737	000676			TST	TMFLG	:SEE IF TM
2658	007700	001005				BNE	RD3	:IF SO: BR
2659	007702	010037	000660			MOV	R0,EOTREC	:GET # OF RECORDS LEFT IN BLOCK TO READ
2660	007706	052737	100000	000660		BIS	#100000,EOTREC	:SET EOT FLAG
2661	007714	032777	000002	170600	RD3:	BIT	#2,ADS	:SEE IF AT LOAD POINT
2662	007722	001407				BEQ	RD4	:IF NOT: BR
2663	007724	004737	022012			JSR	PC,PAPRT	:PRINT CYCLE NUMBER
2664	007730	000004	023713			TYPE,MSG22		:TYPE MSG
2665	007734	000000				HALT		
2666	007736	000137	003144			JMP	STARTA	:RESTART
2667	007742	032777	004000	170640	RD4:	BIT	#4000,ASW	:SEE IF SHOULD CHECK ERRORS
2668	007750	001116				BNE	RD5	:IF NOT: BR
2669	007752	005737	000676			TST	TMFLG	
2670	007756	001470				BEQ	RD4B	:IF NO TM EXPT: BR
2671	007760	032777	000004	170534		BIT	#4,ADS	
2672	007766	001023				BNE	RD4A	:IF TM RECVD: BR
2673	007770	012737	032352	020224		MOV	#RDATA,CADER	:SAVE EXPT BUS ADDRESS
2674	007776	012737	000002	020232		MOV	#2,DRVER	:SET TM STATUS ERROR FLAG
2675	010004	004737	017352			JSR	PC,ERPT	:GO PRINT TM ERROR
2676	010010	013704	000674			MOV	UNP,R4	
2677	010014	005737	000562			TST	RDCMD	:SEE IF READ REVERSE
2678	010020	001403				BEQ	1\$	:IF NOT: BR
2679	010022	005264	001144			INC	RDERR1(R4)	:BUMP READ REVERSE ERROR
2680	010026	000500				BR	RD6	
2681	010030	005264	001104		1\$:	INC	RDER1(R4)	:BUMP READ FORWARD ERROR
2682	010034	000475				BR	RD6	
2683	010036	012703	032352		RD4A:	MOV	#RDATA,R3	
2684	010042	005737	000562			TST	RDCMD	:SEE IF READ REVERSE
2685	010046	001007				BNE	RD4A0	:IF SO: BR
2686	010050	032737	002000	000552		BIT	#2000,UDES	:SEE IF IN PE
2687	010056	001025				BNE	RD4A2	:IF SO: BR
2688	010060	062703	000002			ADD	#2,R3	
2689	010064	000422				BR	RD4A2	
2690	010066	013704	000556		RD4A0:	MOV	FMCNT,R4	
2691	010072	005104				COM	R4	
2692	010074	032737	000020	000552		BIT	#20,UDES	:SEE IF CORE DUMP
2693	010102	001402				BEQ	RD4A1	:IF NOT: BR
2694	010104	000241				CLC		
2695	010106	006004				ROR	R4	:SET TO FC/2
2696	010110	060403			RD4A1:	ADD	R4,R3	:SET EXPT BUS ADDRESS
2697	010112	042703	000001			BIC	#1,R3	:MAKE EXPT ADDRESS EVEN
2698	010116	032737	002000	000552		BIT	#2000,UDES	:SEE IF IN PE
2699	010124	001002				BNE	RD4A2	:IF SO: BR
2700	010126	162703	000002			SUB	#2,R3	
2701	010132	004737	016614		RD4A2:	JSR	PC,ER2	
2702	010136	000402				BR	RD4C	
2703	010140	004737	016522		RD4B:	JSR	PC,ERCHK	:GO CHECK ERRORS
2704	010144	005737	000706		RD4C:	TST	SERFL	
2705	010150	001416				BEQ	RD5	:IF NO ERROR: BR

2706	010152	013704	000674		MOV	UNP,R4	
2707	010156	005737	000562		TST	RDCMD	:SEE IF READ REVERSE
2708	010162	001003			BNE	RD4D	:IF SO: BR
2709	010164	005264	001104		INC	RDER1(R4)	:BUMP READ FORWARD ERROR
2710	010170	000402			BR	RD4E	
2711	010172	005264	001144	RD4D:	INC	RDERR1(R4)	:BUMP READ REVERSE ERROR
2712	010176	004737	010374	RD4E:	JSR	PC,RDRTY	:GO RETRY
2713	010202	005037	000712		CLR	RTYFL	:CLEAR RETRY FLAG
2714	010206	032777	020000	170374	RD5:	BIT	#20000,@SWR
2715	010214	001005			BNE	RD6	:SEE IF SHOULD DO DATA CHECK
2716	010216	005737	000676		TST	TMFLG	:IF NOT; BR
2717	010222	001002			BNE	RD6	
2718	010224	004737	014750		JSR	PC,DCHK	:GO CHECK DATA
2719	010230	005037	000706	RD6:	CLR	SERFL	:CLEAR STATUS ERROR FLAG
2720	010234	004737	013614		JSR	PC,DS3	:CLEAR BUFFER
2721	010240	032777	000040	170342	BIT	#40,@SWR	:SEE IF SHOULD YOZZLE
2722	010246	001402			BEQ	RD7	:IF NOT: BR
2723	010250	004737	010770		JSR	PC,YOZ	:ELSE GO YOZZLE
2724	010254	013737	000574	000666	RD7:	MOV	RSTAL,STAL
2725	010262	004737	011654		JSR	PC,STALL	:SET DELAY
2726	010266	005737	000562		TST	RDCMD	:STALL
2727	010272	001403			BEQ	RD7A	:SEE IF READ REVERSE
2728	010274	005037	000676		CLR	TMFLG	:IF NOT: BR
2729	010300	000405			BR	RD10	:CLEAR TAPE MARK FLAG
2730	010302	005737	000660	RD7A:	TST	EOTREC	:SEE IF EOT FOUND
2731	010306	100002			BPL	RD10	:IF NOT: BR
2732	010310	012700	000001		MOV	#1,R0	:SET TO EOT
2733	010314	005300		RD10:	DEC	R0	
2734	010316	001402			BEQ	RD11	:IF DONE ALL: BR
2735	010320	000137	007556		JMP	RDO	
2736	010324	005737	000562	RD11:	TST	RDCMD	:SEE IF READ REVERSE
2737	010330	001016			BNE	RDEX	:IF SO: BR
2738	010332	005737	000660		TST	EOTREC	:SEE IF FOUND EOT
2739	010336	100413			BMI	RDEX	:IF SO: BR
2740	010340	005737	000564		TST	TMEX	:SEE IF TM EXPECTED
2741	010344	001410			BEQ	RDEX	:IF NOT: BR
2742	010346	005737	000676		TST	TMFLG	:SEE IF TM FOUND
2743	010352	001005			BNE	RDEX	:IF SO: BR
2744	010354	005237	000676		INC	TMFLG	:ELSE SET FLAG
2745	010360	005200			INC	R0	:SET RECORD COUNT TO ONE
2746	010362	000137	007556		JMP	RDO	:GO READ TM
2747	010366	005037	000676	RDEX:	CLR	TMFLG	
2748	010372	000207		RDX:	RTS	PC	:EXIT

2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765  
2766  
2767  
2768  
2769  
2770  
2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789  
2790  
2791  
2792  
2793  
2794  
2795  
2796  
2797  
2798  
2799  
2800  
2801  
2802  
2803  
2804  
2805

```
*****  
:READ ERROR RETRY SUBROUTINE:  
:  
:THIS SUBROUTINE WILL RETRY ALL DATA RELATED  
:READ ERRORS UP TO EIGHT (8) TIMES. IF ALL  
:FOUR RETRIES ARE BAD, IT IS CONSIDERED  
:A HARD ERROR. IF ANY ARE GOOD, IT IS A  
:SOFT ERROR. RETRIES MAY BE INHIBITED  
:VIA SWITCH FOUR (SW4=0: INHIBIT RETRIES)  
:*****  
C10374 032777 000020 170206 RDRTY: BIT #20,@SWR ;SEE IF RETRY INHIBITED  
010402 001001 BNE RDRT0 ;IF NOT: BR  
010404 000207 RTS PC ;ELSE RETURN  
010406 013703 000722 RDRT0: MOV ERSAV,R3  
010412 022703 100000 CMP #100000,R3 ;++B BRANCH IF OTHER THAN CORRECTED READ ERROR  
010416 001011 BNE 1$ ;++B  
010420 032777 000040 170074 BIT #40,@DS ;++B BRANCH IF NRZ  
010426 001405 BEQ 1$ ;++B  
010430 005037 000706 CLR SERFL ;++B CLEAR ERROR FLAG  
010434 000004 026304 TYPE,MSG124 ;++B TYPE 'CORRECTED PE DATA ERROR'  
010440 000447 BR RDRT2 ;++B INC SOFT COUNTS  
010442 042703 102720 1$: BIC #102720,R3 ;MARK NON-RECOVERABLE ERROR BITS  
010446 001407 BEQ RDRT1 ;IF NOT: BR  
010450 004737 022012 JSR PC,PAPRT ;PRINT HEADER  
010454 000004 025371 TYPE,MSG79 ;TYPE MSG  
010460 004737 010754 JSR PC,NRTP ;PRINT ER FOR NON-RETRYABLE ERROR  
010464 000207 RTS PC ;RETURN  
010466 032777 002000 170114 RDRT1: BIT #2000,@SWR ;SEE IF PRINT INHIBITED  
010474 001002 BNE RDRT1B ;IF SO: BR  
010476 000004 025101 TYPE,MSG64 ;TYPE MSG  
010502 005037 000702 RDRT1B: CLR RTCNT ;CLEAR RETRY COUNTER  
010506 005037 000706 RDRTG: CLR SERFL ;CLEAR STATUS ERROR FLAG  
010512 012737 000002 000712 MOV #2,RTYFL ;SET READ RETRY FLAG  
010520 004737 010770 JSR PC,YOZ ;GO TO YOZZLE TO RETRY READ  
010524 005737 000706 TST SERFL ;SEE IF RETRY ERROR  
010530 001026 BNE RDRT5 ;IF SO: BR  
010532 032777 002000 170050 BIT #2000,@SWR  
010540 001007 BNE RDRT2  
010542 000004 025514 TYPE,MSG105 ;TYPE MSG  
010546 000004 025123 TYPE,MSG65 ;TYPE MSG  
010552 013703 000702 MOV RTCNT,R3  
010556 104400 TYPOCT ;PRINT RETRY NUMBER  
010560 013704 000674 RDRT2: MOV UNP,R4  
010564 005737 000562 TST RDCMD ;SEE IF READ REVERSE  
010570 001003 BNE RDRT3 ;IF SO: BR  
010572 005264 002664 INC RFSOFT(R4) ;ELSO BUMP FORWARD SOFT ERROR COUNTER  
010576 000402 BR RDRT4  
010600 005264 002704 RDRT3: INC RRSOFT(R4) ;BUMP ERRORS SOFT CNTR  
010604 000207 RDRT4: RTS PC ;RETURN  
010606 013703 000722 RDRT5: MOV ERSAV,R3 ;GET ER  
010612 005037 000650 CLR TEMP3 ;CLEAR RECOVERABLE ERROR INDICATOR  
010616 042703 102720 BIC #102720,R3 ;MASK RECOVERABLE BITS  
010622 001412 BEQ RDRT5A ;IF RECOVERABLE: BR  
010624 004737 022012 JSR PC,PAPRT ;PRINT HEADER
```

```
2806 010630 000004 025371 TYPE,MSG79 ;TYPE MSG
2807 010634 004737 010754 JSR PC,NRTP ;PRINT ER
2808 010640 012737 000001 000650 MOV #1,TEMP3 ;SET FLAG
2809 010646 000404 BR RDRT5B
2810 010650 032777 002000 167732 RDRT5A: BIT #2000,@SWR ;SEE IF PRINT INHIBITED
2811 010656 001013 BNE RDRT6 ;IF SO: BR
2812 010660 000004 025123 RDRT5B: TYPE,MSG65 ;TYPE MSG
2813 010664 013703 000702 MOV RTCNT,R3
2814 010670 104400 TYPOCT ;PRINT RETRY NUMBER
2815 010672 005737 000650 TST TEMP3 ;SEE IF DID NON-RECOVERABLE
2816 010676 001403 BEQ RDRT6 ;IF NOT: BR
2817 010700 005037 000650 CLR TEMP3 ;CLEAR FLAG
2818 010704 000207 RTS PC ;EXIT
2819 010706 005237 000702 RDRT6: INC RTCNT
2820 010712 023737 000702 000604 CMP RTCNT,RETRY ;SEE IF DONE 8 RETRIES
2821 010720 001272 BNE RDRTG ;IF NOT: BR
2822 010722 000004 026023 TYPE,MSG115 ;TYPE MSG
2823 010726 013704 000674 MOV UNP,R4
2824 010732 005737 000562 TST RDCMD ;SEE IF READ REVERSE
2825 010736 001003 BNE RDRT7 ;IF SO: BR
2826 010740 005264 002724 INC RFHARD(R4) ;BUMP FORWARD HARD ERROR CNTR
2827 010744 000402 BR RDRTX
2828 010746 005264 002744 RDRT7: INC RRHARD(R4) ;BUMP REVERSE HARD ERROR CNTR
2829 010752 000207 RDRTX: RTS PC ;RETURN
2830
2831 010754 013703 000722 NRTP: MOV ERSAV,R3 ;GET ER REGISTER
2832 010760 104400 TYPOCT ;PRINT ER
2833 010762 004737 020250 JSR PC,FRPRT ;PRINT F OR R
2834 010766 000207 RTS PC ;RETURN
2835
2836 ;*****
2837 ;YOZZLE SUBROUTINE:
2838 ;
2839 ;THIS SUBROUTINE, ENTERED VIA SWITCH FIVE (5), IS USED TO PERFORM
2840 ;A CONTINUOUS READ AND SPACE OVER OF THE CURRENT RECORD ON TAPE.
2841 ;FULL STATUS AND DATA CHECKING MAY BE PERFORMED
2842 ;OR NOT VIA CONSOLE SWITCHES ELEVEN (11) AND THIRTEEN (13).
2843 ;A SOFTWARE DELAY IS PERFORMED BETWEEN EACH READ
2844 ;AND SPACE OPERATION AND MAY BE VARIED BY TYPING
2845 ;CNTRL C ON THE TTY AND ENTERING A VALUE IN RESPONSE
2846 ;TO THE PRINTED REQUEST.
2847 ;*****
2848 010770 013737 000602 000666 YOZ: MOV YSTAL,STAL
2849 010776 004737 011654 JSR PC,STALL ;DO YOZZLE STALL
2850 011002 012777 177777 167506 YOZO: MOV #-1,@FC ;SET TO 1 RECORD SPACING
2851 011010 005737 000562 TST RDCMD ;SEE IF READ REVERSE
2852 011014 001404 BEQ YOZA ;IF NOT: BR
2853 011016 112737 000030 000672 MOVB #30,MTC1 ;SET TO SPACE FORWARD
2854 011024 000403 BR YOZB
2855 011026 112737 000032 000672 YOZA: MOVB #32,MTC1 ;SET TO SPACE REVERSE
2856 011034 012737 011054 000662 YOZB: MOV #YOZC,RTRN ;SET RETURN ADDRESS
2857 011042 012737 177775 000666 MOV #177775,STAL ;SET TIME MULTIPLIER
2858 011050 000137 020372 JMP TAPG ;GO YOZZLE
2859 011054 005737 000676 YOZC: TST TMFLG ;SEE IF TM
2860 011060 001404 BEQ 1$ ;IF NOT: BR
2861 011062 012737 040000 000666 MOV #40000,STAL ;SET TM STALL
```

2862	011070	000403				BR	2\$		
2863	011072	013737	000602	000666	1\$:	MOV	YSTAL,STAL		
2864	011100	004737	011654		2\$:	JSR	PC,STALL		:DO YOZZLE STALL
2865	011104	012777	032352	167402		MOV	#RDATA,@BA		:SET BUS ADDRESS
2866	011112	005737	000562			TST	RDCMD		:SEE IF READ REVERSE
2867	011116	001416				BEQ	YOZC1		:IF NOT: BR
2868	011120	013703	000556			MOV	FMCNT,R3		
2869	011124	005103				COM	R3		
2870	011126	032737	000020	000552		BIT	#20,UDES		:SEE IF CORE DUMP
2871	011134	001401				BEQ	YOZC0		:IF NOT: BR
2872	011136	006203				ASR	R3		:R3 = FC/2
2873	011140	060377	167350		YOZC0:	ADD	R3,@BA		:SET REVERSE BUS ADDRESS
2874	011144	012737	000076	000672		MOV	#76,MTC1		:SET READ REVERSE
2875	011152	000403				BR	YOZC2		
2876	011154	012737	000070	000672	YOZC1:	MOV	#70,MTC1		:SET READ FORWARD
2877	011162	013777	000556	167326	YOZC2:	MOV	FMCNT,@FC		:SET CHARACTER COUNT
2878	011170	012737	011202	000662		MOV	#YOZD,RTRN		:SET RETURN ADDRESS
2879	011176	000137	020372			JMP	TAPG		:GO READ
2880	011202	032777	004000	167400	YOZD:	BIT	#4000,@SWR		:SEE IF SHOULD CHECK ERRORS
2881	011210	001047				BNE	YOZE		:IF NOT: BR
2882	011212	005737	000676			TST	TMFLG		:SEE IF TAPE MARK TIME
2883	011216	001442				BEQ	YOZD1		:IF NOT: BR
2884	011220	005737	000562			TST	RDCMD		:SEE IF READ REVERSE
2885	011224	001425				BEQ	YOZD0		:IF NOT: BR
2886	011226	012703	032352			MOV	#RDATA,R3		
2887	011232	013704	000556			MOV	FMCNT,R4		
2888	011236	005104				COM	R4		
2889	011240	032737	000020	000552		BIT	#20,UDES		:SEE IF CORE DUMP
2890	011246	001401				BEQ	YOZD4		:IF NOT: BR
2891	011250	006204				ASR	R4		:SET TO FC/2
2892	011252	060403			YOZD4:	ADD	R4,R3		:SET EXPT BUS ADDRESS
2893	011254	042703	000001			BIC	#1,R3		:MAKE EXPT ADDRESS EVEN
2894	011260	032737	002000	000552		BIT	#2000,UDES		:SEE IF PE
2895	011266	001001				BNE	YOZD2		:IF SO: BR
2896	011270	005743				TST	-(R3)		:SET EXPT BA
2897	011272	004737	016614		YOZD2:	JSR	PC,ER2		:GO CHECK ERRORS
2898	011276	000430				BR	YOZF		
2899	011300	012703	032352		YOZD0:	MOV	#RDATA,R3		
2900	011304	032737	002000	000552		BIT	#2000,UDES		:SEE IF PE
2901	011312	001001				BNE	YOZD3		:IF SO: BR
2902	011314	005723				TST	(R3)+		:SET EXPT BA
2903	011316	004737	016614		YOZD3:	JSR	PC,ER2		:GO CHECK ERRORS
2904	011322	000416				BR	YOZF		
2905	011324	004737	016522		YOZD1:	JSR	PC,ERCHK		:ELSE GO CHECK ERRORS
2906	011330	005737	000712		YOZE:	TST	RTYFL		:SEE IF RETRY
2907	011334	001013				BNE	YOZG		:IF SO: BR
2908	011336	032777	020000	167244		BIT	#20000,@SWR		:SEE IF SHOULD CHECK DATA
2909	011344	001005				BNE	YOZF		:IF NOT: BR
2910	011346	005737	000676			TST	TMFLG		:SEE IF TAPE MARK
2911	011352	001002				BNE	YOZF		:IF SO: BR
2912	011354	004737	014750			JSR	PC,DCHK		:ELSE GO CHECK DATA
2913	011360	004737	013614		YOZF:	JSR	PC,DS3		:GO CLEAR DATA AREA
2914	011364	032777	000040	167216	YOZG:	BIT	#40,@SWR		:SEE IF SHOULD CONTINUE YOZZLE
2915	011372	001402				BEQ	YOZH		:IF NOT: BR
2916	011374	000137	C11002		YOZH:	JMP	YOZO		
2917	011400	000207				RTS	PC		:EXIT



```
2919  
2920  
2921  
2922  
2923  
2924  
2925  
2926  
2927  
2928  
2929  
2930  
2931  
2932  
2933  
2934  
2935 011402 013737 000600 000666 BKSP: MOV TSTAL,STAL  
2936 011410 004737 011654 JSR PC,STALL ;DO TURN AROUND STALL  
2937 011414 012737 023602 000652 MOV #MSG10,EMADDR  
2938 011422 012703 032352 MOV #RDATA,R3 ;SET EXPECTED BA  
2939 011426 010377 167062 MOV R3,@BA  
2940 011432 005737 000564 TST TMEX ;SEE IF TM  
2941 011436 001436 BEQ B0 ;IF NOT: BR  
2942 011440 012777 177777 167050 MOV #-1,@FC  
2943 011446 012737 000032 000672 MOV #32,MTC1  
2944 011454 012737 011466 000662 MOV #1$,RTRN  
2945 011462 000137 020372 JMP TAPG ;SPACE TO TM  
2946 011466 032777 010000 167114 1$: BIT #10000,@SWR ;SEE IF SHOULD CHECK ERROR  
2947 011474 001017 BNE B0 ;IF NOT: BR  
2948 011476 012737 025021 000652 MOV #MSG55,EMADDR  
2949 011504 032777 000004 167010 BIT #4,@DS ;SEE IF TM  
2950 011512 001006 BNE 2$ ;IF SO: BR  
2951 011514 012737 032352 020224 MOV #RDATA,CADER  
2952 011522 004737 017352 JSR PC,ERPT ;PRINT ERROR  
2953 011526 000402 BR B0  
2954 011530 004737 016614 2$: JSR PC,ER2  
2955 011534 013700 000554 B0: MOV RCNT,R0  
2956 011540 005737 000660 TST EOTREC ;BRANCH IF EOT NOT DETECTED  
2957 011544 100007 BPL 1$  
2958 011546 042737 100000 000660 BIC #100000,EOTREC ;CLEAR EOT INDICATOR  
2959 011554 013703 000660 MOV EOTREC,R3 ;GET # OF RECORDS LEFT IN BLOCK  
2960 011560 160300 SUB R3,R0 ;FORM # OF RECORDS TO BACK SPACE  
2961 011562 005200 INC R0  
2962 011564 012737 023602 000652 1$: MOV #MSG10,EMADDR ;SET ERROR MESH ADDRESS  
2963 011572 012737 011630 000662 MOV #2$,RTRN ;SET RETURN PC  
2964 011600 012777 177777 166710 MOV #-1,@FC ;SET TO BACKSPACE 1 RECORD  
2965 011606 012703 032352 MOV #RDATA,R3 ;SET EXPECTED BA  
2966 011612 010377 166676 MOV R3,@BA  
2967 011616 012737 000032 000672 MOV #32,MTC1 ;SET SPACE REVERSE  
2968 011624 000137 020372 JMP TAPG ;GO DO SPACE  
2969 011630 004737 016614 2$: JSR PC,ER2  
2970 011634 013737 000600 000666 MOV TSTAL,STAL ;DO STALL  
2971 011642 004737 011654 JSR PC,STALL ;STALL  
2972 011646 005300 DEC R0 ;DECREMENT # OF RECORD TO BACKSPACE  
2973 011650 001345 BNE 1$  
2974 011652 000207 RTS PC ;EXIT
```

