

TM02, TU16
TE16

TM02/TU16 CTRL LGC
CZTUCGO

AH-9456G-MC
FICHE 1 OF 1

OCT 1983
COPYRIGHT © 74-83
MADE IN USA



A large grid of data tables, likely a control log or schedule, with multiple columns and rows of text. The text is too small to read clearly but appears to be organized in a structured format. A barcode is visible in the bottom right corner of the grid area.

.REM %

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

IDENTIFICATION

PRODUCT CODE: AC-9455G-MC
PRODUCT NAME: CZTUCGO TM02/TU16 CTRL LGC
DATE CREATED: 21 APRIL 76
MAINTAINER: TAPE DIAGNOSTIC ENGINEERING
AUTHOR: R. B. BARNES
REVISED: 11 NOV 1977 BY CLEM WALSH
11 JUL 1983 BY B. LEBLANC

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 DOCUMENT.

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

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

COPYRIGHT (C) 1974, 1983 BY DIGITAL EQUIPMEN CORPORATION

45
46
47
48
49
50
51
52
53
54
55
56
57
58
59

TABLE OF CONTENTS

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

61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115

(PAGE 1)

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO SEQUENTIALLY TEST ALL CONTROL LOGIC AND DATA FORMATTING WITHIN THE TMO2 FORMATTER. EACH TEST WILL ATTEMPT TO ISOLATE FAILURES TO THE MODULE LEVEL AND PROVIDE PRINTOUT INFORMATION WHICH WILL IDENTIFY THE FAILING MODULE. THERE ARE TWO (2) MAJOR AREAS OF TESTING: CONTROL LOGIC AND DATA FORMATTING. THE CONTROL LOGIC SECTION (TEST 1-41 & 57-64) WILL TEST ALL ERROR AND STATUS CONDITIONS AS WELL AS ADDRESSING PROTOCOL AND OPERATIONAL LOGIC SEQUENCES. THE DATA FORMATTING SECTION (TESTS 42-56) WILL TEST ALL DATA FORMATS AND TRANSFER PATHS IN ALL POSSIBLE COMBINATIONS. THE LEVEL OF FAULT ISOLATION IS POSSIBLE BECAUSE OF TMO2 THE STRUCTURE AND ITS MAINTAINENCE MODES.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT A HARDWARE SWITCH REGISTER
- B. 8K OF CORE
- C. CONSOLE TTY
- D. TMO2 MAGTAPE CONTROLLER
- E. MASSBUS CONTROLLER (RH)
- F. TU16 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

THERE ARE TWO (2) STARTING ADDRESSES THAT MAY BE USED: 200(8) AND 210(8).

A. 200(8): STARTING AT THIS ADDRESS WILL CAUSE A PROGRAM IDENTIFICATION HEADER TO BE PRINTED BEFORE TESTING IS BEGUN.

B. 210(8): STARTING AT THIS ADDRESS WILL NOT PRINT THE IDENTIFICATION HEADER AND IS THEREFORE GENERALLY TO BE USED FOR RESTARTS RATHER THAN INITIAL START

***IF THE SOFTWARE SWITCH REGISTER IS USED THEN THE FOLLOWING MESSAGE WILL BE TYPED FIRST ; SWR=XXXXXX NEW= THIS WILL ALLOW THE LOADING OF THE SOFTWARE SWITCH REGISTER LOC. 176 BEFORE THE TESTING IS STARTED. (REFER TO SECTION 5 FOR OPERATOR OPTION)

(PAGE 2)

5. CONSOLE SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER. IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

- 1) TYPE CONTROL G <^G>; THIS WILL ALLOW THE TTY TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.
- 2) THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW= (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)
- 3) 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 LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS WILL BE ALLOWED)
IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGE!.
 - B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

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

155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182

ALL SWITCHES ARE USED (0-15) AND THE NORMAL, OR DEFAULT, RUN
IS DONE WITH ALL SWITCHES SET TO ZERO (0).
ALL SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME.

***BUT, THE SOFTWARE SWITCH REGISTER CAN ONLY BE LOADED DYNAMICALLY
AS STATED ABOVE UNDER CONTROL HEADING.

SW15(100000): 1=HALT ON ERROR
0=CONTINUE
SW14(040000): 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
SW13(020000): 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
SW12(010000): 1=INHIBIT ITERATIONS
0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
SW11(004000): 1=DO CONTINUOUS CYCLE
0=HALT AT END OF PASS
SW10(002000): 1=HALT AT END OF CURRENT TEST
0=CONTINUE TO NEXT TEST
SW9(001000): 1=DO MANUAL INTERVENTION TESTS
0=INHIBIT MANUAL INTERVENTION
SW8(000400): 1=INHIBIT WRAP AROUND DATA CHECK
0=DO DATA CHECKS
SW7(000200): 1=INHIBIT WRAP AROUND STATUS CHECK
0=DO STATUS CHECK
SW6(000100): 1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)
0=AUTO PATTERN
SW5-0: SELECT INDIVIDUAL TEST (1-64)** 00=DO ALL TESTS

(PAGE 3)

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
223
224
225
226
227
228
229
230
231
232

6. ERROR PRINTOUTS

ERROR PRINTOUTS WILL APPEAR IN TWO FORMS, ONE FOR THE CONTROL LOGIC TESTS AND ANOTHER FOR THE DATA TESTS.

CONTROL LOGIC PRINTOUTS WILL CONTAIN A HEADER WHICH CALLS OUT THE TEST NUMBER, FUNCTION BEING TESTED, AND THE SUSPECT MODULE, OR MODULES ON THE FIRST LINE. THE SECOND LINE WILL CONTAIN INFORMATION AS TO THE ACTUAL ERROR. BOTH THE EXPECTED RESULT AND THE ACTUAL RESULT OF THE TEST WILL BE GIVEN. LINE THREE WILL SHOW THE CONTENTS OF THE MAJOR REGISTERS AT THE TIME OF THE ERROR AND LINE FOUR WILL PRINT THE ITERATION NUMBER WHEN APPLICABLE.

DATA TESTS WILL PRINT A HEADER CONTAINING THE TEST NUMBER, AND A DESCRIPTION OF THE WRAP AROUND FUNCTION UNDER TEST. FOLLOWING THE HEADER WILL BE A LIST OF THE MAJOR REGISTERS WITH THE EXPECTED AND ACTUAL VALUES. ANY BAD DATA WILL BE PRINTED (PER CHARACTER) FOLLOWING THE REGISTER INFORMATION OR FOLLOWING THE HEADER IF NO STATUS ERRORS WERE ENCOUNTERED.

EXAMPLES:

1. THE FOLLOWING EXAMPLE SHOWS A TYPICAL ERROR PRINTOUT FOR THE ADDRESS TESTS (LT1-LT3).

LOGIC TEST 1: DRIVE ADDRESSING (M8909 OR RH)
NON-EXIST DRIVE 3 EXPT-NOT RECVD
ITER: 3

THIS PRINTOUT SHOWS THAT THE DRIVE ADDRESS (CS2 BITS 2,1,0) RESULTED IN THE DETECTION OF NED (BIT 12 OF CS2) FOR DRIVE THREE (3) WHEN THAT DRIVE SHOULD BE THERE. THIS ERROR OCCURRED ON ITERATION THREE (3).

2. THIS EXAMPLE WILL SHOW A TYPICAL PRINTOUT OF ONE OF THE REGISTER BIT TESTS.

LOGIC TEST 7: FC BIT TEST (M8705)
FC BITS 15-0 EXPT 177777 RECVD 177577

THIS PRINTOUT SHOWS THAT FRAME COUNT BIT SEVEN (7) WAS NOT SET WHEN IT SHOULD HAVE BEEN. NO ITERATION NUMBER IS DISPLAYED WHEN RUNNING WITH CONSOLE SWITCH TWELVE (12) SET TO A ONE (1).

(PAGE 4)

3. THE FOLLOWING IS A TYPICAL PRINTOUT RESULTING FROM BAD
STATUS DETECTION DURING A MANUAL INTERVENTION TEST (LT14-LT17)

LOGIC TEST 15: MANUAL STATUS TEST 2
BAD STATUS EXPT 100700 RCVD 000700
ITER: 0

THIS SHOWS THAT ON THE FIRST TRY (ITER: 0) THE ACTION TAKEN
BY THE OPERATOR DID NOT RESULT IN THE PROPER STATUS DETECTION
BY THE HARDWARE (ATA IS NOT SET).

4. THE FOLLOWING FOUR (4) EXAMPLES SHOW EACH OF THE ERROR TYPES THAT
CAN BE DETECTED BY ANY OF THE ERROR FORCING TESTS. NOTE THAT ONE OR
MORE OF THE ERROR TYPES COULD BE DETECTED ON A SINGLE EXECUTION OF
THE TEST.

LOGIC TEST 24: DPAR (M8906 RH)
DPAR EXPT EXPT-NOT RCVD

CS1	WC	BA	FC	CS2	DS	ER	AS	MR	TC
004260	000000	033726	000000	000100	010600	000000	000000	177712	140300

THIS MESSAGE SHOWS THAT DPAR (BIT 5 OF ER) DID NOT SET.

LOGIC TEST 26: FCE (M8909)
ERR NOT SET

CS1	WC	BA	FC	CS2	DS	ER	AS	MR	TC
004260	000000	001376	000000	000100	110600	001000	000001	000000	100300

THIS MESSAGE SHOWS THAT WHILE FCE (BIT 9 OF ER) WAS INDEED
SET, THE COMPOSITE ERROR BIT (BIT 14 OF DS) WAS NOT.

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

(PAGE 5)

268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286

LOGIC TEST 30: DTE (M8906 RH)
UNEXPECTED ERROR BITS

CS1	WC	BA	FC	CS2	DS	ER	AS	MR	TC
144260	002006	006600	000000	001300	150600	030000	000001	000017	100300

THIS MESSAGE SHOWS THAT WHILE THE PROPER ERROR BIT (DTE: BIT 12 OF ER) IS SET, OPI (BIT 13 OF ER) IS ALSO SET AND SHOULD NOT BE.

LOGIC TEST 32: UNS (M8909)
NOT RESET BY DRIVE CLEAR

CS1	WC	BA	FC	CS2	DS	ER	AS	MR	TC
144210	002006	006600	000000	001300	150000	040000	000001	000000	140307

THIS MESSAGE SHOWS THAT WHILE THE PROPER ERROR BITS WERE SET, THEY WERE NOT CLEARED BY A DRIVE CLEAR OPERATION.

(PAGE 6)

5. THE FOLOWING ARE TWO EXAMPLES OF ERRORS DETECTED BY THE WRAP AROUND DATA TESTS. NOTE THAT EACH WRAP AROUND TEST MAY BE ACCOMPANIED BY EITHER A STATUS ERROR OF A DATA ERROR OR BOTH.

LOGIC TEST 42: WRAP 3, NRZ, NORMAL, ODD
BAD STATUS
CS1 EXPT 004270 RCVD 144270
CS2 EXPT 000100 RCVD 000100
DS EXPT 010600 RCVD 150600
ER EXPT 000000 RCVD 000100

THIS MESSAGE INDICATES BAD STATUS OF VPE (BIT 6 OF ER)

LOGIC TEST 44: WRAP 2, NRZ, NORMAL, ODD
BAD DATA
CN:0
G: 11111111
B: 11111011
CN:10
G: 00000000
B: 00001000

THIS MESSAGE SHOWS THAT DATA RECEIVED WAS NOT AS EXPECTED. CHARACTER ZERO (CN: 0) SHOWS THAT BIT TWO (2) WAS DROPPED, WHILE CHARACTER TEN (CN: 10) SHOWS BIT THREE (3) HAS BEEN PICKED UP
G: = EXPECTED DATA (GOOD)
B: = ACTUAL DATA (BAD)

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

(PAGE 7)

7. OPERATION

THE PROCEDURES FOR OPERATING THIS PROGRAM ARE QUITE SIMPLE AND REQUIRE ONLY A FEW STEPS:

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE

REFER TO SECTION 5 FOR DYNAMIC LOADING OF SOFTWARE SWITCH REGISTER.

3. PRESS START

ALL CONSOLE SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME. THE NORMAL OPERATING SEQUENCE IS ALL SWITCHES DOWN (0). THE TEST WILL TAKE APPROXIMATELY 3 MINUTES TO RUN; HOWEVER, IF ITERATIONS ARE INHIBITED (SW12=1) THE TEST WILL RUN IN ABOUT 30 SECONDS. THE END OF PASS IS NOTED BY A PRINTOUT STATING END OF PASS, AND THE NUMBER OF THAT PASS.

FOR THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER REFER TO SECTION 5.

SINGLE TEST SELECTION: (SW0-SW5)

WHEN SW0-SW5 ARE SET TO ZERO (00), THE SCHEDULAR WILL EXECUTE ALL TESTS (1-64) IN SEQUENCE. IF SW0-SW5 ARE SET TO SOME SPECIFIC TEST NUMBER (1-64) THEN THAT PARTICULAR TEST ONLY WILL BE EXECUTED UNTIL THE TEST SELECT NUMBER IS CHANGED. WHEN YOU WISH TO SELECT A PARTICULAR TEST, SET SW10 TO A ONE (1) IN ORDER TO STOP AT THE END OF THE CURRENT TEST BEFORE SELECTING A DIFFERENT TEST NUMBER. YOU MAY SELECT THAT NUMBER IN ANY DIRECTION (HIGHER OR LOWER) BECAUSE EACH TEST IS SELF CONTAINED.

WRAP AROUND DATA PATTERNS MAY BE SELECTED VIA SW6 WHEN IN SINGLE TEST MODE. A TELETYPE REQUEST IS MADE FOR THE DESIRED DATA PATTERN WHENEVER SWITCH TEN (SW10) AND SWITCH SIX (SW6) ARE SET TO A ONE (1) WHILE ONE OF THE WRAP TESTS IS SELECTED IN SW0-SW5.

PROGRAM HALTS***

***IF THE SOFTWARE SWITCH REGISTER IS USED AND THE PROGRAM HALTS THEN THE OPERATOR CAN PRESS A <^G> CONTROL G BEFORE HITTING CONTINUE. THIS WILL ALLOW THE OPERATOR TO ENTER DATA INTO LOC. 176 (SWREG). THE FOLLOWING MESSAGE WILL BE TYPED OUT ;
SWR=XXXXXX NEW= (REFER TO SECTION 5 FOR OPERATOR OPTIONS).

318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369

(PAGE 8)

371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422

8. TEST DESCRIPTION

LOGIC TEST #1: DRIVE ADDRESSING

PURPOSE: VERIFY THE PRESENCE OF TMO2 AT THE ADDRESSES SPECIFIED BY THE OPERATOR. TEST OCCURS IMMEDIATELY AFTER DRIVE SELECTION.

PROGRAMMED SEQUENCE: FOR EACH TMO2 ADDRESS (0-7) THE C1 REGISTER IS READ, AND THE NON-EXISTANT DRIVE (NED) BIT IS CHECKED. NED IS SET WHEN THE TMO2 DOES NOT RESPOND TO DEM BY ISSUING TRA. IN THIS TEST, NED IS EXPECTED FOR EACH ADDRESS NOT TYPED BY THE OPERATOR.

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909

CIRCUITS

PRINT REFERENCES

RH-DS BITS	(CSRB)
RH-NED BIT	(CSRB)
MASSBUS CABLE C (DEM,TRA,DS BITS)	(MB3)
DRIVE ADDRESS	(MB12)
DEM-TRA HANDSHAKE	

LOGIC TEST #2: REGISTER ADDRESSING

PURPOSE: CHECK THE REGISTER SELECT LINES

PROGRAMMED SEQUENCE: READ ALL 14 MASSBUS REGISTERS WHICH MAKE UP THE TAPE SYSTEM CHECKING FOR (1) CONTROL BUS PARITY ERROR AND (2) ILR BIT

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909,M8905,M8903

CIRCUITS

PRINT REFERENCE

C-LINES	(MB1,2,3),(MB13),(MB14),(MB15)
RH REGISTER SELECT	(BCTA)
TMO2 REGISTER SELECT	(MB12)
MASSBUS REGISTER SELECT LINES	(MB1,2)
PARITY TREE	(MB14)
CPAR,ILR BITS	(MB11)

424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474

(PAGE 9)

LOGIC TEST #3: CONTROL BUS

PURPOSE: VERIFY THAT ALL CONTROL LINES PROPERLY TRANSMIT ONES AND ZEROS.

PROGRAMMED SEQUENCE: WRITE FC REGISTER AND CHECK CPAR, READ FC AND CHECK MCPE, UPDATE DATA, REPEAT. DATA IS ALL 0'S, WALKING '1' BIT, ALL '0'S, 2 WALKING '1' BITS BEGINNING WITH BIT 0 AND 8 DATA IS CHECKED ALONG WITH ERROR BITS.

LIKELY FAULT LOCATIONS: M5904,CABLES,M5903YA,M8909,M8905,M8903

CIRCUITS

PRINT REFERENCE

C-LINES	(MB1,2,3)
C-BUS MULTIPLEXERS	(MB1,3,4,5,8)(TCCM7)(MR)
ERROR BIT	(MB111)
MCPE BIT	(PACA)

LOGIC TEST #4: SLAVE ADDRESSING

PURPOSE: VERIFY THE FUNCTIONING OF THE SLAVE ADDRESS BITS IN THE TAPE CONTROL REGISTER THE SLAVE ADDRESS BUS LINES, THE ADDRESS DECODE CIRCUIT IN THE TU16 AND THE SPR BIT.

IT IS REQUIRED THAT ONLY ONE SLAVE BE POWERED UP WHEN
THIS TEST IS RUN.

PROGRAMMED SEQUENCE: THE SLAVE ADDRESS BITS IN THE TAPE CONTROL REGISTER ARE LOADED WITH ALL 8 COMBINATIONS AND SPR IS CHECKED FOR EACH ADDRESS.

LIKELY FAULTS LOCATIONS: M8905,M8907,CABLE,M9001,M8910,M9001YA,M8903

CIRCUITS

PRINT REFERENCE

REGISTER SELECT	(MB12)
SLAVE ADDRESS BITS	(MR6)
SLAVE ADDRESS LINES	(M8907,2-2),(LAW6)
TU16 ADDRESS DECODE	(LAW6)
SPR BIT	(LAW6)(M9001YA)(TCCM7)

476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520

LOGIC TEST #5: MAINTENANCE REGISTER BITS

PURPOSE: TO VERIFY THAT THE VARIOUS BITS OF THE MAINTENANCE REGISTER CAN BE WRITTEN INTO AND READ AND OTHERWISE BEHAVE AS EXPECTED.

PROGRAMMED SEQUENCE: IN THE FIRST SEQUENCE AN INCREMENTING DATA WORD (0-37) IS WRITTEN INTO THE MR. WITH THE CONTENTS OF BITS 0-4 BEING CHECKED AFTER EACH OPERATION. THEN 15(OCTAL) IS WRITTEN INTO THE REGISTER WHICH SHOULD PERMIT BITS 7-15 TO BE WRITTEN FROM THE CONTROL BUS. THEN THE DATA WRITTEN INTO BITS 7-15 IS INCREMENTED AND CHECKED.

LIKELY FAULT LOCATIONS: M8905

CIRCUITS

PRINT REFERENCE

C-LINES	
MAINTENANCE REGISTER	(MR2,3,5)
M.R. FUNCTION DECODE	(MR5)
M.R. MULTIPLEXOR	(MR4)

LOGIC TEST #6: TAPE CONTROL REGISTER BITS

PURPOSE: TO VERIFY THAT TAPE CONTROL BITS 0-11 CAN BE WRITTEN INTO AND READ AND THAT TCW BEHAVES AS EXPECTED:

PROGRAMMED SEQUENCE: ALL 0'S DATA PATTERN IS WRITTEN TO AND READ FROM THE TAPE CONTROL REGISTER. TCW IS CHECKED FOR A 'ONE'. THIS SEQUENCE IS REPEATED WITH ALL '1' DATA AND AGAIN WITH ALL '0'S.

LIKELY FAULT LOCATIONS: M8909,M8905

CIRCUITS

PRINT REFERENCE

TM02 REGISTER SELECT	(MB12)
TC FLIP-FLOPS, MULTIPLEXERS	(MR6)

522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565

LOGIC TEST #7: FRAME COUNT BIT TEST

PURPOSE: TO VERIFY THAT THE FRAME COUNT BITS CAN BE WRITTEN INTO AND READ FROM AND ARE NEITHER STUCK AT 0 NOR STUCK AT 1.

PROGRAMMED SEQUENCE: DATA IS WRITTEN INTO THE FRAME COUNT REGISTER AND READ FROM IT. THE DATA PATTERN IS ALL ZEROS FOLLOWED BY ALL ONES FOLLOWED BY ALL ZEROS.