2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991  
2992  
2993  
2994  
2995  
2996

011654 005337 000666  
011660 001375  
011662 000207

STALL: DEC STAL  
BNE STALL ;DELAY  
RTS PC ;EXIT

```
*****  
:STALL ROUTINE:  
:  
:THIS ROUTINE IS USED TO PROVIDE SOFTWARE DELAYS  
:DURING READ, WRITE, TURN AROUND, AND YOZZLE.  
:THE DELAY TIMES MAY BE SET BY THE OPERATOR AT  
:INITIAL START FROM 200(8) OR MAY BE MODIFIED  
:AT ANY TIME BY ENTERING CNTRL C ON THE TTY AND  
:INSERTING NEW VALUES IN RESPONSE TO THE REQUEST.  
:THE READ STALL AND THE WRITE STALL ARE DELAYS  
:EXECUTED BETWEEN EACH RECORD OF THE DATA BLOCK.  
:THE TURN AROUND STALL IS EXECUTED EACH TIME  
:THE DIRECTION OF TAPE MOVEMENT IS CHANGED AND  
:ALSO EACH TIME THE TAPE OPERATION CHANGES FROM  
:WRITE TO READ OR READ TO WRITE. THE YOZZLE  
:STALL IS EXECUTED ONLY DURING THE YOZZLE ROUTINE.  
:*****
```

2998  
2999  
3000  
3001  
3002  
3003  
3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012  
3013  
3014  
3015  
3016  
3017  
3018  
3019  
3020  
3021  
3022  
3023  
3024  
3025  
3026  
3027  
3028  
3029  
3030  
3031  
3032  
3033  
3034  
3035

011664 012701 177760  
011670 012702 175000  
011674 004737 022314  
011700 042737 000001 000630  
011706 013737 000630 000556  
011714 012737 177777 013652  
011722 000207

CCNTR:

```
*****  
:RANDOM CHARACTER COUNT GENERATOR:  
:  
:THIS ROUTINE ENTERED VIA CONSOLE SWITCH  
:SEVEN (7) IS USED TO GENERATE A RANDOM  
:CHARACTER COUNT FOR EACH DATA BLOCK.  
:ALL RECORDS WITHIN A GIVEN BLOCK WILL BE  
:THE SAME, BUT EACH BLOCK WILL VARY.  
:THE LIMITS ARE TWENTY (20) TO FOUR THOUSAND  
:(4000) OCTAL CHARACTERS PER RECORD.  
:*****  
MOV #20,R1 ;SET HIGH LIMIT  
MOV #3000,R2 ;SET LOW LIMIT  
JSR PC,RANG ;GO GENERATE NUMBER  
BIC #1,RANSAV  
MOV RANSAV,FMCNT ;SET CHAR COUNT  
MOV #-1,PATS ;PRESET DATA PATTERN  
RTS PC ;EXIT
```

RCNTR:

```
*****  
:RANDOM RECORD COUNT GENERATOR:  
:  
:THIS ROUTINE ENTERED VIA CONSOLE SWITCH SIX (6)  
:IS USED TO GENERATE A RANDOM NUMBER OF RECORDS  
:FOR EACH BLOCK OF DATA.  
:THE LIMITS ARE ONE (1) TO FIVE HUNDRED (500) OCTAL  
:RECORDS PER BLOCK.  
:*****  
MOV #1,R2 ;SET LOW LIMIT  
MOV #500,R1 ;SET HIGH LIMIT  
JSR PC,RANG ;GO GENERATE NUMBER  
MOV RANSAV,RCNT ;SET RECORD COUNT  
RTS PC ;EXIT
```

3037  
3038  
3039  
3040  
3041  
3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092

```
*****  
:TEST CONDITION ENTRY ROUTINE:  
:  
:THIS ROUTINE IS USED TO ALLOW THE OPERATOR  
:TO ENTER, AT THE TTY, THE NECESSARY PARAMETERS  
:TO RUN THE PROGRAM AS HE WISHES. THE  
:ROUTINE IS ONLY ENTERED UPON INITIAL STARTING  
:FROM LOCATION 200(8).  
:THE MAIN PURPOSE OF THIS ROUTINE IS TO ESTABLISH  
:A TABLE OF DEVICES TO BE TESTED. THIS TABLE  
:CONSISTS OF AN ENTRY FOR EACH OF ONE (1) TO  
:EIGHT (8) DEVICES. EACH ENTRY CONTAINS THE  
:SLAVE NUMBER, DENSITY, PARITY, AND  
:FORMAT. THE INFORMATION IS ENTERED  
:IN RESPONSE TO PRINTED REQUESTS AT THE TTY.  
:SLAVES MAY BE ENTERED IN ANY ORDER. EACH  
:PARAMETER IS CHECKED FOR LEGALITY BEFORE BEING  
:SET INTO THE TABLE.  
:THE DRIVE NUMBER REQUEST WILL ALSO CHECK THE MASSBUS  
:FOR THE PRESENCE OF THE REQUESTED DRIVE. IF IT IS NOT FOUND,  
:A NON-EXIST DRIVE MESSAGE WILL BE PRINTED AND ANOTHER DRIVE  
:REQUEST MADE. WHEN THE DRIVE IS FOUND, THE RESPONSE IS STORED  
:AND CONTROL PASSED TO THE SLAVE SELECT ROUTINE.  
:THE SLAVE SELECT ROUTINE ALSO CHECKS FOR THE PRESENCE OF THE  
:SLAVE. IF IT IS NOT PRESENT, A MESSAGE IS PRINTED AND ANOTHER  
:REQUEST IS ISSUED. WHEN THE SELECTED SLAVE IS FOUND TO BE  
:PRESENT, A MESSAGE IS PRINTED IF IT IS A 7 CHANNEL DRIVE  
:TO ASSIST IN SELECTING DENSITY, PARITY, AND FORMAT.  
:UPON COMPLETION OF THE DEVICE TABLE, REQUESTS  
:ARE PRINTED FOR ENTRY OF THE NUMBER OF CHARACTERS  
:PER RECORD AND THE NUMBER OF RECORDS PER BLOCK. THE  
:NEXT REQUEST IS FOR A PATTERN NUMBER TO BE USED  
:FOR WRITING AND CHECKING OF READ DATA.  
:FOLLOWING THE PATTERN REQUEST IS THE TAPE MARK OPTION.  
:RESPONDING TO THE REQUEST (TM=) WITH A ONE (1)  
:WILL CAUSE THE PROGRAM TO WRITE A TM AT THE  
:END OF EACH DATA BLOCK AND TO EXPECT THE  
:TM TO BE DETECTED IN EITHER READ FORWARD AND REVERSE  
:OR DURING SPACE OPERATION. A RESPONSE OF ZERO (TM=0)  
:DISALLOWS WRITING OF THE TM AND CAUSES THE READ  
:AND SPACE ROUTINES TO EXPECT NO TM TO BE PRESENT.  
:THE LAST REQUESTS ARE FOR ENTRY OF THE DESIRED  
:WRITE, READ, AND TURN AROUND STALLS.  
*****
```

011750	005737	000636	TINP:	TST	TINF	;SEE IF SHOULD INPUT FROM TTY
011754	001002			BNE	1\$	;IF SO: BR
011756	000137	013270		JMP	TINP4	;GET SWITCHES
011762	005037	000674	1\$:	CLR	UNP	;CLEAR TABLE POINTER
011766	005037	004730		CLR	REOTC	;CLEAR EOT UNIT COUNTER
011772	012737	024231	012016	MOV	#MSG31,41\$	;GET TITLE MSG
012000	005737	000734		TST	ASEQF	;SEE IF AUTO SEQ
012004	001403			BEQ	4\$	;IF NOT: BR
012006	012737	024157	012016	MOV	#MSG30,41\$	;SET AUTO SEQ HDR
012014	000004		4\$:	TYPE		;TYPE MSG

3093	012016	000000		41\$:	.WORD 0	:ADDRESS OF APPROPRIATE TITLE MSG	
3094	012020	105077	177772		CLRB @41\$	:DO NOT TYPE TITLE ON RESTART	
3095	012024	000004	024313		TYPE,MSG31A	:TYPE INSTRUCTIONS	
3096	012030	105037	024313		CLRB MSG31A	:DO NOT TYPE STARTUP INSTRUCTIONS ON RESTART	
3097	012034	005737	013444		TST SCVFL	:SEE IF SHORT CONVERSATION	
3098	012040	001065			BNE 6\$	:IF SO: BR	
3099	012042	000004	025245		TYPE,MSG74	:REQUEST REGISTER START	
3100	012046	013703	000544		MOV REGS,R3		
3101	012052	104400			TYPOCT	:PRINT CURRENT REG START	
3102	012054	012705	000544		MOV #REGS,R5	:SAVE ADDRESS LOCATION	
3103	012060	012701	000007		MOV #7,R1	:SET SIZE OF ENTRY	
3104	012064	012702	176400		MOV #176400,R2	:SET UPPER LIMIT	
3105	012070	012703	172300		MOV #172300,R3	:SET LOWER LIMIT	
3106	012074	004737	022476		JSR PC,TTR	:GO GET RESPONSE	
3107							
3108	012100	000004	025270		TYPE,MSG75	:REQUEST INTERRUPT VECTOR ADDRESS	
3109	012104	013703	000546		MOV VECT,R3		
3110	012110	104400			TYPOCT	:PRINT CURRENT VECTOR	
3111	012112	012705	000546		MOV #VECT,R5	:SET SAVE LOCATION	
3112	012116	012701	000004		MOV #4,R1	:SET SIZE OF ENTRY	
3113	012122	012702	000224		MOV #224,R2	:SET UPPER LIMIT	
3114	012126	012703	000150		MOV #150,R3	:SET LOWER LIMIT	
3115	012132	004737	022476		JSR PC,TTR	:GO GET RESPONSE	
3116	012136	013700	000546		MOV VECT,R0	:GET VECTOR ADDRESS	
3117	012142	012720	021160		MOV #MTINT,(R0)+	:LOAD VECTOR WITH HANDLER ADDRESS	
3118	012146	012710	000340		MOV #340,(R0)	:LOAD PRIORITY LEVEL	
3119	012152	013700	000544		MOV REGS,R0	:GET STARTING REGISTER ADDRESS	
3120	012156	012701	000016		MOV #16,R1	:SET NUMBER OF REGISTERS	
3121	012162	012702	000510		MOV #C1,R2	:GET FIRST ADDRESS LOCATION	
3122	012166	010022		5\$:	MOV R0,(R2)+	:BUILD TABLE OF ADDRESSES	
3123	012170	062700	000002		ADD #2,R0	:BUMP ADDRESS	
3124	012174	005301			DEC R1	:SEE IF DONE	
3125	012176	001373			BNE 5\$	:IF NOT: BR	
3126	012200	005737	000734		TST ASEQF	:SEE IF AUTO SEQ	
3127	012204	001403			BEQ 6\$	:IF NOT: BR	
3128	012206	005726			TST (SP)+	:RESET STACK POINTER	
3129	012210	000137	021176		JMP ASEQ	:GO TO AUTO SEQUENCE	
3130							
3131	012214	012777	000040	166276	6\$:	MOV #40,@CS	:INITIALIZE
3132	012222	000004	024756		TYPE,MSG52A	:REQUEST DRIVE (TMO3) #	
3133	012226	012705	000550		MOV #DVN,R5	:GET ADDRESS	
3134	012232	012701	000002		MOV #2,R1	:SET SIZE OF RESPONSE	
3135	012236	012702	000007		MOV #7,R2	:SET UPPER LIMIT	
3136	012242	012703	000000		MOV #0,R3	:SET LOWER LIMIT	
3137	012246	004737	022476		JSR PC,TTR	:GO GET DRIVE NUMBER	
3138	012252	013777	000550	166240	MOV DVN,@CS		
3139	012260	005777	166224		TST @C1	:ACCESS DRIVE	
3140	012264	032777	010000	166226	BIT #10000,@CS	:SEE IF NED	
3141	012272	001403			BEQ TINPO	:IF NOT: BR	
3142	012274	000004	025202		TYPE,MSG71	:TYPE 'NON-EXISTANT DRIVE'	
3143	012300	000745			BR 6\$	:RETRY DVN	
3144							
3145	012302	012705	000646		TINPO: MOV #TEMP2,R5	:SET ADDRESS FOR RESPONSE	
3146	012306	000004	024400		TYPE,MSG32	:REQUEST SLAVE (TE16,TU77) #	
3147	012312	005037	000646		CLR TEMP2	:CLEAR BUFFER	
3148	012316	012701	000002		MOV #2,R1	:SET NUMBER OF CHARACTERS TO INPUT	

3149	012322	012702	000007		MOV #7,R2	:SET MAXIMUM LIMIT
3150	012326	012703	000000		MOV #0,R3	:SET MINIMUM LIMIT
3151	012332	004737	022476		JSR PC,TTR	:GO GET UNIT NUMBER
3152	012336	005737	000644		TST TEMP1	:SEE IF HAVE NEW PARAMETER
3153	012342	001010			BNE TINPOB	:IF SO: BR
3154	012344	013700	000674		MOV UNP,R0	
3155	012350	001754			BEQ TINPO	:BRANCH IF FIRST ENTRY
3156	012352	012760	177777	000742	MOV #-1,UN1(R0)	:SET END UNIT TABLE
3157	012360	000137	012700		JMP TINP2C	:GO GET RECORD COUNT
3158	012364	013700	000674		TINPOB: MOV UNP,R0	
3159	012370	011560	000742		MOV (R5),UN1(R0)	:SET NEW SLAVE #
3160	012374	012777	000040	166116	MOV #40,ACS	:DO A MASS BUS CLEAR
3161	012402	013777	000550	166110	MOV DVN,ACS	:LOAD DRIVE #
3162	012410	016077	000742	166124	MOV UN1(R0),ATC	:LOAD SLAVE NUMBER
3163	012416	032777	002000	166112	BIT #2000,ADT	:SEE IF SLAVE PRESENT
3164	012424	001003			BNE TINPOD	:IF SO: BR
3165	012426	000004	025034		TYPE,MSG57	:TYPE NON-EXISTANT SLAVE'
3166	012432	000723			BR TINPO	:REDO
3167	012434	017703	166076		TINPOD: MOV ADT,R3	:GET CONTENTS OF DT REG
3168	012440	042703	000007		BIC #7,R3	:CLEAR DRIVE TYPE #
3169	012444	022703	142050		CMP #142050,R3	:SEE IF 9TRK TM03
3170	012450	001407			BEQ TINPOE	:IF SO: BR
3171	012452	000004	024727		TYPE,MSG50	:TYPE 'ILLEGAL DRIVE TYPE'
3172	012456	017703	166054		MOV ADT,R3	
3173	012462	042703	000007		BIC #7,R3	:CLEAR SLAVE #
3174	012466	104400			TYPOCT	:PRINT DRIVE TYPE REGISTER
3175	012470	004737	023362		TINPOE: JSR PC,SNPT	:PRINT SERIAL NUMBER
3176						
3177	012474	000004	024413		TINP1: TYPE,MSG33	:REQUEST DENSITY
3178	012500	005037	000646		CLR TEMP2	:CLEAR BUFFER
3179	012504	012701	000002		MOV #2,R1	:SET NUMBER OF CHARACTERS TO INPUT
3180	012510	012702	000004		MOV #4,R2	:SET MAXIMUM LIMIT
3181	012514	012703	000003		MOV #3,R3	:SET MINIMUM LIMIT
3182	012520	004737	022476		JSR PC,TTR	:GO GET DENSITY
3183	012524	012703	000010		MOV #10,R3	:SET POSITION FACTOR
3184	012530	004737	013446		JSR PC,TPOS	:GO LOAD DENSITY INTO PROPER POSITION
3185						
3186	012534	000315			TINP2: SWAB (R5)	:IF DENSITY
3187	012536	022715	000004		CMP #4,(R5)	:IS 1600BPI
3188	012542	001415			BEQ 1\$	:THEN SKIP PARITY REQUEST
3189	012544	000004	024426		TYPE,MSG34	:REQUEST PARITY
3190	012550	005037	000646		CLR TEMP2	:CLR BFR
3191	012554	012701	000002		MOV #2,R1	:SET NUMBER OF CHAR. TO INPUT
3192	012560	012702	000001		MOV #1,R2	:SET HIGH LIMIT
3193	012564	012703	000000		MOV #0,R3	:SET LOW LIMIT
3194	012570	004737	022476		JSR PC,TTR	:GO INPUT PARITY
3195	012574	000402			BR 2\$	:SKIP 1600 BPI PAROTY SETTING
3196	012576	012715	000000		1\$: MOV #0,(R5)	:SET ODD PARITY FOR 1600 BPI
3197	012602	012703	000003		2\$: MOV #3,R3	:SET POSITION FACTOR
3198	012606	004737	013446		JSR PC,TPOS	:GO POSITION PARITY
3199						
3200	012612	000004	025000		TINP2A: TYPE,MSG53	:REQUEST FORMAT
3201	012616	005037	000646		CLR TEMP2	
3202	012622	012701	000003		MOV #3,R1	
3203	012626	012702	000017		MOV #17,R2	
3204	012632	012703	000000		MOV #0,R3	

3205	012636	004737	022476		JSR	PC,TTR		:GO GET FORMAT
3206	012642	012703	000004		MOV	#4,R3		
3207	012646	004737	013446		JSR	PC,TPOS		
3208	012652	005237	004730		TINP2B: INC	REOTC		:BUMP EOT UNIT COUNTER
3209	012656	022737	000016	000674	CMP	#16,UNP		:SEE IF DONE UNITS
3210	012664	001405			BEQ	TINP2C		:IF SO: BR
3211	012666	062737	000002	000674	ADD	#2,UNP		:POINT TO NEXT UNIT
3212	012674	000137	012302		JMP	TINPO		:ELSE LOOK FOR NEXT UNIT
3213								
3214								
3215	012700	005037	000674		TINP2C: CLR	UNP		:CLEAR UNIT POINTER
3216	012704	113737	004730	004731	MOVB	REOTC,REOTC+1		:SET # OF UNITS TO TEST
3217								
3218	012712	000004	024440		TINP3: TYPE,MSG35			:REQUEST RECORDS PER BLOCK
3219	012716	013703	000554		MOV	RCNT,R3		
3220	012722	104400			TYPOCT			:PRINT RECORD COUNT
3221	012724	012705	000554		MOV	#RCNT,R5		:SET RECORD COUNT ADDRESS
3222	012730	012701	000007		MOV	#7,R1		:SET NUMBER OF CHARACTERS TO INPUT
3223	012734	012702	177777		MOV	#177777,R2		:SET MAXIMUM LIMIT
3224	012740	012703	000001		MOV	#1,R3		:SET MINIMUM LIMIT
3225	012744	004737	022476		JSR	PC,TTR		:GO GET RECORD COUNT
3226	012750	013737	000554	000632	MOV	RCNT,RCSAV		:SAVE RECORD COUNT
3227								
3228	012756	000004	024460		TYPE,MSG36			:REQUEST CHARACTERS PER RECORD
3229	012762	005437	000556		NEG	FMCNT		
3230	012766	013703	000556		MOV	FMCNT,R3		
3231	012772	104400			TYPOCT			:PRINT CHAR COUNT
3232	012774	012705	000556		MOV	#FMCNT,R5		:SET CHARACTER COUNT ADDRESS
3233	013000	012701	000007		MOV	#7,R1		:SET NUMBER OF CHARACTERS TO INPUT
3234	013004	012702	004000		MOV	#4000,R2		:SET MAXIMUM LIMIT
3235	013010	012703	000004		MOV	#4,R3		:SET MINIMUM LIMIT
3236	013014	004737	022476		JSR	PC,TTR		:GO GET CHARACTER COUNT
3237	013020	005437	000556		NEG	FMCNT		:SET TO TWO'S COMPLIMENT
3238	013024	013737	000556	000634	MOV	FMCNT,FCSAV		:SAVE FRAME COUNT
3239								
3240	013032	000004	024476		TYPE,MSG37			:REQUEST PATTERN #
3241	013036	013703	000560		MOV	PATRN,R3		
3242	013042	104400			TYPOCT			:PRINT PATTERN
3243	013044	005037	014014		CLR	DOFL		:CLEAR EXTERNAL DATA FLAG
3244	013050	012705	000560		MOV	#PATRN,R5		:SET PATTERN NUMBER ADDRESS
3245	013054	012701	000003		MOV	#3,R1		:SET NUMBER OF CHARACTERS TO INPUT
3246	013060	012702	000015		MOV	#15,R2		:SET MAXIMUM LIMIT
3247	013064	012703	000000		MOV	#0,R3		:SET MINIMUM LIMIT
3248	013070	004737	022476		JSR	PC,TTR		:GO GET PATTERN NUMBER
3249								
3250	013074	000004	025166		TYPE,MSG69			:REQUEST TAPE MARK
3251	013100	013703	000564		MOV	TMEX,R3		
3252	013104	104400			TYPOCT			:PRINT CURRENT TM FLAG SETTING
3253	013106	012705	000564		MOV	#TMEX,R5		:GET TM FLAG ADDRESS
3254	013112	012701	000002		MOV	#2,R1		:SET SIZE OF RESPONSE
3255	013116	012702	000001		MOV	#1,R2		:SET UPPER LIMIT
3256	013122	012703	000000		MOV	#0,R3		:SET LOWER LIMIT
3257	013126	004737	022476		JSR	PC,TTR		:TM 1=YES
3258								
3259	013132	000004	023670		TYPE,MSG21			:REQUEST INTERCHANGE READ
3260	013136	013703	000570		MOV	INTRF,R3		

3261	013142	104400		TYPOCT		:PRINT CURRENT SETTING
3262	013144	012705	000570	MOV	#INTRF,R5	:GET FLAG ADDRESS
3263	013150	012701	000002	MOV	#2,R1	:SET SIZE OF RESPONSE
3264	013154	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
3265	013160	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
3266	013164	004737	022476	JSR	PC,TTR	:GO GET RESPONSE
3267						
3268	013170	000004	024513	TYPE,MSG38		:REQUEST SINGLE PASS
3269	013174	013703	000572	MOV	SPFLG,R3	
3270	013200	104400		TYPOCT		:PRINT CURRENT SETTING
3271	013202	012705	000572	MOV	#SPFLG,R5	:SET ADDRESS OF FLAG
3272	013206	012701	000002	MOV	#2,R1	:SET SIZE OF RESPONSE
3273	013212	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
3274	013216	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
3275	013222	004737	022476	JSR	PC,TTR	:GO GET RESPONSE
3276						
3277	013226	000004	024531	TINP3A: TYPE,MSG39		:REQUEST CRC CORRECTION
3278	013232	013703	000566	MOV	CRCC,R3	
3279	013236	104400		TYPOCT		
3280	013240	012705	000566	MOV	#CRCC,R5	
3281	013244	012701	000002	MOV	#2,R1	
3282	013250	012702	000001	MOV	#1,R2	
3283	013254	012703	000000	MOV	#0,R3	
3284	013260	004737	022476	JSR	PC,TTR	
3285	013264	004737	022346	JSR	PC,GTSWR	:GET SWITCHES
3286	013270	005737	013444	TINP4: TST	SCVFL	:BRANCH IF SHORT CONVERSATION
3287	013274	001060		BNE	TINPX	
3288	013276	005737	000636	TS:	TINF	:BRANCH IF NO TTY INPUT
3289	013302	001455		BEQ	TINPX	
3290	013304	000004	024567	TYPE,MSG40		:REQUEST READ STALL
3291	013310	013703	000574	MOV	RSTAL,R3	
3292	013314	104400		TYPOCT		:PRINT READ STALL
3293	013316	012705	000574	MOV	#RSTAL,R5	:SET READ STALL ADDRESS
3294	013322	012701	000007	MOV	#7,R1	:SET NUMBER OF CHARACTERS TO INPUT
3295	013326	012702	177777	MOV	#-1,R2	:SET MAXIMUM LIMIT
3296	013332	012703	000001	MOV	#1,R3	:SET MINIMUM LIMIT
3297	013336	004737	022476	JSR	PC,TTR	:GO GET READ STALL
3298						
3299	013342	000004	024616	TYPE,MSG41		:REQUEST WRITE STALL
3300	013346	013703	000576	MOV	WSTAL,R3	
3301	013352	104400		TYPOCT		:PRINT READ STALL
3302	013354	012705	000576	MOV	#WSTAL,R5	:SET WRITE STALL ADDRESS
3303	013360	012701	000007	MOV	#7,R1	:SET NUMBER OF CHARACTERS TO INPUT
3304	013364	012702	177777	MOV	#-1,R2	:SET MAXIMUM LIMIT
3305	013370	012703	000001	MOV	#1,R3	:SET MINIMUM LIMIT
3306	013374	004737	022476	JSR	PC,TTR	:GO GET WRITE STALL
3307						
3308	013400	000004	024627	TYPE,MSG42		:REQUEST TURN AROUND STALL
3309	013404	013703	000600	MOV	TSTAL,R3	
3310	013410	104400		TYPOCT		:PRINT TA STALL
3311	013412	012705	000600	MOV	#TSTAL,R5	:SET TURN AROUND STALL ADDRESS
3312	013416	012701	000007	MOV	#7,R1	:SET NUMBER OF CHARACTERS TO INPUT
3313	013422	012702	177777	MOV	#-1,R2	:SET MAXIMUM LIMIT
3314	013426	012703	000001	MOV	#1,R3	:SET MINIMUM LIMIT
3315	013432	004737	022476	JSR	PC,TTR	:GO GET TURN AROUND STALL
3316	013436	005037	013444	TINPX: CLR	SCVFL	:CLEAR SHORT CONVERSATION FLAG



```
3317 013442 000207          RTS      PC          :EXIT
3318 013444 000000          SCVFL:  0          :SHORT CONVERSATION FLAG
3319
3320          :UNIT DESCRIPTION POSITIONING SUBROUTINE*****
3321
3322 013446 006337 000646          TPOS:  ASL      TEMP2      :POSITION CHARACTER
3323 013452 005303          DEC      R3          :SEE IF DONE
3324 013454 001374          BNE      TPOS       :IF NOT: BR
3325 013456 013700 000674          MOV      UNP,R0     :LOAD UNIT POINTER
3326 013462 053760 000646 000742          BIS      TEMP2,UN1(R0) :LOAD CHARACTER INTO UN1(R0)
3327 013470 000207          RTS      PC          :EXIT
3328
```

3330  
3331  
3332  
3333  
3334  
3335  
3336  
3337  
3338  
3339  
3340  
3341  
3342  
3343  
3344  
3345  
3346  
3347  
3348  
3349  
3350  
3351  
3352  
3353  
3354  
3355  
3356  
3357  
3358  
3359  
3360  
3361  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376  
3377  
3378  
3379  
3380  
3381  
3382  
3383  
3384  
3385

```
*****  
:DATA SETUP ROUTINE:  
:  
:THIS ROUTINE IS USED TO GENERATE INTO THE ENTIRE  
:WRITE BUFFER (4000 OCTAL CHARACTERS) THE DATA PATTERN  
:SELECTED BY THE OPERATOR. THERE ARE 15 (8) FIXED  
:DATA PATTERNS AVAILABLE AND ONE SELECTION (DATA PATTERN 0)  
:WHICH WILL READ ANY PATTERN PRESENTED AT THE  
:HIGH SPEED PAPER TAPE READER. THIS TAPE MUST BE PREPARED  
:BY USING THE PROGRAM CALLED DTC. (MAINDEC-11-DZTUF-A-D)  
:RANDOM DATA MAY ALSO BE USED VIA CONSOLE  
:SWITCH EIGHT (8).  
:THIS ROUTINE IS ALSO USED TO CLEAR OUT THE  
:READ BUFFER (4000 OCTAL CHARACTERS) BEFORE EACH  
:RECORD IS READ.  
*****
```

```
013472 005737 014404 DSUP: TST RDFL :SEE IF DID RANDOM DATA  
013476 001044 BNE DS2A :IF NOT: BR  
013500 005737 000734 DSO: TST ASEQF :SEE IF AUTO SEQ  
013504 001406 BEQ DSOC :IF NOT: BR  
013506 005737 000560 TST PATRN :SEE IF AUTO RANDOM  
013512 100003 BPL DSOC :IF NOT: BR  
013514 004737 014342 JSR PC,DATR :ELSE GO GENERATE RANDOM DATA  
: RTS PC :++B DELETED  
013520 000433 BR DS2A :++B GENERATE EXPECTED LRC/CRC & CLEAR READ BFR  
013522 023737 000560 013652 DSOC: CMP PATRN,PATS :SEE IF NEW PATTERN  
013530 001014 BNE DS0A :IF SO: BR  
013532 013703 000552 MOV UDES,R3 :GET UNIT DESCRIPTION  
013536 042703 177767 BIC #177767,R3 :MASK EVEN PARITY  
013542 023703 013654 CMP PARS,R3 :SEE IF SAME AS LAST TIME  
013546 001404 BEQ DS0B :IF SO: BR  
013550 010337 013654 MOV R3,PARS :SAVE PARITY  
013554 004737 014406 JSR PC,CRCLRC :GO GENERATE EXPT CRC/LRC  
013560 000207 DS0B: RTS PC  
013562 012703 026344 DSOA: MOV #WDATA,R3 :R3 = ADDRS OF WRITE BUFFER  
013566 013701 000560 MOV PATRN,R1 :R1 = PATTERN SELECTOR  
013572 010137 013652 MOV R1,PATS  
013576 062701 000001 ADD #1,R1 :BUMP POINTER  
013602 006301 ASL R1 :MAKE PATTERN SELECTOR EVEN  
013604 004771 002764 JSR PC,@DATBL(R1) :GO GENERATE PATTERN  
013610 004737 014406 DS2A: JSR PC,CRCLRC :GO GENERATE EXPT CRC/LRC  
013614 013702 000556 DS3: MOV FMCNT,R2 :R2=BUFFER SIZE  
013620 006202 ASR R2 :R2=FRAME CMT/2  
013622 012701 032352 MOV #RDATA,R1 :R1=READ DATA START  
013626 005021 DS4: CLR (R1)+ :CLEAR BUFFER  
013630 005202 INC R2 :SEE IF DONE ALL  
013632 001375 BNE DS4 :IF NOT: BR  
013634 013737 000552 013654 MOV UDES,PARS :GET UNIT DESCRIPTION  
013642 042737 177767 013654 BIC #177767,PARS :MASK PARITY  
013650 000207 RTS PC :EXIT  
013652 177777 PATS: -1 :PATTERN NUMBER SAVE  
013654 000000 PARS: 0
```

```
3387
3388 ;EXTERNAL DATA INPUT FROM H/S READER (256 CHARACTER MAXIMUM)
3389
3390 013656 005737 014014 DATO: TST DOFL ;BRANCH IF SHOULD DO EXTERNAL INPUT
3391 013662 001401 BEQ 1$
3392 013664 000207 RTS PC ;++B RETURN
3393 013666 012737 000001 014014 1$: MOV #1,DOFL ;SET EXTERNAL FLAG
3394 013674 005077 164722 CLR @PRS ;CLEAR READER STATUS
3395 013700 005037 000644 CLR TEMP1 ;CLEAR FOR USE AS CHARACTER FLAG
3396 013704 052777 000001 164710 DATOA: BIS #1,@PRS ;START READER
3397 013712 105777 164704 DATOB: TST@PRS ;SEE IF DONE
3398 013716 100375 BPL DATOB ;IF NOT : BR
3399 013720 005001 CLR R1 ;CLEAR SAVE LOCATION
3400 013722 117701 164676 MOV@PRB,R1 ;SAVE CHARACTER
3401 013726 005737 000644 TST TEMP1 ;SEE IF HAVE FOUND START CHARACTER
3402 013732 001011 BNE DATOC ;IF SO : BR
3403 013734 105701 TSTB R1 ;SEE IF CHARACTER IS 0
3404 013736 001762 BEQ DATOA ;IF SO : BR
3405 013740 012737 000001 000644 MOV #1,TEMP1 ;ELSE SET CHARACTER FOUND FLAG
3406 013746 010137 000646 MOV R1,TEMP2 ;SAVE DATA SIZE
3407 013752 010102 MOV R1,R2 ;SAVE DATA SIZE
3408 013754 000753 BR DATOA ;GO GET FIRST DATA CHAR
3409 013756 110123 DATOC: MOV@R1,(R3)+ ;LOAD BUFFER
3410 013760 005302 DEC R2 ;SEE IF READ ALL
3411 013762 001350 BNE DATOA ;IF NOT : BR
3412 013764 012701 026344 DATOD: MOV #WDATA,R1 ;R1 = START OF WRITE BUFFER
3413 013770 013702 000646 MOV TEMP2,R2 ;R2 = SIZE OF DATA FIELD
3414 013774 112123 DATOE: MOV@R1+,(R3)+ ;REPEAT LOAD OF DATA FIELD
3415 013776 022703 032352 CMP #RDATA,R3 ;SEE IF DONE
3416 014002 003001 BGT DATOF ;IF NOT: BR
3417 014004 000207 RTS PC ;++B RETURN
3418 014006 005302 DATOF: DEC R2 ;SEE IF AT END OF DATA FIELD
3419 014010 001371 BNE DATOE ;IF NOT : BR
3420 014012 000764 BR DATOD ;ELSE RESTART FILL
3421 014014 000000 DOFL: 0 ;EXTERNAL DATA FLAG=1 IF ALREADY DONE
3422
```

```
3424                                     ;ALL ONES*****
3425
3426 014016 012701 177777  DAT1:  MOV    #-1,R1      ;R1=DATA
3427 014022 012702 002002  DAT1A: MOV    #2002,R2    ;R2=WORD COUNT +2
3428 014026 010123          1$:  MOV    R1,(R3)+    ;LOAD BUFFER
3429 014030 005302          DEC    R2              ;SEE IF DONE
3430 014032 001375          BNE   1$              ;IF NOT: BR
3431 014034 000207          RTS    PC
3432
3433                                     ;ALL ZEROS*****
3434
3435 014036 005001  DAT2:  CLR    R1          ;R1=DATA
3436 014040 000770          BR     DAT1A        ;LOAD BUFFER
3437
3438                                     ;WALKING ONE*****
3439
3440 014042 012701 000001  DAT3:  MOV    #1,R1      ;R1=DATA
3441 014046 000241          CLC
3442 014050 012702 004004  DAT3A: MOV    #4004,R2    ;R2=CHARACTER COUNT+4
3443 014054 110123          1$:  MOVB   R1,(R3)+    ;LOAD BUFFER
3444 014056 106101          ROLB  R1            ;SET NEXT CHARACTER
3445 014060 005302          DEC    R2              ;SEE IF DONE
3446 014062 001374          BNE   1$              ;IF NOT: BR
3447 014064 000207          RTS    PC
3448
3449                                     ;WALKING ZERO*****
3450
3451 014066 012701 000376  DAT4:  MOV    #376,R1    ;R1=START OF DATA
3452 014072 000261          SEC
3453 014074 000765          BR     DAT3A        ;LOAD BUFFER
3454
3455                                     ;ALTERNATING ONE/ZERO*****
3456
3457
3458 014076 012701 052525  DAT5:  MOV    #52525,R1  ;R1=DATA
3459 014102 000747          BR     DAT1A        ;LOAD BUFFER
3460
3461                                     ;ALTERNATING ZERO/ONE*****
3462
3463 014104 012701 125252  DAT6:  MOV    #125252,R1 ;R1=DATA
3464 014110 000744          BR     DAT1A        ;LOAD BUFFER
3465
3466                                     ;ONE/ZERO IN ALTERNATING WORDS*****
3467
3468 014112 012701 125252  DAT7:  MOV    #125252,R1  ;SET WORD 1
3469 014116 012702 052525  MOV    #52525,R2      ;SET WORD 2
3470 014122 012704 001002  MOV    #1002,R4       ;SET NUMBER OF ENTRIES
3471 014126 010123          1$:  MOV    R1,(R3)+    ;LOAD WORD 1
3472 014130 010223          MOV    R2,(R3)+    ;LOAD WORD 2
3473 014132 005304          DEC    R4           ;SEE IF DONE
3474 014134 001374          BNE   1$           ;IF NOT: BR
3475 014136 000207          RTS    PC
3476
```

```
3478 ;WALKING ONE/ALL ONE IN ALTERNATING CHARS****
3479
3480 014140 012702 002002 DAT10: MOV #2002,R2 ;SET BUFFER SIZE
3481 014144 012701 000001 MOV #1,R1 ;SET WALK BASE
3482 014150 000241 CLC
3483 014152 012713 177400 1$: MOV #177400,(R3) ;LOAD ALL ONE BYTE
3484 014156 050123 BIS R1,(R3)+ ;LOAD WALK BYTE
3485 014160 106101 ROLB R1 ;WALK ONE
3486 014162 005302 DEC R2
3487 014164 001372 BNE 1$ ;DO FULL BUFFER
3488 014166 000207 RTS PC
3489
3490 ;ALL BITS 0-377*****
3491
3492 014170 005001 DAT11: CLR R1 ;R1=STARTING DATA
3493 014172 012702 004004 MOV #4004,R2 ;R2=CHARACTER COUNT+4
3494 014176 110123 1$: MOVB R1,(R3)+ ;LOAD BUFFER
3495 014200 105201 INCB R1 ;BUMP DATA
3496 014202 005302 DEC R2 ;SEE IF DONE
3497 014204 001374 BNE 1$ ;IF NOT: BR
3498 014206 000207 RTS PC ;RETURN
3499
3500 ;ALL BITS 377-0*****
3501
3502 014210 012701 000377 DAT12: MOV #377,R1 ;R1=STARTING DATA
3503 014214 012702 004004 MOV #4004,R2 ;R2=CHARACTER COUNT+4
3504 014220 110123 1$: MOVB R1,(R3)+ ;LOAD BUFFER
3505 014222 105301 DECB R1 ;BUMP DATA
3506 014224 005302 DEC R2 ;SEE IF DONE
3507 014226 001374 BNE 1$ ;IF NOT: BR
3508 014230 000207 RTS PC ;RETURN
3509
3510 ;ALTERNATING CHARACTERS 0 AND 377*****
3511
3512 014232 012701 000377 DAT13: MOV #377,R1 ;R1 = DATA
3513 014236 000137 014022 JMP DAT1A ;LOAD BUFFER
3514
3515 ;WALKING ZERO/ALL ZERO IN ALTERNATING CHARS*****
3516
3517 014242 012702 002002 DAT14: MOV #2002,R2 ;SET BUFFER SIZE
3518 014246 012701 000376 MOV #376,R1 ;SET WALK BASE
3519 014252 000261 SEC
3520 014254 010113 1$: MOV R1,(R3) ;LOAD WALK BYTE
3521 014256 042723 177400 BIC #177400,(R3)+ ;CLEAR HIGH BYTE
3522 014262 106101 ROLB R1 ;WALK ZERO BIT
3523 014264 005302 DEC R2
3524 014266 001372 BNE 1$ ;FILL BUFFER
3525 014270 000207 RTS PC ;RETURN
3526
```

```
3531                                     ;AUTO SEQUENCE PATTERN*****
3532
3533 014272 012702 000200          DAT15: MOV      #200,R2          ;SET NUMBER OF ENTRIES
3534 014276 012701 014322          1$:  MOV      #APATS,R1        ;SET START OF PATTERN
3535 014302 012704 000010          MOV      #10,R4             ;SET SIZE OF PATTERN
3536 014306 012123                2$:  MOV      (R1)+,(R3)+    ;FILL BUFFER
3537 014310 005304                DEC      R4                 ;SEE IF DONE PATTERN
3538 014312 001375                BNE     2$                 ;IF NOT: BR
3539 014314 005302                DEC      R2                 ;SEE IF DONE BUFER
3540 014316 001367                BNE     1$                 ;IF NOT: BR
3541 014320 000207                RTS      PC                 ;RETURN
3542
3543 014322 000000          APATS: 0
3544 014324 177400          177400
3545 014326 000377          377
3546 014330 000000          0
3547 014332 177777          -1
3548 014334 000377          377
3549 014336 177400          177400
3550 014340 177777          -1
3551
3552                                     ;RANDOM DATA GENERATOR SUBROUTINE*****
3553
3554 014342 013704 000556          DATR: MOV      FMCNT,R4       ;SET NUMBER OF FRAMES
3555 014346 012703 026344          MOV      #WDATA,R3        ;SET ADDRESS OF START OF BUFFER
3556 014352 012701 177777          MOV      #-1,R1           ;SET HIGH LIMIT
3557 014356 005002                CLR      R2                ;SET LOW LIMIT
3558 014360 004737 022314          1$:  JSR      PC,RANG        ;GO GENERATE NUMBER
3559 014364 013723 000630          MOV      RANSV,(R3)+     ;LOAD BUFFER
3560 014370 005204                INC      R4                 ;SEE IF DONE WHOLE BUFFER
3561 014372 001372                BNE     1$                 ;IF NOT: BR
3562 014374 012737 000001 014404  MOV      #1,RDFL          ;SET RANDOM DATA FLAG
3563 014402 000207                RTS      PC                 ;EXIT
3564 014404 000000          RDFL: 0                   ;RANDOM DATA SELECT FLAG
```

```
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575 014406 013700 000556
3576 014412 005400
3577 014414 012701 026344
3578 014420 005037 014742
3579 014424 111104
3580 014426 004737 014614
3581 014432 004737 014716
3582 014436 000241
3583 014440 006004
3584 014442 103014
3585 014444 052704 000400
3586 014450 000241
3587 014452 010405
3588 014454 042705 177703
3589 014460 005105
3590 014462 042705 177703
3591 014466 042704 000074
3592 014472 050504
3593 014474 010437 014742
3594 014500 005300
3595 014502 001350
3596 014504 013704 014742
3597 014510 005137 014742
3598 014514 042737 177050 014742
3599 014522 042704 177727
3600 014526 050437 014742
3601 014532 013737 014742 014744
3602 014540 013700 000556
3603 014544 005400
3604 014546 012701 026344
3605 014552 005037 014742
3606 014556 111104
3607 014560 004737 014614
3608 014564 004737 014716
3609 014570 005300
3610 014572 001371
3611 014574 013704 014744
3612 014600 004737 014716
3613 014604 013737 014742 014746
3614 014612 000207
3615 014614 005704
3616 014616 001010
3617 014620 032737 000010 000552
3618 014626 001404
3619 014630 012704 000420
3620 014634 005201
3621 014636 000207

:*****
:CRC/LRC CHARACTER BUILD:
:
:THIS ROUTINE WILL CONSTRUCT AND SAVE THE EXPECTED
:CRC AND LRC CHARACTERS ACCORDING TO DATA AND
:RECORD SIZE IF OPERATING IN NRZ MODE
:*****

CRCLRC: MOV FMCNT,R0 ;SET RECORD SIZE
        NEG R0
        MOV #WDATA,R1 ;SET START OF BUFFER
        CLR XORS
CL0:    MOVB (R1),R4 ;GET CHARACTER
        JSR PC,CLP ;GO GET PARITY OF CHARACTER
        JSR PC,XOR ;XOR CHARACTER
        CLC
        ROR R4 ;ROTATE 1 RIGHT
        BCC CL2 ;IF NO CARRY: BR
        BIS #400,R4 ;SET BIT NINE
CL1:    MOV R4,R5 ;SAVE CHARACTER
        BIC #177703,R5
        COM R5
        BIC #177703,R5
        BIS #74,R4 ;COMPLIMENT BITS 2,3,4,5
CL2:    MOV R4,XORS
        DEC R0
        BNE CLO ;BRANCH IF NOT LAST CHAR
CLLAST: MOV XORS,R4
        COM XORS
        BIC #177050,XORS
        BIC #177727,R4 ;COMPLIMENT ALL BUT BITS 3&5
        BIS R4,XORS
        MOV XORS,EXCRC ;SAVE EXPECTED CRC
        MOV FMCNT,R0
        NEG R0
        MOV #WDATA,R1 ;DO EXPT LRC
        CLR XORS
CL3:    MOVB (R1),R4
        JSR PC,CLP ;GET PARITY
        JSR PC,XOR ;XOR CHARACTER
        DEC R0
        BNE CL3 ;DO ALL FOR LRC
        MOV EXCRC,R4
        JSR PC,XOR ;XOR CRC TO DATA
        MOV XORS,EXLRC ;SAVE EXPT LRC
        RTS PC ;RETURN
CLP:    TST R4 ;SEE IF 0 CHAR
        BNE CLPE ;IF NOT: BR
        BIT #10,UDES ;SEE IF EVEN PARITY
        BEQ CLPE ;IF NOT: BR
        MOV #420,R4 ;SET 0 CHAR EVEN PARITY
        INC R1 ;BUMP POINTER
        RTS PC ;RETURN
```





```
3652  
3653  
3654  
3655  
3656  
3657  
3658  
3659  
3660  
3661  
3662  
3663  
3664  
3665  
3666  
3667 014750 005037 000656 DCHK: CLR BBC ;CLEAR BAD RECORD CNTR  
3668 014754 005037 000704 CLR DERFL ;CLEAR DATA ERROR FLAG  
3669 014760 013705 000556 MOV FMCNT,R5 ;LOAD CHAR COUNT  
3670 014764 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP  
3671 014772 001401 BEQ DCHKO ;IF NOT: BR  
3672 014774 006205 ASR R5 ;R5 = FC/2  
3673 014776 012701 026344 DCHKO: MOV #WDATA,R1 ;SET WRITE DATA ADDR  
3674 015002 012702 032352 MOV #RDATA,R2 ;SET READ DATA ADDR  
3675 015006 032737 000010 000552 BIT #10,UDES ;SEE IF EVEN PARITY  
3676 015014 001430 BEQ DFOCO ;IF NOT: BR  
3677 015016 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP PARITY  
3678 015024 001024 BNE DFOCO ;IF SO: BR  
3679 015026 032737 002000 000552 BIT #2000,UDES ;SEE IF PE MODE  
3680 015034 001020 BNE DFOCO ;IF SO: BR  
3681 015036 105711 DFOF: TSTB (R1) ;SEE IF 0 CHAR  
3682 015040 001404 BEQ DFOD ;IF SO: BR  
3683 015042 005201 INC R1 ;BUMP POINTER  
3684 015044 005205 DFOE: INC R5 ;SEE IF DONE  
3685 015046 001373 BNE DFOF ;IF NOT: BR  
3686 015050 000406 BR DFOC ;ELSE CONTINUE  
3687 015052 112721 000020 DFOD: MOVB #20,(R1)+ ;SET 20 IN PLACE OF 0  
3688 015056 012737 177777 013652 MOV #-1,PATS ;SET PATTERN GENERATE FLAG  
3689 015064 000767 BR DFOE  
3690 015066 013705 000556 DFOC: MOV FMCNT,R5 ;RESET CHAR CNT  
3691 015072 012701 026344 MOV #WDATA,R1 ;RESET DATA ADDRESS  
3692 015076 005737 000562 DFOCO: TST RDCMD ;SEE IF READ REVERSE  
3693 015102 001462 BEQ DFO ;IF NOT: BR  
3694 015104 013704 000556 DFOB: MOV FMCNT,R4 ;GET FRAME COUNT  
3695 015110 005404 NEG R4 ;SET TO WHOLE NUMBER  
3696 015112 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP  
3697 015120 001402 BEQ DFOBO ;IF NOT: BR  
3698 015122 000241 CLC  
3699 015124 006004 DFOBO: ROR R4 ;SET TO FC/2  
3700 015126 060401 ADD R4,R1 ;POINT TO START OF WRITE DATA  
3701 015130 060402 ADD R4,R2 ;POINT TO START OF READ DATA  
3702 015132 032737 000001 000556 BIT #1,FMCNT ;SEE IF ODD FRAME COUNT  
3703 015140 001401 BEQ DFOA ;IF NOT: BR  
3704 015142 105722 TSTB (R2)+ ;BUMP POINTER  
3705 015144 032737 000020 000552 DFOA: BIT #20,UDES ;SEE IF CORE DUMP  
3706 015152 001431 BEQ DFOA4 ;IF NOT: BR  
3707 015154 000241 CLC
```

```
3708 015156 132742 000001      BITB  #1,-(R2)      ;SEE IF BIT 0 = 1
3709 015162 001401              BEQ   DF0A0        ;IF NOT: BR
3710 015164 000261              SEC
3711 015166 106012      DFOA0: RORB  (R2)
3712 015170 000241              CLC
3713 015172 132712 000001      BITB  #1,(R2)
3714 015176 001401              BEQ   DF0A1
3715 015200 000261              SEC
3716 015202 106012      DFOA1: RORB  (R2)      ;POSITION BITS FOR REVERSE CORE DUMP
3717 015204 000241              CLC
3718 015206 132712 000001      BITB  #1,(R2)
3719 015212 001401              BEQ   DF0A2
3720 015214 000261              SEC
3721 015216 106012      DFOA2: RORB  (R2)
3722 015220 000241              CLC
3723 015222 132712 000001      BITB  #1,(R2)
3724 015226 001401              BEQ   DF0A3
3725 015230 000261              SEC
3726 015232 106012      DFOA3: RORB  (R2)
3727 015234 005202      DFOA4: INC   R2      ;RESET POINTER
3728 015236 124142      DFOA4: CMPB  -(R1),-(R2) ;TEST DATA CHARACTER
3729 015240 001010              BNE   DF1          ;IF NOT GOOD: BR
3730 015242 105037 000656      CLRB  BBC          ;CLEAR BAD RECORD COUNTER
3731 015246 000411              BR    DF2
3732 015250 122122      DFO:  CMPB  (R1)+,(R2)+ ;CHECK DATA
3733 015252 001003              BNE   DF1          ;IF BAD: BR
3734 015254 105037 000656      CLRB  BBC          ;CLEAR BAD RECORD CNTR
3735 015260 000404              BR    DF2
3736 015262 004737 016020      DF1:  JSR   PC,DRPKF ;GO GET DROPS AND PICKS
3737 015266 004737 015354              JSR   PC,DERR      ;GO DO PRINT
3738 015272 005205      DF2:  INC   R5      ;BUMP CHAR CNTR
3739 015274 001404              BEQ   DF3          ;IF DONE ALL: BR
3740 015276 005737 000562      TST   RDCMD       ;SEE IF READ REVERSE
3741 015302 001762              BEQ   DFO         ;IF NOT: BR
3742 015304 000717              BR    DFOA        ;ELSE CONTINUE READ REV
3743 015306 005037 000664      DF3:  CLR   HDRFL   ;CLEAR HEADER FLAG
3744 015312 005737 000704              TST   DERFL       ;SEE IF HAD DATA ERROR
3745 015316 001415              BEQ   DFX         ;IF NOT: BR
3746 015320 005737 000706      TST   SERFL
3747 015324 001012              BNE   DFX         ;IF NOT DATA ERROR ONLY: BR
3748 015326 013704 000674      MOV   UNP,R4
3749 015332 005737 000562      TST   RDCMD       ;SEE IF READ REVERSE
3750 015336 001003              BNE   DF4         ;IF SO: BR
3751 015340 005264 001124              INC   DATER1(R4)  ;BUMP DATA ERROR FORWARD COUNTER
3752 015344 000402              BR    DFX
3753 015346 005264 001164      DF4:  INC   DEREV1(R4) ;BUMP REVERSE DATA ERROR
3754 015352 000207      DFX:  RTS   PC     ;EXIT
3755
```