LIKELY FAULT LOCATIONS: M8909

CIRCUITS PRINT REFERENCE

TM02 REGISTER SELECT	(MBI2)
FRAME COUNT REGISTER	(MBI8)
FRAME COUNT MULTIPLEXERS	(MBI10)

LOGIC TEST #10: FUNCTION CODE BIT TEST

PURPOSE: TO VERIFY THAT THE FUNCTION CODE BITS CAN BE WRITTEN INTO AND READ FROM AND ARE NEITHER STUCK AT 0 NOR STUCK AT 1.

PROGRAMMED SEQUENCE: THE C1 REGISTER IS WRITTEN WITH ALL ZEROS. DATA IS CHECKED ON THE 5 FUNCTION CODE BITS (BITS 1-5). BITS 1-5 ARE WRITTEN WITH ONES, CHECK AND REPEAT WITH ALL ZEROS.

LIKELY FAULT LOCATION: M8909, M8905

CIRCUITS PRINT REFERENCE

TM02 REGISTER SELECTION	(MBI2)
FUNCTION CODE FLOPS	(MBI5)
FUNCTION CODE MULTIPLEXERS	(MR6)

(PAGE 12)

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

LOGIC TEST #11: GO BIT SET, RESET

PURPOSE: TO VERIFY THAT THE GO BIT CAN BE SET IN A SIMULATED READ OPERATION AND CLEARED WITH AN INIT.

PROGRAMMED SEQUENCE: INIT AND CHECK THAT GO=0. SET UP A SIMULATED READ OPERATION BY LOADING A WAM3 15(OCTAL) INTO THE MAINTENANCE REGISTER, CLEARING THE FRAME COUNT REGISTER TO SET FCS, LOAD 1700 (FORMAT) INTO THE TAPE CONTROL REGISTER, SETTING READ COMMAND AND GO BIT. CHECK FOR GO=1. INIT AND CHECK THAT GO BIT=0.

LIKELY FAULT LOCATION: MASSBUS CABLE B(INIT),M8909,M8905

CIRCUIT

PRINT REFERENCE

FCS	MB18
SET ILF	MB17
SET NEF	MB17
GO BIT	MB15
GO BIT MULTIPLEXER	MR6
SET ILR	MB12

LOGIC TEST #12: DRIVE READY BIT

TEST 12 IS AN EXACT REPEAT OF TEST 11 EXCEPT THAT DRIVE READY (DRY) IS CHECKED INSTEAD OF THE GO BIT. DRY IS SIMPLY GO L MULTIPLEXED ONTO THE C-LINES AS BIT SEVEN OF THE STATUS REGISTER.

PRINT REF TCCM7

(PAGE 13)

606
607
608
609
610
611
612
613
614
615
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
641
642
643
644
645
646

LOGIC TEST #13: INTERRUPT TEST

PURPOSE: TO VERIFY THE OPERATION OF THE RH INTERRUPT LOGIC.

PROGRAMMED SEQUENCE: THE C1 REGISTER IS CLEARED, PRIORITY IS SET,
THE INTERRUPT ENABLE BIT IS SET AND THE INTERRUPT IS AWAITED.

LIKELY FAULT LOCATION:

CIRCUITS

PRINT REFERENCE

INTERRUPT CONTROL

BCTF

MANUAL INTERVENTION TESTS 14,15,16,17

LOGIC TEST #14: STATUS AT BOT, ON LINE, LOADED, NO WRITE RING

PURPOSE: TO TEST FOR THE PRESENCE OF MOL,WRL,DPR,DRY,BOT.

PROGRAMMED SEQUENCE: THE OPERATOR IS INSTRUCTED TO LOAD THE
DRIVE WITH A TAPE MINUS THE WRITE ENABLE RING AND PLACE
THE DRIVE ON LINE AT BOT MOL,WRL,DPR,DRY,BOT ARE CHECKED.

LIKELY FAULT LOCATION: M8910,SLAVE CABLE, M8903

CIRCUIT

PRINT REFERENCE

MOL
WRL
DPR
DRY
BOT

LAW6,TCCM7,M8908,M9001YA,YC
LAW8,TCCM7,M8908,M9001YA,YC
TCCM7
TCCM7
LAW6,TCCM7,M8908YA,M8913,YA

(PAGE 14)

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

LOGIC TEST #15: STATUS AT BOT,OFFLINE,LOADED, NO WRITE RING

PURPOSE: TO TEST ATA,DPR,DRY,SSC

PROGRAMMED SEQUENCE: OPERATOR IS INSTRUCTED TO TAKE DRIVE
OFFLINE: ATA,SSC,DPR,DRY ARE CHECKED.

LIKELY FAULT LOCAT:ON: M8910,M8903,M8909,SLAVE CABLE

CIRCUIT

PRINT REFERENCE

SSC
ATA

LAW8,M8913,M8913YA,TCCM7
MB13

LOGIC TEST #16: STATUS AT EOT,ON LINE, LOADED, NO WRITE RING

PURPOSE: TO TEST EOT,SSC,SLA

PROGRAMMED SEQUENCE: THE OPERATOR IS INSTRUCTED TO MOVE TO EOT
AND PLACE THE DRIVE ON LINE. EOT,SSC,SLA ARE CHECKED IN
ADDITION TO ATA,MOL,WEL,DPR,DRY

LIKELY FAULT LOCATION: M8910,SLAVE CABLE,M8903

CIRCUIT

PRINT REFERENCE

SSC
EOT
SLA

LAW8,M8913,M8913YA,TCCM7
LAW6,TCCM7,M8908YA,M8913YA
LAW8,TCCM7,M9001YA,YC,M8908

687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715

LOGIC TEST #17: STATUS AT ONLINE LOADED

TEST 17 IS EXACTLY LIKE TEST 16 EXCEPT THAT THE DRIVE IS
REVERSED OFF OF EOT AND THE WRITE ENABLE RING IS INSTALLED.

EACH OF THE NEXT 11 TESTS ARE DESIGNED TO VERIFY
THE ABILITY TO SET SPECIFIC ERROR BITS.

LOGIC TEST #20: ILLEGAL FUNCTION

PROGRAMMED SEQUENCE: THE WORD COUNT IS SET TO -1. ALL CODES
STORED IN THE ILLEGAL FUNCTION TABLE ARE LOADED AND ILF IS
CHECKED FOR EACH ONE. THEN UNEXPECTED ERRORS ARE CHECKED.

LIKELY FAULT LOCATION: M8909

CIRCUIT

PRINT REFERENCE

SET ILF DECODE
ILF FLOP
ILF MULTIPLEXER

MB15,MB17
MB111
MB110

(PAGE 16)

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
743
744
745
746
747
748
749
750
751
752
753
754
755

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

PROGRAMMED SEQUENCE: INIT, SELECT SLAVE AND DRIVE. LOAD 300
@ TAPE CONTROL REGISTER LOAD WAM3 IN THE MAINTENANCE
REGISTER. LOAD THE C1 REGISTER WITH A READ COMMAND AND GO
BIT. ATTEMPT TO WRITE THE FRAME COUNT REGISTER. READ
ERROR REGISTER. CHECKING FOR RMR. CHECK FOR UNEXPECTED ERRORS
WAIT FOR ACCL. DELAY. DO EOP CLEAR.

LIKELY FAULT LOCATION: M8909

CIRCUIT -----	PRINT REFERENCE -----
RMR DECODE	MBI2
RMR FLOP	MBI11
RMR MULTIPLEXER	MBI10

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)

PROGRAMMED SEQUENCE: WRITE 20(8) INTO CS2. ENABLING THE
WRITING OF EVEN PARITY ON MASSBUS. WRITE ALL ONES TO
FRAME COUNT. RESET PAT. CHECK ERROR REGISTER FOR CPAR CHECK
FOR OTHER UNEXPECTED ERRORS.

LIKELY FAULT LOCATIONS: M8909

CIRCUIT -----	PRINT REFERENCE -----
MASSBUS PARITY TREE	MBI4
CPAR FLOP	MBI11
CPAR MULTIPLEXER	MBI10

(PAGE 17)

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
791
792
793
794
795
796
797
798
799

LOGIC TEST #23: FORMAT ERROR (FMT)

PROGRAMMED SEQUENCE: AN ILLEGAL FORMAT CODE IS LOADED INTO THE TAPE CONTROL REGISTER. WAM3 IS LOADED INTO THE MR READ COMMAND AND THE GO BIT IS SET. THE ERROR REGISTER IS CHECKED FOR FORMAT ERROR AND UNEXPECTED ERROR BITS. THIS SEQUENCE IS REPEATED FOR ALL ILLEGAL FORMAT CODES

LIKELY FAULT LOCATIONS: M8905,M8906,M8909

CIRCUIT -----	PRINT REFERENCE -----
FORMAT BITS	MR6
ILF DECODE	BF3
ILF FLOP	MBI11
ILF MULTIPLEXERS	MBI10

LOGIC TEST #24: DATA BUS PARITY ERROR (DPAR)

PROGRAMMED SEQUENCE: SET UP A WRAP 2 AS FOLLOWS:
NORMAL FORMAT ----> TAPE CONTROL REGISTER, -10 ----> WORD COUNT, -20 ----> FRAME COUNT, WAM2 ----> MAINTENANCE REGISTER,. LOAD WRITE COMMAND AND GO BIT. SET PAT BIT IN CS2. AFTER A DELAY MR IS LOADED 4 TIMES CAUSING 2 DATA BUS TRANSFERS. DPAR AND CPAR ARE CHECKED. THEN A CHECK FOR UNEXPECTED ERRORS IS MADE MASKING OPI.

LIKELY FAULT LOCATIONS: DBUS LINES, M8905, M8906

CIRCUIT -----	PRINT REFERENCE -----
MM CLK	MRS
WRT CLK GENERATION	TCCM4
DPAR FLOP	MBI11
DATA BUS PARITY TREE	BF3

(PAGE 18)

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

LOGIC TEST #25: NON-EXECUTABLE FUNCTION (NEF)

PROGRAMMED SEQUENCE: LOAD FC WITH -1. SET WAM 2. SET
WRITE AND GO. ILF SHOULD SET DUE TO TOO SMALL INITIAL
FRAME COUNT. CHECK ILF. CHECK FOR UNEXPECTED ERRORS.

LIKELY FAULT LOCATION: M8909

CIRCUIT PRINT REFERENCE

NEF FLOP	MB111
NEF MULTIPLEXER	MB110
SET NEF	MB17

LOGIC TEST #26: FRAME COUNT ERROR

PROGRAMMED SEQUENCE: SET WC TO -10, FC TO -20 WAM3 IN
MAINTENANCE REGISTER, LOAD WRITE AND GO, DELAY ISSUE MM OR
CLEAR. CHECK FCE AND CHECK FOR UNEXPECTED ERRORS. FRAME
COUNT ERROR SHOULD BE SET BECAUSE A WRITE OPERATION WAS
TERMINATED PRIOR TO A WORD COUNT OVERFLOW.

LIKELY FAULT LOCATIONS: M8909, MB CABLE, M8903, M8905

CIRCUITS PRINT REFERENCE

RUN LINE	MB1
EBL PLS	MB19
FCE FLOP	MB111
SHUTDOWN LOGIC	TCCM5
MAINT. FUNCTION DECODE	MR5

(PAGE 19)

842
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

LOGIC TEST #27: ILLEGAL REGISTER

PROGRAMMED SEQUENCE: IF THE RH HAS ALL MASSBUS REGISTER OPEN (MOST SYSTEM IN THE FIELD DON'T), ALL THE ILLEGAL REGISTER ADDRESSES ARE READ, CHECKING THE ILR BIT AFTER EACH ATTEMPT.

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

CIRCUITS -----	PRINT REFERENCE -----
REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MBI2
ILR FLOP	MBI11

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

THE MAINTENANCE REGISTER IS LOADED WITH A FUNCTION THAT IS DESIGNED TO CRIPPLE OCCUPIED. FRAME COUNT REGISTER IS CLEARED TO SET FCS LOAD WRITE COMMAND AND GO BIT. CHECK FOR DTE. THEN DRIVE IS INITIALIZED. FCS IS SET AND WRP 3 CODE IS LOADED INTO MR. WRITE COMMAND AND GO BIT ARE SET. AFTER DELAY FOR ACCELERATION, THE MR CLOCK IS GENERATED AND ANOTHER CHECK IS MADE FOR DTE. FINAL CHECK IS MADE FOR ERRORS OTHER THAN OPI. THE FIRST MAINTENANCE REGISTER CODE WHICH CRIPPLES THE OCCUPIED RECEIVER CAUSES OCCUPIED TO BE ASSERTED AND TESTS THE CIRCUITRY WHICH CHECKS FOR OCCUPIED WHEN A DATA TRANSFER COMMAND IS INITIATED. THE SECOND TEST UTILIZES THE FACT THAT THE WRP 3 CODE INHIBITS THE MASSBUS WCLK RECEIVER CREATING A SITUATION WHERE SCLK IS NOT FOLLOWED BY A WRITE CLOCK.

LIKELY FAULT LOCATIONS: M8909, M8905, M8906, MB CABLES

CIRCUITS -----	PRINT REFERENCES -----
DTE FLOP	MBI11
CRIPPLE OCCUPIED FUNCTION	MR5
WRP 3 FUNCTION	MR5
PREVIOUS OCCUPIED CHECK	MBI7
CHECK FOR WCLK	BF2
MM CLK	MR5

(PAGE 20)

895
896
897
898
899
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
932
933
934
935
936
937
938

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

SET UP INCLUDES FORMAT, WRP 2 (BIT FIDDLER WRITE), FCS.
WRITE COMMAND AND GO BIT ARE SET AND THE PROGRAM DELAYS
FOR OPI. A SECOND TEST INVOLVES SETTING UP WRP 3 AND
ISSUING A READ COMMAND. ESSENTIALLY THIS TEST UTILIZES
THE WRAPAROUND CODES TO PREVENT ANY RECORDS BEING DETECTED
AFTER A READ OR A WRITE COMMAND IS ISSUED.

LIKELY FAULT LOCATIONS: M8903, M8909

CIRCUITS

PRINT REFERENCES

OPI TIMER
OPI FLOP
OPI TIMER CONTROL

TCCM5
MB11
MB17

LOGIC TEST 32: UNSAFE (UNS)

PROGRAMMED SEQUENCE:

A NON-EXISTANT SLAVE IS SELECTED AND A READ COMMAND IS
ISSUED. UNSAFE ERROR IS CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8910, SLAVE CABLE

CIRCUITS

PRINT REFERENCES

UNSAFE FLOP
SET UNSAFE
MOL GENERATION

MB111
MB17
LAW6

(PAGE 21)

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
979
980

LOGIC TEST 33: POSITIONING IN PROGRESS (PIP)

PROGRAMMED SEQUENCE:

SET UP DRIVE AND SLAVE ARE SELECTED, FCS IS SET. A SPACE
COMMAND IS ISSUED AND PIP IS CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8903

CIRCUITS

PRINT REFERENCES

SPACE FUNCTION DECODE
PIP GENERATION
STATUS REGISTER

MBIS
TCCM7
TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

DENSITY CODES 0 - 4 ARE LOADED AND PES IS CHECKED FOR EACH
CODE. IT IS EXPECTED ONLY FOR DENSITY 4.

LIKELY FAULT LOCATIONS: M8905, SLAVE BUS, M8911, M8903

CIRCUITS

PRINT REFERENCES

DENSITY BITS
DENSITY LINES
PES CIRCUIT
PES STATUS BIT

MR6
SBC
SC3
TCCM7

(PAGE 22)

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

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

SETUP FORMAT AND WRP-3 ARE SET, READ COMMAND IS ISSUED.
TCW IS CHECKED. DRIVE IS INITIALIZED, TAPE CONTROL REG-
ISTER IS WRITTEN TO AND TCW IS CHECKED.

LIKELY FAULT LOCATION: M8905

CIRCUIT

PRINT REFERENCES

TCW

MR6

LOGIC TEST 36: FRAME COUNTER STATUS (FCS)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, FCS IS CHECKED, DRIVE IS INITIALIZED,
FRAME COUNTER IS WRITTEN TO, AND FCS IS CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8903

CIRCUITS

PRINT REFERENCES

FCS BIT
FCS MULTIPLEXER

MB18
TCCM7

(PAGE 23)

1021
1022
1023
1024
1025
1026
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

LOGIC TEST 37: ACCELERATION (ACCL)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, FORMAT IS SET AND ACCL IS CHECKED
FOR ONE. WAM 3 CODE IS LOADED, READ COMMAND IS ISSUED.
AFTER A DELAY ACCL IS CHECKED FOR ZERO.

LIKELY FAULT LOCATIONS: M8903, M8911

CIRCUITS

PRINT REFERENCES

ACCL BIT, MOTION DELAY COUNTER	TCCM3
CLOCK	SC2

LOGIC TEST 40: PE TAPE MARK (TM)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, WAMO IS SET, WRITE TAPE MARK IS SET.
AFTER DELAY TAPE MARK BIT IS CHECKED. WAMO MULTIPLEXES
THE OUTPUT OF THE WRITE DATA GENERATOR ONTO THE RDA LINES.
THE DATA SYNC MODULES SYNC ON THE DATA AND SEND ENVELOPE
INFORMATION TO THE TAPE MARK DETECTOR ON M8902.

LIKELY FAULT LOCATIONS: M8902, M8901, M8903, M8905

CIRCUITS

PRINT REFERENCES

TAPE MARK DETECTOR	TCPE4, TCPE5
TAPE MARK MULTIPLEXER	TCCM7
ENVELOPE SIGNALS	DS 3, 5, 7
WRITE DATA BUFFER	TCCM2
RDA MULTIPLEXERS	TCCM6
WRITE TAPE MARK FUNCTION	MBI5
WAMO SIGNAL	MR5

1069
1070
1071
1072
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

LOGIC TEST 41: NRZ TAPE MARK (TM VPE, ITM)

PROGRAMMED SEQUENCE:

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8903, M8904

CIRCUITS

PRINT REFERENCES

WRITE DATA BUFFER	TCCM2
RSDO MULTIPLEXER	TCCM6
RDA MULTIPLEXERS	TCCM6
TM DETECTOR	CNRZ4
ILLEGAL TAPE MARK FLOP	CNRZ4

SEE NOTE ON PAGE 22 FOR TESTS 42-56

LOGIC TEST 42: WRP3, NRZ, NORMAL, ODD (BIT FIDDLER READ)

PROGRAMMED SEQUENCE:

TAPE CONTROL REGISTER IS LOADED WITH DENSITY 3, FORMAT 14,
ODD PARITY WRP3 IS LOADED IN MAINT. REGISTER. READ FUNCTION
IS LOADED, EXECUTING WRAP3 CONSISTS LOADING DATA CHARACTERS
INTO MAINT. REGISTER DATA FIELD, WHERE THERE ARE MULTI-
PLEXERS TO BIT FIDDLER, MM CLK IS TOGGLED TO CREATE RDS.
THE BIT FIDDLER TRANSMIT DATA ACCESS MASSBUS DATA LINES.
WHEN ALL THE DATA HAS BEEN TRANSMITTED AN EOR CLK IS
TRANSMITTED TO N REGISTER WHICH BRINGS OPERATION TO A CLOSE.

LIKELY FAULT LOCATIONS: M8906, M8905, MASSBUS P-LINES

CIRCUITS

PRINT REFERENCES

MASSBUS CHAR. ASSEMBLE	BF5
CLK. GENERATOR	BF2
MAINT. REGISTER DATA FIELD	MR2, MR3
RDS GENERATION	MR5

1120
1121
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

LOGIC TEST 43: WARP3, PE, NORMAL, ODD

JUST LIKE TEST 42 EXCEPT FOR DENSITY BITS.

LOGIC TEST 44: WRAP2, NRZ, NORMAL, ODD

PROGRAMMED SEQUENCE:

WRAP2 IS BIT FIDDLER WRITE. MM CLOCK IS MULTIPLEXED INTO WRT CLK SO THAT IT FORMS WRT STROBE. THE OUTPUT OF THE BIT FIDDLER IS CLOCKED INTO THE DATA FIELD OF THE MAINTENANCE REGISTER. SET UP CONSISTS OF MOVING NRZ, NORMAL FORMAT, ODD PARITY TO UNIT DESCRIPTION MAINT. REGISTER IS LOADED WITH WAM2 WRITE COMMAND IS ISSUED. AFTER THE ACCELERATION DELAY, MM CLOCK ARE GENERATED UNTIL ALL THE DATA HAS BEEN CLOCKED. SEQUENCE IS COMPLETED BY LOADING MAINTENANCE REGISTER WITH EOR CLR. THE SEQUENCE IS REPEATED WITH VARYING DATA PATTERNS.

LIKELY FAULT LOCATIONS: M8906, M8905, M8903

CIRCUITS

PRINT REFERENCES

BIT FIDDLER CHAR UNPACK
BIT FIDDLER DATA REQUEST
WRT STRB.
MAINT. REG. DATA FIELD

BF4
BF2
TCCM4
MR2, MR3

(PAGE 26)

1156
1157
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

LOGIC TEST 45: WRP2, PE, NORMAL, ODD

THE TEST IS EXACTLY LIKE TEST 44 EXCEPT THAT PE WRT CLK ENBL L MUST BE ASSERTED BY M8902 TO ENABLE WR TO STROBE. THIS DOES NOT HAPPEN UNTIL THE PE WRITE CONTROL CIRCUIT HAS CLOCKED THROUGH THE PREAMBLE.