3757  
3758  
3759  
3760  
3761  
3762  
3763  
3764  
3765  
3766  
3767  
3768  
3769  
3770  
3771  
3772  
3773  
3774  
3775  
3776  
3777  
3778  
3779  
3780  
3781  
3782  
3783  
3784  
3785  
3786  
3787  
3788  
3789  
3790  
3791  
3792  
3793  
3794  
3795  
3796  
3797  
3798  
3799  
3800  
3801  
3802  
3803  
3804  
3805  
3806  
3807  
3808  
3809  
3810  
3811  
3812

015354 032777 002000 163226  
015362 001057  
015364 005237 000670  
015370 005737 000664  
015374 001006  
015376 004737 022012  
015402 000004 023522  
015406 004737 020250  
015412 000004 023541  
015416 010203  
015420 162703 032352  
015424 005303  
015426 005737 000562  
015432 001402  
015434 010503  
015436 005103  
015440 104400  
015442 000004 023527  
015446 005737 000562  
015452 001402  
015454 111103  
015456 000401  
015460 114103  
015462 004737 023304  
015466 000004 023534  
015472 005737 000562  
015476 001402  
015500 111203

```
*****  
:DATA ERROR SUBROUTINE:  
:  
:THIS SUBROUTINE IS USED TO PRINT OUT ANY  
:ERRORS FOUND DURING THE DATA CHECK.  
:EACH CHARACTER FOUND BAD WILL BE PRINTED  
:IN BIT FORMAT ALONG WITH ITS EXPECTED CHARACTER.  
:AN ERROR HEADER CONSISTING OF THE UNIT NUMBER,  
:BLOCK NUMBER, RECORD NUMBER, SIZE OF RECORD, AND  
:ERROR TYPE (READ FORWARD, READ REVERSE, WRITE, ETC)  
:IS PRINTED ONLY ONCE FOR EACH RECORD FOUND BAD.  
:A COUNT IS MADE OF THE NUMBER OF SUCCESSIVE BAD  
:CHARACTERS, AND IF TEN (10) SUCCESSIVE BAD CHARACTERS  
:ARE FOUND IN A SINGLE RECORD, A MESSAGE INDICATING  
:A BAD RECORD CONDITION IS PRINTED AND THE NEXT  
:TWENTY (20) CHARACTERS ARE SKIPPED BEFORE CHECKING  
:IS RESUMED. IF THE BAD RECORD CONDITION IS FOUND  
:THREE TIMES IN A RECORD, ALL REMAINING DATA IS  
:SKIPPED EXCEPT THE FINAL TEN (10) CHARACTERS.  
:THIS SKIPPING IS OF COURSE ONLY POSSIBLE IN  
:RECORDS WHICH CONTAIN A SUFFICIENT NUMBER OF CHARACTERS.  
:PRINTING OF ERRORS MAY BE DISALLOWED AT ANY TIME  
:BY SETTING CONSOLE SWITCH TEN (10) TO A ONE.  
:THE OPERATOR MAY CAUSE THE PROGRAM TO HALT ON ANY ERROR  
:BY SETTING CONSOLE SWITCH FIFTEEN (15) TO A ONE.  
*****  
  
DERR: BIT #2000,@SWR ;BRANCH IF NO ERROR  
BNE DERR4 ;PRINTOUT DESIRED  
DERR0: INC PFLG ;SET PRINT FLAG  
TST HDRFL ;SEE IF HAVE PRINTED HEADER  
BNE DERR0A ;IF SO: BR  
JSR PC,PAPRT ;PRINT CYCLE NUMBER  
TYPE,MSG1 ;TYPE DATA ERROR TAG '*DE'  
DERR0A: JSR PC,FRPRT ;PRINT F OR R  
TYPE,MSG4 ;TYPE CHAR # TAG 'CN'  
MOV R2,R3 ;POINT TO CHAR  
SUB #RDATA,R3  
DEC R3  
TST RDCMD ;SEE IF READ REVERSE  
BEQ DERR0B ;IF NOT: BR  
MOV R5,R3 ;GET CHAR NUMBER  
COM R3  
DERR0B: TYPOCT ;PRINT CHAR NUMBER  
TYPE,MSG2 ;TYPE GOOD CHAR TAG 'G'  
TST RDCMD ;SEE IF READ REVERSE  
BEQ DERR0C ;IF NOT: BR  
MOVB (R1),R3 ;GET CHAR  
BR DERR0D  
DERR0C: MOVB -(R1),R3 ;LOAD EXPECTED DATA  
DERR0D: JSR PC,DOUT ;GO PRINT CHAR  
TYPE,MSG3 ;TYPE BAD CHARACTER TAG 'B'  
TST RDCMD ;SEE IF READ REVERSE  
BEQ DERR1 ;IF NOT: BR  
MOVB (R2),R3 ;GET CHAR
```

3813	015502	000401				BR	DERR2		
3814	015504	114203				DERR1: MOV	-(R2),R3		
3815	015506	004737	023304			DERR2: JSR	PC,DOUT	:	PRINT BAD CHAR
3816	015512	005737	000562			TST	RDCMD	:	BRANCH IF READ
3817	015516	001001				BNE	DERR4	:	REVERSE
3818	015520	122122				DERR3: CMPB	(R1)+,(R2)+	:	RESET POINTERS
3819	015522	105237	000656			DERR4: INCB	BBC	:	BUMP BAD RECORD CNTR
3820	015526	122737	000010	000656		CMPB	#10,BBC	:	SEE IF BLD BTH
3821	015534	001107				BNE	DEREX	:	IF NOT: BR
3822	015536	032777	002000	163044		BIT	#2000,@SWR	:	SEE IF PRINT INHIBIT
3823	015544	001002				BNE	1\$	:	IF SO: BR
3824	015546	000004	023622			TYPE,MSG15		:	TYPE 'BAD RECORD'
3825	015552	105037	000656			1\$: CLRB	BBC	:	RESET BAD RECORD CNTR
3826	015556	105237	000657			INCB	BBC+1	:	BUMP AMOUNT
3827	015562	122737	000003	000657		CMPB	#3,BBC+1	:	SEE IF HAD 3 BLD BTHS
3828	015570	101047				BHI	DERR4B	:	IF NOT: BR
3829	015572	022705	177767			CMP	#177767,R5	:	SEE IF ON LAST EIGHT CHARS
3830	015576	101464				BLOS	DERR6	:	IF SO: BR
3831	015600	012705	177767			MOV	#177767,R5	:	SET CHAR CNTR TO 8
3832	015604	005737	000562			TST	RDCMD	:	SEE IF READ REVERSE
3833	015610	001416				BEQ	DERR4A	:	IF NOT: BR
3834	015612	012701	026344			MOV	#WDATA,R1	:	GET START OF BUFFER
3835	015616	012702	032352			MOV	#RDATA,R2	:	GET START OF BUFFER
3836	015622	062701	000010			ADD	#10,R1		
3837	015626	062702	000010			ADD	#10,R2	:	POINT TO START +10
3838	015632	032737	000001	000556		BIT	#1,FMCNT	:	SEE IF ODD FRAME COUNT
3839	015640	001445				BEQ	DEREX	:	IF NOT: BR
3840	015642	105722				TSTB	(R2)+	:	BUMP POINTER
3841	015644	000443				BR	DEREX		
3842	015646	013737	000556	000644		DERR4A: MOV	FMCNT,TEMP1	:	LOAD CHAR COUNT
3843	015654	005437	000644			NEG	TEMP1	:	++B
3844	015660	162737	000010	000644		SUB	#10,TEMP1	:	POINT TO BUFFER -8
3845	015666	013701	000644			MOV	TEMP1,R1	:	POINT TO NEXT CHAR
3846	015672	062701	026344			ADD	#WDATA,R1	:	POINT TO NEXT WRITE CHAR
3847	015676	013702	000644			MOV	TEMP1,R2	:	POINT TO END OF READ DATA -8 FORWARD
3848	015702	062702	032352			ADD	#RDATA,R2	:	POINT TO NEXT CHAR
3849	015706	000422				BR	DEREX	:	EXIT
3850	015710	062705	000024			DERR4B: ADD	#24,R5	:	SKIP 20 CHARS
3851	015714	103415				BCS	DERR6	:	IF EXCEED RECORD SIZE: BR
3852	015716	005737	000562			TST	RDCMD	:	SEE IF READ REVERSE
3853	015722	001405				BEQ	DERR5	:	IF NOT: BR
3854	015724	162701	000024			SUB	#24,R1		
3855	015730	162702	000024			SUB	#24,R2	:	RESET POINTERS
3856	015734	000407				BR	DEREX		
3857	015736	062701	000024			DERR5: ADD	#24,R1	:	SKIP 20 CHARS
3858	015742	062702	000024			ADD	#24,R2	:	SKIP FORWARD 20 CHARS
3859	015746	000402				BR	DEREX		
3860	015750	012705	177777			DERR6: MOV	#-1,R5	:	SET TO EOR
3861	015754	005777	162630			DEREX: TST	@SWR	:	BRANCH IF NOT HALT ON ERROR
3862	015760	100012				BPL	DEREX1		
3863	015762	000000				HALT			
3864	015764	005737	000670			TST	PFLG	:	SEE IF PRINTED
3865	015770	001006				BNE	DEREX1	:	IF SO: BR
3866	015772	032777	002000	162610		BIT	#2000,@SWR	:	SEE IF SHOULD PRINT
3867	016000	001002				BNE	DEREX1	:	IF NOT: BR
3868	016002	000137	015364			JMP	DERRO	:	ELSE PRINT

CZTEDDO TMO3-TE16/TU77 DRT  
CZTEDD.P11 06-JUL-83 14:42

MACY11 30(1046) 06-JUL-83 21:03 <sup>6 7</sup>PAGE 55-2

SEQ 0084

3869 016006 005037 000670  
3870 016012 005237 000704  
3871 016016 000207  
3872

DEREX1: CLR PFLG ;CLEAR FLAG  
INC DERFL ;BUMP DATA ERROR FLAG  
RTS PC ;RETURN

3874  
3875  
3876  
3877  
3878  
3879  
3880  
3881  
3882  
3883  
3884  
3885  
3886  
3887  
3888  
3889  
3890  
3891  
3892  
3893  
3894  
3895  
3896  
3897  
3898  
3899  
3900  
3901  
3902  
3903  
3904  
3905  
3906  
3907  
3908  
3909  
3910  
3911  
3912  
3913  
3914  
3915  
3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924  
3925  
3926  
3927  
3928  
3929

016020 005037 000644  
016024 005037 000646  
016030 005037 000650  
016034 111137 000644  
016040 111237 000646  
016044 013704 000674  
016050 016437 000764  
016056 016437 001004  
016064 005737 000562  
016070 001005  
016072 124142  
016074 112137 000644  
016100 112237 000646  
016104 004737 016116  
016110 004737 016324  
016114 000207  
016116 113703 000644  
016122 113704 000646  
016126 140403  
016130 001001  
016132 000207  
016134 012737 000010  
016142 132703 000001  
016146 001451  
016150 105737 000650  
016154 001014  
016156 005277 162534  
016162 100043  
016164 032777 002000  
016172 001402  
016174 004737 022012  
016200 004737 016370  
016204 000413  
016206 005277 162506  
016212 100027  
016214 032777 002000  
016222 001402

000720  
000716

000710

162366

```
DRPKF: CLR TEMP1
        CLR TEMP2
        CLR TEMP3
        MOV (R1),TEMP1 ;LOAD GOOD CHAR
        MOV (R2),TEMP2 ;LOAD BAD CHAR
        MOV UNP,R4
        MOV PIK1(R4),BPKP
        MOV DRP1(R4),BDPP
        TST RDCMD ;SEE IF READ REVERSE
        BNE DRPK ;IF SO: BR
        CMPB -(R1),-(R2) ;POINT TO CHAR
        MOV (R1)+,TEMP1 ;LOAD GOOD CHAR
        MOV (R2)+,TEMP2 ;LOAD BAD CHAR
DRPK: JSR PC,DROP ;GET DROPS
      JSR PC,PICK ;GET PICKS
      RTS PC ;EXIT

DROP: MOVB TEMP1,R3 ;R3 = GOOD CHAR
      MOVB TEMP2,R4 ;R4 = BAD CHAR
DPC: BICB R4,R3 ;GET DROPS/PICKS
     BNE DPCG ;IF SOME: BR
     RTS PC ;RETURN
DPCG: MOV #10,BCNT ;SET NUMBER TO CHECK
DPC0: BITB #1,R3 ;SEE IF DROPPED OR PICKED THIS BIT
     BEQ DPC2 ;IF NOT: BR
     TSTB TEMP3 ;SEE IF ON PICKS
     BNE DPC1 ;IF SO: BR
     INC @BDPP ;BUMP DROP CNTR
     BPL DPC2 ;IF NO OVERFLOW: BR
     BIT #2000,@SWR ;SEE IF HAVE PRINTED DATA
     BEQ DPC0A ;IF SO: BR
     JSR PC,PAPRT ;PRINT CYCLE NUMBER
     JSR PC,DPPRT ;PRINT DROPS AND PICKS
DPC0A: BR DPC2A
DPC1: INC @BPKP ;BUMP PICK CNTR
      BPL DPC2 ;& BR IF NO OVERFLOW
      BIT #2000,@SWR ;SEE IF HAVE PRINTED DATA
      BEQ DPC1A ;IF SO: BR
```

```
*****
:DROPS AND PICKS SUBROUTINE:
:
:THIS SUBROUTINE IS USED TO ACCUMULATE FROM
: EACH BAD DATA CHARACTER FOUND THE NUMBER
: OF BITS WHICH WERE EITHER DROPPED OR PICKED UP.
: TWO COUNTERS PER SLAVE ARE USED TO ACCUMULATE THIS
: INFORMATION AND CAN STORE UP TO 32K DROPS
: OR PICKS BEFORE OVERFLOWING. IF OVERFLOW IS
: ABOUT TO OCCUR, THESE ACCUMULATORS ARE
: PRINTED IN OCTAL AND RESET TO ZERO.
: THE CONTENTS OF THE ACCUMULATORS MAY BE
: DISPLAYED AT ANY TIME BY SETTING CONSOLE
: SWITCH FOURTEEN TO A ONE (1). THE PRINTOUT WILL OCCUR
: AT THE END OF THE CURRENT BLOCK CYCLE.
*****
```

```

3930 016224 004737 022012          JSR      PC,PAPRT      ;PRINT CYCLE NUMBER
3931 016230 004737 016370          DPC1A: JSR      PC,DPPRT ;PRINT DROPS AND PICKS
3932 016234 013704 000674          DPC2A: MOV      UNP,R4
3933 016240 016403 001004          MOV      DRP1(R4),R3 ;SET DROP POINTER
3934 016244 016404 000764          MOV      PIK1(R4),R4 ;SET PICK POINTER
3935 016250 012737 000010 000710 MOV      #10,BCNT     ;SET NUMBER OF BITS
3936 016256 005023          DPC2B: CLR      (R3)+   ;CLEAR DROPS
3937 016260 005024          CLR      (R4)+   ;CLEAR PICK
3938 016262 005337 000710          DEC      BCNT     ;SEE IF DONE
3939 016266 001373          BNE      DPC2B   ;IF NOT: BR
3940 016270 000207          RTS      PC      ;EXIT
3941 016272 000241          DPC2:  CLC
3942 016274 106003          RORB     R3      ;GET NEXT BIT
3943 016276 005337 000710          DEC      BCNT     ;SEE IF DONE
3944 016302 001407          BEQ      DPC3
3945 016304 062737 000002 000720 ADD      #2,BPKP
3946 016312 062737 000002 000716 ADD      #2,BDPP
3947 016320 000710          BR       DPC0
3948 016322 000207          DPC3:  RTS      PC      ;CONTINUE
3949 016324 013704 000674          PICK:  MOV      UNP,R4 ;RETURN
3950 016330 016437 000764 000720 MOV      PIK1(R4),BPKP ;GET UNIT POINTER
3951 016336 016437 001004 000716 MOV      DRP1(R4),BDPP ;SET PICK POINTER
3952 016344 113704 000644          MOVB    TEMP1,R4 ;SET DROP POINTER
3953 016350 113703 000646          MOVB    TEMP2,R3 ;R4 = GOOD CHAR
3954 016354 112737 000001 000650 MOVB    #1,TEMP3 ;R3 = BAD CHAR
3955 016362 004737 016126          JSR      PC,DPC   ;SET PICK FLAG
3956 016366 000207          RTS      PC      ;GO CHECK PICKS
3957 016370 000004 024133          DPPRT: TYPE,MSG26 ;EXIT
3958 016374 013704 000674          MOV      UNP,R4 ;TYPE 'DROPS'
3959 016400 016437 001004 000716 MOV      DRP1(R4),BDPP ;SET DROP POINTER
3960 016406 016437 000764 000720 MOV      PIK1(R4),BPKP ;SET PICK POINTER
3961 016414 062757 000016 000716 ADD      #16,BDPP
3962 016422 062737 000016 000720 ADD      #16,BPKP
3963 016430 012737 000010 000710 MOV      #10,BCNT ;SET NUMBER TO PRINT
3964 016436 017703 162254          DPPRT0: MOV     @BDPP,R3
3965 016442 104400          TYPOCT
3966 016444 005337 000710          DEC      BCNT     ;PRINT DROPS
3967 016450 001404          BEQ      DPPRT1 ;SEE IF DONE
3968 016452 162737 000002 000716 SUB      #2,BDPP   ;IF NOT: BR
3969 016460 000766          BR       DPPRT0 ;BUMP POINTER
3970 016462 012737 000010 000710 DPPRT1: MOV     #10,BCNT ;CONTINUE FOR ALL 8 BITS
3971 016470 000004 024144          TYPE,MSG27 ;SET NUMBER TO PRINT
3972 016474 017703 162220          DPPRT2: MOV     @BPKP,R3 ;TYPE 'PICKS'
3973 016500 104400          TYPOCT
3974 016502 005337 000710          DEC      BCNT     ;PRINT PICKS
3975 016506 001404          BEQ      DPPRTX ;SEE IF DONE
3976 016510 162737 000002 000720 SUB      #2,BPKP   ;IF SO: BR
3977 016516 000766          BR       DPPRT2 ;BUMP POINTER
3978 016520 000207          DPPRTX: RTS     PC ;CONTINUE FOR ALL 8 BITS
                                     ;RETURN

```

3980  
3981  
3982  
3983  
3984  
3985  
3986  
3987  
3988  
3989  
3990  
3991  
3992  
3993  
3994  
3995  
3996  
3997  
3998  
3999  
4000  
4001  
4002  
4003  
4004

```
*****  
:STATUS CHECK SUBROUTINE:  
:THIS SUBROUTINE IS USED TO PERFORM A CHECK OF  
:BOTH THE MASSBUS CONTROLLER (RH11) AND THE TAPE  
:CONTROLLER (TM02). THE RH11 IS CHECKED FOR ERRORS  
:AS REFLECTED IN REGISTERS CS1 AND CS2 AND ALSO THAT  
:THE BUS ADDRESS (BA) AND WORD COUNT (WC) ARE  
:CORRECT. THE TM02 IS CHECKED FOR DRIVE STATIS (DS),  
:DRIVE ERRORS (ER), AND PROPER FRAME COUNT. THE SPECIAL  
:CHECK CHARACTERS (CRC+LRC) ARE ALSO CHECKED WHEN  
:APPROPRIATE (IE: NRZ READ OR WRITE). CERTAIN TYPES  
:OF DRIVE ERRORS IN PE OPERATION WILL BE ACCOMPANIED  
:BY THE DISPLAY OF THE DEAD TRACK REGISTER (CC). THESE  
:TYPES ARE ER BITS 15,10,7,6. THE PRINTOUTS OF BAD  
:CRC,LRC,FC, AND BA WILL SHOW BOTH THE EXPECTED AND  
:RECEIVED VALUES (IE: EXPT-RCVD). ONLY THOSE REGISTERS  
:WHICH ARE IN ERROR WILL BE PRINTED AND ALL PRINTOUTS  
:ARE IN OCTAL FORMAT WITH NO LEADING ZEROS. AS IN  
:DATA ERRORS, STATUS ERRORS ARE PRECEDED BY HEADER  
:DESCRIBING THE HARDWARE UNDER TEST, THE BLOCKING  
:INFORMATION, AND THE ERROR TYPE.  
*****
```

4005	016522	013703	000556		ERCHK:	MOV	FMCNT,R3		:GET FRAME COUNT
4006	016526	032703	000001			BIT	#1,R3		:SEE IF ODD
4007	016532	001401				BEQ	1\$		:IF NOT: BR
4008	016534	005303				DEC	R3		:BUMP COUNT
4009	016536	005403			1\$:	NEG	R3		
4010	016540	032737	000020	000552		BIT	#20,UDES		:SEE IF CORE DUMP
4011	016546	001401				BEQ	2\$		:IF NOT: BR
4012	016550	006203				ASR	R3		:SET TO FC/2
4013	016552	032737	000010	000672	2\$:	BIT	#10,MTC1		:SEE IF WRITE OP
4014	016560	001413				BEQ	4\$		:IF SO: BR
4015	016562	005737	000562			TST	RDCMD		
4016	016566	001405				BEQ	3\$		
4017	016570	012703	032352			MOV	#RDATA,R3		
4018	016574	162703	000002			SUB	#2,R3		:SET POINTER
4019	016600	000405				BR	ER2		
4020	016602	062703	032352		3\$:	ADD	#RDATA,R3		:BUILD EXPT READ ADDRESS
4021	016606	000402				BR	ER2		
4022	016610	062703	026344		4\$:	ADD	#WDATA,R3		:BUILD EXPT WRITE ADDRESS
4023									
4024	016614	032777	040000	161702	ER2:	BIT	#40000,@ER		:BRANCH IF NOT UNSAFE
4025	016622	001403				BEQ	1\$		
4026	016624	005726				TST	(SP)+		:ADJUST STACK
4027	016626	000137	020312			JMP	OFFLINE		:GO MARK UNIT OFFLINE
4028	016632	010337	020224		1\$:	MOV	R3,CADER		:SAVE ADDRESS
4029	016636	012704	000007			MOV	#7,R4		
4030	016642	012701	020226			MOV	#BAER,R1		
4031	016646	005021			2\$:	CLR	(R1)+		:CLEAR FLAGS
4032	016650	005304				DEC	R4		
4033	016652	001375				BNE	2\$		
4034	016654	020377	161634			CMP	R3,@BA		:SEE IF ADDRESS OK
4035	016660	001402				BEQ	3\$		:IF SO: BR



4036	016662	005237	020226			INC	BAER	:SET BUS ADDRESS ERROR
4037	016666	032737	000010	000672	3\$:	BIT	#10,MTC1	:SEE IF WRITE OPER
4038	016674	001006				BNE	5\$	:IF NOT: BR
4039	016676	005777	161614		4\$:	TST	@FC	:SEE IF FC=0
4040	016702	001440				BEQ	ER3	:IF SO: BR
4041	016704	005237	020234			INC	FCER	:SET FC ERROR
4042	016710	000435				BR	ER3	
4043	016712	032737	000040	000672	5\$:	BIT	#40,MTC1	:SEE IF SPACE OPER
4044	016720	001766				BEQ	4\$	:IF SO: BR
4045	016722	005737	000676			TST	TMFLG	:SEE IF TM TIME
4046	016726	001011				BNE	7\$	:IF SO: BR
4047	016730	013703	000556			MOV	FMCNT,R3	
4048	016734	005403				NEG	R3	:R3 = EXPT RECORD SIZE
4049	016736	020377	161554		6\$:	CMP	R3,@FC	:SEE IF FC = EXPT
4050	016742	001420				BEQ	ER3	:IF SO: BR
4051	016744	005237	020234			INC	FCER	:SET FC ERROR FLAG
4052	016750	000415				BR	ER3	
4053	016752	032737	002000	000552	7\$:	BIT	#2000,UDES	:SEE IF PE
4054	016760	001346				BNE	4\$	:IF SO: BR
4055	016762	005737	000562			TST	RDCMD	:SEE IF READ REVERSE
4056	016766	001003				BNE	8\$	:IF SO: BR
4057	016770	012703	000002			MOV	#2,R3	
4058	016774	000760				BR	6\$	:LOOK FOR EXPT = 2
4059	016776	012703	000001		8\$:	MOV	#1,R3	
4060	017002	000755				BR	6\$	:GO CHECK FC FOR TM
4061								
4062	017004	032777	160000	161476	ER3:	BIT	#160000,@C1	:SEE IF COUNT ERROR
4063	017012	001437				BEQ	ER4	
4064	017014	017703	161500			MOV	@CS,R3	:GET CONT STATUS REG
4065	017020	042703	000307			BIC	#307,R3	:MASK OUT IR,OR,UNIT NO. & SEE IF OTHER ERRORS
4066	017024	001406				BEQ	1\$	:IF NOT: BR
4067	017026	005737	000676			TST	TMFLG	:SEE IF TAPE MARK TIME
4068	017032	001425				BEQ	3\$	:IF NOT: BR
4069	017034	042703	001000			BIC	#1000,R3	:MASK MISSED TRANS & BR IF OTHER ERRORS
4070	017040	001022				BNE	3\$	
4071	017042	032777	060000	161440	1\$:	BIT	#60000,@C1	:SEE IF EITHER TRE OR MCPE
4072	017050	001420				BEQ	ER4	:IF NOT: BR
4073	017052	005737	000676			TST	TMFLG	:SEE IF TM TIME
4074	017056	001413				BEQ	3\$	:IF NOT: BR
4075	017060	017703	161440			MOV	@ER,R3	:GET ERROR REGISTER
4076	017064	032737	000010	000552		BIT	#10,UDES	:SEE IF EVEN PARITY
4077	017072	001402				BEQ	2\$	:IF NOT: BR
4078	017074	042703	000100			BIC	#100,R3	:MASK PAR
4079	017100	042703	001000		2\$:	BIC	#1000,R3	:MASK FCE
4080	017104	001402				BEQ	ER4	:IF NO ERRORS EXCEPT FCE: BR
4081	017106	005237	020230		3\$:	INC	CONER	:SET CONT ERROR FLAG
4082								
4083	017112	032777	040000	161402	ER4:	BIT	#40000,@DS	:SEE IF DRIVE ERROR
4084	017120	001420				BEQ	ER6	:IF NOT: BR
4085	017122	005737	000676			TST	TMFLG	:SEE IF TAPE MARK TIME
4086	017126	001413				BEQ	2\$	:IF NOT: BR
4087	017130	017703	161370			MOV	@ER,R3	:GET ER
4088	017134	032737	000010	000552		BIT	#10,UDES	:SEE IF EVEN PARITY
4089	017142	001402				BEQ	1\$	:IF NOT: BR
4090	017144	042703	000100			BIC	#100,R3	:MASK PAR
4091	017150	042703	001000		1\$:	BIC	#1000,R3	:MASK OUT FCE & BRANCH IF

4092	017154	001402				BEQ	ER6	:NO OTHER ERRORS
4093	017156	005237	020232		2\$:	INC	DRVER	:SET DRIVER ERROR FLAG
4094								
4095	017162	013737	014744	020246	ER6:	MOV	EXCRC,CRCSV	:SAVE EXPECTED CRC
4096	017170	013737	014746	020244		MOV	EXLRC,LRCV	:AND EXPECTED LRC
4097	017176	032737	002000	000552		BIT	#2000,UDES	
4098	017204	001062				BNE	ERPT	:IF IN PE MODE: BR
4099	017206	032777	020000	161374		BIT	#20000,@SWR	:SEE IF NO DATA CHECK
4100	017214	001056				BNE	ERPT	:IF NOT: BR (ALLOW READ OF UNKNOWN TAPES)
4101	017216	032737	000040	000672		BIT	#40,MTC1	:SEE IF WRITE OR READ OP
4102	017224	001452				BEQ	ERPT	:IF NOT: BR
4103	017226	005737	000676			TST	TMFLG	:SEE IF TAPE MARK TIME
4104	017232	001405				BEQ	1\$	:IF NOT: BR
4105	017234	005037	014744			CLR	EXCRC	
4106	017240	012737	000023	014746		MOV	#23,EXLRC	:SET CRC/LRC FOR TM
4107	017246	032737	000060	000552	1\$:	BIT	#60,UDES	:SEE IF FORMAT 14
4108	017254	001036				BNE	ERPT	:IF NOT: BR
4109	017256	017703	161246			MOV	@CC,R3	:GET CRC CHARACTER
4110	017262	042703	177000			BIC	#177000,R3	
4111	017266	023703	014744			CMP	EXCRC,R3	
4112	017272	001402				BEQ	2\$	:IF CRC GOOD: BR
4113	017274	005237	020240			INC	CR CER	:SET ERROR FLAG
4114	017300	017703	161230		2\$:	MOV	@MR,R3	:GET LRC
4115	017304	000303				SWAB	R3	
4116	017306	005703				TST	R3	
4117	017310	100002				BPL	3\$	
4118	017312	052703	000400			BIS	#400,R3	
4119	017316	042703	177000		3\$:	BIC	#177000,R3	
4120	017322	023703	014746			CMP	EXLRC,R3	
4121	017326	001411				BEQ	ERPT	:IF LRC GOOD: BR
4122	017330	010337	020242			MOV	R3,ACTLRC	:SAVE ACTUAL LRC
4123	017334	005237	020236			INC	LRCER	:SET LRC ERROR FLAG
4124	017340	005737	000562			TST	RDCMD	:SEE IF READ REVERSE
4125	017344	001402				BEQ	ERPT	:IF NOT: BR
4126	017346	005037	020236			CLR	LRCER	:ELSE CLEAR LRC ERROR
4127	017352	012703	000006		ERPT:	MOV	#6,R3	
4128	017356	005037	000706			CLR	SERFL	:CLEAR ERROR FLAG
4129	017362	005037	000722			CLR	ERSAV	
4130	017366	012704	020226			MOV	#BAER,R4	
4131	017372	005724			ERPTT:	TST	(R4)+	:SEE IF ANY ERROR
4132	017374	001004				BNE	ERPTG	:IF SO: BR
4133	017376	005303				DEC	R3	
4134	017400	001374				BNE	ERPTT	
4135	017402	000137	020170			JMP	ERPX1	
4136	017406	005237	000706		ERPTG:	INC	SERFL	:SET ERROR FLAG
4137	017412	017737	161106	000722		MOV	@ER,ERSAV	:SAVE ERROR REGISTER
4138	017420	032777	002000	161162		BIT	#2000,@SWR	:SEE IF PRINT
4139	017426	001420				BEQ	ERPT0	:IF SO: BR
4140	017430	022737	000002	000712		CMP	#2,RTYFL	:SEE IF READ RETRY
4141	017436	001006				BNE	ERPTG1	:IF NOT: BR
4142	017440	013703	000702			MOV	RTCNT,R3	
4143	017444	005203				INC	R3	:BUMP RETRY COUNT
4144	017446	020337	000604			CMP	R3,RETRY	:SEE IF LAST RETRY
4145	017452	001406				BEQ	ERPT0	:IF SO: BR
4146	017454	022737	000002	020232	ERPTG1:	CMP	#2,DRVER	:SEE IF TM STATUS ERROR
4147	017462	001402				BEQ	ERPT0	:IF SO: BR

4148	017464	000137	020050		JMP	ERPX0		
4149	017470	005237	000670		INC	PFLG		
4150	017474	004737	022012		JSR	PC,PAPRT		:PRINT HEADER
4151	017500	013737	000652	017510	MOV	EMADDR,1\$		:GET ADDRESS OF ERROR MSG HEADER
4152	017506	000004			TYPE			
4153	017510	000000			1\$:	.WORD 0		:ADDRESS OF ERROR MESSAGE HEADER
4154	017512	004737	020250		JSR	PC,FRPRT		:PRINT F OR R
4155	017516	005737	000676		TST	TMFLG		
4156	017522	001406			BEQ	ERPT1		
4157	017524	022737	025012	000652	CMP	#MSG54,EMADDR		
4158	017532	001402			BEQ	ERPT1		
4159	017534	000004	025030		TYPE,MSG56			:TYPE 'TM'
4160	017540	005737	020230		ERPT1:	TST CONER		
4161	017544	001412			BEQ	ERPT2		:IF NO CONT ERROR: BR
4162	017546	000004	023737		TYPE,MSG23			:TYPE 'CS1'
4163	017552	017703	160732		MOV	@C1,R3		
4164	017556	104400			TYPOCT			:PRINT CONTROL 1
4165	017560	000004	023764		TYPE,MSG23D			:TYPE CS TAG
4166	017564	017703	160730		MOV	@CS,R3		
4167	017570	104400			TYPOCT			:PRINT CONT STATUS
4168	017572	005737	020232		ERPT2:	TST DRIVER		
4169	017576	001412			BEQ	ERPT3		:IF SO DRIVE ERROR: BR
4170	017600	000004	023772		TYPE,MSG23E			:TYPE DS TAG
4171	017604	017703	160712		MOV	@DS,R3		
4172	017610	104400			TYPOCT			:PRINT DRIVE STATUS
4173	017612	000004	023777		TYPE,MSG23F			:TYPE ER TAG
4174	017616	017703	160702		MOV	@ER,R3		
4175	017622	104400			TYPOCT			:PRINT DRIVE ERROR
4176	017624	005737	020226		ERPT3:	TST BAER		
4177	017630	001412			BEQ	ERPT4		:IF NO BA ERROR: BR
4178	017632	000004	023752		TYPE,MSG23B			:TYPE BA TAG
4179	017636	017703	160652		MOV	@BA,R3		
4180	017642	104400			TYPOCT			:PRINT BUS ADDRESS
4181	017644	000004	023520		TYPE,DASH			
4182	017650	013703	020224		MOV	CADER,R3		
4183	017654	104400			TYPOCT			:PRINT EXPT BUS ADDRESS
4184	017656	005737	020234		ERPT4:	TST FCER		
4185	017662	001405			BEQ	ERPT5		:IF NO FC ERROR: BR
4186	017664	000004	023757		TYPE,MSG23C			:TYPE FC TAG
4187	017670	017703	160622		MOV	@FC,R3		
4188	017674	104400			TYPOCT			:PRINT FRAME COUNT
4189	017676	000004	023745		ERPT5:	TYPE,MSG23A		:TYPE WC TAG
4190	017702	017703	160604		MOV	@WC,R3		
4191	017706	104400			TYPOCT			:PRINT WORD COUNT
4192	017710	005737	020240		TST	CRCER		
4193	017714	001414			BEQ	ERPT5A		:IF NO CRC ERROR: BR
4194	017716	000004	025055		TYPE,MSG58			:TYPE CRC TAG
4195	017722	017703	160602		MOV	@CC,R3		
4196	017726	042703	177000		BIC	#177000,R3		
4197	017732	104400			TYPOCT			:PRINT ACTUAL CRC
4198	017734	000004	023520		TYPE,DASH			
4199	017740	013703	014744		MOV	EXCRC,R3		
4200	017744	104400			TYPOCT			:PRINT EXPECTED CRC
4201	017746	005737	020236		ERPT5A:	TST LRCER		
4202	017752	001412			BEQ	ERPT6		:IF NO LRC ERROR: BR
4203	017754	000004	025063		TYPE,MSG59			:TYPE LRC ERR TAG



4260  
4261  
4262  
4263  
4264  
4265  
4266  
4267  
4268  
4269  
4270  
4271  
4272  
4273  
4274  
4275  
4276  
4277  
4278  
4279  
4280  
4281  
4282  
4283  
4284  
4285  
4286  
4287  
4288  
4289  
4290  
4291  
4292  
4293  
4294  
4295

```
*****  
:F FOR FORWARD/R FOR REVERSE PRINT SUBROUTINE:  
:THIS SUBROUTINE IS USED TO PRINT OUT THE  
:TAPE DIRECTION USED WHEN ANY ERROR IS  
:DETECTED IN STATUS OF READ OR WRITE, DATA, OR  
:SPACING OPERATIONS.  
*****
```

```
020250 032737 000010 000672 FRPRT: BIT #10,MTC1 ;SEE IF WRITE COMMAND  
020256 001414 BEQ 3$ ;IF SO: BR  
020260 012737 023655 020306 MOV #MSG17,2$ ;PRSET MESSAGE TO READ REVERSE  
020266 032737 000002 000672 BIT #2,MTC1 ;BRANCH IF REVERSE  
020274 001003 BNE 1$  
020276 012737 023652 020306 MOV #MSG16,2$ ;SET FORWARD MESSAGE  
020304 000004 1$: TYPE ;TYPE MSG  
020306 000000 2$: .WORD 0  
020310 000207 3$: RTS PC ;EXIT
```

:ROUTINE TO MARK UNIT OFF LINE

```
020312 013701 000674 OFFLINE:MOV UNP,R1 ;GET UNIT POINTER  
020316 052761 040000 000742 BIS #40000,UN1(R1) ;MARK UNIT OFF LINE  
020324 000004 024061 TYPE,MSG25 ;TYPE 'SLAVE UNSAFE-NO FURTHER TESTING ON SLAVE  
020330 005737 000734 TST ASEQF ;BRANCH IF NOT IN AUTO SEQUENCE  
020334 001406 BEQ 1$  
020336 000004 026246 TYPE,MSG123 ;TYPE 'AUTO-SEQ TEST WILL RESTART  
020342 012706 000500 MOV #500,SP ;RESET STACK PTR  
020346 000137 021226 JMP ASEQ0 ;RESTART AUTO-SEQ  
020352 105337 004731 1$: DECB REOTC+1 ;DECREMENT UNITS TO TEST CTR  
020356 001003 BNE 2$  
020360 000004 026210 TYPE,MSG122 ;TYPE 'NO UNITS LEFT TO TEST: HALT'  
020364 000000 HALT  
020366 000137 004154 2$: JMP REOT
```