CIRCUITS PRINT REFERENCES

(IN ADDITION TO TEST 44)
PE WRITE CONTROL TCPE3

LOGIC TEST 46: WRP1, NRZ, NORMAL, ODD

THIS TEST IS EXACTLY LIKE TEST 44 EXCEPT THE WRITE BUFFER (TCCM2) IS MULTIPLEXED TO THE MAINTENANCE REGISTER.

LIKELY FAULT LOCATIONS: M8903, M8904 (CRC GENERATOR)

CIRCUITS PRINT REFERENCES

WRITE BUFFER TCCM2
CRC GENERATOR CNRZ2

LOGIC TEST 47: WRAP1, PE, NORMAL, ODD

IN PE MODE BOTH THE PREAMBLE AND POSAMBLE ARE CLOCKED THROUGH THE WRITE BUFFER IN ADDITION TO PHASE ENCODED DATA.

LIKELY FAULT LOCATIONS: M8902 (WRITE CONTROL STATES), M8903

CIRCUITS PRINT REFERENCE

WRITE BUFFER TCCM2
WRITE CONTROL TCPE3

1205
1206
1207
1208
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
1251
1252
1253
1254

LOGIC TEST 50: WRAP0, NORMAL, ODD

WRAP 0 IS THE MOST COMPLETE OF THE TMO2 WRAPAROUND DATA PATH. IT CONSISTS OF A WRITE OPERATION IN WHICH THE OUTPUT OF THE WRITE DATA BUFFER IS MULTIPLEXED TO THE READ DATA INPUTS, CHECKED AND LOADED INTO THE MAINTENANCE REGISTER FOR RETRIEVAL BY THE PROCESSOR. THE WHOLE OPERATION USES THE TYPE SYSTEM CLOCKS AND HAPPENS AT THE PROPER DATA RATES. MM CLK SERVES AS A FLAG ANNOUNCING WHEN A NEW CHARACTER HAS BEEN LOADED INTO THE MAINTENANCE REGISTER. IN PE MODE EVERY OTHER CHARACTER IS READ TO ALLOW SUFFICIENT PROCESSOR LOOP TIME. IN NRZ WRAP 0 IS EXPECTED TO PRODUCE LRC ERRORS BECAUSE THE TMO2 DOES NOT WRITE THE LRC CHARACTER.

LIKELY FAULT LOCATIONS: M8904, M8903

CIRCUITS

PRINT REFERENCES

CRC GENERATOR	CNR22
CRC CHECKOUT	CNR23
CRC, CRC STROBE	TCCM4
READ LINE MULTIPLEXERS	TCCM6
MM CLK	MR5
CRC READ TIMING	CNR24
SHUTDOWN CIRCUITRY	TCCM5

LOGIC TEST 51: WRP0, PE, NORMAL, ODD

REPEAT OF TEST 50 IN PE MODE.

LIKELY FAULT LOCATIONS: M8901, M8902, M8903

CIRCUITS

PRINT REFERENCES

DATA DISCRIMINATOR	DS2, DS4, DS6
PHASE LOCKED CLOCK	DS3, DS5, DS7
SKREW BUFFER	DS3, DS5, DS7
PE WRITE MAJOR STATES	TCPE3
PE READ MAJOR STATES	TCPE5
WRAP 0 CIRCUIT TO BLOCK RLT RDS	TCPE3
DESKEW BUFFER READ COUNTER	TCPE4

(PAGE 28)

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
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295

LOGIC TEST 52: CORE DUMP WRITE, WAM2

REPEAT OF TEST 44 EXCEPT BIT FIDDLER OPERATES IN CORE DUMP
MODE!

LIKELY FAULT LOCATION: M8906

LOGIC TEST 53: CORE DUMP READ, WAM3

REPEAT OF TEST 42 EXCEPT BIT FIDDLER OPERATES IN CORE DUMP
MODE!

LIKELY FAULT LOCATION: M8906

LOGIC TEST 54: EVEN PARITY WRITE - WAM1

REPEAT OF TEST 46 EXCEPT EVEN PARITY IS SPECIFIED.

LIKELY FAULT LOCATION: M8903

LOGIC TEST 55: EVEN PARITY READ: WAM0,

REPEAT OF TEST 50 EXCEPT EVEN PARITY IS USED.

LIKELY FAULT LOCATIONS: M8903, M8904

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

(PAGE 29)

LOGIC TEST 56: READ REVERSE, WAM3 (M8906)

REPEAT OF TEST 42 EXCEPT READ REVERSE COMMAND IS ISSUED.

LIKELY FAULT LOCATIONS: M8908, M8909

NOTE: FOR TESTS 42-56

FOR THE MOST PART, THIS DIAGNOSTIC TESTS PARTICULAR
AREAS OF THE TMO2 LOGIC INDEPENDENT OF THE TU16. HOWEVER
THERE ARE A FEW SIGNALS WHICH ARE REQUIRED FROM THE TU16
TO COMPLETE THE TESTS, AND AT LEAST ONE CASE WHERE TU16
FAILURES INTERFERE WITH THE TESTS. THE KNOWN CASES ARE
LISTED HERE AND SHOULD BE CHECKED AS PART OF THE DEBUGGING.

1. MOL(SB)L: REQUIRED TO ENABLE CLOCK
2. CLOCK(SB)L: REQUIRED TO GENERATE ACCELERATION AND SHUTDOWN.
3. WRITE CLOCK(SB)L: USED IN WAMO TO GENERATE DATA AND REC(SB)L
4. RSDO(SB)L: SHOULD NOT OCCUR DURING WRAP AROUND TESTS, BUT WILL
INTERFERE WITH THEIR OPERATION IF CAUSED BY A FAILURE
SUCH AS A GROUNDED OUTPUT FROM THE G056.

(PAGE 30)

1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360

THE NEXT 5 TESTS CONSISTS OF WRITING ON TAPE USING MAINTENANCE MODE FUNCTIONS TO FORCE ERROR CONDITIONS TO CHECK THE ERROR CHECKING CAPABILITIES. OCCASIONAL ERRORS MAY RESULT FROM TAPE DEFECTS. CONSTANT ERROR MAY BE THE RESULT OF PROBLEMS WITH ERROR CHECKING CIRCUITRY OR PROBLEMS WITH THE DRIVE. DEBUG OF THE PROBLEMS MAY BE EASIER USING DATA RELIABILITY OF UTILITY DRIVER.

LOGIC TEST 57: CYCLIC REDUNDANCY ERROR

PROGRAMMED SEQUENCE:

FIRST THE DIAGNOSTIC PERFORMS A WRAP0 DESIGNED TO LOAD THE CRC CHECKER IN A KNOWN MANNER. CHECK ARE MADE FOR LRC ERROR AND THE CONTENT OF CRC REGISTER. THEN A WRITE OPERATION IS PERFORMED USING A MAINT. MODE (IICC) WHICH INHIBITS THE INITIALIZATION OF THE CRC CHECKER. THE CRC CHECKER LOGIC WHICH HAS NOT BEEN CLEARED SHOULD DETECT A CRC ERROR. UNEXPECTED ERROR BITS MAY INDICATE PROBLEMS WITH THE WRITE OPERATION.

LIKELY FAULT LOCATIONS: M8905, M8904, G056, SLAVE CABLE,
----- M8910

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
CRC CHECK CIRCUIT

MRS
CNRZ3

1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386

LOGIC TEST 60: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL)
WHICH ASSERTS WD(SB) 5L THROUGHOUT THE RECORD. ALL ONES
DATA IS USED SO THAT THE FUNCTION ONLY INTERFERES WITH
THE WRITING OF THE LRC CHARACTER WHEN NONE OF THE TMO2
WRITE DATA LINES SHOULD BE ASSERTED.

LIKELY FAULT LOCATIONS: M8505, M8903, M8910, M8904

CIRCUITS

MM FUNCTION DECODE
WRITE LINE DRIVERS
WRITE HEAD DRIVERS
LRC CHECKING

PRINT REFERENCES

MRS
TCCM2
LAW3, 4
CNRZ3

1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433

(PAGE 31)

LOGIC TEST 61: PE CORRECTABLE DATA

PROGRAMMED SEQUENCE:

A PE WRITE OPERATION IS PERFORMED USING A FUNCTION WHICH WILL GROUND THE BIT STROBE LINE ON BIT 1. THIS SHOULD CAUSE THE BIT1 DEAD TRACK FLOP TO ASSERT AND CAUSE CORRECTABLE DATA ERROR. THE DEAD TRACK REGISTER IS CHECKED FOR BIT 1.

LIKELY FAULT LOCATIONS: M8905, M8901, M8902

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE	MRS
BIT STROBE CIRCUIT	DS4
DEAD TRACK FLOP	DS5, TCPE2
DEAD TRACK REGISTER	MR4

LOGIC TEST 62: PE INCORRECTABLE DATA

REPEAT OF TEST 61, EXCEPT THAT THE MAINT. MODE FUNCTION GOUNDS BITS STROBE FOR BITS 1, 2 AND THE WD LINE FOR BIT 5 IN HELD ASSERTED. INC. DATA AND PCF ERRORS ARE EXPECTED.

LIKELY FAULT LOCATIONS: M8902, M8901

CIRCUIT

PRINT REFERENCE

INC ERROR, PEF,	TCPE2
-----------------	-------

1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464

(PAGE 32)

LOGIC TEST 63: PE FORMAT

THE MM FUNCTION USED IN THIS TEST INVERTS THE DATA USED
IN PREAMBLE AND POSTAMBLE OF BIT ONE.

LIKELY FAULT LOCATIONS: M8902, M8903, M8905

CIRCUITS

PRINT REFERENCES

PEF.
WRITE BUFFER
MM DECODE

TCPE2
TCCM2
MR5

LOGIC TEST 64: FRAME COUNT OVERFLOW

THIS TEST USES A WRAP2 TO CHECK THE OVERFLOW OF FRAME
COUNT REGISTER.

LIKELY FAULT LOCATION: M8909

FRAME COUNT REGISTER MB18

1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508

9. LISTING

%

```
.TITLE CZTUGO TM02/TU16 CTRL LGC
:ZZ - CZTUC-G-0
:21 OCT 75
:R. BARNES
:REVISED 21 APR 76 BY S. CARPENTER
:REVISED 11 NOV 77 BY CLEM WALSH
:REVISED 11 JUL 83 BY B. LEBLANC

:      TO SUPPORT THE DYNAMIC LOADING OF THE
:      SOFTWARE SWITCH REGISTER
:ABS

:CONSOLE SWITCHES*****
:SW15: 1=HALT ON ERROR
:      0=CONTINUE
:SW14: 1=LOOP ON ERROR
:      0=CONTINUE
:SW13: 1=DO NOT PRINT ERRORS
:      0=PRINT ERRORS
:SW12: 1=INHIBIT ITERATIONS
:      0=DO ITERATIONS
:SW11: 1=CONTINUOUS CYCLE
:      0=HALT AT END OF PASS
:SW10: 1=HALT AT END OF EACH TEST
:      0=CONTINUE
:SW9:  1=DO MANUAL INTERVENTION TESTS
:      0=INHIBIT MANUAL INTERVENTION
:SW8:  1=NO WRAP DATA CHECK
:      0=DO WRAP DATA CHECK
:SW7:  1=NO WRAP STATUS CHECK
:      0=DO WRAP STATUS CHECK
:SW6:  1=SELECTABLE WRAP DATA PATTERN (IN SINGLE TEST)
:      0=AUTO PATTERNS
:SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
```



```
1557 ;REGISTER EQUIVS*****
1558
1559 000000 R0=%0
1560 000001 R1=%1
1561 000002 R2=%2
1562 000003 R3=%3
1563 000004 R4=%4
1564 000005 R5=%5
1565 000006 SP=%6
1566 000007 PC=%7
1567
1568 ;TRAP CATCHERS*****
1569
1570 000000 .=0
1571 000200 .REPT 200
1572 .+2
1573 HALT
1574 .ENDR
1575
1576 :
1577 :
1578 :
1579 :
1580 001000 $SVPC=.
1581
1582 000040 .=40
1583 000040 000 DRIVE: .BYTE 0 ;DRIVE # FOR XXDP LOAD MEDIUM
1584 ;ASSEMBLE AS A 0
1585
1586 000041 .=41
1587 000041 000 MEDIUM: .BYTE 0 ;XXDP LOAD MEDIUM
1588 ;ASSEMBLE AS A 0
1589
1590 000042 .=42
1591 000042 000000 .WORD 0 ;LOCATION INDICATOR - AUTOM/MAN MODE
1592 ;ASSEMBLE AS A 0
1593
1594 000046 .=46
1595 000046 002654 .WORD $ENDAD ;SET TO $ENDAD IN .SEOP
1596
1597 000052 .=52
1598 000052 000000 .WORD 0 ;CHARACTERISTICS OF PROGRAM
1599 ;SET TO 0
1600
1601 001000 .=$SVPC ;RESTORE PC
1602
1603 :
1604 :
1605 :
```

1607
1608
1609
1610
1611
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646

```
*****  
:                               MACRO FOR SIZING DRIVES AND SLAVES  
:                               IN CHAIN MODE.  
:                               *****  
:                               .MACRO $SCHNMODE  
:                               MOV #1,DRVN ;INITIALIZE DRIVE #  
:                               :INITIALIZE SLAVE #  
NXTDRV: MOV #1,SLVN ;INITIALIZE SLAVE #  
1$: MOV #40,ACS ;INIT CONTROLLER  
: INC DRVN ;STEP DRIVE #  
: CMP #10,DRVN ;ALL DRIVES TESTED?  
: BEQ TEND0 ;BRANCH - IF YES  
: MOV DRVN,ACS ;LOAD DRIVE #  
: TST @C1 ;ACCESS DRIVE  
: BIT #10000,ACS ;NON-EXISTANT DRIVE  
: BNE 1$ ;BRANCH - IF YES (NED=1)  
NXTSLV: INC SLVN ;STEP SLAVE #  
: BNE 1$ ;BRANCH IF NO? SLAVE C  
: TST DRVN ;DRIVE 0?  
: BNE 1$ ;BRANCH - IF NO  
: TSTB XXDPM ;XXDP?  
: BEQ 1$ ;BRANCH - IF NO  
: INC SLVN ;STEP TO SLAVE #1  
1$: CMP #10,SLVN ;ALL SLAVES TESTED?  
: BEQ NXTDRV ;BRANCH - IF YES  
: MOV SLVN,@C ;LOAD SLAVE UNIT #  
: BIT #2000,@DT ;SLAVE PRESENT?  
: BEQ NXTSLV ;BRANCH - IF NO (SPR=0)  
: CMPB #2,@DT ;IS DRIVE A TE16 OR TU16?  
: BEQ 2$ ;BRANCH IF NO  
: BIT #140000,@DT ;IS DRIVE A TAPE UNIT?  
: BEQ NXTSLV ;BRANCH IF NO  
2$: MOV #MSGC,R4 ;GET MESSAGE  
: JSR PC,TTOUT ;TYPE MESSAGE  
: HALT  
: .ENDM $SCHNMODE  
:                               *****
```

C
C

```
1648  
1649  
1650  
1651 000060 000060  
1652 000060 024260  
1653 000062 000000  
1654  
1655  
1656  
1657 000176 000176  
1658 000176 000000  
1659  
1660  
1661  
1662  
1663 000200 000200  
1664 000200 005000  
1665 000202 000167 001222  
1666  
1667 000210 000210  
1668 000210 012700 000001  
1669 000214 000167 000210  
1670  
1671  
1672  
1673 000224 000224  
1674 000224 024246  
1675 000226 000340  
1676
```

```
;TTY INTERRUPT VECTOR*****  
.=60  
TTINT ;TTY INTERRUPT HEADER ADDRESS  
0  
;SOFTWARE SWITCH REGISTER LOC. 176*****  
.=176  
SWREG: 0 ;SOFTWARE SWITCH REGISTER  
;START ADDRESS*****  
.=200  
CLR R0  
JMP START ;PROGRAM START  
.=210  
MOV #1,R0 ;SET NO HEADER FLAG  
JMP START  
;TM02 INTERRUPT VECTOR*****  
.=224  
MTINT ;TAPE INTERRUPT HANDLER ADDRESS  
340
```

1678 000510

.=510
:MASS BUS REGISTER EQUIVS*****

1679
1680
1681 000510 172440
1682 000512 172442
1683 000514 172444
1684 000516 172446
1685 000520 172450
1686 000522 172452
1687 000524 172454
1688 000526 172456
1689 000530 172460
1690 000532 172462
1691 000534 172464
1692 000536 172466
1693 000540 172470
1694 000542 172472
1695 000544 172474

C1: 172440
WC: 172442
BA: 172444
FC: 172446
CS: 172450
DS: 172452
ER: 172454
AS: 172456
CC: 172460
DB: 172462
MR: 172464
DT: 172466
SN: 172470
TC: 172472
BAE: 172474

:ILLEGAL FUNCTION CODES

1696
1697
1698
1699 000546 005405
1700 000550 007415
1701 000552 016423
1702 000554 020437
1703 000556 022443
1704 000560 025447
1705 000562 031455
1706 000564 033465
1707 000566 036473

ILFT: 5405
7415
16423
20437
22443
25447
31455
33465
36473

:CONSTANTS*****

1708
1709
1710
1711 000570 177776
1712 000572 177570
1713 000574 177560
1714 000576 177562
1715 000600 177564
1716 000602 177566
1717 000604 177777
1718 000606 000011
1719 000610 000020
1720 000612 000224
1721 000614 172440

PSW: 177776 :PROCESSOR STATUS
SWR: 177570 :SWITCH REGISTER
TKS: 177560 :TTY READER STATUS
TKB: 177562 :TTY READ BUFFER
TPS: 177564 :TTY PUNCH STATUS
TPB: 177566 :TTY PUNCH BUFFER
SERNUM: 177777 :SERIAL NUMBER
DRVTP: 011 :DRIVE TYPE
ITAMT: 20 :ITERATION AMOUNT
VECT: 224 :INTERRUPT VECTOR(RH)
REGS: 172440 :STARTING REGISTER ADDRESS

ACT11 MODE INDICATORS

1722
1723
1724
1725
1726 000616 000000
1727 000620 000
1728 000621 000
1729 000622 000
1730 000623 000
1731

AUTOM: .WORD 0 :AUTOMATIC MODE INDICATOR
ACT11M: .BYTE 0 :ACT11 AUTO MODE INDICATOR
XXDPM: .BYTE 0 :XXDP AUTOM MODE INDICATOR
ADUMPM: .BYTE 0 :ACT11 DUMP MODE INDICATOR
XDUMPM: .BYTE 0 :XXDP DUMP MODE INDICATOR

```
1733 ;FLAGS AND COUNTERS*****
1734
1735 000624 000000 TCB: 0
1736 000626 000000 TIB: 0
1737 000630 000000 HDRFL: 0
1738 000632 000000 EMADDR: 0
1739 000634 000000 DRVN: 0
1740 000636 000000 TR00: 0
1741 000640 000000 TR01: 0
1742 000642 000000 TR02: 0
1743 000644 000000 TR03: 0
1744 000646 000000 TR04: 0
1745 000650 000000 TR05: 0
1746 000652 000000 TR06: 0
1747 000654 000000 TR07: 0
1748 000656 000000 TR10: 0
1749 000660 000000 TR11: 0
1750 000662 000000 TR12: 0
1751 000664 000000 TR13: 0
1752 000666 000000 TR14: 0
1753 000670 000000 TR15: 0
1754 000672 000000 NRZOF: 0
1755 000674 000000 SLVN: 0
1756 000676 000000 PFLG: 0
1757 000700 000000 RTRN: 0
1758 000702 000000 ERADD: 0
1759 000704 000000 TEMP1: 0
1760 000706 000000 TEMP2: 0
1761 000710 000000 TEMP3: 0
1762 000712 000000 ITCNT: 0
1763 000714 000000 SAV1: 0
1764 000716 000000 SAV2: 0
1765 000720 000000 SAV3: 0
1766 000722 000000 SCOLP: 0
1767 000724 000000 ITRLP: 0
1768 000726 000000 EXFL: 0
1769 000730 000000 ATAF: 0
1770 000732 000000 SLAF: 0
1771 000734 000000 SSCF: 0
1772 000736 000000 ERRF: 0
1773 000740 000000 ASF: 0
1774 000742 000000 SCF: 0
1775 000744 000000 TREF: 0
1776 000746 000000 PEXFL: 0
1777 000750 000000 STFLG: 0
1778 000752 000000 LTADD: 0
1779 000754 000000 T24FL: 0
1780 000756 000000 ADDFL: 0
1781 000760 000000 WAM: 0
1782 000762 000000 FUN: 0
1783 000764 000000 DATC: 0
1784 000766 000000 WTAD: 0
1785 000770 000000 DATAD: 0
1786 000772 000000 RDAD: 0
1787 000774 000000 W2FLG: 0
1788 000776 000000 DERFL: 0
```

1789	001000	000000	PREFL:	0	
1790	001002	000000	SERFL:	0	
1791	001004	000000	CRCNT:	0	
1792	001006	000000	UDES:	0	
1793	001010	000000	WPGFL:	0	
1794	001012	000000	PATRN:	0	
1795	001014	000000	STATF:	0	
1796	001016	000000	RDRVF:	0	
1797	001020	000000	RCDP:	0	
1798	001022	000000	STATC:	0	
1799	001024	000000	SKAT:	0	
1800	001026	000000	PCNTR:	0	:PASS COUNTER
1801					
1802			:*****		
1803	001030	000000	PAFLG:	0	
1804	001032	000000	RH17F:	0	
1805			:*****		
1806					
1807			:EXPT WRAP STATUS*****		
1808					
1809	001034	000000	WCS1:	0	
1810	001036	000000	WCS2:	0	
1811	001040	000000	WDS:	0	
1812	001042	000000	WER:	0	
1813					
1814			:DATA PATTERN GENERATORS*****		
1815					
1816	001044	000000	DATBL:	0	
1817	001046	016050	DATA0:	DAT1	:ALL ONE BITS
1818	001050	016072	DATA1:	DAT2	:ALL ZERO BITS
1819	001052	016100	DATA2:	DAT3	:ALTERNATING ONE/ZERO BITS
1820	001054	016110	DATA3:	DAT4	:ALL BITS 0-377
1821					
1822			:CORE DUMP PATTERNS*****		
1823					
1824	001056	000005	WCDP2:	5	
1825	001060	000005		5	
1826	001062	000012		12	
1827	001064	000012		12	
1828	001066	000000		0	
1829	001070	000017	WCDP0:	17	
1830	001072	000017		17	
1831	001074	000017		17	
1832	001076	000017		17	
1833	001100	000000		0	

1835
1836
1837
1838 001102 000000
1839 001104 000000
1840 001106 002724
1841 001110 002724
1842 001112 003154
1843 001114 003154
1844 001116 003362
1845 001120 003364
1846 001122 003546
1847 001124 003546
1848 001126 004024
1849 001130 004032
1850 001132 004226
1851 001134 004230
1852 001136 004440
1853 001140 004442
1854 001142 004564
1855 001144 004566
1856 001146 004720
1857 001150 004722
1858 001152 005162
1859 001154 005164
1860 001156 005370
1861 001160 005400
1862 001162 005472
1863 001164 005540
1864 001166 005610
1865 001170 005656
1866 001172 005726
1867 001174 005774
1868 001176 006044
1869 001200 006112
1870 001202 006162
1871 001204 006176
1872 001206 006326
1873 001210 006342
1874 001212 006472
1875 001214 006506
1876 001216 006616
1877 001220 006632
1878 001222 006746
1879 001224 006762
1880 001226 007272
1881 001230 007306
1882 001232 007424
1883 001234 007432
1884 001236 007640
1885 001240 007664
1886 001242 007756
1887 001244 010000
1888 001246 010274
1889 001250 010310
1890 001252 010624

:LOGIC TEST ENTRY TABLE*****

TSTTBL: 0
0
T1AD: LT1
T1IAD: LT1
T2AD: LT2
T2IAD: LT2
T3AD: LT3
T3IAD: LT3IT
T4AD: LT4
T4IAD: LT4
T5AD: LT5
T5IAD: LT5IT
T6AD: LT6
T6IAD: LT6IT
T7AD: LT7
T7IAD: LT7IT
T10AD: LT10
T10IAD: LT10IT
T11AD: LT11
T11IAD: LT11IT
T12AD: LT12
T12IAD: LT12IT
T13AD: LT13
T13IAD: LT13IT
T14AD: LT14
T14IAD: LT14IT
T15AD: LT15
T15IAD: LT15IT
T16AD: LT16
T16IAD: LT16IT
T17AD: LT17
T17IAD: LT17IT
T20AD: LT20
T20IAD: LT20IT
T21AD: LT21
T21IAD: LT21IT
T22AD: LT22
T22IAD: LT22IT
T23AD: LT23
T23IAD: LT23IT
T24AD: LT24
T24IAD: LT24IT
T25AD: LT25
T25IAD: LT25IT
T26AD: LT26
T26IAD: LT26IT
T27AD: LT27
T27IAD: LT27IT
T30AD: LT30
T30IAD: LT30IT
T31AD: LT31
T31IAD: LT31IT
T32AD: LT32

1891	001254	010640	T32IAD:	LT32IT
1892	001256	010756	T33AD:	LT33
1893	001260	010772	T33IAD:	LT33IT
1894	001262	011060	T34AD:	LT34
1895	001264	011102	T34IAD:	LT34IT
1896	001266	011222	T35AD:	LT35
1897	001270	011236	T35IAD:	LT35IT
1898	001272	011374	T36AD:	LT36
1899	001274	011410	T36IAD:	LT36IT
1900	001276	011514	T37AD:	LT37
1901	001300	011530	T37IAD:	LT37IT
1902	001302	011664	T40AD:	LT40
1903	001304	011706	T40IAD:	LT40IT
1904	001306	012012	T41AD:	LT41
1905	001310	012026	T41IAD:	LT41IT
1906	001312	012260	T42AD:	LT42
1907	001314	012260		LT42
1908	001316	012372	T43AD:	LT43
1909	001320	012372		LT43
1910	001322	012454	T44AD:	LT44
1911	001324	012454		LT44
1912	001326	012562	T45AD:	LT45
1913	001330	012562		LT45
1914	001332	012644	T46AD:	LT46
1915	001334	012644		LT46
1916	001336	012752	T47AD:	LT47
1917	001340	012752		LT47
1918	001342	013040	T50AD:	LT50
1919	001344	013040		LT50
1920	001346	013146	T51AD:	LT51
1921	001350	013146		LT51
1922	001352	013230	T52AD:	LT52
1923	001354	013230		LT52
1924	001356	013342	T53AD:	LT53
1925	001360	013342		LT53
1926	001362	013470	T54AD:	LT54
1927	001364	013470		LT54
1928	001366	013540	T55AD:	LT55
1929	001370	013540		LT55
1930	001372	013610	T56AD:	LT56
1931	001374	013610		LT56
1932	001376	013666	T57AD:	LT57
1933	001400	013716	T57IAD:	LT57IT
1934	001402	014224	T60AD:	LT60
1935	001404	014244	T60IAD:	LT60IT
1936	001406	014452	T61AD:	LT61
1937	001410	014500	T61IAD:	LT61IT
1938	001412	014730	T62AD:	LT62
1939	001414	014756	T62IAD:	LT62IT
1940	001416	015202	T63AD:	LT63
1941	001420	015230	T63IAD:	LT63IT
1942	001422	015450	T64AD:	LT64
1943	001424	015464	T64IAD:	LT64IT
1944	001426	000000	TADX:	0


```
1946 .EVEN
1947 ;PROGRAM START AND HOUSEKEEPING*****
1948
1949 001430 012777 000340 177132 START: MOV #340,@PSW ;SET PRIORITY
1950 001436 012706 000500 MOV #500,SP ;SET STACK POINTER
1951
1952 : *****
1953 : DIAGNOSTIC SETUP FOR EXECUTION
1954 : UNDER ACT11.
1955 : *****
1956
1957 001442 004767 024202 JSR PC,CKMODE ;CHECK FOR MODE OF OPERATION
1958 001446 005767 177144 TST AUTOM ;IS IT AUTOMATIC MODE
1959 001452 001001 BNE 1$ ;BRANCH - IF YES
1960 001454 000412 BR SUSWR ;CHECK SWR IN DUMPM
1961 001456 032737 020000 000052 1$: BIT #20000,@#52 ;SET UP FOR MANUAL INTERVENTION
1962 001464 001406 BEQ SUSWR ;BRANCH - IF NO
1963 001466 012704 025776 MO #MSGC,R4 ;GET MESSAGE
1964 001472 004767 023024 JSR PC,TTOUT ;TYPE MESSAGE
1965 001476 000167 024240 JMP ABORT ;AND ABORT THE PROGRAM
1966
1967 : *****
1968
1969 001502 013746 000006 SUSWR: MOV @#6,-(SP) ;SAVE VECTORS
1970 001506 013746 000004 MOV @#4,-(SP)
1971 001512 012737 001532 000004 MOV #1$,@#4 ;SET UP FOR TIMEOUT
1972 001520 022777 177777 177044 CMP #-1,@SWR ;REFERENCE HARDWARE SWITCH REGISTER
1973 001526 001402 BEQ 2$
1974 001530 000404 BR 3$
1975 001532 022626 1$: CMP (SP)+,(SP)+ ;ADJUST STACK
1976 001534 012767 000176 177030 2$: MOV #SWREG,SWR ;POINT TO SOFTWARE SWITCH REG
1977 001542 012637 000004 3$: MOV (SP)+,@#4 ;RESTORE VECTORS
1978 001546 012637 000006 MOV (SP)+,@#6
1979 001552 022767 000176 177012 CMP #SWREG,SWR ;IS SWREG SELECTED
1980 001560 001002 BNE 4$
1981 001562 004767 023404 JSR PC,CNTLU ;CHECK FOR CONTROL G
1982
1983 001566 4$: *****
1984 : IF IN ACT11 MODE INHIBIT TYPING PROGRAM
1985 : IDENTIFICATION AND MANUAL INTERVENTION
1986 : *****
1987
1988 001566 005767 177026 TST ACT11M ;AUTOMATIC MODE?
1989 001572 001142 BNE TSCD ;BRANCH - IF YES
1990
1991 : *****
1992
1993 001574 005700 TST R0
1994 001576 001136 BNE ST2
1995 001600 005067 177220 CLR SKAT ;CLEAR SKIP ADDRESS TEST FLAG
1996 001604 012704 026112 MOV #MSG1,R4
1997 001610 004767 022706 JSR PC,TTOUT ;PRINT TITLE
1998 001614 012704 027714 MOV #MSG44,R4
1999 001620 004767 022676 JSR PC,TTOUT ;REQUEST REGISTER ADDRESS
2000 001624 016703 176764 MOV REGS,R3
2001 001630 004767 023030 JSR PC,OCTP ;PRINT CURRENT ADDRESS
```

2002	001634	012705	000614	MOV	#REGS,R5	:SET ADDRESS SAVE LOC
2003	001640	012701	000006	MOV	#6,R1	:SET SIZE OF RESPONSE
2004	001644	012702	176400	MOV	#176400,R2	:SET UPPER LIMIT
2005	001650	012703	172300	MOV	#172300,R3	:SET LOWER LIMIT
2006	001654	004767	022404	JSR	PC,TTR	:GO GET RESPONSE
2007	001660	012704	027736	MOV	#MSG45,R4	
2008	001664	004767	022632	JSR	PC,TTOUT	:REQUEST VECTOR
2009	001670	016703	176716	MOV	VECT,R3	
2010	001674	004767	022764	JSR	PC,OCTP	:PRINT CURRENT VECTOR
2011	001700	012705	000612	MOV	#VECT,R5	:SET ADDRESS SAVE LOC
2012	001704	012701	000003	MOV	#3,R1	:SET SIZE OF RESPONSE
2013	001710	012702	000224	MOV	#224,R2	:SET UPPER LIMIT
2014	001714	012703	000150	MOV	#150,R3	:SET LOWER LIMIT
2015	001720	004767	022340	JSR	PC,TTR	:GO GET RESPONSE
2016	001724	016700	176662	MOV	VECT,R0	:GET VECTOR
2017	001730	012720	024246	MOV	#MTINT,(R0)+	:LOAD INTERRUPT ADDRESS IN VECTOR
2018	001734	012710	000340	MOV	#340,(R0)	:LOAD PRIORITY
2019	001740	016700	176650	MOV	REGS,R0	:GET START OF REGS
2020	001744	012701	000016	MOV	#16,R1	:SET NUMBER OF REGS
2021	001750	012702	000510	MOV	#C1,R2	:GET START OF TABLE
2022	001754	010022		MOV	RO,(R2)+	:BUILD TABLE
2023	001756	062700	000002	ADD	#2,R0	:BUMP ADDRESS
2024	001762	005301		DEC	R1	:SEE IF DONE
2025	001764	001373		BNE	ST0	:IF NOT: BR
2026	001766	012702	000624	MOV	#TOB,R2	
2027	001772	012700	000077	MOV	#77,R0	
2028	001776	005022		CLR	(R2)+	:CLEAR FLAGS + COUNTERS
2029	002000	005300		DEC	R0	
2030	002002	001375		BNE	ST1	
2031	002004	012704	030164	MOV	#MSG52,R4	
2032	002010	004767	022506	JSR	PC,TTOUT	:PRINT NRZ ONLY REQUEST
2033	002014	012705	000672	MOV	#NRZOF,R5	
2034	002020	012701	000001	MOV	#1,R1	:SET SIZE OF ENTRY
2035	002024	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
2036	002030	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
2037	002034	004767	022224	JSR	PC,TTR	:GO GET RESPONSE
2038	002040	012704	030273	MOV	#MSG56,R4	
2039	002044	004767	022452	JSR	PC,TTOUT	:REQUEST STATIC ONLY
2040	002050	012705	001022	MOV	#STATC,R5	:SET ADDRESS OF STATIC FLAG
2041	002054	012701	000001	MOV	#1,R1	:SET SIZE OF RESPONSE
2042	002060	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
2043	002064	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
2044	002070	004767	022170	JSR	PC,TTR	:GET RESPONSE
2045	002074	005067	176726	CLR	PCNTR	:CLEAR PASS COUNTER

C
C

```
2047
2048 ;TEST SCHEDULAR*****
2049
2050 002100 005067 176704 TSCD: CLR WPGFL ;CLEAR WRAP PATRN FLAG
2051 002104 052777 000100 176462 BIS #100,@TKS ;SET KEYBOARD IE BIT
2052 002112 005067 176632 CLR STFLG ;CLEAR SINGLE TEST FLAG
2053
2054 ; *****
2055
2056 002116 005067 176710 CLR RH17F ;SET RH11 INDICATOR
2057 002122 013745 000004 MOV @#4,-(SP) ;SAVE ERROR TRAP AND VECTORS
2058 002126 013745 000006 MOV @#6,-(SP) ;SAVE PRIORITY
2059 002132 012737 002156 000004 MOV #1$,@#4 ;SET TIME OUT
2060 002140 005037 000006 CLR @#6 ;SET LOW PRIORITY
2061 002144 005777 176374 TST @BAE ;REFERENCE BAE REGISTER
2062 002150 012767 000001 176654 MOV #1,RH17F ;SET RH70 INDICATOR
2063 002156 012637 000006 1$: MOV (SP)+,@#6 ;RESTORE ERROR TYPE
2064 002162 012637 000004 MOV (SP)+,@#4
2065
2066 ; *****
2067
2068 002166 017700 176400 MOV @SWR,R0
2069 002172 042700 177700 BIC #177700,R0
2070 002176 005700 TST R0
2071 002200 001151 BNE STSCD ;GO SELECT SINGLE TEST
2072
2073 ; *****
2074
2075 002202 005767 176410 TST AUTOM ;AUTOMATIC MODE?
2076 002206 001473 BEQ TSCDA ;BRANCH - IF NOT
2077 002210 $CHNMODE ;MACRO FOR CHAINMODE
(1) 002210 012767 177777 176416 MOV #-1,DRVN ;INITIALIZE DRIVE #
(1) 002216 012767 177777 176450 NXTDRV: MOV #-1,SLVN ;INITIALIZE SLAVE #
(1) 002224 012777 000040 176266 1$: MOV #40,@CS ;INIT CONTROLLER
(1) 002232 005267 176376 INC DRVN ;STEP DRIVE #
(1) 002236 022767 000010 176370 CMP #10,DRVN ;ALL DRIVES TESTED?
(1) 002244 001562 BEQ TEND0 ;BRANCH - IF YES
(1) 002246 016777 176362 176244 MOV DRVN,@CS ;LOAD DRIVE #
(1) 002254 005777 176230 TST @C1 ;ACCESS DRIVE
(1) 002260 032777 010000 176232 BIT #10000,@CS ;NON-EXISTANT DRIVE
(1) 002266 001356 BNE 1$ ;BRANCH - IF YES (NED=1)
(1) 002270 005267 176400 NXTSLV: INC SLVN ;STEP SLAVE #
(1) 002274 001010 BNE 1$ ;BRANCH IF NOT SLAVE 0
(1) 002276 005767 176332 TST DRVN ;DRIVE 0?
(1) 002302 001005 BNE 1$ ;BRANCH - IF NO
(1) 002304 105767 176311 TSTB XXDPM ;XXDP?
(1) 002310 001402 BEQ 1$ ;BRANCH - IF NO
(1) 002312 005267 176356 INC SLVN ;STEP TO SLAVE #1
(1) 002316 022767 000010 176350 1$: CMP #10,SLVN ;ALL SLAVES TESTED?
(1) 002324 001734 BEQ NXTDRV ;BRANCH - IF YES
(1) 002326 016777 176342 176206 MOV SLVN,@C ;LOAD SLAVE UNIT #
(1) 002334 032777 002000 176174 BIT #2000,@DT ;SLAVE PRESENT?
(1) 002342 001752 BEQ NXTSLV ;BRANCH - IF NO (SPR=0)
(1) 002344 122777 000002 176164 CMPB #2,@DT ;IS DRIVE A TE16 OR TU16?
(1) 002352 001404 BEQ 2$ ;BRANCH IF NO
(1) 002354 032777 140000 176154 BIT #140000,@DT ;IS DRIVE A TAPE UNIT?
```

```
(1) 002362 001742          BEQ      NXTSLV          ;BRANCH IF NO
(1) 002364 012704 025776    2$:      MOV      #MSGC,R4      ;GET MESSAGE
(1) 002370 004767 022126    JSR      PC,TTOUT        ;TYPE MESSAGE
(1) 002374 000000          HALT
2078
2079      ;      *****
2080
2081 002376 012767 001102 176346 TSCDA:  MOV      #TSTTBL,LTADD
2082 002404 062767 000004 176340 TSCDO:  ADD      #4,LTADD
2083 002412 016767 176334 176304      MOV      LTADD,ITRLP
2084 002420 062767 000002 176276      ADD      #2,ITRLP          ;SET ITERATION ADDRESS
2085 002426 005777 176320      TST      @LTADD
2086 002432 001002          BNE      TSCD1
2087 002434 000167 000136      JMP      TEND              ;GO TO END ROUTINE
2088 002440 005067 176164      TSCD1:  CLR      HDRFL          ;CLEAR PRINT HEADER FLAG
2089 002444 017700 176302      MOV      @LTADD,R0        ;SET POINTER TO TEST
2090 002450 000110          JMP      (R0)              ;GO TO TEST
2091 002452 004767 022442      TSCD2:  JSR      PC,CKSWR        ;CHECK FOR CNTL G
2092 002456 032777 002000 176106      BIT      #2000,@SWR      ;SEE IF HALT ON TEST
2093 002464 001404          BEQ      TSCD3            ;IF NOT: BR
2094 002466 004767 023146      JSR      PC,STOP
2095 002472 005067 176312      CLR      WPGFL          ;CLEAR WRAP DATA GENERATOR FLAG
2096 002476 005767 176246      TSCD3:  TST      STFLG          ;SE IF SINGLE TEST
2097 002502 001740          BEQ      TSCDO            ;IF NOT: BR
2098 002504 017700 176062      MOV      @SWR,R0
2099 002510 042700 177700      BIC      #177700,R0      ;MASK TEST NUMBER
2100 002514 005700          TST      R0              ;SEE IF RETURN TO ALL
2101 002516 001002          BNE      STSCD           ;BRANCH - IF NOT
2102 002520 000167 177354      JMP      TSCD            ;JUMP - IF YES
2103 002524 012767 000001 176216 STSCD:  MOV      #1,STFLG        ;SET SINGLE TEST FLAG
2104 002532 022700 000065      CMP      #65,R0         ;SEE IF EXCEEDED TESTS
2105 002536 003417          BLE      TEND            ;IF SO: BR
2106 002540 000241          CLC
2107 002542 006100          ROL      R0
2108 002544 006100          ROL      R0              ;SET TABLE MODIFIER
2109 002546 012767 001102 176176      MOV      #TSTTBL,LTADD
2110 002554 060067 176172      ADD      R0,LTADD        ;SET TEST POINTER
2111 002560 016767 176166 176136      MOV      LTADD,ITRLP
2112 002566 062767 000002 176130      ADD      #2,ITRLP        ;SET ITERATION POINTER
2113 002574 000721          BR      TSCD1
2114
2115      ;      *****
2116
2117 002576 000240          TEND:  NOP
2118 002600 005767 176012      TST      AUTOM          ;AUTOMATIC MODE?
2119 002604 001402          BEQ      TENDO           ;BRANCH - IF NO
2120 002606 000167 177456      JMP      NXTSLV          ;GET ANOTHER SLAVE DEVICE
2121
2122
2123 002612          TENDO:
2124
2125      ;      *****
2126
2127 002612 012704 027554          MOV      #MSG41,R4
2128 002616 004767 021700          JSR      PC,TTOUT        ;PRINT END OF PASS
2129 002622 016703 176200          MOV      PCNTR,R3
```

```
2130 002626 004767 022032 JSR PC,OCTP ;PRINT PASS NUMBER
2131
2132 : *****
2133 : AUTOMATIC MODE END OF PASS
2134 : *****
2135
2136 002632 005767 176172 TST PAFLG ;PASS INDICATOR SET?
2137 002636 001002 BNE 3$ ;BRANCH - IF YES
2138 002640 005267 176164 INC PAFLG ;SET PASS INDICATOR
2139 002644 013704 000042 3$: MOV @#42,R4 ;CONTENTS OF 42 TO R4
2140 002650 001405 BEQ HERE ;BRANCH - IF NOT AUTO MODE
2141 002652 000005 RESET ;CLEAR THE WORLD
2142 002654 004714 $ENDAD: JSR PC,(R4) ;RETURN TO MONITOR
2143 002656 000240 NOP
2144 002660 000240 NOP
2145 002662 000240 NOP
2146 002664 HERE:
2147 002664 005767 175726 TST AUTOM ;CHECK FOR AUTOM MODE
2148 002670 001006 BNE TENDX ;BRANCH - IF YES
2149 002672 032777 004000 175672 BIT #4000,@SWR ;SEE IF HALT ON PASS
2150 002700 001002 BNE TENDX ;IF NOT: BR
2151 002702 004767 022732 JSR PC,STOP
2152 002706 012767 000001 176110 TENDX: MOV #1,SKAT ;SET SKIP ADDRESS TEST FLAG
2153 002714 005267 176106 INC PCNTR ;BUMP PASS COUNTER
2154 002720 000167 177154 JMP TSCD ;RESTART
```

```

2156                                     ;LOGIC TEST 1: DRIVE ADDRESSING*****
2157
2158 002724 005767 176074          LT1:  TST      SKAT          ;SEE IF SKIP ADDRESS TESTS
2159 002730 001403                    BEQ      LT1G0         ;IF NOT: BR
2160 002732 005767 176012          TST      STFLG        ;SEE IF SINGLE TEST
2161 002736 001504                    BEQ      LT1X         ;IF NOT: BR
2162 002740 012704 026225          LT1G0:  MOV      #MSG2A,R4
2163 002744 004767 021552          JSR      PC,TTOUT     ;PRINT TEST INSTRUCTIONS
2164 002750 012767 030320 175554  LT1G:  MOV      #MSLT1,EMADDR ;SET HEADER ADDRESS
2165 002756 012704 026206          MOV      #MSG2,R4
2166 002762 004767 021534          JSR      PC,TTOUT     ;REQUEST DRIVE NUMBER
2167 002766 012705 000634          MOV      #DRVN,R5
2168 002772 012701 000001          MOV      #1,R1
2169 002776 012702 000007          MOV      #7,R2
2170 003002 012703 000000          MOV      #0,R3
2171 003006 004767 021252          JSR      PC,TTR
2172 003012 005767 175666          TST      TEMP1
2173 003016 001454                    BEQ      LT1X
2174 003020 005001                    CLR      R1
2175 003022 012700 000010          MOV      #10,R0
2176 003026 012777 000040 175464  LT1A:  MOV      #40,@CS
2177 003034 010177 175460          MOV      R1,@CS
2178 003040 005777 175444          TST      @C1
2179 003044 032777 010000 175446  BIT      #10000,@CS
2180 003052 001005                    BNE      LT1B
2181 003054 026701 175554          CMP      DRVN,R1
2182 003060 001407                    BEQ      LT1C
2183 003062 000167 000022          JMP      LT1ER
2184 003066 026701 175542          LT1B:  CMP      DRVN,R1
2185 003072 001002                    BNE      LT1C
2186 003074 000167 000020          JMP      LT1ER1
2187 003100 005300          LT1C:  DEC      R0
2188 003102 001722                    BEQ      LT1G
2189 003104 005201                    INC      R1
2190 003106 000747                    BR
2191 003110 012767 000001 175610  LT1ER:  MOV      #1,EXFL
2192 003116 000403                    BR      LT1ER2
2193 003120 012767 000002 175600  LT1ER1:  MOV      #2,EXFL
2194 003126 012767 026354 175546  LT1ER2:  MOV      #MSG3,ERADD
2195 003134 012767 003026 175560          MOV      #LT1A,SCOLP
2196 003142 004767 017316          JSR      PC,LTGER
2197 003146 000754                    BR
2198 003150 000167 177276          LT1X:  JMP      TSCD2
2199