4298  
4299  
4300  
4301  
4302  
4303  
4304  
4305  
4306  
4307  
4308  
4309  
4310  
4311  
4312  
4313  
4314  
4315  
4316  
4317  
4318  
4319  
4320  
4321  
4322  
4323  
4324  
4325  
4326

```
*****  
:TAPE COMMAND EXECUTE SUBROUTINE:  
:  
:THIS SUBROUTINE IS USED TO EXECUTE THE  
:MAG TAPE COMMAND DESCRIBED BY THE READ  
:OR WRITE ROUTINE. THE FINAL COMMAND IS  
:SENT TO THE DEVICE REGISTER ALONG WITH THE  
:INTERRUPT ENABLE AND GO BITS.  
:ONCE THE COMMAND IS ISSUED, AN INTERRUPT  
:TIMER IS STARTED AND IF NO INTERRUPT IS RETURNED  
:BEFORE TIME OUT OCCURS, AN ERROR WILL BE  
:PRINTED AND THE PROGRAM STOPPED. TESTING MAY  
:BE RESUMED BY PRESSING THE CONTINUE SWITCH.  
:TWO INTERRUPT HANDLERS ARE USED, ONE FOR MAG TAPE  
:AND ANOTHER FOR TELETYPE (TTY).  
:UPON RECEIPT OF A MAG TAPE INTERRUPT, HOUSEKEEPING  
:IS PERFORMED AND CONTROL RETURNED TO THE CALLING  
:ROUTINE (READ,WRITE,ETC).  
:RECEIPT OF A TTY INTERRUPT WILL CAUSE THE  
:PROGRAM TO CHECK FOR ENTRY OF A CNTRL C CHARACTER.  
:IF NOT CNTRL C, THEN CONTINUATION OF WAIT FOR MAG  
:TAPE INTERRUPT IS RETURNED. IF, HOWEVER, THE TTY  
:INTERRUPT WAS CAUSED BY ENTRY OF A CNTRL C,  
:THEN AT THIS TIME REQUESTS FOR NEW STALL VALUES  
:ARE PRINTED AND THE RESPONSES ENTERED. RESUMPTION  
:OF TAPE INTERRUPT WAIT IS THEN RESUMED.  
*****
```

```
4327 020372 005037 000644 TAPG: CLR TEMP1  
4328 020376 013777 000550 160114 MOV DVN,@CS ;SET DRIVE NO.  
4329 020404 032777 040000 160112 1$: BIT #40000,@ER ;SEE IF UNIT SAFE  
4330 020412 001402 BEQ TAPG3 ;IF SO: BR  
4331 020414 000137 020312 JMP OFFLINE ;GO MARK UNIT OFF-LINE  
4332 020420 032777 020000 160074 TAPG3: BIT #20000,@DS ;SEE IF PIP RESET  
4333 020426 001410 BEQ TAPG3F ;IF SO: BR  
4334 020430 004737 022012 JSR PC,PAPRT ;PRINT HEADER  
4335 020434 000004 026044 TYPE,MSG116 ;TYPE MSG  
4336 020440 032777 020000 160054 1$: BIT #20000,@DS  
4337 020446 001374 BNE 1$ ;AWAIT PIP RESET  
4338 020450 022737 000026 000672 TAPG3F: CMP #26,MTC1 ;SEE IF WRITE TM  
4339 020456 001003 BNE TAPG3A ;IF NOT: BR  
4340 020460 012704 177777 MOV #-1,R4 ;ELSE SET FC FOR -1  
4341 020464 000406 BR TAPG3B  
4342 020466 013704 000556 TAPG3A: MOV FMCNT,R4  
4343 020472 032704 000001 BIT #1,R4  
4344 020476 001401 BEQ TAPG3B  
4345 020500 005304 DEC R4  
4346 020502 000261 TAPG3B: SEC  
4347 020504 006004 ROR R4 ;SET WC = FC/2 FOR NORMAL FORMAT  
4348 020506 032737 000020 000552 BIT #20,UDES ;SEE IF CORE DUMP FORMAT  
4349 020514 001402 BEQ TAPG3C ;IF NOT: BR  
4350 020516 000261 SEC  
4351 020520 006004 ROR R4 ;SET WC = FC/4 FOR CORE DUMP  
4352 020522 010477 157764 TAPG3C: MOV R4,@WC ;SET WORD COUNT  
4353 020526 012777 000011 157754 MOV #11,@C1 ;DRIVE CLEAR
```



```
4389
4390
4391 020734 017746 157654
4392 020740 042716 000200
4393 020744 122716 000003
4394 020750 001005
4395 020752 000005
4396 020754 005077 157626
4397 020760 000137 000200
4398 020764 122716 000001
4399 020770 001015
4400 020772 022737 000176 000610
4401 021000 001014
4402 021002 012737 177570 000610
4403 021010 004737 022432
4404 021014 000004 026162
4405 021020 004737 022454
4406 021024 022716 000007
4407 021030 001005
4408 021032 012737 000176 000610
4409 021040 004737 022346
4410 021044 022716 000002
4411 021050 001041
4412 021052 004737 022432
4413 021056 005237 013444
4414 021062 004737 013226
4415 021066 032777 000040 157514
4416 021074 001425
4417 021076 000004 024652
4418 021102 013703 000602
4419 021106 104400
4420 021110 012705 000602
4421 021114 012701 000007
4422 021120 012702 177777
4423 021124 012703 002000
4424 021130 004737 022476
4425 021134 004737 022454
4426 021140 005726
4427 021142 012716 010770
4428 021146 000002
4429 021150 004737 022454
4430 021154 005726
4431 021156 000002
4432
4433
4434 021160 000240
4435 021162 042777 000037 157344
4436 021170 013716 000662
4437 021174 000002

;TTY INTERRUPT HANDLER
TTINT: MOV @TKB,-(SP) ;GET CHARACTER
        BIC #200,(SP) ;STRIP PARITY BIT
        CMPB #3,(SP) ;BRANCH IF NOT ^C
        BNE 1$
        RESET ;RESET ALL I/O
        CLR @PSW ;CLEAR PSW
        JMP @#200 ;RESTART PROGRAM
1$: CMPB #1,(SP) ;BRANCH IF NOT ^A
    BNE 2$
    CMP #SWREG,SWR ;BRANCH IF HARDWARE SWR IS INVOKED
    BNE 3$
    MOV #177570,SWR ;INVOKE HARWARE SWR
    JSR PC,,SAVE ;SAVE REGISTERS ON THE STACK
    TYPE,MSG121 ;TYPE 'HARDWARE SWR IN USE'
    JSR PC,,RESTORE ;RESTORE REGISTERS
2$: CMP #7,(SP) ;BRANCH IF NOT ^G
    BNE 4$
    MOV #SWREG,SWR ;INVOKE SOFTWARE SWR
    JSR PC,GTSWR ;GET SWITCHES
4$: CMP #2,(SP) ;BRANCH IF NOT ^B
    BNE 6$
    JSR PC,,SAVE ;SAVE REGISTERS ON THE STACK
    INC SCVFL ;SET FLAG
    JSR PC,TINP3A ;GO CHECK CRC CORRECTION
    BIT #40,@SWR ;BRANCH IF NOT YOZZLING
    BEQ 5$
    TYPE,MSG44 ;REQUEST NEW YOZZLE STALL
    MOV YSTAL,R3
    TYPOCT ;PRINT PRESENT STALL
    MOV #YSTAL,R5 ;SET ADDRESS OF YSTL
    MOV #7,R1 ;SET NUMBER OF CHAR TO INPUT
    MOV #-1,R2 ;SET MAXIMUM LIMIT
    MOV #2000,R3 ;SET MINIMUM LIMIT
    JSR PC,TTR ;GO GET VALUE
    JSR PC,,RESTORE ;RESTORE REGISTERS
    TST (SP)+ ;POP CHARACTER OF THE STACK
    MOV #YOZ,(SP) ;RETURN TO 'YOZ'
    RTI ;RETURN TO YOZ
5$: JSR PC,,RESTORE ;POP CHARACTER OFF THE STACK
6$: TST (SP)+
    RTI ;RETURN

;MAG TAPE INTERRUPT HANDLER
MTINT: NOP
MTINTA: BIC #37,@MR ;CLEAR MAINT MODE
        MOV RTRN,(SP) ;SET RETURN TO (RTRN)
        RTI ;RETURN
```



```
4439                                     :*****  
4440                                     :AUTO SEQUENCE  
4441                                     :  
4442                                     :THIS ROUTINE ,ENTERED VIA STARTING ADDRESS 240  
4443                                     :WILL EXERCISE ALL AVAILABLE SLAVES ON ALL AVAILABLE  
4444                                     :DRIVES IN BOTH PE AND NRZ ACCORDING TO THE PRESELECTED  
4445                                     :TEST PLAN. IF NRZ ONLY, PE TESTING WILL NOT BE ATTEMPTED.  
4446                                     :*****  
4447  
4448 021176 000004 025477 ASEQ: TYPE,MSG104 ;REQUEST 'AUTO CONT'  
4449 021202 012705 000740 MOV #ASEQCF,R5 ;SET ADDRESS OF ENTRY  
4450 021206 012701 000002 MOV #2,R1 ;SET SIZE OF ENTRY  
4451 021212 012702 000001 MOV #1,R2 ;SET UPPER LIMIT  
4452 021216 012703 000000 MOV #0,R3 ;SET LOWER LIMIT  
4453 021222 004737 022476 JSR PC,TTR ;GO GET INPUT  
4454 021226 005037 000550 ASEQ0: CLR DVN ;SET DRIVE # 0  
4455 021232 004737 021340 ASEQ1: JSR PC,HRDS ;GO SELECT HARDWARE CONFIGURATION  
4456 021236 000004 025447 TYPE,MSG101 ;TYPE '*****...***'  
4457 021242 000004 024756 TYPE,MSG52A ;TYPE 'DRIVE (TM03) = '  
4458 021246 013703 000550 MOV DVN,R3  
4459 021252 104400 TYPOCT ;PRINT DRIVE #  
4460 021254 000004 026335 TYPE,SPACE  
4461 021260 000004 024400 TYPE,MSG32 ;TYPE ' SLAVE # = '  
4462 021264 012700 000742 1$: MOV #UN1,R0 ;POINT TO START OF SLAVE TABLE  
4463 021270 012003 MOV (R0)+,R3  
4464 021272 100402 BMI 2$  
4465 021274 104400 TYPOCT ;PRINT SLAVE TABLE  
4466 021276 000774 BR 1$ ;DO ALL  
4467 021300 004737 021524 2$: JSR PC,AMOD1 ;GO DO MODE 1(NRZ)  
4468 021304 004737 021656 JSR PC,AMOD2 ;GO DO MODE 2(PE)  
4469 021310 022737 000007 000550 ASEQ4: CMP #7,DVN ;SEE IF DONE ALL DRIVES  
4470 021316 001403 BEQ ASEQX ;IF SO: BR  
4471 021320 005237 000550 INC DVN ;BUMP DRIVE NUMBER  
4472 021324 000742 BR ASEQ1 ;CONTINUE  
4473 021326 005737 000740 ASEQX: TST ASEQCF ;SEE IF CONTINUOUS AUTO SEQ  
4474 021332 001335 BNE ASEQ0 ;++B CONTINUE TESTING  
4475 021334 000137 004662 JMP TEND
```

```
4477
4478
4479
4480 021340 005037 004730 HRDS: CLR REOTC ;CLEAR EOT UNIT CNTR
4481 021344 012777 000040 157146 MOV #40,@CS ;INIT
4482 021352 013777 000550 157140 MOV DVN,@CS ;SET DRIVE
4483 021360 005777 157124 TST @C1 ;ACCESS DRIVE
4484 021364 032777 010000 157126 BIT #10000,@CS ;TEST FOR NON-EXISTANT DRIVE
4485 021372 001403 BEQ 2$ ;IF DRIVE AVAIL: BR
4486 021374 005726 1$: TST (SP)+ ;RESET STACK POINTER
4487 021376 000137 021310 JMP ASEQ4 ;GO SEE IF TRIED ALL DRIVES
4488 021402 017700 157130 2$: MOV @DT,R0 ;++B GET CONTENTS OF DRIVE TYPE REG
4489 021406 042700 002007 BIC #2007,R0 ;++B CLEAR SPR AND SPEED BITS
4490 021412 022700 140050 CMP #140050,R0 ;++B BRANCH IF NOT TMO3 MAGTAPE DRIVE
4491 021416 001366 BNE 1$
4492 021420 005000 CLR R0
4493 021422 012701 000742 MOV #UN1,R1 ;SET START OF SLAVE TABLE
4494 021426 005737 003034 TST CHNFLG ;BRANCH IF NOT IN CHAIN MODE
4495 021432 001410 BEQ 3$
4496 021434 122737 000006 000041 CMPB #6,@#41 ;BRANCH IF NOT LOADED VIA TMDP
4497 021442 001004 BNE 3$
4498 021444 005737 000550 TST DVN ;BRANCH IF NOT DRIVE 0
4499 021450 001001 BNE 3$
4500 021452 005200 INC R0 ;DO NOT TEST SLAVE 0
4501 021454 010077 157062 3$: MOV R0,@TC ;SELECT SLAVE
4502 021460 032777 010000 157034 BIT #10000,@DS ;SEE IF SLAVE AVAIL FOR TEST(MOL)
4503 021466 001404 BEQ 4$ ;IF NOT: BR
4504 021470 062737 000401 004730 ADD #401,REOTC ;INCREMENT UNITS TO TEST COUNT
4505 021476 010021 MOV R0,(R1)+ ;LOAD SLAVE # INTO SLAVE TABLE
4506 021500 005200 4$: INC R0 ;STEP TO NEXT SLAVE
4507 021502 022700 000010 CMP #10,R0 ;BRANCH IF ALL SLAVE NOT DONE
4508 021506 001362 BNE 3$
4509 021510 005737 004730 5$: TST REOTC ;SEE IF FOUND ANY SLAVES
4510 021514 001727 BEQ 1$ ;IF NOT: BR
4511 021516 012711 177777 MOV #-1,(R1) ;TERMINATE SLAVE TABLE
4512 021522 000207 RTS PC ;RETURN TO SEQ
```

```
4514  
4515  
4516  
4517 021524 005037 000654  
4518 021530 012701 000742  
4519 021534 052721 001700  
4520 021540 022711 177777  
4521 021544 001373  
4522 021546 004737 004744  
4523 021552 012737 000006 000736  
4524 021560 012737 174000 000556  
4525 021566 012737 000100 000554  
4526 021574 012737 000001 000560  
4527 021602 005037 000564  
4528 021606 005037 000570  
4529 021612 004737 003352  
4530 021616 012737 000010 000560  
4531 021624 004737 003352  
4532 021630 012737 000014 000560  
4533 021636 004737 003352  
4534 021642 012737 177777 000560 3$:  
4535 021650 004737 003352  
4536 021654 000207
```

;SUBROUTINE TO SELECT NRZ AUTO TEST MODE\*\*\*\*\*

```
AMOD1: CLR BLCNTR ;ASSURE BLOCK COUNTER IS 0  
MOV #UN1,R1 ;GET START OF SLAVE TABLE  
1$: BIS #1700,(R1)+ ;SET ALL SLAVE TO NRZ,NORM,ODD  
CMP #-1,(R1) ;LOOP UNTIL REACED END OF TABLE  
BNE 1$  
JSR PC,RWDA ;GO REWIND ALL AVAIL SLAVES  
MOV #6,ABLCNT ;SET NUMBER OF BLOCKS FOR MODE 1  
MOV #-4000,FMCNT ;SET FC = 4000  
MOV #100,RCNT ;SET REC CNTR = 100  
MOV #1,PATRN ;SELECT PATTERN 1  
CLR TMEX ;ASSURE NO TMK  
CLR INTRF ;ASSURE NORMAL READ  
JSR PC,STAUTO ;GO DO AUTO MODE 1  
MOV #10,PATRN ;SELECT PATTERN 10  
JSR PC,STAUTO ;GO DO PATTERN 10  
MOV #14,PATRN ;SELECT PATTERN 14  
JSR PC,STAUTO  
3$: MOV #-1,PATRN ;SELECT AUTO RANDOM DATA  
JSR PC,STAUTO  
RTS PC ;RETURN TO SEQ
```

```
4538  
4539  
4540  
4541 021656 005037 000654  
4542 021662 012701 000742  
4543 021666 042711 001700  
4544 021672 052721 002300  
4545 021676 022711 177777  
4546 021702 001371  
4547 021704 004737 004744  
4548 021710 012737 000006 000736  
4549 021716 012737 174000 000556  
4550 021724 012737 000100 000554  
4551 021732 012737 000010 000560  
4552 021740 004737 003352  
4553 021744 012737 000014 000560  
4554 021752 004737 003352  
4555 021756 012737 000015 000560  
4556 021764 004737 003352  
4557 021770 012737 177777 000736  
4558 021776 012737 177777 000560  
4559 022004 004737 003352  
4560 022010 000207  
4561  
4562
```

```
          :SUBROUTINE TO SELECT PE AUTO TEST MODE*****  
AMOD2: CLR      BLCNTR      ;CLEAR BLOCK CNTR  
        MOV      #UN1,R1    ;SET START OF SLAVE TABLE  
1$:     BIC      #1700,(R1)  ;CLEAR NRZ  
        BIS      #2300,(R1)+ ;SET TO PE NORM, ODD  
        CMP      #-1,(R1)   ;LOOP UNTIL END OF TABLE  
        BNE     1$  
        JSR     PC,RWDA     ;REWIND ALL SLAVES  
        MOV      #6,ABLCNT  ;SET AUTO BLOCK COUNT  
        MOV      #-4000,FMCNT ;SET FC = 4000  
        MOV      #100,RCNT  ;SET REC CNTR TO 100  
        MOV      #10,PATRN  ;SELECT PATTERN 10  
        JSR     PC,STAUTO   ;GO DO AUTO SEQ  
        MOV      #14,PATRN  ;SELECT PATTERN 14  
        JSR     PC,STAUTO  
        MOV      #15,PATRN  ;SELECT PATTERN 15  
        JSR     PC,STAUTO  
        MOV      #-1,ABLCNT ;FORCE TO END OF TAPE  
        MOV      #-1,PATRN  ;SELECT AUTO RANDOM DATA  
        JSR     PC,STAUTO  
3$:     RTS      PC        ;RETURN TO SEQ
```

```
4564  
4565  
4566  
4567  
4568  
4569  
4570  
4571  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580 022012 000004 024754  
4581 022016 013703 000550  
4582 022022 104400  
4583 022024 000004 024400  
4584 022030 013703 000552  
4585 022034 042703 177770  
4586 022040 104400  
4587 022042 000004 023522  
4588 022046 013703 000552  
4589 022052 000303  
4590 022054 042703 177770  
4591 022060 104400  
4592 022062 000004 025071  
4593 022066 005003  
4594 022070 032737 000010 000552  
4595 022076 001401  
4596 022100 005203  
4597 022102 104400  
4598 022104 000004 025075  
4599 022110 013703 000552  
4600 022114 006003  
4601 022116 006003  
4602 022120 006003  
4603 022122 006003  
4604 022124 042703 177760  
4605 022130 104400  
4606 022132 000004 023565  
4607 022136 005737 000560  
4608 022142 100003  
4609 022144 000004 023655  
4610 022150 000403  
4611 022152 013703 000560  
4612 022156 104400  
4613 022160 000004 023607  
4614 022164 013703 000654  
4615 022170 104400  
4616 022172 000004 023615  
4617 022176 010003  
4618 022200 032737 000010 000672  
4619 022206 001416
```

```
*****  
:ERROR HEADER PRINT SUBROUTINE:  
:THIS ROUTINE IS USED TO PRINT OUT A HEADER  
:WITH EACH ERROR MESSAGE. THE PRINT IS IN TWO  
:LINES AND CONTAINS THE FOLLOWING INFORMATION.  
:LINE 1: DRIVE NO. SLAVE NO. DENSITY PARITY FORMAT  
:LINE 2: CURRENT BLOCK NUMBER, RECORD NUMBER IN  
:WHICH THE ERROR OCCURED PLUS THE TOTAL NUMBER  
:OF RECORDS IN THIS BLOCK, THE RECORD SIZE (NUMBER  
:OF CHARACTERS), AND THE ERROR TYPE (READ,WRITE, SPACE, ETC)  
:PLUS THE TAPE DIRECTION (FORWARD OR REVERSE).  
:ALL NUMBERS ARE IN OCTAL.  
*****  
PAPRT: TYPE,MSG52 ;TYPE 'DRIVE # = '  
MOV DVN,R3  
TYPOCT ;PRINT DRIVE NUMBER  
TYPE,MSG32 ;TYPE 'SLAVE # = '  
MOV UDES,R3  
BIC #77770,R3  
TYPOCT ;PRINT SLAVE NUMBER  
TYPE,MSG1 ;TYPE DENSITY TAG '*DE'  
MOV UDES,R3  
SWAB R3  
BIC #177770,R3  
TYPOCT ;PRINT DENSITY  
TYPE,MSG61 ;TYPE PARITY TAG '*P'  
CLR R3  
BIT #10,UDES  
BEQ PAPRT0  
INC R3 ;SET PARITY INDICATOR = EVEN  
PAPRT0: TYPOCT ;PRINT PARITY BIT STATE  
TYPE,MSG62 ;TYPE FORMAT TAG '*F'  
MOV UDES,R3  
ROR R3  
ROR R3  
ROR R3 ;POSTION FORMAT BITS  
ROR R3  
BIC #177760,R3  
TYPOCT ;PRINT FORMAT  
TYPE,MSG8 ;TYPE PATTERN # TAG '*PATRN'  
TST PATRN ;BRANCH IF NOT RANDOM PATTERN  
BPL PAPRTC  
PAPRTA: TYPE,MSG17 ;TYPE 'R' FOR RANDOM  
BR PAPRTD  
PAPRTC: MOV PATRN,R3  
TYPOCT ;PRINT PATRN NUMBER  
PAPRTD: TYPE,MSG13 ;TYPE BLOCK # TAG '*BN'  
MOV BLCNTR,R3  
TYPOCT ;PRINT NUMBER  
TYPE,MSG14 ;TYPE RECORD # TAG '*RN'  
MOV R0,R3 ;GET # OF RECORDS LEFT TO PROCESS  
BIT #10,MTC1 ;SEE IF WRITE OPERATION  
BEQ PAPRT1 ;IF SO: BR
```



```
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652 022314 063737 000630 000626 RANG: ADD RANSAV,RANBAS
4653 022322 063737 000626 000630 ADD RANBAS,RANSAV ;GET NEW NUMBER
4654 022330 023701 000630 CMP RANSAV,R1 ;SEE IF NUMBER TOO BIG
4655 022334 101367 BHI RANG ;IF SO: BR
4656 022336 020237 000630 CMP R2,RANSAV ;SEE IF NUMBER TOO SMALL
4657 022342 101364 BHI RANG ;IF SO: BR
4658 022344 000207 RT^ PC ;EXIT
4659
4660 ;SUBROUTINE TO GET NEW SOFTWARE SWR
4661
4662 022346 022737 000176 000610 GTSWR: CMP #SWREG,SWR ;BRANCH IF SOFTWARE SWR
4663 022354 001025 BNE 1$ ;NOT INVOKED
4664 022356 004737 022432 JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
4665 022362 000004 023500 TYPE,$MSWR
4666 022366 017703 156216 MOV @SWR,R3 ;GET CURRENT SWR
4667 022372 104400 TYPOCT
4668 022374 000004 023510 TYPE,$MNEW ;REQUEST NEW SWR SETTING
4669 022400 013705 000610 MOV SWR,R5 ;TTR ROUTINE RETURNS VALUE TO (R5)
4670 022404 012701 000007 MOV #7,R1 ;LIMIT RESPONSE TO 7 CHARS
4671 022410 012702 177777 MOV #177777,R2 ;BETWEEN 0 AND 177777
4672 022414 012703 000000 MOV #0,R3
4673 022420 004737 022476 JSR PC,TTR ;GET RESPONSE
4674 022424 004737 022454 JSR PC,.RESTORE ;RESTORE REGISTERS
4675 022430 000207 1$: RTS PC ;RETURN
4676
4677 ;:ROUTINE TO SAVE REGISTERS ON THE STACK
;SAVE: MOV %5,-(SP) ;:R5 IS SAVED AT 12(SP)
MOV %4,-(SP) ;:R4 IS SAVED AT 10(SP)
MOV %3,-(SP) ;:R3 IS SAVED AT 6(SP)
MOV %2,-(SP) ;:R2 IS SAVED AT 4(SP)
MOV %1,-(SP) ;:R1 IS SAVED AT 2(SP)
MOV %0,-(SP) ;:R0 IS SAVED AT (SP)
MOV 14(SP),-(SP) ;:PUSH RETURN PC ON THE STACK
RTS PC ;:RETURN TO CALLER
(1) 022432 010546
(1) 022434 010446
(1) 022436 010346
(1) 022440 010246
(1) 022442 010146
(1) 022444 010046
(1) 022446 016646 000014
(1) 022452 000207
(1)
4678 ;:ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
;.RESTORE:MOV (SP)+,14(SP) ;:STORE RETURN PC ON STACK
MOV (SP)+,%0
MOV (SP)+,%1
MOV (SP)+,%2
MOV (SP)+,%3
MOV (SP)+,%4
MOV (SP)+,%5
(1) 022454 012666 000014
(1) 022460 012600
(1) 022462 012601
(1) 022464 012602
(1) 022466 012603
(1) 022470 012604
(1) 022472 012605
(1) 022474 000207
(1) RTS PC ;:RETURN
```