```

```

2201                                     ;LOGIC TEST 2: REGISTER ADDRESSING*****
2202
2203 003154 000240                      LT2:  NOP
2204 003156 012777 000040 175334      LT2IT: MOV    #40,@CS          ;INIT
2205 003164 016777 175444 175326      MOV    DRVN,@CS          ;SELECT DRIVE
2206 003172 012767 030374 175432      MOV    #MSLT2,EMADDR    ;SAVE LT2 HEADER ADDRESS
2207 003200 012705 000510              MOV    #C1,R5           ;SET ADDRESS OF FIRST REGISTER
2208 003204 012700 000016              MOV    #16,R0          ;SET NUMBER OF REGISTERS
2209 003210 012702 000636              MOV    #TR00,R2        ;SET START OF REGISTER BUFFER
2210 003214 011501                      LT2A: MOV    (R5),R1
2211 003216 011112                      MOV    (R1),(R2)        ;READ REGISTER
2212 003220 032777 020000 175262      BIT    #20000,@C1      ;SEE IF ERROR
2213 003226 001402                      BEQ    LT2B             ;IF NOT: BR
2214 003230 004767 000024              JSR    PC,LT2ER1        ;ELSE GO TO ERROR 1
2215 003234 032777 000002 175262      LT2B: BIT    #2,@ER      ;SEE IF ILR
2216 003242 001402                      BEQ    LT2C             ;IF NOT: BR
2217 003244 004767 000026              JSR    PC,LT2ER2        ;ELSE GO TO ERROR 2
2218 003250 022225                      LT2C: CMP    (R2)+,(R5)+ ;BUMP ADDRESS
2219 003252 005300                      DEC    R0
2220 003254 001357                      BNE    LT2A             ;CONTINUE FOR ALL REGISTERS
2221 003256 000435                      BR     LT2X
2222 003260 012767 000002 175440      LT2ER1: MOV   #2,EXFL    ;FLAG NOT EXPECTED
2223 003266 012767 026376 175406      MOV   #MSG4,ERADD      ;POINT TO CONTROLLER ERROR
2224 003274 000415                      BR     LT2ERG           ;GO TO ERROR
2225 003276 012767 000002 175422      LT2ER2: MOV   #2,EXFL    ;FLAG NOT EXPECTED
2226 003304 012767 026414 175370      MOV   #MSG5,ERADD      ;POINT TO DRIVE ERROR
2227 003312 000406                      BR     LT2ERG           ;GO TO ERROR
2228 003314 012767 000001 175404      LT2ER3: MOV   #1,EXFL    ;FLAG EXPECTED
2229 003322 012767 026376 175352      MOV   #MSG4,ERADD      ;POINT TO DRIVE
2230 003330 012767 003344 175364      LT2ERG: MOV   #LT2LP,SCOLP ;SET SCOPE ADDRESS
2231 003336 004767 017122              JSR   PC,LTGER         ;GO PRINT
2232 003342 000207                      RTS   PC               ;ELSE CONTINUE
2233 003344 005726                      LT2LP: TST   (SP)+      ;RESET STACK
2234 003346 000167 177642              JMP   LT2A             ;LOOP
2235 003352 004767 020520              LT2X: JSR   PC,ITER     ;GO SEE IF ITERATIONS
2236 003356 000167 177070              JMP   TSCD2           ;RETURN TO SCHED

```

```
2238 ;LOGIC TEST 3: CONTROL BUS*****
2239
2240 003362 000240 LT3: NOP
2241 003364 012767 030453 175240 LT3IT: MOV #MSLT3,EMADDR ;SET TEST HEADER
2242 003372 012701 000001 MOV #1,R1 ;PRESET PATTERN 1
2243 003376 012700 000020 MOV #20,R0 ;SET PATTERN CHANGE NUMBER
2244 003402 004767 C20552 LT3A: JSR PC,INIT1 ;GO INIT
2245 003406 010177 175104 MOV R1,@FC ;WRITE TO FC
2246 003412 032777 000010 175104 BIT #10,@ER ;SEE IF CPAR (TM02)
2247 003420 001013 BNE LT3ER1 ;IF SO: BR
2248 003422 017702 175070 LT3B: MOV @FC,R2 ;READ FC
2249 003426 032777 020000 175054 BIT #20000,@C1 ;SEE IF MCPE (RH)
2250 003434 001020 BNE LT3ER2 ;IF SO: BR
2251 003436 005300 LT3C: DEC R0 ;SEE IF DONE PATTERN CHANGES
2252 003440 001427 BEQ LT3X ;IF SO: BR
2253 003442 000241 CLC
2254 003444 006101 ROL R1 ;CHANGE PATTERN
2255 003446 000755 BR LT3A ;CONTINUE
2256 003450 012767 026705 175224 LT3ER1: MOV #MSG11,ERADD ;SET ERROR CODE
2257 003456 012767 003402 175236 MOV #LT3A,SCOLP ;SET SCOPE ADDRESS
2258 003464 017702 175026 MOV @FC,R2 ;GET DATA
2259 003470 004767 020130 JSR PC,LTGER1 ;GO DO ERROR
2260 003474 000752 BR LT3B
2261 003476 012767 026661 175176 LT3ER2: MOV #MSG10,ERADD ;SET ERROR CODE
2262 003504 012767 003422 175210 MOV #LT3B,SCOLP ;SET SCOPE ADDRESS
2263 003512 004767 020106 JSR PC,LTGER1 ;GO DO ERROR
2264 003516 000747 BR LT3C
2265 003520 105701 LT3X: TSTB R1 ;SEE IF DONE PATTERN 2
2266 003522 100405 BMI LT3XX ;IF SO: BR
2267 003524 012701 000401 MOV #401,R1 ;SET PATTERN 2
2268 003530 012700 000010 MOV #10,R0 ;SET PATTERN CHANGE NUMBER
2269 003534 000722 BR LT3A ;DO PATTERN 2
2270 003536 004767 020334 LT3XX: JSR PC,ITER ;GO SEE IF ITERATIONS
2271 003542 000167 176704 JMP TSCD2 ;RETURN TO SCHEDULAR
```



```
2273
2274
2275
2276 003546 005767 175252 LT4: TST SKAT ;SEE IF SKIP ADDRESS TESTS
2277 003552 001403 BEQ LT4G0 ;IF NOT: BR
2278 003554 005767 175170 TST STFLG ;SEE IF SINGLE TEST
2279 003560 001517 BEQ LT4X ;IF NOT: BR
2280 003562 012704 026510 LT4G0: MOV #MSG8A,R4
2281 003566 004767 020730 JSR PC,TTOUT ;PRINT TEST INSTRUCTIONS
2282 003572 012704 026471 LT4G: MOV #MSG8,R4
2283 003576 004767 020720 JSR PC,TTOUT ;REQUEST SLAVE
2284 003602 012705 000674 MOV #SLVN,R5
2285 003606 012701 000001 MOV #1,R1
2286 003612 012702 000007 MOV #7,R2
2287 003616 012703 000000 MOV #0,R3
2288 003622 004767 020436 JSR PC,TTR ;GET SLAVE NUMBER
2289 003626 005767 175052 TST TEMP1 ;SEE IF SLAVE
2290 003632 001472 BEQ LT4X ;IF NOT: BR
2291 003634 005001 CLR R1 ;SELECT SLAVE 0
2292 003636 012700 000010 MOV #10,R0 ;SET NUMBER OF SLAVES
2293 003642 012777 000040 174650 LT4A: MOV #40,@CS ;INIT
2294 003650 016777 174760 174642 MOV DRVN,@CS ;SELECT DRIVE
2295 003656 010177 174660 MOV R1,@TC ;SELECT SLAVE
2296 003662 017703 174650 MOV @DT,R3 ;GET DT
2297 003666 020167 175002 CMP R1,SLVN ;SEE IF SHOULD HAVE SPR
2298 003672 001405 BEQ LT4B ;IF SO: BR
2299 003674 032703 002000 BIT #2000,R3 ;SEE IF SPR
2300 003700 001417 BEQ LT4D ;IF NOT: BR
2301 003702 000167 000044 JMP LT4ER1 ;GO TO ERROR 1
2302 003706 032703 002000 LT4B: BIT #2000,R3 ;SEE IF SPR
2303 003712 001002 BNE LT4C ;IF SO: BR
2304 003714 000167 000042 JMP LT4ER2 ;ELSE GO TO ERROR
2305 003720 012704 027376 LT4C: MOV #MSG30,R4
2306 003724 004767 020572 JSR PC,TTOUT ;PRINT SERIAL NUMBER TAG
2307 003730 017703 174604 MOV @SN,R3
2308 003734 004767 021576 JSR PC,SNPT ;PRINT SERIAL NUMBER
2309 003740 005300 LT4D: DEC R0
2310 003742 001713 BEQ LT4G ;IF DONE ALL: BR
2311 003744 005201 INC R1 ;BUMP SLAVE
2312 003746 000167 177670 JMP LT4A ;CONTINUE
2313 003752 012767 000001 174746 LT4ER1: MOV #1,EXFL ;FLAG EXPT: NOT RECEIVED
2314 003760 000403 BR LT4ERG
2315 003762 012767 000002 174736 LT4ER2: MOV #2,EXFL ;FLAG RECVD: NOT EXPT
2316 003770 012767 030535 174634 LT4ERG: MOV #MSLT4,EMADDR ;SET LT4 HEADER
2317 003776 012767 026637 174676 MOV #MSG9,ERADD ;SET ERROR CONDITION
2318 004004 012767 003642 174710 MOV #LT4A,SCOLP ;SET SCOPE ADDRESS
2319 004012 004767 016446 JSR PC,LTGER ;GO TO ERROR
2320 004016 000750 BR LT4D ;IF NO SCOPE: BR
2321 004020 000167 176426 LT4X: JMP TSCD2 ;RETURN TO SCHED
2322
```

```
2324 ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST*****
2325
2326 004024 012767 030614 174600 LT5: MOV #MSLT5,EMADDR ;SET TEST HEADER
2327 004032 004767 020122 LT5IT: JSR PC,INIT1 ;GO INIT
2328 004036 012700 000032 MOV #32,R0 ;SET LOOP FOR BITS 4-0
2329 004042 005001 CLR R1 ;SET TEST WORD
2330 004044 010177 174464 LT5A: MOV R1,@MR ;SEND TEST WORD TO MR
2331 004050 017702 174460 MOV @MR,R2 ;READ MR
2332 004054 042702 177740 BIC #177740,R2 ;MASK BITS 4-0
2333 004060 020102 CMP R1,R2 ;SEE IF EXPT = RECVD
2334 004062 001402 BEQ LT5B ;IF SO: BR
2335 004064 000167 000056 JMP LT5ER1 ;ELSE GO TO ERROR 1
2336 004070 005300 LT5B: DEC R0
2337 004072 001402 BEQ LT5C ;IF DONE LOOP: BR
2338 004074 005201 INC R1 ;BUMP TEST WORD
2339 004076 000762 BR LT5A ;CONTINUE LOOP
2340 004100 012701 000015 LT5C: MOV #15,R1 ;SET TEST WORD + WAM 3
2341 004104 012700 001000 MOV #1000,R0 ;SET LOOP FOR BITS 15-7
2342 004110 010177 174420 LT5D: MOV R1,@MR ;LOAD MR
2343 004114 017702 174414 MOV @MR,R2 ;READ MR
2344 004120 042702 000140 BIC #140,R2 ;MASK OUT BITS 5,6
2345 004124 020102 CMP R1,R2 ;SEE IF EXPT = RECVD
2346 004126 001402 BEQ LT5E ;IF SO: BR
2347 004130 000167 000036 JMP LT5ER2 ;ELSE GO TO ERR 2
2348 004134 005300 LT5E: DEC R0
2349 004136 001427 BEQ LT5X ;IF DONE LOOP: BR
2350 004140 062701 000200 ADD #200,R1 ;BUMP TEST WORD
2351 004144 000761 BR LT5D ;CONTINUE LOOP
2352 004146 012767 026750 174526 LT5ER1: MOV #MSG14,ERADD ;SET ERROR CODE
2353 004154 012767 004044 174540 MOV #LT5A,SCOLP ;SET SCOPE ADDRESS
2354 004162 004767 017436 JSR PC,LTGER1 ;GO TO ERROR
2355 004166 000167 177676 JMP LT5B ;CONTINUE
2356 004172 012767 026765 174502 LT5ER2: MOV #MSG15,ERADD ;SET ERROR CODE
2357 004200 012767 004110 174514 MOV #LT5D,SCOLP ;SET SCOPE ADDRESS
2358 004206 004767 017412 JSR PC,LTGER1 ;GO TO ERROR
2359 004212 000167 177716 JMP LT5E ;CONTINUE
2360 004216 004767 017654 LT5X: JSR PC,ITER ;GO SEE IF ITERATIONS
2361 004222 000167 176224 IMP TSCD2 ;RETURN TO SCHED
2362
```

```
2364 ;LOGIC TEST 6: TC REGISTER BIT TEST*****
2365
2366 004226 000240          LT6:  NOP
2367 004230 012767 030660 174374 LT6IT: MOV    #MSLT6,EMADDR ;POINT TO LT6 HEADER
2368 004236 012700 000003          LT6IT: MOV    #3,R0 ;SET NUMBER OF TESTS
2369 004242 005001          LT6A1: CLR
2370 004244 004767 017710          LT6A: JSR    PC,INIT1 ;GO INIT
2371 004250 010177 174266          LT6B: MOV    R1,@TC ;WRITE TC
2372 004254 017702 174262          LT6B: MOV    @TC,R2 ;READ TC
2373 004260 042702 160000          LT6B: BIC    #160000,R2 ;MASK OUT TCW
2374 004264 020102          LT6B: CMP    R1,R2 ;SEE IF EXPT = RECVD
2375 004266 001402          LT6B: BEQ    LT6C ;IF SO: BR
2376 004270 000167 000036          LT6B: JMP    LT6ER1 ;ELSE GO TO ERROR
2377 004274 032777 020000 174240 LT6C: BIT    #20000,@TC ;SEE IF TCW SET
2378 004302 001002          LT6C: BNE    LT6D ;IF SO: BR
2379 004304 000167 000046          LT6C: JMP    LT6ER2 ;ELSE GO TO ERROR
2380 004310 005300          LT6D: DEC    R0
2381 004312 001446          LT6D: BEQ    LT6X ;IF DONE ALL: BR
2382 004314 022700 000001          LT6D: CMP    #1,R0 ;SEE IF RESET TEST
2383 004320 001750          LT6D: BEQ    LT6A1 ;IF SO: BR
2384 004322 012701 017777          LT6D: MOV    #17777,R1 ;SET TEST WORD
2385 004326 000167 177712          LT6D: JMP    LT6A ;DC SET TEST
2386 004332 012767 027027 174342 LT6ER1: MOV    #MSG18,ERADD ;SET ERROR CODE
2387 004340 012767 004250 174354 LT6ER1: MOV    #LT6B,SCOLP ;SET SCOPE ADDRESS
2388 004346 004767 017252          LT6ER1: JSR    PC,LTGER1 ;GO TO ERROR
2389 004352 000167 177716          LT6ER1: JMP    LT6C ;CONTINUE
2390 004356 012767 027013 174316 LT6ER2: MOV    #MSG17,ERADD ;SET ERROR CODE
2391 004364 010167 174324          LT6ER2: MOV    R1,SAV1 ;SAVE R1
2392 004370 012701 000001          LT6ER2: MOV    #1,R1 ;SET EXPT
2393 004374 005002          LT6ER2: CLR    R2 ;SET RECVD 0
2394 004376 012767 004420 174316 LT6ER2: MOV    #LT6ER4,SCOLP ;SET SCOPE ADDRESS
2395 004404 004767 017214          LT6ER2: JSR    PC,LTGER1 ;GO TO ERROR
2396 004410 016701 174300          LT6ER2: MOV    SAV1,R1
2397 004414 000167 177670          LT6ER2: JMP    LT6D ;ELSE CONTINUE
2398 004420 005077 174116          LT6ER4: CLR    @TC ;WRITE TO TC
2399 004424 000167 177644          LT6ER4: JMP    LT6C ;LOOP ON ERROR
2400 004430 004767 017442          LT6X: JSR    PC,ITER ;GO SEE IF ITERATIONS
2401 004434 000167 176012          LT6X: JMP    TSCD2 ;RETURN TO SCHED
2402
```

```
2404                                     ;LOGIC TEST 7: FRAME COUNT BIT TEST*****
2405
2406 004440 000240                      LT7:  NOP
2407 004442 012700 000003              LT7IT: MOV    #3,R0          :SET TEST NUMBER
2408 004446 012767 030724 174156      LT7C:  MOV    #MSLT7,EMADDR :SET TEST HEADER
2409 004454 005001                      CLR    R1                 :SET TEST WORD
2410 004456 004767 017476              LT7A:  JSR    PC,INIT1     :GO INIT
2411 004462 010177 174030              MOV    R1,@FC            :CLEAR FRAME COUNT
2412 004466 017702 174024              MOV    @FC,R2            :READ FC
2413 004472 020102                      CMP    R1,R2             :SEE IF EXPT = RECDV
2414 004474 001402                      BEQ    LT7B               :IF SO: BR
2415 004476 000167 000022              JMP    LT7ER1            :ELSE GO TO ERROR
2416 004502 005300                      LT7B:  DEC    R0           :SEE IF DONE ALL
2417 004504 001421                      BEQ    LT7X              :IF SO: BR
2418 004506 022700 000001              CMP    #1,R0             :SEE IF RESET TEST
2419 004512 001755                      BEQ    LT7C              :IF SO: BR
2420 004514 012701 177777              MOV    #-1,R1            :SET TEST WORD TO -1
2421 004520 000167 177732              JMP    LT7A              :CONTINUE
2422 004524 012767 027046 174150      LT7ER1: MOV    #MSG19,ERADD  :SET ERROR CODE
2423 004532 012767 004456 174162      MOV    #LT7A,SCOLP       :SET SCOPE ADDRESS
2424 004540 004767 017060              JSR    PC,LTGER1        :GO PRINT ERROR
2425 004544 000167 177732              JMP    LT7B              :ELSE CONTINUE
2426 004550 012700 000003              LT7X:  MOV    #3,R0       :RESET TEST AMT
2427 004554 004767 017316              JSR    PC,ITER           :GO SEE IF ITERATIONS
2428 004560 000167 175666              JMP    TSCD2            :RETURN TO SCHED
2429
```



```
2458
2459 ;LOGIC TEST 11: GO BIT SET RESET*****
2460
2461 004720 000240 LT11: NOP
2462 004722 012767 031043 173702 LT11IT: MOV #MSLT11,EMADDR ;SET TEST HEADER
2463 004730 004767 017224 JSR PC,INIT1 ;GO INIT
2464 004734 017702 173550 MOV @C1,R2 ;READ C1
2465 004740 032702 000001 BIT #1,R2 ;SEE IF GO=0
2466 004744 001402 BEQ LT11B ;IF SO: BR
2467 004746 000167 000066 JMP LT11E1 ;ELSE GO TO ERROR 1
2468 004752 012777 000015 173554 LT11B: MOV #15,@MR ;SELECT WAM 3
2469 004760 005077 173532 CLR @FC ;ASSURE FCS = 1
2470 004764 052777 001700 173550 BIS #1700,@TC ;ASSURE FMT OK
2471 004772 012777 000071 173510 MOV #71,@C1 ;SET READ+GO
2472 005000 017702 173504 MOV @C1,R2 ;READ C1
2473 005004 032702 000001 BIT #1,R2 ;SEE IF GO =1
2474 005010 001002 BNE LT11C ;IF SO: BR
2475 005012 000167 000054 JMP LT11E2 ;ELSE GO TO ERROR 2
2476 005016 004767 017136 LT11C: JSR PC,INIT1 ;GO INIT
2477 005022 017702 173462 MOV @C1,R2 ;READ C1
2478 005026 032702 000001 BIT #1,R2 ;SEE IF GO=0
2479 005032 001447 BEQ LT11X ;IF SO:BR
2480 005034 000167 000064 JMP LT11E3 ;ELSE GO TO ERROR 3
2481 005040 012767 027117 173634 LT11E1: MOV #MSG21,ERADD ;SET ERROR CODE
2482 005046 012702 000001 MOV #1,R2 ;SET REVD
2483 005052 005001 CLR R1 ;SET EXPT
2484 005054 012767 004722 173640 MOV #LT11IT,SCOLP ;SET SCOPE ADDRESS
2485 005062 004767 016536 JSR PC,LTGER1 ;GO PRINT ERROR
2486 005066 000167 177660 JMP LT11B ;ELSE CONTINUE
2487 005072 012767 027155 173602 LT11E2: MOV #MSG22,ERADD ;SET ERROR CODE
2488 005100 005002 CLR R2 ;SET RCVD
2489 005102 012701 000001 MOV #1,R1 ;SET EXPT
2490 005106 012767 004752 173606 MOV #LT11B,SCOLP ;SET SCOPE ADDRESS
2491 005114 004767 016504 JSR PC,LTGER1 ;GO PRINT ERROR
2492 005120 000167 177672 JMP LT11C ;ELSE CONTINUE
2493 005124 012767 027176 173550 LT11E3: MOV #MSG23,ERADD ;SET ERROR CODE
2494 005132 005001 CLR R1 ;SET EXPT
2495 005134 012702 000001 MOV #1,R2 ;SET RCVD
2496 005140 012767 005016 173554 MOV #LT11C,SCOLP ;SET SCOPE ADDRESS
2497 005146 004767 016452 JSR PC,LTGER1 ;GO PRINT ERROR
2498 005152 004767 016720 LT11X: JSR PC,ITER ;GO SEE IF ITERATIONS
2499 005156 000167 175270 JMP TSCD2 ;RETURN TO SCHED
```

```
2501
2502
2503
2504 005162 000240
2505 005164 012767 031110 173440
2506 005172 004767 016762
2507 005176 032777 000200 173316
2508 005204 001002
2509 005206 000167 000062
2510 005212 012777 000015 173314
2511 005220 005077 173272
2512 005224 052777 001700 173310
2513 005232 012777 000071 173250
2514 005240 032777 000200 173254
2515 005246 001402
2516 005250 000167 000042
2517 005254 004767 016700
2518 005260 032777 000200 173234
2519 005266 001034
2520 005270 000167 000044
2521 005274 012767 027231 173400
2522 005302 012767 005164 173412
2523 005310 004767 016302
2524 005314 000736
2525 005316 012767 027257 173356
2526 005324 012767 005212 173370
2527 005332 004767 016260
2528 005336 000746
2529 005340 012767 027306 173334
2530 005346 012767 005254 173346
2531 005354 004767 016236
2532 005360 004767 016512
2533 005364 000167 175062

;LOGIC TEST 12: DRIVE READY BIT*****
LT12: NOP
LT12IT: MOV #MSLT12,EMADDR ;SET TEST HEADER
JSR PC,INIT1 ;GO INIT
BIT #200,@DS ;SEE IF DRY=1
BNE LT12B ;IF SO: BR
JMP LT12E1 ;ELSE GO TO ERROR 1
LT12B: MOV #15,@MR ;SET WAM3
CLR @FC ;ASSURE FCS = 1
BIS #1700,@TC ;ASSURE FMT OK
MOV #71,@C1 ;SET READ+GO
BIT #200,@DS ;SEE IF DRY=0
BEQ LT12C ;IF SO: BR
JMP LT12E2 ;ELSE GO TO ERROR 2
LT12C: JSR PC,INIT1 ;GO INIT
BIT #200,@DS ;SEE IF DRY=1
BNE LT12X ;IF SO: BR
JMP LT12E3 ;ELSE GO TO ERROR 3
LT12E1: MOV #MSG24,ERADD ;SET ERROR CODE
MOV #LT12IT,SCOLP ;SET SCOPE ADDRESS
JSR PC,LTGER2 ;GO TO ERROR
BR LT12B ;CONTINUE
LT12E2: MOV #MSG25,ERADD ;SET ERROR CODE
MOV #LT12B,SCOLP ;SET LOOP ADDRESS
JSR PC,LTGER2 ;GO PRINT ERROR
BR LT12C ;CONTINUE
LT12E3: MOV #MSG25A,ERADD ;SET ERROR CODE
MOV #LT12C,SCOLP ;SET ERROR LOOP
JSR PC,LTGER2 ;GET PRINT ERROR
LT12X: JSR PC,ITER ;GO TO ITERATION SUBROUTINE
JMP TSCD2 ;RETURN TO SCHED
```

```
2535
2536
2537
2538 005370 005000
2539 005372 012767 031161 173232
2540 005400 004767 016554
2541 005404 012767 005462 173266
2542 005412 005077 173072
2543 005416 005077 173146
2544 005422 052777 000100 173060
2545 005430 005300
2546 005432 001376
2547 005434 012777 000340 173126
2548 005442 012767 027333 173232
2549 005450 012767 005400 173244
2550 005456 004767 016134
2551 005462 004767 016410
2552 005466 000167 174760

;LOGIC TEST 13: INTERRUPT TEST*****
LT13: CLR R0
MOV #MSLT13,EMADDR ;SET TEST HEADER
LT13IT: JSR PC,INIT1 ;GO INIT,SELECT DRIVE, SELECT ABOVE
MOV #LT13X,RTRN ;SET RETURN ADDRESS
CLR @C1 ;CLEAR CS1
CLR @PSW ;SET PRIORITY
BIS #100,@C1 ;BIT SET IE
LT13A: DEC R0
BNE LT13A ;AWAIT INTERRUPT
LT13E1: MOV #340,@PSW ;RESET PRIORITY
MOV #MSG26,ERADD ;SET ERROR CODE
MOV #LT13IT,SCOLP ;SET LOOP ADDRESS
LT13X: JSR PC,LTGER2 ;GO PRINT ERROR
JSR PC,ITER ;GO TO ITERATION SUBROUTINE
JMP TSCD2 ;RETURN TO SCHED
```


2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589

:THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS.
:THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TU16
:CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.

:LOGIC TEST 14: STATUS AT BOT ON LINE, LOADED, NO WRITE RING*****

FOR AUTOMATIC MODE SKIP TESTS
WITH MANUAL INTERVENTION

LT14: TST AUTOM ;CHECK FOR AUTOMATIC MODE
BNE LT14XX ;BRANCH - IF YES

BIT #1000,BSUR ;SEE IF INHIB MAN TST
BNE LT14A ;IF NOT: BR
TST STFLG ;SEE IF SINGLE TEST
BEQ LT14XX ;IF NOT: BR
JMP INMT ;ELSE GO PRINT INHIB MSG
LT14A: MOV #MSLT14,EMADDR ;SET TEST HEADER
MOV #MSG1,R4 ;SET INSTRUCTION ONE
JSR PC,INST ;GO DO INSTRUCTION
LT14IT: JSR PC,INIT1 ;INIT, SELECT DRIVE + SLAVE
MOV #14602,R1 ;SET TEST WORD
MOV @DS,R2 ;ASSURE MOL,WRL,DPR,DRY,BOT
CMP R1,R2
BEQ LT14X ;IF SO: BR
MOV #LT14IT,SCOLP ;SET LOOP ADDRESS
MOV #MSG27,ERRADD ;SET ERROR CODE
JSR PC,LTGER1 ;GO PRINT ERROR
LT14X: JSR PC,ITER ;GO SEE IF ITERATION
LT14XX: JMP TSCD2 ;RETURN TO SCHED

```
2591                                     ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING*****
2592                                     ;
2593                                     ;*****
2594                                     ;
2595                                     ;
2596 005610                               LT15:
2597 005610 005767 173002                TST     AUTOM          ;AUTOMATIC MODE?
2598 005614 001042                BNE     LT15XX        ;BRANCH - IF YES
2599                                     ;
2600                                     ;*****
2601                                     ;
2602 005616 032777 001000 172746         BIT     #1000,@SWR    ;SEE IF INHIB MAN TST
2603 005624 001005                               BNE     LT15A        ;IF NOT: BR
2604 005626 005767 173116                TST     STFLG        ;SEE IF SINGLE TEST
2605 005632 001433                               BEQ     LT15XX        ;IF NOT: BR
2606 005634 000167 016300                JMP     INMT         ;ELSE GO PRINT INHIB MSG
2607 005640 012767 031274 172764 LT15A: MOV     #MSLT15,EMADDR ;SET TEST HEADER
2608 005646 012704 034245                MOV     #MSG2,R4
2609 005652 004767 016326                JSR     PC,INST     ;PRINT INSTRUCTION
2610 005656 004767 016304 LT15IT: JSR    PC,INIT2 ;GO INIT, SELECT DRIVE, SLAV
2611 005662 012701 100700                MOV     #100700,R1  ;SET TEST WORD
2612 005666 017702 172630                MOV     @DS,R2     ;READ STATUS
2613 005672 020102                               CMP     R1,R2      ;SEE OF EXPT=RCVD
2614 005674 001410                               BEQ     LT15X
2615 005676 012767 005656 173016         MOV     #LT15IT,SCOLP ;SET LOOP ADDRESS
2616 005704 012767 027362 172770         MOV     #MSG27,ERADD ;SET ERROR CODE
2617 005712 004767 015706                JSR     PC,LTGER1   ;GO PRINT ERROR
2618 005716 004767 016154 LT15X: JSR    PC,ITER  ;GO SEE IF ITERATIONS
2619 005722 000167 174524 LT15XX: JMP    TSCD2  ;RETURN TO SCHED
```

```
2621                                     ;LOGIC TEST 16: STATUS AT EOT, OFFLINE LOADED, NO WRITE RING*****
2622                                     ;
2623                                     ;*****
2624                                     ;
2625                                     ;
2626 005726                               LT16: TST     AUTOM      ;AUTO MODE?
2627 005726 005767 172664                BNE     LT16XX    ;BRANCH - IF YES
2628 005732 001042                        ;
2629                                     ;*****
2630                                     ;
2631                                     ;
2632 005734 032777 001000 172630          BIT     #1000,@SWR ;SEE IF INHIB MAN TST
2633 005742 001005                        BNE     LT16A     ;IF NOT: BR
2634 005744 005767 173000                TST     STFLG     ;SEE IF SINGLE TEST
2635 005750 001433                        BEQ     LT16XX    ;IF NOT: BR
2636 005752 000167 016162                JMP     INMT      ;ELSE GO PRINT INHIB MSG
2637 005756 012767 031342 172646 LT16A: MOV     #MSLT16,EMADDR ;SET TEST HEADER
2638 005764 012704 034266                MOV     #MMSG3,R4
2639 005770 004767 016210                JSR     PC,INST   ;GO PRINT INSTRUCTION
2640 005774 004767 016166                LT16IT: JSR    PC,INIT2 ;SELECT DRIVE,SLAVE
2641 006000 012701 116701                MOV     #116701,R1 ;SET TEST WORD
2642 006004 017702 172512                MOV     @DS,R2    ;READ STATUS
2643 006010 020102                        CMP     R1,R2     ;SEE IF EXPT=RCVD
2644 006012 001410                        BEQ     LT16X     ;IF SO: BR
2645 006014 012767 005774 172700          MOV     #LI16IT,SCOLP ;SET LOOP ADDRESS
2646 006022 012767 027362 172652          MOV     #MSG27,ERADD ;SET ERROR CODE
2647 006030 004767 015570                JSR     PC,LTGER1 ;GO PRINT ERROR
2648 006034 004767 016036                LT16X: JSR     PC,ITER ;GO SEE IF ITERATION
2649 006040 000167 174406                LT16XX: JMP     TSCD2 ;RETURN TO SCHED
2650
```

```
2652
2653 ;LOGIC TEST 17: STATUS AT ON LINE, LOADED*****
2654
2655 ; *****
2656
2657 LT17:
2658 006044 005767 172546 TST AUTOM ;AUTOMATIC MODE?
2659 006050 001042 BNE LT17XX ;BRANCH - IF YES
2660
2661 ; *****
2662
2663 006052 032777 001000 172512 BIT #1000,@SWR ;SEE IF INHIB MAN TST
2664 006060 001005 BNE LT17A ;IF NOT: BR
2665 006062 005767 172662 TST STFLG ;SEE IF SINGLE TEST
2666 006066 001433 BEQ LT17XX ;IF NOT: BR
2667 006070 000167 016044 JMP INMT ;ELSE GO PRINT INHIB MSG
2668 006074 012767 031410 172530 LT17A: MOV #MSLT17,EMADDR ;SET TEST HEADER
2669 006102 012704 034324 MOV #MMSG4,R4
2670 006106 004767 016072 JSR PC,INST ;GO PRINT INSTRUCTION
2671 006112 004767 016050 LT17IT: JSR PC,INIT2 ;SELECT DRIVE, SLAVE
2672 006116 012701 110701 MOV #110701,R1 ;SET TEST WORD
2673 006122 017702 172374 MOV @DS,R2 ;READ STATUS
2674 006126 020102 CMP R1,R2 ;SEE IF EXPT=RCVD
2675 006130 001410 BEQ LT17X ;IF SO: BR
2676 006132 012767 006112 172562 MOV #LT17IT,SCOLP ;SET LOOP ADDRESS
2677 006140 012767 027362 172534 MOV #MSG27,ERADD ;SET ERROR CODE
2678 006146 004767 015452 JSR PC,LTGER1 ;YES PRINT ERROR
2679 006152 004767 015720 LT17X: JSR PC,ITER ;GO SEE IF ITERATIONS
2680 006156 000167 174270 LT17XX: JMP TSCD2 ;RETURN TO SCHED
```

```
2682 :THE FOLLOWING 11 TESTS WILL TEST ALL POSSIBLE ERROR BITS
2683 :BY FORCING THEIR CONDITIONS THROUGH VARIOUS ILLEGAL PROGRAMMING
2684 :SEQUENCES AND USING THE MAINTENANCE WILL MODES AVAILABLE WITH TMO2
2685 :FOR EACH ERROR CONDITION SET THE APPROPRIATE STATUS WILL BE
2686 :CHECKED. IE: ERR, ATA, SLA, SC ETC.
2687
2688 :LOGIC TEST 20: ILLEGAL FUNCTION (ILF)*****
2689
2690 006162 012767 031456 172442 LT20: MOV #MSLT20,EMADDR ;SET TEST HEADER
2691 006170 012767 006210 172524 MOV #LT20A,SCOLP ;SET LOOP ADDRESS
2692 006176 012700 000022 LT20IT: MOV #22,R0 ;SET NUMBER OF ILL CODES
2693 006202 012767 000546 172474 MOV #ILFT,TEMP1 ;POINT TO START IF TABLE
2694 006210 004767 015744 LT20A: JSR PC,INIT1 ;GO INIT, SELECT SLAVE + DRIVE
2695 006214 012777 177777 172270 MOV #-1,AWC ;SET WC= -1
2696 006222 012701 090001 MOV #1,R1 ;SET TEST WORD
2697 006226 117777 172452 172254 MOVB @TEMP1,@C1 ;SET ILL CODE
2698 006234 017702 172264 MOV @ER,R2 ;READ ER
2699 006240 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2700 006242 001011 BNE LT20B ;IF SO: BR
2701 006244 012767 034607 172430 MOV #TMS17,ERADD ;SET ERROR CODE
2702 006252 012767 000001 172446 MOV #1,EXFL ;SET EXPT FLG
2703 006260 004767 014172 JSR PC,LTGER0 ;GO PRINT ERROR
2704 006264 000404 BR LT20C
2705 006266 020102 LT20B: CMP R1,R2 ;SEE UNEXPECTED ERRORS
2706 006270 001402 BEQ LT20C ;IF NOT: BR
2707 006272 004767 014146 JSR PC,LTGER3 ;ELSE PRINT ERROR
2708 006276 005300 LT20C: DEC R0 ;SEE IF DONE ALL ILL CODES
2709 006300 001404 BEQ LT20X ;IF SO: BR
2710 006302 005267 172376 INC TEMP1 ;BUMP ADDRESS
2711 006306 000167 177676 JMP LT20A ;CONTINUE
2712 006312 004767 015560 LT20X: JSR PC,ITER ;GO SEE IF ITERATION
2713 006316 004767 014612 JSR PC,DRVCLR
2714 006322 000167 174124 JMP TSCD2 ;RETURN TO SCHED
```

```
2716  
2717  
2718  
2719 006326 012767 031535 172276 LT21: MOV #MSLT21,EMADDR ;SET TEST HEADER  
2720 006334: 012767 006342 172360 MOV #LT21IT,SCOLP ;SET SCOPE LOOP ADDRESS  
2721 006342 004767 015612 LT21IT: JSR PC,INIT1 ;GO INIT, SELECT SLAVE, DRIVE  
2722 006346 052777 000300 172166 BIS #300,@TC ;SET FORMAT  
2723 006354 012777 000015 172152 MOV #15,@MR ;SET WAM3  
2724 006362 012777 000071 172120 MOV #71,@C1 ;SET READ+GO  
2725 006370 005077 172122 CLR @FC ;ATTEMPT WRITE TO FC  
2726 006374 012701 000004 MOV #4,R1 ;SET TEST WORD  
2727 006400 017702 172120 MOV @ER,R2 ;GET ER  
2728 006404 030102 BIT R1,R2 ;SEE IF EXPT=RCVD  
2729 006406 001011 BNE LT21A ;IF SO: BR  
2730 006410 012767 034623 172264 MOV #TMS19,ERADD ;SET ERROR CODE  
2731 006416 012767 000001 172302 MOV #1,EXFL ;SET EXPT FLG  
2732 006424 004767 014026 'SR PC,LTGERO ;GO PRINT ERROR  
2733 006430 000404 JR LT21B  
2734 006432 020102 LT21A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
2735 006434 001402 BEQ LT21B ;IF NOT: BR  
2736 006436 004767 014002 JSR PC,LTGER3 ;ELSE GO PRINT ERROR  
2737 006442 004767 015430 LT21B: JSR PC,ITER ;GO SEE IF ITERATION  
2738 006446 012703 040000 MOV #40000,R3  
2739 006452 005303 LT21XA: DEC R3 ;DELAY FOR ALPHA  
2740 006454 001376 BNE LT21XA  
2741 006456 004767 013564 JSR PC,EORPA ;GO DO EOR CLEAR  
2742 006462 004767 014446 JSR PC,DRVCLR  
2743 006466 000167 173760 JMP TSCD2 ;RETURN TO SCHED
```

```
2745  
2746 ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)*****  
2747  
2748 006472 012767 031571 172132 LT22: MOV #MSLT22,EMADDR ;SET TEST HEADER  
2749 006500 012767 006506 172214 MOV #LT22IT,SCOLP ;SET SCOPE LOOP ADDRESS  
2750 006506 004767 015446 LT22IT: JSR PC,INIT1 ;INIT, SELECT SLAVE+DRIVE  
2751 006512 052777 000020 172000 BIS #20,@CS ;ENABLE EVEN PARITY ON MB  
2752 006520 012777 177777 171770 MOV #-1,@FC ;WRITE TO FC  
2753 006526 012701 000010 MOV #10,R1 ;SET TEST WORD  
2754 006532 042777 000020 171760 BIC #20,@CS ;RESET PARITY TO ODD  
2755 006540 017702 171760 MOV @ER,R2 ;GET ER  
2756 006544 030102 BIT R1,R2 ;SEE IF EXPT=RCVD  
2757 006546 001011 BNE LT22A ;IF SO: BR  
2758 006550 012767 034631 172124 MOV #TMS20,ERADD ;SET ERROR CODE  
2759 006556 012767 000001 172142 MOV #1,EXFL ;SET EXPT FLG  
2760 006564 004767 013666 JSR PC,LTGERO ;GO PRINT ERROR  
2761 006570 000404 BR LT22X  
2762 006572 020102 LT22A: CMP R1,R2 ;SEE IF UNEXPECTED ERROR  
2763 006574 001402 BEQ LT22X ;IF NOT: BR  
2764 006576 004767 013642 JSR PC,LTGER3 ;ELSE GO PRINT ERROR  
2765 006602 004767 015270 LT22X: JSR PC,ITER ;GO SEE IF ITERATION  
2766 006606 004767 014322 JSR PC,DRVCLR  
2767 006612 000167 173634 JMP TSCD2 ;RETURN TO SCHED
```

```
2769
2770
2771
2772 006616 012767 031626 172006 LT23: MOV #MSLT23,EMADDR ;SET TEST HEADER
2773 006624 012767 006632 172070 MOV #LT23IT,SCOLP ;SET SCOPE ADDRESS
2774 006632 004767 015322 LT23IT: JSR PC,INIT1 ;GO INIT SELECT DRIVE+SLAVE
2775 006636 042777 000360 171676 BIC #360,@TC ;SET ILLEGAL FORMAT
2776 006644 012701 000020 MOV #20,R1 ;SET TEST WORD
2777 006650 012777 000015 171656 MOV #15,@MR ;SET WAM 3
2778 006656 012777 000071 171624 MOV #71,@C1 ;SET READ+GO
2779 006664 017702 171634 MOV @ER,R2 ;READ ER
2780 006670 030102 BIT R1,R2 ;SEE IF EXPT=RCVD
2781 006672 001011 BNE LT23A ;IF SO: BR
2782 006674 012767 034640 172000 MOV #TMS21,ERADD ;SET ERROR CODE
2783 006702 012767 000001 172016 MOV #1,EXFL ;SET EXPT FLG
2784 006710 004767 013542 JSR PC,LTGERO ;GO PRINT ERROR
2785 006714 000404 BR LT23X
2786 006716 020102 LT23A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2787 006720 001402 BEQ LT23X ;IF NOT: BR
2788 006722 004767 013516 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2789 006726 004767 015144 LT23X: JSR PC,ITER ;GO SEE IF ITERATION
2790 006732 004767 013310 JSR PC,EORPA
2791 006736 004767 014172 JSR PC,DRVCLR
2792 006742 000157 173504 JMP TSCD2 ;RETURN TO SCHED
```