4680  
4681  
4682  
4683  
4684  
4685  
4686  
4687  
4688  
4689  
4690  
4691  
4692  
4693  
4694  
4695  
4696  
4697  
4698  
4699  
4700  
4701  
4702  
4703  
4704  
4705  
4706  
4707  
4708  
4709  
4710  
4711  
4712  
4713  
4714  
4715  
4716  
4717  
4718  
4719  
4720  
4721  
4722  
4723  
4724  
4725  
4726  
4727  
4728  
4729  
4730  
4731  
4732  
4733  
4734  
4735

022476 010146  
022500 011601  
022502 005037 000644  
022506 005000  
022510 004737 022722  
022514 122737 000003 000642  
022522 001003  
022524 000005  
022526 000137 000200  
022532 122737 000015 000642 11\$:  
022540 001004  
022542 005737 000644  
022546 001455  
022550 000447  
022552 122737 000025 000642 2\$:  
022560 001003  
022562 000004 024155  
022566 000744  
022570 122737 000177 000642 21\$:  
022576 001010  
022600 000241  
022602 006000  
022604 006200  
022606 006200  
022610 000004 026112  
022614 005201  
022616 000734  
022620 122737 000060 000642 3\$:  
022626 101027  
022630 122737 000070 000642 4\$:  
022636 101423  
022640 005237 000644 5\$:  
022644 006300  
022646 006300  
022650 006300  
022652 042737 177770 000642  
022660 053700 000642  
022664 005301  
022666 001310

```
*****  
: TTY ENTRY SUBROUTINE:  
: THIS SUBROUTINE IS USED BY THE TEST CONDITION  
: ENTRY ROUTINE TO READ THE RESPONSE ENTERED  
: AT THE TTY AND CHECK THEM FOR LEGALITY AND  
: LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL  
: (0-7) AND MUST FALL WITHIN THE LIMITS SET BY  
: THE CALLING ROUTINE.  
: IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,  
: A QUESTION MARK IS TYPED (?) AND THE RESPONSE  
: MAY BE REENTERED.  
: ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND  
: MAY BE TERMINATED AT LESS THAN SIX BY TYPING A  
: CARRIAGE RETURN  
*****  
TTR: MOV R1, -(SP) ;SAVE CHAR COUNT  
10$: MOV (SP), R1 ;RESTORE CHAR COUNT (FOR ^U)  
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
CLR R0  
1$: JSR PC, TTIN ;GO READ CHARACTER  
CMPB #3, TIB ;BRANCH IF NOT ^C  
BNE 11$  
RESET  
JMP @#200 ;RESTART AT 200  
11$: CMPB #15, TIB ;SEE IF CR  
BNE 2$ ;IF NOT: BR  
TST TEMP1 ;SEE IF FIRST CHARACTER  
BEQ 9$ ;IF SO: BR  
BR 6$ ;ELSE GO LOAD VALUE  
2$: CMPB #25, TIB ;BRANCH IF NOT CONTROL U  
BNE 21$  
TYPE, MSG28 ;TYPE <CR><LF>  
BR 10$  
21$: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'  
BNE 3$  
CLC ;REMOVE LAST CHARACTER  
ROR R0  
ASR R0  
ASR R0  
TYPE, MSG118 ;TYPE '\'  
INC R1 ;DEC CHAR RECEIVED COUNT  
BR 1$ ;GET NEXT CHARACTER  
3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0  
BHI TIBR  
4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7  
BLOS TIBR  
5$: INC TEMP1 ;SET FIRST CHARACTER FLAG  
ASL R0  
ASL R0 ;SHIFT 3 LEFT  
ASL R0  
BIC #177770, TIB ;STRIP ASCII  
BIS TIB, R0 ;LOAD CHARACTER  
DEC R1 ;SEE IF DONE  
BNE 1$ ;IF NOT: BR
```



4736	022670	020002		6S:	CMP	R0,R2		;SEE IF EXCEEDED MAXIMUM LIMIT
4737	022672	101005			BHI	TINER		
4738	022674	020300		7S:	CMP	R3,R0		;SEE IF BELOW MINIMUM LIMIT
4739	022676	101003			BHI	TINER		
4740	022700	010015		8S:	MOV	R0,(R5)		;LOAD VALUE
4741	022702	005726		9S:	TST	(SP)+		;POP CHAR COUNT OFF STACK
4742	022704	000207			RTS	PC		;EXIT
4743								
4744	022706	000004	024646	TINER:	TYPE,#MSG43			;TYPE '?'
4745	022712	005726			TST	(SP)+		;POP CHAR COUNT OFF STACK
4746	022714	162716	000020		SUB	#20,(SP)		;RESET SP TO START OF VALUE ROUTINE
4747	022720	000207			RTS	PC		;REDO VALUE ENTRY

```
4749
4750 ;TTY READ SUBROUTINE*****
4751
4752 022722 005277 155664 TFIN: INC @TKS
4753 022726 105777 155660 1$: TSTB @TKS
4754 022732 100375 BPL 1$
4755 022734 017737 155654 000642 MOV @TKB,TIB
4756 022742 042737 177600 000642 BIC #177600,TIB ;STRIP PARITY BIT
4757 022750 022737 000015 000642 CMP #15,TIB ;BRANCH IF NOT <CR>
4758 022756 001003 BNE 2$
4759 022760 000004 024155 TYPE,MSG28 ;TYPE '<CR><LF>'
4760 022764 000402 BR 3$
4761 022766 000004 000642 2$: TYPE,TIB ;ECHO CHARACTER
4762 022772 000207 3$: RTS PC
4763
4764 ;TTY OUTPUT SUBROUTINE*****
4765
4766 022774 010446 TTOUT: MOV R4,-(SP) ;SAVE R4 ON THE STACK
4767 022776 010346 MOV R3,-(SP)
4768 023000 017604 000004 MOV @4(SP),R4 ;GET ADDRESS OF MESSAGE TO TYPE
4769 023004 062766 000002 000004 ADD #2,4(SP) ;ADJUST RETURN PC
4770 023012 111437 000640 10$: MOV (R4),TOB ;GET A CHARACTER
4771 023016 001431 BEQ 3$ ;AND BRANCH IF END OF MSG
4772 023020 122724 000045 CMPB #45,(R4)+ ;BRANCH IF CRLF CHARACTER (%)
4773 023024 001403 BEQ 1$
4774 023026 004737 023110 JSR PC,TOG ;ECHO CHARACTER
4775 023032 000767 BR 10$
4776
4777 023034 112737 000015 000640 1$: MOVB #15,TOB
4778 023042 004737 023110 JSR PC,TOG
4779 023046 012703 000006 MOV #6,R3
4780 023052 005037 000640 2$: CLR TOB
4781 023056 004737 023110 JSR PC,TOG
4782 023062 005303 DEC R3
4783 023064 001372 BNE 2$ ;DO FILLERS
4784 023066 112737 000012 000640 MOVB #12,TOB
4785 023074 004737 023110 JSR PC,TOG
4786 023100 000744 BR 10$
4787 023102 012603 3$: MOV (SP)+,R3 ;RESTORE REGISTERS
4788 023104 012604 MOV (SP)+,R4
4789 023106 000002 RTI ;RETURN
4790
4791 023110 105777 155502 TOG: TSTB @TPS
4792 023114 100375 BPL TOG
4793 023116 113777 000640 155474 MOVB TOB,@TPB
4794 023124 000207 RTS PC ;RETURN
4795
```

```

4797                                     :OCIAL OUTPUT SUBROUTINE*****
4798
4799 023126 005037 023302   OCTP:  CLR      OFL          ;CLEAR FLAG FOR LEADING ZERO
4800 023132 010304          MOV      R3,R4        ;SEE IF NUMBER IS ZERO
4801 023134 001003          BNE     1$           ;IF NOT ZERO: BR
4802 023136 000004 026337   TYPE,DIGIT0
4803 023142 000434          BR      4$           ;SPACE AND EXIT
4804 023144 100004          1$:  BPL      3$           ;BRANCH IF MSD IS A '0'
4805 023146 012704 000001   MOV      #1,R4
4806 023152 004737 023242   JSR     PC,OCTPG    ;PRINT 1
4807 023156 006004          3$:  ROR      R4
4808 023160 006004          ROR     R4
4809 023162 006004          ROR     R4          ;POSITION DIGIT
4810 023164 006004          ROR     R4
4811 023166 000304          SWAB   R4
4812 023170 004737 023242   JSR     PC,OCTPG    ;PRINT DIGIT 2
4813 023174 006004          ROR     R4
4814 023176 000304          SWAB   R4
4815 023200 004737 023242   JSR     PC,OCTPG    ;PRINT DIGIT 3
4816 023204 006104          ROL     R4
4817 023206 006104          ROL     R4
4818 023210 000304          SWAB   R4
4819 023212 004737 023242   JSR     PC,OCTPG    ;PRINT DIGIT 4
4820 023216 006004          ROR     R4
4821 023220 006004          ROR     R4
4822 023222 006004          ROR     R4
4823 023224 004737 023242   JSR     PC,OCTPG    ;PRINT DIGIT 5
4824 023230 004737 023242   JSR     PC,OCTPG    ;PRINT DIGIT 6
4825 023234 000004 026335   4$:  TYPE,SPACE ;TYPE A SPACE
4826 023240 000002          RTI     ;EXIT
4827
4828 023242 042704 177770   OCTPG: BIC     #177770,R4
4829 023246 001003          BNE     1$
4830 023250 005737 023302   TST     OFL
4831 023254 001410          BEQ     2$
4832 023256 005237 023302   1$:  INC     OFL
4833 023262 052704 000260   BIS     #260,R4
4834 023266 010437 000640   MOV     R4,TOB
4835 023272 004737 023110   JSR     PC,TOG
4836 023276 010304          2$:  MOV     R3,R4
4837 023300 000207          RTS     PC
4838 023302 000000          OFL:  0           ;FIRST CHAR FLAG
4839
4840                                     ;DATA CHARACTER OUTPUT SUBROUTINE*****
4841
4842
4843 023304 012704 000010   DOUT: MOV     #10,R4    ;SET NUMBER TO PRINT
4844 023310 110346          MOVB   R3,-(SP)     ;GET CHAR TO OUTPUT
4845 023312 106316          1$:  ASLB   (SP)       ;BRANCH IF BIT IS A ZERO
4846 023314 103003          BCC    2$
4847 023316 000004 026341   TYPE,DIGIT1
4848 023322 000402          BR     3$
4849 023324 000004 026337   2$:  TYPE,DIGIT0
4850 023330 005304          3$:  DEC     R4
4851 023332 001367          BNE    1$
4852 023334 005726          TST   (SP)+        ;POP STACK

```

```
4853 023336 000207          RTS      PC
4854
4855 023340 113703 000651    DOUTD:  MOVB   TEMP3+1,R3
4856 023344 004737 023304      JSR     PC,DOUT
4857 023350 013703 000650      MOV     TEMP3,R3
4858 023354 004737 023304      JSR     PC,DOUT
4859 023360 000207          RTS      PC
4860
4861                          ;TU16 SERIAL NUMBER PRINT SUBROUTINE*****
4862
4863 023362 017703 155152    SNPT:   MOV     @SN,R3          ;GET CONTENTS OF SERIAL # REG
4864 023366 000004 023575      TYPE,MSG9          ;TYPE SN TAG
4865 023372 010304          MOV     R3,R4
4866 023374 000304          SWAB    R4
4867 023376 006004          ROR     R4
4868 023400 006004          ROR     R4
4869 023402 006004          ROR     R4
4870 023404 006004          ROR     R4
4871 023406 004737 023454    JSR     PC,SNPG      ;PRINT FIRST DIGIT
4872 023412 010304          MOV     R3,R4
4873 023414 000304          SWAB    R4
4874 023416 004737 023454    JSR     PC,SNPG      ;PRINT SECOND DIGIT
4875 023422 010304          MOV     R3,R4
4876 023424 006004          ROR     R4
4877 023426 006004          ROR     R4
4878 023430 006004          ROR     R4
4879 023432 006004          ROR     R4
4880 023434 004737 023454    JSR     PC,SNPG      ;PRINT THIRD DIGIT
4881 023440 010304          MOV     R3,R4
4882 023442 004737 023454    JSR     PC,SNPG      ;PRINT FOURTH DIGIT
4883 023446 000004 024155      TYPE,MSG28         ;TYPE <CR><LF>
4884 023452 000207          RTS      PC          ;EXIT
4885 023454 012737 000260 000640 SNPG:  MOV     #260,TOB      ;SET NUMBER BASE
4886 023462 042704 177760      BIC     #177760,R4   ;MASK NUMBER
4887 023466 050437 000640      BIS     R4,TOB       ;BUILD DIGIT
4888 023472 004737 023110      JSR     PC,TOG       ;GO TYPE
4889 023476 000207          RTS      PC          ;RETURN
4890
```

4892  
4893  
4894  
4895 023500 051445 051127 036440 \$MSWR: .ASCIZ /%SWR = /  
023506 000040  
4896 023510 047040 053505 036440 \$MNEW: .ASCIZ / NEW = /  
023516 000040  
4897 023520 000055 DASH: .ASCIZ /-/  
4898 023522 042052 020105 000 MSG1: .ASCIZ /\*DE /  
4899 023527 045 035507 000040 MSG2: .ASCIZ /%G: /  
4900 023534 041045 020073 000 MSG3: .ASCIZ /%B: /  
4901 023541 045 047103 000040 MSG4: .ASCIZ /%CN /  
4902 023546 053452 020105 000 MSG5: .ASCIZ /\*WE /  
4903 023553 052 042522 000040 MSG6: .ASCIZ /\*RE /  
4904 023560 051052 020123 000 MSG7: .ASCIZ /\*RS /  
4905 023565 052 040520 051124 MSG8: .ASCIZ /\*PATRN /  
023572 020116 000  
4906 023575 123 035116 000040 MSG9: .ASCIZ /SN: /  
4907 023602 051452 020105 000 MSG10: .ASCIZ /\*SE /  
4908 023607 045 041052 020116 MSG13: .ASCIZ /%\*BN /  
023614 000  
4909 023615 052 047122 000040 MSG14: .ASCIZ /\*RN /  
4910 023622 020045 020040 020040 MSG15: .ASCIZ /% BAD RECORD%/  
023630 020040 020040 041040  
023636 042101 051040 041505  
023644 051117 022504 000045  
4911 023652 043040 000 MSG16: .ASCIZ / F/  
4912 023655 040 000122 MSG17: .ASCIZ / R/  
4913 023660 042440 052117 021440 MSG20: .ASCIZ / EOT # /  
023666 000040  
4914 023670 047111 042524 041522 MSG21: .ASCIZ /INTERCHANGE READ? /  
023676 040510 043516 020105  
023704 042522 042101 020077  
023712 000  
4915 023713 045 046111 042514 MSG22: .ASCIZ /%ILLEGAL BOT: HALT%/  
023720 040507 020114 047502  
023726 035124 044040 046101  
023734 022524 000  
4916 023737 045 051503 020061 MSG23: .ASCIZ /%CS1 /  
023744 000  
4917 023745 045 041527 000040 MSG23A: .ASCIZ /%WC /  
4918 023752 041045 020101 000 MSG23B: .ASCIZ /%BA /  
4919 023757 045 041506 000040 MSG23C: .ASCIZ /%FC /  
4920 023764 041445 031123 000040 MSG23D: .ASCIZ /%CS2 /  
4921 023772 042045 020123 000 MSG23E: .ASCIZ /%DS /  
4922 023777 045 051105 000040 MSG23F: .ASCIZ /%ER /  
4923 024004 040445 020123 000 MSG23G: .ASCIZ /%AS /  
4924 024011 045 045503 000040 MSG23H: .ASCIZ /%CK /  
4925 024016 042045 020102 000 MSG23I: .ASCIZ /%DB /  
4926 024023 045 051115 000040 MSG23J: .ASCIZ /%MR /  
4927 024030 042045 020124 000 MSG23K: .ASCIZ /%DT /  
4928 024035 045 041524 000040 MSG23L: .ASCIZ /%TC /  
4929 024042 047045 020117 047111 MSG24: .ASCIZ /%NO INTERRUPT%/  
024050 042524 051122 050125  
024056 022524 000  
4930 024061 045 046123 053101 MSG25: .ASCIZ /%SLAVE UNSAFE-TEST DISCONTINUED ON SLAVE%/

	024066	020105	047125	040523		
	024074	042506	052055	051505		
	024102	020124	044504	041523		
	024110	047117	044524	052516		
	024116	042105	047440	020116		
	024124	046123	053101	022505		
	024132	000				
4931	024133	045	051104	050117	MSG26:	.ASCIZ /%DROPS: /
	024140	035123	000040			
4932	024144	050045	041511	051513	MSG27:	.ASCIZ /%PICKS: /
	024152	020072	000			
4933	024155	045	000		MSG28:	.ASCIZ /%/
4934	024157	045	052045	047515	MSG30:	.ASCIZ '%XTM03-TE16/TU77 AUTO SEQUENCE (CZTEDDO)%';++B
	024164	026463	042524	033061		
	024172	052057	033525	020067		
	024200	052501	047524	051440		
	024206	050505	042525	041516		
	024214	020105	041450	052132		
	024222	042105	030104	022451		
	024230	000				
4935	024231	045	052045	030115	MSG31:	.ASCIZ '%XTM03-TE16/TU77 DATA RELIABILITY TEST (CZTEDDO)%';++B
	024236	026463	042524	033061		
	024244	052057	033525	020067		
	024252	040504	040524	051040		
	024260	046105	040511	044502		
	024266	044514	054524	052040		
	024274	051505	020124	041450		
	024302	052132	042105	030104		
	024310	022451	000			
4936	024313	124	050131	020105	MSG31A:	.ASCIZ /TYPE <CR> TO TERMINATE ALL REQUESTS & ^C TO RESTART%/
	024320	041474	037122	052040		
	024326	020117	042524	046522		
	024334	047111	052101	020105		
	024342	046101	020114	042522		
	024350	052521	051505	051524		
	024356	023040	057040	020103		
	024364	047524	051040	051505		
	024372	040524	052122	000045		
4937	024400	046123	053101	020105	MSG32:	.ASCIZ /SLAVE # = /
	024406	020043	020075	000		
4938	024413	104	047105	044523	MSG33:	.ASCIZ /DENSITY = /
	024420	054524	036440	000040		
4939	024426	040520	044522	054524	MSG34:	.ASCIZ /PARITY = /
	024434	036440	000040			
4940	024440	042522	047503	042122	MSG35:	.ASCIZ /RECORD COUNT = /
	024446	041440	052517	052116		
	024454	036440	000040			
4941	024460	044103	051101	041440	MSG36:	.ASCIZ /CHAR COUNT = /
	024466	052517	052116	036440		
	024474	000040				
4942	024476	040520	052124	051105	MSG37:	.ASCIZ /PATTERN # = /
	024504	020116	020043	020075		
	024512	000				
4943	024513	123	047111	046107	MSG38:	.ASCIZ /SINGLE PASS? /
	024520	020105	040520	051523		
	024526	020077	000			