;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)*****

```
2794  
2795  
2796 006746 012767 031670 171656 LT24: MOV #MSLT24,EMADDR ;SET TEST HEADER  
2797 006754 012767 006762 171740 MOV #LT24IT,SCOLP ;SET SCOPE ADDRESS  
2798 006762 012767 000005 171620 LT24IT: MOV #5,ITAMT  
2799 006770 004767 015164 JSR PC,INIT1 ;GO INIT, SELECT DRIVE+SLAVE  
2800 006774 052777 000300 171540 BIS #300,@TC ;SET NORMAL FORMAT  
2801 007002 012777 035054 171504 MOV #WDATA,@BA ;SET BA  
2802 007010 012777 177760 171500 MOV #-20,@FC ;SET FC  
2803 007016 012777 177770 171466 MOV #-10,@WC ;SET WC  
2804 007024 012777 000013 171502 MOV #13,@MR ;SELECT WAM 2  
2805 007032 012777 000061 171450 MOV #61,@C1 ;SET WRITE+GO  
2806 007040 052777 000020 171452 BIS #20,@CS ;FORCE EVEN PARITY  
2807 007046 012701 000040 MOV #40,R1 ;SET TEST WORD  
2808 007052 012703 000004 MOV #4,R3  
2809 007056 005000 CLR R0  
2810 007060 005300 LT24A: DEC R0  
2811 007062 001376 BNE LT24A ;DELAY  
2812 007064 005303 DEC R3  
2813 007066 001374 BNE LT24A  
2814 007070 012700 000004 MOV #4,R0  
2815 007074 012777 000013 171432 LT24B: MOV #13,@MR ;CLOCK MR 4 TIMES  
2816 007102 005300 DEC R0  
2817 007104 022700 000002 CMP #2,R0 ;SEE IF DONE 1 BYTE  
2818 007110 001002 BNE LT24B0 ;IF NOT: BR  
2819 007112 017701 171416 MOV @MR,R1 ;ELSE GET BYTE 1  
2820 007116 005700 LT24B0: TST R0 ;SEE IF BYTE 2  
2821 007120 001365 BNE LT24B ;IF NOT: BR  
2822 007122 017704 171406 MOV @MR,R4 ;GET BYTE 2  
2823 007126 005000 CLR R0  
2824 007130 005300 LT24C: DEC R0  
2825 007132 001376 BNE LT24C ;DELAY  
2826 007134 032777 000040 171362 BIT #40,@ER ;SEE IF DPAR IS SET  
2827 007142 001023 BNE LT24D ;IF SO: BR  
2828 007144 000301 SWAB R1  
2829 007146 042701 177400 BIC #177400,R1 ;GET LOW BYTE  
2830 007152 02704 000377 BIC #377,R4  
2831 007156 050401 BIS R4,R1 ;GET HIGH BYTE  
2832 007160 005267 171570 INC T24FL ;SET T24 FLAG  
2833 007164 012767 034646 171510 MOV #TMS22,ERADD ;SET ERROR CODE  
2834 007172 012767 000001 171526 MOV #1,EXFL ;SET EXPT FLG  
2835 007200 004767 013252 JSR PC,LTGERO ;GO PRINT ERROR  
2836 007204 005067 171544 CLR T24FL ;CLEAR FLAG  
2837 007210 000412 BR LT24X  
2838 007212 012701 000050 LT24D: MOV #50,R1  
2839 007216 017702 171302 MOV @ER,R2 ;GET ERROR REGISTER  
2840 007222 042702 020000 BIC #20000,R2 ;MASK OPI  
2841 007226 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
2842 007230 001402 BEQ LT24X ;IF NOT: BR  
2843 007232 004767 013206 JSR PC,LTGER3 ;ELSE GO PRINT ERROR  
2844 007236 042777 000020 171254 LT24X: BIC #20,@CS ;RESET EVEN PARITY  
2845 007244 004767 012776 JSR PC,EORPA ;GO DO EOR CLEAR  
2846 007250 004767 013660 JSR PC,DRVCLR ;GO SEE IF DRIVE CLEAR OK  
2847 007254 004767 014616 JSR PC,ITER ;GO SEE IF ITERATION  
2848 007260 012767 000020 171322 MOV #20,ITAMT  
2849 007266 000167 173160 JMP TSCD2 ;RETURN TO SCHED
```

```
2851  
2852  
2853 ;LOGIC TEST 25: NON-EXECUTABLE FUNCTION(NEF)*****  
2854 007272 012767 031730 171332 LT25: MOV #MSLT25,EMADDR ;SET TEST HEADER  
2855 007300 012767 007306 171414 MOV #LT25IT,SCOLP ;SET LOOP ADDRESS  
2856 007306 004767 014646 LT25IT: JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE  
2857 007312 052777 000300 171222 BIS #300,@TC ;SET NORMAL FORMAT  
2858 007320 012777 177777 171170 MOV #-1,@FC ;SET ITLLEGAL FC  
2859 007326 012777 000013 171200 MOV #13,@MR ;SET WAM 2  
2860 007334 012777 000061 171146 MOV #61,@C1 ;LOAD WRITE+GO  
2861 007342 012701 004000 MOV #4000,R1 ;SET TEST WORD  
2862 007346 017702 171152 MOV @ER,R2 ;GET ER  
2863 007352 030102 BIT R1,R2 ;SEE IF EXPT=RCVD  
2864 007354 001011 BNE LT25A ;IF SO: BR  
2865 007356 012767 034734 171316 MOV #TMS31,ERADD ;SET ERROR CODE  
2866 007364 012767 000001 171334 MOV #1,EXFL ;SET EXPT FLAG  
2867 007372 004767 013060 JSR PC,LTGERO ;GO PRINT ERROR  
2868 007376 000404 BR LT25X  
2869 007400 020102 LT25A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
2870 007402 001402 RFN LT25X ;IF NOT: BR  
2871 007404 004767 013034 JSR PC,LTGER3 ;ELSE GO PRINT ERROR  
2872 007410 004767 014462 LT25X: JSR PC,ITER ;GO SEE IF ITERATION  
2873 007414 004767 013514 JSR PC,DRVCLR  
2874 007420 000167 173026 JMP TSCD2 ;RETURN TO SCHED
```

```

2876
2877
2878
2879 007424 012767 031764 171200 LT26: MOV #MSLT26,EMADDR ;SET TEST HEADER
2880 007432 004767 014522 LT26IT: JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE
2881 007436 005000 CLR R0
2882 007440 005300 LT26W: DEC R0
2883 007442 001376 BNE LT26W ;AWAIT OPI RESET
2884 007444 052777 000300 171070 BIS #300,@TC ;SET NORMAL FORMAT
2885 007452 012777 177770 171032 MOV #-10,@WC ;SET WC=-10
2886 007460 012777 177760 171030 MOV #-20,@FC ;SET FC=-20
2887 007466 012777 000013 171040 MOV #13,@MR ;SET WAM 3
2888 007474 012777 000061 171006 MOV #61,@C1 ;LOAD WRITE+GO
2889 007502 012701 001000 MOV #1000,R1 ;SET TEST WORD
2890 007506 005000 CLR R0
2891 007510 005300 LT26A: DEC R0
2892 007512 001376 BNE LT26A ;DELAY
2893 007514 012777 000025 171012 MOV #25,@MR ;LOAD MM EOR CLEAR
2894 007522 105077 171006 CLR @MR ;RESET MR
2895 007526 012703 000004 MOV #4,R3
2896 007532 005000 CLR R0
2897 007534 032777 001000 170762 LT26B: BIT #1000,@ER ;SEE IF FCE SET
2898 007542 001022 BNE LT26C ;IF SO: BR
2899 007544 005300 DEC R0
2900 007546 001372 BNE LT26B ;DELAY
2901 007550 005303 DEC R3
2902 007552 001370 BNE LT26B
2903 007554 017702 170744 MOV @ER,R2 ;GET ER
2904 007560 012767 007432 171134 MOV #LT26IT,SCOLP ;SET SCOPE ADDRESS
2905 007566 012767 034713 171106 MOV #TMS28,ERADD
2906 007574 012767 000001 171124 MOV #1,EXFL ;SET EXPT FLG
2907 007602 004767 012650 JSR PC,LTGERO ;GO PRINT ERROR
2908 007606 000406 BR LT26X
2909 007610 017702 170710 LT26C: MOV @ER,R2 ;GET ERROR REGISTER
2910 007614 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2911 007616 001402 BEQ LT26X ;IF NOT: BR
2912 007620 004767 012620 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2913 007624 004767 014246 LT26X: JSR PC,ITER ;GO SEE IF ITERATION
2914 007630 004767 013300 JSR PC,DRVCLR
2915 007634 000167 172612 JMP TSCD2 ;RETURN TO SCHED

```

```
2917  
2918  
2919  
2920 007640 022767 172400 170642 LT27:  CMP #172400,C1 ;SEE IF ADDRESSES OPEN  
2921 007646 001041 BNE LT27XX ;IF NOT: BR  
2922 007650 012767 007674 171044 MOV #LT27A,SCOLP ;SET SCOPE ADDRESS  
2923 007656 012767 032020 170746 MOV #MSLT27,EMADDR ;SET TEST HEADER  
2924 007664 012700 000020 LT27IT: MOV #20,R0 ;SET NUMBER OF ILR TESTS  
2925 007670 012701 172434 MOV #172434,R1 ;SET FIRST ILR ADDRESS  
2926 007674 004767 014260 LT27A: JSR PC,INIT1 ;GO INIT, SELECT DRIVE+SLAVE  
2927 007700 011103 MOV (R1),R3 ;ATTEMPT ILR READ  
2928 007702 032777 000002 170614 BIT #2,ER ;SEE IF ILR=1  
2929 007710 001010 BNE LT27B ;IF SO: BR  
2930 007712 012767 000001 171006 MOV #1,EXFL ;SET EXPT-NOT R'VD FLAG  
2931 007720 012767 034535 170754 MOV #TMS10,ERADD ;SET ERROR CODE  
2932 007726 004767 012532 JSR PC,LTGER ;GO PRINT ERROR  
2933 007732 005300 LT27B: DEC R0 ;SEE IF DONE ALL  
2934 007734 001402 BEQ LT27X ;IF SO: BR  
2935 007736 005721 TST (R1)+ ;BUMP ADDRESS  
2936 007740 000755 BR LT27A ;CONTINUE TESTS  
2937 007742 004767 014130 LT27X: JSR PC,ITER ;GO SEE IF ITERATIONS  
2938 007746 004767 013162 JSR PC,DRVCLR  
2939 007752 000167 172474 LT27XX: JMP TSCD2 ;RETURN TO SCHED
```

```

2941
2942
2943
2944 007756 012767 034742 170716 LT30: MOV #TMS32,ERADD ;SET ERROR CODE
2945 007764 012767 032054 170640 MOV #MSLT30,EMADDR ;SET TEST HEADER
2946 007772 012767 010000 170722 MOV #LT30IT,SCOLP ;SET SCOPE ADDRESS
2947 010000 004767 014154 LT30IT: JSR PC,INIT1 ;INIT, SELECT DRIVE + SLAVE
2948 010004 052777 000300 170530 BIS #300,@TC ;SET NORMAL FORMAT
2949 010012 012701 010000 MOV #10000,R1 ;SET TEST WORD
2950 010016 012777 000017 170510 MOV #17,@MR ;CRIPPLE OCCUPIED
2951 010024 005077 170466 CLR @FC ;SET FC3
2952 010030 012777 000061 170452 MOV #61,@C1 ;LOAD WRITE+GO
2953 010036 032777 010000 170460 BIT #10000,@ER ;SEE IF DTE SET
2954 010044 001005 BNE LT30A ;IF SO: BR
2955 010046 012767 000001 170652 MOV #1,EXFL ;SET EXPT FLG
2956 010054 004767 012376 JSR PC,LTGERO ;GO PRINT ERROR
2957 010060 004767 014074 LT30A: JSR PC,INIT1 ;GO INIT SELECT DRIVE,SLAVE
2958 010064 052777 000300 170450 BIS #300,@TC ;SET FORMAT
2959 010072 012701 010000 MOV #10000,R1 ;SET TEST WORD
2960 010076 005077 170414 CLR @FC ;SET FCS
2961 010102 012777 000015 170424 MOV #15,@MR ;SET WRAP 3
2962 010110 012777 000061 170372 MOV #61,@C1 ;LOAD WRITE+GO
2963 010116 012704 040000 MOV #40000,R4
2964 010122 005777 170414 LT30B: TST @TC ;SEE IF ALPHA
2965 010126 100015 BPL LT30C ;AWAIT ALPHA
2966 010130 005300 DEC R0
2967 010132 001373 BNE LT30B
2968 010134 016704 170472 MOV EMADDR,R4
2969 010140 004767 014356 JSR PC,TTOUT ;PRINT HEADER
2970 010144 012704 030116 MOV #MSG50,R4
2971 010150 004767 014346 JSR PC,TTOUT ;PRINT ALPHA ERROR
2972 010154 004767 013664 JSR PC,SCOPE
2973 010160 000435 BR LT30X
2974 010162 012777 000015 170344 LT30C: MOV #15,@MR ;CLOCK MR
2975 010170 012777 000015 170336 MOV #15,@MR ;CLOCK MR
2976 010176 005000 CLR R0
2977 010200 005300 LT30D: DEC R0
2978 010202 001376 BNE LT30D ;DELAY
2979 010204 032777 010000 170312 BIT #10000,@ER ;SEE IF DTE SET
2980 010212 001006 BNE LT30E ;IF SO: BR
2981 010214 012767 000001 170504 MOV #1,EXFL ;SET EXPT FLG
2982 010222 004767 012230 JSR PC,LTGERO ;GO PRINT ERROR
2983 010226 000412 BR LT30X
2984 010230 012701 010000 LT30E: MOV #10000,R1 ;SET TEST WORD
2985 010234 017702 170264 MOV @ER,R2 ;GET ERROR REGISTER
2986 010240 042702 020000 BIC #20000,R2 ;MASK OPI
2987 010244 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2988 010246 001402 BEQ LT30X ;IF NOT: BR
2989 010250 004767 012170 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2990 010254 004767 013616 LT30X: JSR PC,ITER ;GO SEE IF ITERATION
2991 010260 004767 011762 JSR PC,EORPA ;GO CLEAR GO BIT
2992 010264 004767 012644 JSR PC,DRVCLR
2993 010270 000167 172156 JMP TSCD2 ;RETURN TO SCHED
2994