4944	024531	103	041522	041440	MSG39:	.ASCIZ	/CRC CORRECTION (YES=1,NO=0)? /
	024536	051117	042522	052103			
	024544	047511	020116	054450			
	024552	051505	030475	047054			
	024560	036517	024460	020077			
	024566	000					
4945	024567	045	042445	052116	MSG40:	.ASCIZ	/%ENTER STALLS%READ = /
	024574	051105	051440	040524			
	024602	046114	022523	042522			
	024610	042101	036440	000040			
4946	024616	051127	052111	020105	MSG41:	.ASCIZ	/WRITE = /
	024624	020075	000				
4947	024627	124	051125	020116	MSG42:	.ASCIZ	/TURN AROUND = /
	024634	051101	052517	042116			
	024642	036440	000040				
4948	024646	037445	000045		MSG43:	.ASCIZ	/%?%/
4949	024652	042445	052116	051105	MSG44:	.ASCIZ	/%ENTER YOZZLE STALL = /
	024660	054440	055117	046132			
	024666	020105	052123	046101			
	024674	020114	020075	000			
4950	024701	045	051105	020122	MSG45:	.ASCIZ	/%ERR AMT /
	024706	046501	020124	000			
4951	024713	045	047516	020124	MSG49:	.ASCIZ	/%NOT AVAIL /
	024720	053101	044501	020114			
	024726	000					
4952	024727	045	046111	042514	MSG50:	.ASCIZ	/%ILLEGAL DRIVE TYPE /
	024734	040507	020114	051104			
	024742	053111	020105	054524			
	024750	042520	000040				
4953	024754	022445			MSG52:	.ASCII	/%%/
4954	024756	051104	053111	020105	MSG52A:	.ASCIZ	/DRIVE (TM03) # = /
	024764	052050	030115	024463			
	024772	021440	036440	000040			
4955	025000	047506	046522	052101	MSG53:	.ASCIZ	/FORMAT = /
	025006	036440	000040				
4956	025012	053452	020105	046524	MSG54:	.ASCIZ	/*WE TM/
	025020	000					
4957	025021	052	042523	052040	MSG55:	.ASCIZ	/*SE TM/
	025026	000115					
4958	025030	052040	000115		MSG56:	.ASCIZ	/ TM/
4959	025034	047045	047117	042455	MSG57:	.ASCIZ	/%NON-EXIST SLAVE/
	025042	044530	052123	051440			
	025050	040514	042526	000			
4960	025055	045	051103	020103	MSG58:	.ASCIZ	/%CRC /
	025062	000					
4961	025063	045	051114	020103	MSG59:	.ASCIZ	/%LRC /
	025070	000					
4962	025071	052	020120	000	MSG61:	.ASCIZ	/*P /
4963	025075	052	020106	000	MSG62:	.ASCIZ	/*F /
4964	025101	045	047452	044522	MSG64:	.ASCIZ	/%*ORIGINAL ERROR*/
	025106	044507	040516	020114			
	025114	051105	047522	025122			
	025122	000					
4965	025123	045	042522	051124	MSG65:	.ASCIZ	/%RETRY: /
	025130	035131	000040				
4966	025134	051452	020105	052122	MSG66:	.ASCIZ	/*SE RTRY /

4967	025142	054522	000040						
	025146	042452	040522	042523	MSG67:	.ASCIZ	/*ERASE/		
	025154	000							
4968	025155	045	042522	042522	MSG68:	.ASCIZ	/%REREV: /		
	025162	035126	000040						
4969	025166	040524	042520	046440	MSG69:	.ASCIZ	/TAPE MARK? /		
	025174	051101	037513	000040					
4970	025202	047045	047117	042455	MSG71:	.ASCIZ	/%NON-EXIST DRIVE/		
	025210	044530	052123	042040					
	025216	044522	042526	000					
4971	025223	045	042522	053506	MSG72:	.ASCIZ	/%REFWD: /		
	025230	035104	000040						
4972	025234	053445	042524	051122	MSG73:	.ASCIZ	/%WTERR: /		
	025242	020072	000						
4973	025245	045	042522	044507	MSG74:	.ASCIZ	/%REGISTER START = /		
	025252	052123	051105	051440					
	025260	040524	052122	036440					
	025266	000040							
4974	025270	042526	052103	051117	MSG75:	.ASCIZ	/VECTOR ADRS = /		
	025276	040440	051104	020123					
	025304	020075	000						
4975	025307	045	042504	042522	MSG76:	.ASCIZ	/%DEREV: /		
	025314	035126	000040						
4976	025320	042045	043105	042127	MSG77:	.ASCIZ	/%DEFWD: /		
	025326	020072	000						
4977	025331	045	047516	026516	MSG78:	.ASCIZ	/%NON-RETRYABLE WRITE ERROR: ER /		
	025336	042522	051124	040531					
	025344	046102	020105	051127					
	025352	052111	020105	051105					
	025360	047522	035122	042440					
	025366	020122	000						
4978	025371	045	047516	026516	MSG79:	.ASCIZ	/%NON-RETRYABLE READ ERROR: ER /		
	025376	042522	051124	040531					
	025404	046102	020105	042522					
	025412	042101	042440	051122					
	025420	051117	020072	051105					
	025426	000040							
4979	025430	042445	042116	047440	MSG100:	.ASCIZ	/%END OF PASS %/		
	025436	020106	040520	051523					
	025444	022440	000						
4980	025447	045	025045	025052	MSG101:	.ASCIZ	/%*****%/		
	025454	025052	025052	025052					
	025462	025052	025052	025052					
	025470	025052	025052	022452					
	025476	000							
4981	025477	101	052125	020117	MSG104:	.ASCIZ	/AUTO CONT.? /		
	025504	047503	052116	037456					
	025512	000040							
4982	025514	051045	041505	053117	MSG105:	.ASCIZ	/%RECOVERED/		
	025522	051105	042105	000					
4983	025527	052	040502	020104	MSG106:	.ASCIZ	/*BAD TAPE OVERFLOW/		
	025534	040524	042520	047440					
	025542	042526	043122	047514					
	025550	000127							
4984	025552	051045	053505	047111	MSG16A:	.ASCIZ	/%REWIND TAPE; RESTART AT BLOCK 1/		
	025560	020104	040524	042520					



	025566	020073	042522	052123	
	025574	051101	020124	052101	
	025602	041040	047514	045503	
	025610	030440	000		
4985	025613	045	047125	042522	MSG107: .ASCII /%UNRECOVERABLE BAD SPOT/
	025620	047503	042526	040522	
	025626	046102	020105	040502	
	025634	020104	050123	052117	
4986	025642	041045	042101	051040	.ASCIZ /%BAD RECORD LEFT ON TAPE%/
	025650	041505	051117	020104	
	025656	042514	052106	047440	
	025664	020116	040524	042520	
	025672	000045			
4987	025674	050052	051517	052111	MSG109: .ASCIZ /*POSITION LOST IN RETRY/
	025702	047511	020116	047514	
	025710	052123	044440	020116	
	025716	042522	051124	000131	
4988	025724	051445	051525	042520	MSG110: .ASCIZ /%SUSPECT BAD TAPE/
	025732	052103	041040	042101	
	025740	052040	050101	000105	
4989	025746	051045	050105	040505	MSG111: .ASCIZ /%REPEAT: /
	025754	035124	000040		
4990	025760	041040	042101	052040	MSG112: .ASCIZ / BAD TAPE SPOTS%/
	025766	050101	020105	050123	
	025774	052117	022523	000	
4991					
4992	026001	045	051440	043117	MSG113: .ASCIZ /% SOFT: /
	026006	035124	000040		
4993					
4994	026012	020045	040510	042122	MSG114: .ASCIZ /% HARD: /
	026020	020072	000		
4995					
4996	026023	045	040510	042122	MSG115: .ASCIZ /%HARD READ ERROR/
	026030	051040	040505	020104	
	026036	051105	047522	000122	
4997	026044	051445	040514	042526	MSG116: .ASCIZ /%SLAVE REWINDING: WILL RESTART AT BOT/
	026052	051040	053505	047111	
	026060	044504	043516	020072	
	026066	044527	046114	051040	
	026074	051505	040524	052122	
	026102	040440	020124	047502	
	026110	000124			
4998	026112	000134			MSG118: .ASCIZ /\//
4999	026114	051045	046505	053117	MSG120: .ASCIZ /%REMOVE TMDP FROM SLAVE TO BE TESTED%/
	026122	020105	046524	050104	
	026130	043040	047522	020115	
	026136	046123	053101	020105	
	026144	047524	041040	020105	
	026152	042524	052123	042105	
	026160	000045			
5000	026162	044045	051101	053504	MSG121: .ASCIZ /%HARDWARE SWR IN USE%/
	026170	051101	020105	053523	
	026176	020122	047111	052440	
	026204	042523	000045		
5001	026210	047516	051440	040514	MSG122: .ASCIZ /NO SLAVES LEFT TO TEST: HALT%/
	026216	042526	020123	042514	

5002	026224	052106	052040	020117	
	026232	042524	052123	020072	
	026240	040510	052114	000045	
	026246	040445	052125	026517	MSG123: .ASCIZ /%AUTO-SEQ: TEST WILL RESTART%/
	026254	042523	035121	052040	
	026262	051505	020124	044527	
	026270	046114	051040	051505	
5003	026276	040524	052122	000045	
	026304	041445	051117	042522	MSG124: .ASCIZ /%CORRECTED PE DATA ERROR/
	026312	052103	042105	050040	
	026320	020105	040504	040524	
	026326	042440	051122	051117	
	026334	000			
5004	026335	040	000		SPACE: .ASCIZ ' '
5005	026337	060	000		DIGIT0: .ASCIZ '0'
5006	026341	061	000		DIGIT1: .ASCIZ '1'
5007					
5008		026344			
5009	026344	000000			WDATA: 0 .EVEN ;WRITE BUFFER
5010					
5011		032352			
5012	032352	000000			RDATA: 0 .=. +4004 ;READ BUFFER
5013					
5014		000001			.END





DAT14	014242	1847	3517#					
DAT15	014272	1848	3533#					
DAT2	014036	1837	3435#					
DAT3	014042	1838	3440#					
DAT3A	014050	3442#	3453					
DAT4	014066	1839	3451#					
DAT5	014076	1840	3458#					
DAT6	014104	1841	3463#					
DAT7	014112	1842	3468#					
DB	000532	1528#						
DCHK	014750	2718	2912	3667#				
DCHKO	014776	3671	3673#					
DEREV1	001164	1707#	2023	3753*				
DEREX	015754	3821	3839	3841	3849	3856	3859	3861#
DEREX1	016006	3862	3865	3867	3869#			
DERFL	000704	1588#	3668*	3744	3870*			
DERR	015354	3737	3785#					
DERRO	015364	3787#	3868					
DERROA	015412	3789	3793#					
DERROB	015440	3798	3801#					
DERROC	015460	3804	3807#					
DERROD	015462	3806	3808#					
DERR1	015504	3811	3814#					
DERR2	015506	3813	3815#					
DERR3	015520	3818#						
DERR4	015522	3786	3817	3819#				
DERR4A	015646	3833	3842#					
DERR4B	015710	3828	3850#					
DERR5	015736	3853	3857#					
DERR6	015750	3830	3851	3860#				
DFX	015352	3745	3747	3752	3754#			
DF0	015250	3693	3732#	3741				
DFOA	015144	3703	3705#	3742				
DFOA0	015156	3709	3711#					
DFOA1	015202	3714	3716#					
DFOA2	015216	3719	3721#					
DFOA3	015232	3724	3726#					
DFOA4	015236	3706	3728#					
DF0B	015104	3694#						
DF0B0	015126	3697	3700#					
DF0C	015066	3686	3690#					
DF0C0	015076	3676	3678	3680	3692#			
DF0D	015052	3682	3687#					
DF0E	015044	3684#	3689					
DF0F	015036	3681#	3685					
DF1	015262	3729	3733	3736#				
DF2	015272	3731	3735	3738#				
DF3	015306	3739	3743#					
DF4	015346	3750	3753#					
DIGITO	026337	4802	4849	5005#				
DIGIT1	026341	4847	5006#					
DOUT	023304	3808	3815	4843#	4856	4858		
DOUTD	023340	4855#						
DPC	016126	3911#	3955					
DPCG	016134	3912	3914#					
DPCO	016142	3915#	3947					





MSG14	023615	2488	4616	4909#				
MSG15	023622	3824	4910#					
MSG16	023652	4274	4911#					
MSG16A	025552	2077	4984#					
MSG17	023655	4271	4609	4912#				
MSG2	023527	3802	4899#					
MSG20	023660	2072	4913#					
MSG21	023670	3259	4914#					
MSG22	023713	2664	4915#					
MSG23	023737	4162	4916#					
MSG23A	023745	4189	4917#					
MSG23B	023752	4178	4918#					
MSG23C	023757	4186	4919#					
MSG23D	023764	4165	4920#					
MSG23E	023772	4170	4921#					
MSG23F	023777	4173	4922#					
MSG23G	024004	4923#						
MSG23H	024011	4216	4924#					
MSG23I	024016	4925#						
MSG23J	024023	4926#						
MSG23K	024030	4927#						
MSG23L	024035	4928#						
MSG24	024042	4382	4929#					
MSG25	024061	4284	4930#					
MSG26	024133	3957	4931#					
MSG27	024144	3971	4932#					
MSG28	024155	2496	2511	4713	4759	4883	4933#	
MSG3	023534	3809	4900#					
MSG30	024157	1879	3091	4934#				
MSG31	024231	3088	4935#					
MSG31A	024313	3095	3096*	4936#				
MSG32	024400	3146	4461	4583	4937#			
MSG33	024413	3177	4938#					
MSG34	024426	3189	4939#					
MSG35	024440	3218	4940#					
MSG36	024460	3228	4941#					
MSG37	024476	3240	4942#					
MSG38	024513	3268	4943#					
MSG39	024531	3277	4944#					
MSG4	023541	3793	4901#					
MSG40	024567	3290	4945#					
MSG41	024616	3299	4946#					
MSG42	024627	3308	4947#					
MSG43	024646	4744	4948#					
MSG44	024652	4417	4949#					
MSG45	024701	4950#						
MSG49	024713	4951#						
MSG5	023546	2262	4902#					
MSG50	024727	3171	4952#					
MSG52	024754	4580	4953#					
MSG52A	024756	3132	4457	4954#				
MSG53	025000	3200	4955#					
MSG54	025012	2312	4157	4956#				
MSG55	025021	2948	4957#					
MSG56	025030	4159	4958#					
MSG57	025034	3165	4959#					









RWND1A	005064	2161	2167	2171#										
RWND2	005102	2159	2174#											
RWND3	005106	2175#	2189											
RWNS5	005172	2180	2185	2187#										
SCVFL	013444	1513*	3097	3286	3316*	3318#	4413*							
SERFL	000706	1589#	2284	2288*	2331	2373	2437*	2449	2459*	2468	2704	2719*	2770*	2783*
		2786	3746	4128*	4136*									
SN	000540	1531#	4863											
SNPG	023454	4871	4874	4880	4882	4885#								
SNPT	023362	3175	4863#											
SPACE	026335	4460	4825	5004#										
SPFLG	000572	1547#	2130	3269	3271									
STAL	000666	1581#	1968*	1975*	1978*	2086*	2168*	2279*	2438*	2561*	2583*	2724*	2848*	2857*
		2861*	2863*	2935*	2970*	2994*	4364*	4372*						
STALL	011654	1976	1979	2280	2439	2448	2562	2584	2725	2849	2864	2936	2971	2994#
		2995												
START	003022	1489	1514	1869#										
STARTA	003144	1496	1892#	2666										
STARTB	003150	1885	1893#											
STARTC	003136	1492	1888#											
STARTD	003250	1889	1912#											
STARTE	003242	1911#	2138											
START1	003406	1935#	1989											
START2	003522	1953	1955#											
START3	003536	1956	1958#											
START4	003552	1940	1959	1961#										
START7	003734	1962	1981	1988#	2095									
START8	003742	1989#												
STAR1A	003420	1937#												
STAR1B	003440	1938	1941#											
STAR1C	003474	1943	1945	1949#										
STAUT	003122	1508	1877	1883#										
STAUTO	003352	1928#	4529	4531	4533	4535	4552	4554	4556	4559				
STFLG	000640	1569#	1893	1894										
STP	003744	1984	1993#	2079										
STTBL	001044	1651#	1900	1901										
SWR	000610	1554#	1871*	1914	1918	1922*	1952	1955	1958	1980	2150	2258	2276	2289
		2300	2318	2335	2348	2356	2378	2394	2548	2551	2553	2557	2564	2572
		2574	2579	2667	2714	2721	2761	2779	2788	2810	2880	2908	2914	2946
		3785	3822	3861	3866	3921	3928	4099	4138	4221	4226	4375	4383	4400
		4402*	4408*	4415	4662	4666	4669							
SWREG	000176	1485#	1871	1914	1922	4400	4408	4662						
TAPG	020372	2267	2317	2446	2466	2652	2858	2879	2945	2968	4327#			
TAPG3	020420	4330	4332#											
TAPG3A	020466	4339	4342#											
TAPG3B	020502	4341	4344	4346#										
TAPG3C	020522	4349	4352#											
TAPG3D	020566	4356	4358	4360#										
TAPG3E	020612	4363	4365#											
TAPG3F	020450	4333	4338#											
TAPG4	020640	4370#	4371	4373										
TAPG5	020654	4374#												
TAPG6	020720	4376	4383#											
TAPG7	020730	4384	4386#											
TC	000542	1532#	1965*	2048*	2100*	2163*	2181*	2194*	2208*	3162*	4241*	4501*		
TEMP1	000644	1572#	3152	3395*	3401	3405*	3842*	3843*	3844*	3845	3847	3892*	3895*	3903*

TEMP2	000646	3909	3952	4327*	4369*	4370*	4699*	4708	4728*	3406*	3413	3893*	3896*	3904*
		1573#	3145	3147*	3178*	3190*	3201*	3322*	3326					
TEMP3	000650	3910	3953											
		1574#	2386*	2392*	2403	2405*	2802*	2808*	2815	2817*	3894*	3917	3954*	4855
		4857												
TEND	004662	2132#	4475											
TIB	000642	1571#	4702	4706	4711	4715	4724	4726	4732*	4733	4755*	4756*	4757	4761
TINER	022706	4725	4727	4737	4739	4744#								
TINF	000636	1568#	1883*	1888*	1892*	2122*	3083	3288						
TINP	011750	1926	3083#											
TINPX	013436	3287	3289	3316#										
TINPO	012302	3141	3145#	3155	3166	3212								
TINPOB	012364	3153	3158#											
TINPOD	012434	3164	3167#											
TINPOE	012470	3170	3175#											
TINP1	012474	3177#												
TINP2	012534	3186#												
TINP2A	012612	3200#												
TINP2B	012652	3208#												
TINP2C	012700	3157	3210	3215#										
TINP3	012712	3218#												
TINP3A	013226	3277#	4414											
TINP4	013270	3085	3286#											
TKB	000614	1556#	4391	4755										
TKS	000612	1555#	1935*	4752*	4753									
TMEX	000564	1544#	2309	2633	2740	2940	3251	3253	4527*					
TMFLG	000676	1585#	2311*	2353*	2368	2630*	2635*	2657	2669	2716	2728*	2742	2744*	2747*
		2859	2882	2910	4045	4067	4073	4085	4103	4155	4244			
TOB	000640	1570#	4770*	4777*	4780*	4784*	4793	4834*	4885*	4887*				
TOG	023110	4774	4778	4781	4785	4791#	4792	4835	4888					
TPB	000620	1558#	4793*											
TPOS	013446	3184	3198	3207	3322#	3324								
TPS	000616	1557#	4791											
TSTAL	000600	1550#	1975	1978	2438	2561	2583	2935	2970	3309	3311			
TTIN	022722	4701	4752#											
TTINT	020734	1479	4391#											
TTOUT	022774	1468	4766#											
TTR	022476	3106	3115	3137	3151	3182	3194	3205	3225	3236	3248	3257	3266	3275
		3284	3297	3306	3315	4424	4453	4673	4697#					
TYPE = 000004		1471#	1874	1879	1994	1998	2001	2004	2007	2010	2013	2016	2019	2022
		2069	2072	2077	2132	2295	2302	2341	2350	2380	2381	2390	2396	2397
		2400	2485	2488	2496	2511	2518	2527	2664	2771	2776	2781	2790	2791
		2806	2812	2822	3092	3095	3099	3108	3132	3142	3146	3165	3171	3177
		3189	3200	3218	3228	3240	3250	3259	3268	3277	3290	3299	3308	3791
		3793	3802	3809	3824	3957	3971	4152	4159	4162	4165	4170	4173	4178
		4181	4186	4189	4194	4198	4203	4206	4216	4275	4284	4287	4292	4335
		4379	4382	4404	4417	4448	4456	4457	4460	4461	4580	4583	4587	4592
		4598	4606	4609	4613	4616	4632	4635	4665	4668	4713	4721	4744	4759
		4761	4802	4825	4847	4849	4864	4883						
TYPOCT= 104400		1475#	1997	2000	2003	2006	2009	2012	2015	2018	2021	2024	2076	2383
		2399	2402	2484	2487	2491	2517	2793	2814	2832	3101	3110	3174	3220
		3231	3242	3252	3261	3270	3279	3292	3301	3310	3801	3965	3973	4164
		4167	4172	4175	4180	4183	4188	4191	4197	4200	4205	4208	4219	4419
		4459	4465	4582	4586	4591	4597	4605	4612	4615	4631	4634	4638	4667
UDES 000552		1539#	1939*	1951*	1965	2048	2099*	2100	2162*	2163	2178*	2181	2193*	2194
		2205	2643	2686	2692	2698	2870	2889	2894	2900	3359	3379	3617	3631





CZTEDDO TMO3-TE16/TU77 DRT  
CZTEDD.P11 06-JUL-83 14:42

MACY11 30(1046) 06-JUL-83 21:03 PAGE 71  
CROSS REFERENCE TABLE -- MACRO NAMES

K 10

SEQ 0127

\$CHAIN	1356#	1871
\$RESTO	1356#	4678
\$SAVE	1356#	4677
.\$ACT1	1356#	1476
.\$EOP	1356#	2133

. ABS. 032354 000

ERRORS DETECTED: 0

CZTEDD,CZTEDD/CRF=CZTEAD.SML/ML,CZTEDD.P11  
RUN-TIME: 5 11 1 SECONDS  
RUN-TIME RATIO: 25/18=1.3  
CORE USED: 14K (28 PAGES)