```

```
2996  
2997  
2998  
2999 010274 012767 032112 170350 LT31: MOV #MSLT31,EMADDR ;SET TEST HEADER  
3000 010302 012767 010310 170412 MOV #LT31IT,SCOLP ;SET SCOPE ADDRESS  
3001 010310 012767 000005 170272 LT31IT: MOV #5,ITAMT ;SET REDUCED ITER COUNT  
3002 010316 004767 013636 JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE  
3003 010322 005000 CLR R0  
3004 010324 005300 LT31W: DEC R0  
3005 010326 001376 BNE LT31W ;AWAIT OPI RESET  
3006 010330 052777 000300 170204 BIS #300,@TC ;SET FORMAT  
3007 010336 012777 000013 170170 MOV #13,@MR ;SET WAM 2  
3008 010344 005077 170146 CLR @FC ;SET FCS  
3009 010350 012777 000061 170132 MOV #61,@C1 ;LOAD WRITE+GO  
3010 010356 012701 020000 MOV #20000,R1 ;SET TEST WORD  
3011 010362 012703 000004 MOV #4,R3  
3012 010366 005000 CLR R0  
3013 010370 032777 020000 170126 LT31A: BIT #20000,@ER ;SEE IF OPI SET  
3014 010376 001015 BNE LT31B ;IF SO: BR  
3015 010400 005300 DEC R0  
3016 010402 001372 BNE LT31A ;DELAY  
3017 010404 005303 DEC R3  
3018 010406 001370 BNE LT31A  
3019 010410 012767 034756 170264 MOV #TMS33A,ERADD ;SET ERROR CODE  
3020 010416 012767 000001 170302 MOV #1,EXFL ;SET EXPT FLG  
3021 010424 004767 012026 JSR PC,LTGERO ;GO PRINT ERROR  
3022 010430 000464 BR LT31X  
3023 010432 017702 170066 LT31B: MOV @ER,R2 ;GET ERROR REGISTER  
3024 010436 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
3025 010440 001403 BEQ LT31C ;IF NOT: BR  
3026 010442 004767 011776 JSR PC,LTGER3 ;ELSE PRINT ERROR  
3027 010446 000455 BR LT31X  
3028 010450 004767 013504 LT31C: JSR PC,INIT1 ;GO INIT  
3029 010454 005000 CLR R0  
3030 010456 005300 LT31W1: DEC R0  
3031 010460 001376 BNE LT31W1 ;AWAIT OPI RESET  
3032 010462 052777 000300 170052 BIS #300,@TC ;SET FORMAT  
3033 010470 012777 000015 170036 MOV #15,@MR ;SET WRAP 3  
3034 010476 012777 000071 170004 MOV #71,@C1 ;LOAD READ+GO  
3035 010504 012701 020000 MOV #20000,R1 ;SET TEST WORD  
3036 010510 012703 000100 MOV #100,R3  
3037 010514 005000 CLR R0  
3038 010516 032777 020000 170000 LT31D: BIT #20000,@ER ;SEE IF OPI SET  
3039 010524 001020 BNE LT31E ;IF SO: BR  
3040 010526 005300 DEC R0  
3041 010530 001372 BNE LT31D ;DELAY  
3042 010532 005303 DEC R3  
3043 010534 001370 BNE LT31D  
3044 010536 012767 010450 170156 MOV #LT31C,SCOLP ;SET SCOPE ADDRESS  
3045 010544 012767 034772 170130 MOV #TMS33B,ERADD ;SET ERROR CODE  
3046 010552 012767 000001 170146 MOV #1,EXFL ;SET EXPT FLG  
3047 010560 004767 011672 JSR PC,LTGERO ;GO PRINT ERROR  
3048 010564 000406 BR LT31X  
3049 010566 017702 167732 LT31E: MOV @ER,R2 ;GET ERROR REGISTER  
3050 010572 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
3051 010574 001403 BEQ LT31X ;IF NOT: BR
```

CZTUGO TMO2/TU16 CTRL LGC
CZTUCG.P11 12-JUL-83 10:41

MACY11 30(1046) 12-JUL-83 10:47 M 6
PAGE 69-1

SEQ 0077

3052	010576	004767	011642		JSR	PC,LTGER3	:ELSE PRINT ERROR
3053	010602	004767	013270		JSR	PC,ITER	:GO SEE IF ITERATIONS
3054	010606	004767	012322		JSR	PC,DRVCLR	
3055	010612	012767	000020	167770	MOV	#20,ITAMT	
3056	010620	000167	171626		JMP	TSCD2	:RETURN TO SCHED

```
3058  
3059  
3060  
3061 010624 012767 032146 170000 LT32: MOV #MSLT32,EMADDR ;SET TEST HEADER  
3062 010652 012767 010640 170062 MOV #LT32IT,SCOLP ;SET SCOPE ADDRESS  
3063 010640 004767 013314 LT32IT: JSR PC,INIT1 ;INIT, SELECT DRIVE +SLAVE  
3064 010644 016700 170024 MOV SLVN,R0 ;GET SLAVE NUMBER  
3065 010650 005100 COM R0 ;SET NONEXISTANT SLAVE  
3066 010652 042700 177770 BIC #177770,R0 ;MASK SLAVE NUMBER  
3067 010656 052700 000300 BIS #300,R0 ;SET FORMAT  
3068 010662 010077 167654 MOV R0,@TC ;SELECT ILLEGAL SLAVE  
3069 010666 012777 000071 167614 MOV #71,@C1 ;LOAD READ+GO  
3070 010674 012701 040000 MOV #40000,R1 ;SET TEST WORD  
3071 010700 017702 167620 MOV @ER,R2 ;READ ER  
3072 010704 030102 BIT R1,R2 ;SEE IF EXPT=RCVD  
3073 010706 001011 BNE LT32A ;IF SO: BR  
3074 010710 012767 035005 167764 MOV #TMS34,ERADD ;SET ERROR CODE  
3075 010716 012767 000001 170002 MOV #1,EXFL ;SET ERROR CODE  
3076 010724 004767 011526 JSR PC,LTGERO ;GO PRINT ERROR  
3077 010730 000404 BR LT32X  
3078 010732 020102 LT32A: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS  
3079 010734 001402 BEQ LT32X ;IF NOT: BR  
3080 010736 004767 011502 JSR PC,LTGER3 ;ELSE PRINT ERROR  
3081 010742 004767 013130 LT32X: JSR PC,ITER ;GO SEE IF ITERATIONS  
3082 010746 004767 012162 JSR PC,DRVCLR  
3083 010752 000167 171474 JMP TSCD2 ;RETURN TO SCHED
```


3085
3086
3087
3088
3089
3090
3091
3092
3093
3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105

010756	012767	032202	167646
010764	012767	010772	167730
010772	004767	013162	
010776	012777	000013	167530
011004	012777	177777	167504
011012	012777	000031	167470
011020	032777	020000	167474
011026	001010		
011030	012767	034565	167644
011036	012767	000001	167662
011044	004767	011406	
011050	004767	013022	
011054	000167	171372	

```
;THE FOLLOWING 6 TESTS WILL LOOK AT VARIOUS BITS IN THE  
;DRIVE STATUS(DS) AND TAPE CONTROL(TC)  
;REGISTERS BY FORCING CERTAIN CONDITONS WHICH DO NOT  
;REQUIRE TAPE MOVEMENT.  
  
;LOGIC TEST 33: POSITIONING IN PROGRESS(PIP)*****  
  
LT33:  MOV      #MSLT33,EMADDR  ;SET TEST HEADER  
      MOV      #LT33IT,SCOLP   ;SET SCOPE ADDRESS  
LT33IT: JSR     PC,INIT1       ;INIT, SELECT DRIVE+SLAVE  
      MOV      #13,@MR        ;SET WAM 2  
      MOV      #-1,@FC        ;SET FCS  
      MOV      #31,@C1        ;LOAD SPACE FORWARD+GO  
      BIT      #20000,@DS     ;SEE IF PIP=1  
      BNE     LT33X           ;IF SO: BR  
      MOV      #TMS14,ERADD    ;SET ERROR CODE  
      MOV      #1,EXFL        ;SET ERROR CODE  
      JSR     PC,LTGERO       ;GO PRINT ERROR  
LT33X: JSR     PC,ITER        ;GO SEE IF ITERATIONS  
      JMP     TSCD2          ;RETURN TO SCHED
```

```

3107
3108
3109
3110 011060 005767 167606      LT34:  TST      NRZOF      ;SEE IF NRZ ONLY
3111 011064 001054              BNE      LT34XX     ;IF SO: BR
3112 011066 012767 034505 167606  MOV      #TMS6,ERADD ;SET ERROR CODE
3113 011074 012767 032236 167530  MOV      #MSLT34,EMADDR ;SET TEST HEADER
3114 011102 012700 000004      LT34IT: MOV      #4,R0
3115 011106 004767 013046      LT34A1: JSR     PC,INIT1 ;GO INIT, SELECT DRIVE+SLAVE
3116 011112 042777 003400 167422  BIC      #3400,@TC ;SELECT NRZI
3117 011120 032777 000040 167374  LT34A:  BIT      #40,@DS ;SEE IF PES=0
3118 011126 001410              BEQ      LT34B      ;IF SO: BR
3119 011130 012767 000002 167570  MOV      #2,EXFL ;SET RCVD-NOT EXPT
3120 011136 012767 011106 167556  MOV      #LT34A1,SCOLP ;SET SCOPE ADDRESS
3121 011144 004767 011306      JSR     PC,LTGERO ;GO PRINT ERROR
3122 011150 062777 000400 167364  LT34B:  ADD      #400,@TC ;BUMP DENSITY
3123 011156 005300              DEC      R0 ;SEE IF DONE ALL NRZI
3124 011160 001357              BNE      LT34A      ;IF NOT: BR
3125 011162 032777 000040 167332  LT34C:  BIT      #40,@DS ;SEE IF PES=1
3126 011170 001010              BNE      LT34X      ;IF SO: BR
3127 011172 012767 011162 167522  MOV      #LT34C,SCOLP ;SET SCOPE ADDRESS
3128 011200 012767 000001 167520  MOV      #1,EXFL ;SET EXPT-NOT RCVD FLAG
3129 011206 004767 011244      JSR     PC,LTGERO ;GO PRINT ERROR
3130 011212 004767 012660      LT34X:  JSR     PC,ITER ;GO SEE IF ITERATION
3131 011216 000167 171230      LT34XX: JMP     TSCD2 ;RETURN TO SCHED
  
```

```
3133
3134
3135 ;LOGIC TEST 35: TAPE CONTROL WRITE(TCW)*****
3136 011222 012767 035030 167452 LT35: MOV #TMS37,ERADD
3137 011230 012767 032272 167374 MOV #MSLT35,EMADDR
3138 011236 004767 012716 LT35IT: JSR PC,INIT1 ;INIT SELECT DRIVE, SLAVE
3139 011242 032777 000020 167252 1$: BIT #20,@DS ;SEE IF SDWN IS RESET
3140 011250 001374 BNE 1$ ;IF NOT: BR
3141 011252 052777 000300 167262 BIS #300,@TC ;SET FORMAT
3142 011260 012777 000015 167246 MOV #15,@MR ;SET WAM 3
3143 011266 012777 000071 167214 MOV #71,@C1 ;LOAD READ+GO
3144 011274 032777 020000 167240 BIT #20000,@TC ;SEE IF TCW=0
3145 011302 001410 BEQ LT35A ;IF SO: BR
3146 011304 012767 000002 167414 MOV #2,EXFL ;SET RCV-NOT EXPT FLAG
3147 011312 012767 011236 167402 MOV #LT35IT,SCOLP ;SET SCOPE ADDRESS
3148 011320 004767 011132 JSR PC,LTGERO ;GO PRINT ERROR
3149 011324 004767 012630 LT35A: JSR PC,INIT1 ;INIT
3150 011330 005077 167206 CLR @TC ;WRITE TO TC
3151 011334 032777 020000 167200 BIT #20000,@TC ;SEE IF TCW=1
3152 011342 001010 BNE LT35X ;IF SO: BR
3153 011344 012767 011324 167350 MOV #LT35A,SCOLP ;SET SCOPE ADDRESS
3154 011352 012767 000001 167346 MOV #1,EXFL ;SE EXPT-NOT RCVD FLAG
3155 011360 004767 011072 JSR PC,LTGERO ;GO PRINT ERROR
3156 011364 004767 012506 LT35X: JSR PC,ITER
3157 011370 000167 171056 JMP TSCD2 ;RETURN TO SCHED
```

```
3159
3160 ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)*****
3161
3162 011374 012767 032334 167230 LT36: MOV #MSLT36,EMADDR
3163 011402 012767 035036 167272 MOV #TMS38,ERADD ;SET ERROR CODE
3164 011410 004767 012544 LT36IT: JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE
3165 011414 032777 040000 167120 BIT #40000,@TC ;SEE IF FCS=0
3166 011422 001410 BEQ LT36A ;IF SO: BR
3167 011424 012767 011410 167270 MOV #LT36IT,SCOLP ;SET SCOPE ADDRESS
3168 011432 012767 000002 167266 MOV #2,EXFL ;SET RCVD-NOT EXPT
3169 011440 004767 011012 JSR PC,LTGERO ;GO PRINT ERROR
3170 011444 004767 012510 LT36A: JSR PC,INIT1 ;INIT
3171 011450 005077 167042 CLR @FC ;WRITE TO FC
3172 011454 032777 040000 167060 BIT #40000,@TC ;SEE IF FCS=1
3173 011462 001010 BNE LT36X ;IF SO: BR
3174 011464 012767 011444 167230 MOV #LT36A,SCOLP ;SET SCOPE ADDRESS
3175 011472 012767 000001 167226 MOV #1,EXFL ;SET EXPT-NOT RCVD
3176 011500 004767 010752 JSR PC,LTGERO ;GO PRINT ERROR
3177 011504 004767 012366 LT36X: JSR PC,ITER
3178 011510 000167 170736 JMP TSCD2 ;RETURN TO SCHED
```

```
3180
3181 ;LOGIC TEST 37: ACCELERATION(ACCL)*****
3182
3183 011514 012767 032376 167110 LT37: MOV #MSLT37,EMADDR
3184 011522 012767 035044 167152 MOV #TMS39,ERADD ;SET ERROR CODE
3185 011530 004767 012424 LT37IT: JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE
3186 011534 052777 000300 167000 BIS #300,@TC ;SET FORMAT
3187 011542 005777 166774 TST @TC ;SEE IF ACCL=1
3188 011546 100410 BMI LT37A ;IF SO: BR
3189 011550 012767 000001 167150 MOV #1,EXFL
3190 011556 012767 011530 167136 MOV #LT37IT,SCOLP ;SET SCOPE ADDRESS
3191 011564 004767 010666 JSR PC,LTGERO ;GO PRINT ERROR
3192 011570 004767 012364 LT37A: JSR PC,INIT1 ;INIT
3193 011574 052777 000300 166740 BIS #300,@TC ;SET FORMAT
3194 011602 012777 000015 166724 MOV #15,@MR ;SET WAM 3
3195 011610 012777 000071 166672 MOV #71,@C1 ;LOAD READ+GO
3196 011616 012700 100000 MOV #100000,R0 ;SET ACCL DELAY
3197 011622 005777 166714 LT37B: TST @TC ;SEE IF ACCL=0
3198 011626 100012 BPL LT37X ;IF SO: BR
3199 011630 005300 DEC R0
3200 011632 001373 BNE LT37B ;DELAY
3201 011634 012767 011570 167060 MOV #LT37A,SCOLP ;SET SCOPE ADDRESS
3202 011642 012767 000002 167056 MOV #2,EXFL
3203 011650 004767 010602 JSR PC,LTGERO ;GO PRINT ERROR
3204 011654 004767 012216 LT37X: JSR PC,ITER
3205 011660 000167 170566 JMP TSCD2 ;RETURN TO SCHED
```

```
3207
3208 ;LOGIC TEST 40: PE TAPE MARK (TM)*****
3209
3210 011664 005767 167002 LT40: TST NRZOF ;SEE IF NRZ ONLY
3211 011670 001046 BNE LT40XX ;IF SO: BR
3212 011672 012767 011706 167022 MOV #LT40IT,SCOLP ;SET SCOPE ADDRESS
3213 011700 012767 032441 166724 MOV #MSLT40,EMADDR
3214 011706 004767 012246 LT40IT: JSR PC,INITI ;INIT, SELECT DRIVE+SLAVE
3215 011712 005000 CLR R0
3216 011714 005300 LT40W: DEC R0
3217 011716 001376 BNE LT40W ;DELAY FOR OPI RESET
3218 011720 052777 002300 166614 BIS #2300,@TC
3219 011726 012777 000007 166600 MOV #7,@MR ;SET WAM 0
3220 011734 012777 000027 166546 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
3221 011742 012700 100000 MOV #100000,R0 ;SET DELAY
3222 011746 032777 000004 166546 LT40A: BIT #4,@DS ;SEE IF TM=1
3223 011754 001012 BNE LT40X ;IF SO: BR
3224 011756 005300 DEC R0
3225 011760 001372 BNE LT40A ;DELAY
3226 011762 012767 034463 166712 MOV #TMS3,ERADD
3227 011770 012767 000001 166730 MOV #1,EXFL
3228 011776 004767 010454 JSR PC,LTGERO ;GO PRINT ERROR
3229 012002 004767 012070 LT40X: JSR PC,ITER
3230 012006 000157 170440 LT40XX: JMP TSCD2 ;RETURN TO SCHED
```

```

3232
3233           ;LOGIC TEST 41: NRZ TAPE MARK (TM,VPE,ITM)*****
3234
3235 012012 012767 012026 166702 LT41: MOV #LT41IT,SCOLP ;SET SCOPE ADDRESS
3236 012020 012767 032506 166604 MOV #MSLT41,EMADDR
3237 012026 004767 012126 LT41IT: JSR PC,INITI ;INIT, SELECT DRIVE,SLAVE
3238 012032 004767 005232 JSR PC,BOTT ;GO ASSURE NOT AT BOT
3239 012036 052777 001700 166476 BIS #1700,@TC ;SET NRZ+NORMAL FORMAT
3240 012044 012777 177760 166444 MOV #-20,@FC ;SET FCS
3241 012052 012777 000007 166454 MOV #7,@MR ;SET WAM 0
3242 012060 012777 000027 166422 MOV #27,@C1 ;LOAD WRITE TAPE MARK+GO
3243 012066 005000 CLR R0
3244 012070 032777 000004 166424 LT41A: BIT #4,@DS ;SEE IF TM=1
3245 012076 001012 BNE LT41B ;IF SO: BR
3246 012100 005300 DEC R0
3247 012102 001372 BNE LT41A ;DELAY
3248 012104 012767 034463 166570 MOV #TMS3,ERADD ;SET ERROR CODE
3249 012112 012767 000001 166606 MOV #1,EXFL
3250 012120 004767 010332 JSR PC,LTGERO ;GO PRINT ERROR
3251 012124 032777 002000 166372 LT41B: BIT #200,@ER ;SEE IF ITM=1
3252 012132 001010 BNE LT41C ;IF SO: BR
3253 012134 012767 034726 166540 MOV #TMS30,ERADD ;SET ERROR CODE
3254 012142 012767 000001 166556 MOV #1,EXFL
3255 012150 004767 010302 JSR PC,LTGERO ;GO PRINT ERROR
3256 012154 032777 000100 166342 LT41C: BIT #100,@ER ;SEE IF VPE=1
3257 012162 001011 BNE LT41D ;IF SO: BR
3258 012164 012767 034713 166510 MOV #TMS28,ERADD ;SET ERROR CODE
3259 012172 012767 000001 166526 MOV #1,EXFL
3260 012200 004767 010252 JSR PC,LTGERO ;GO PRINT ERROR
3261 012204 000410 BR LT41X
3262 012206 012701 002100 LT41D: MOV #2100,R1 ;SET EXPT ERROR BITS
3263 012212 017702 166306 MOV @ER,R2 ;GET ERROR REGISTER
3264 012216 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3265 012220 001402 BEQ LT41X ;IF NOT: BR
3266 012222 004767 010216 JSR PC,LTGER3 ;ELSE PRINT ERROR
3267 012226 005002 LT41X: CLR R2 ;SET TIMER
3268 012230 032777 000200 166264 1$: BIT #200,@DS ;SEE IF DRY SET
3269 012236 001002 BNE 2$ ;IF SO: BR
3270 012240 005302 D'C R2 ;AWAIT DRY
3271 012242 001372 BNE 1$ ;DELAY
3272 012244 004767 011626 2$: JSR PC,ITER ;GO SEE IF ITERATIONS
3273 012250 004767 010660 JSR PC,DRVCLR ;GO DO DRIVE CLEAR
3274 012254 000167 170172 JMP TSCD2 ;RETURN TO SCHED
  
```

3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295

012260 012767 004270 166546
012266 012767 000100 166542
012274 012767 010600 166536
012302 012767 000000 166532
012310 012767 032555 166314
012316 012767 001700 166462
012324 005067 166462
012330 012767 012336 166364
012336 004767 003712
012342 005267 166444
012346 032767 000004 166436
012354 001770
012356 004767 011514
012362 005067 166430
012366 000167 170060

LT42:

LT42A:

LT42B:

```
;THE FOLLOWING 13 TESTS WILL CHECK DATA FORMATTING  
;AND TRANSFER THROUGH THE TMO2 WRAP AROUND MODES  
  
;LOGIC TEST 42: WRAP 3, NRZ, NORMAL ODD *****  
MOV #4270,WCS1 ;SET EXPT CS1  
MOV #100,WCS2 ;SET EXPT CS2  
MOV #10600,WDS ;SET EXPT DS  
MOV #0,WER ;SET EXPT ER  
MOV #MSLT42,EMADDR ;SET HEADER  
MOV #1700,UDES ;SET NRZ,NORMAL, ODD  
CLR PATRN ;POINT TO PATTERN 0  
MOV #LT42B,SCOLP ;SET SCOPE ADDRESS  
JSR PC,WAM3 ;GO DO WRAP 3  
INC PATRN ;BUMP PATTERN POINTER  
BIT #4,PATRN ;SEE IF DONE  
BEQ LT42B ;IF NOT: BR  
JSR PC,ITER ;GO SEE IF ITERATIONS  
CLR RDRVF ;CLEAR REVENUE FLAG  
JMP TSCD2 ;RETURN TO SCHEDULAR
```



```
3297
3298
3299
3300 012372 005767 166274
3301 012376 001402
3302 012400 000167 170046
3303 012404 012767 004270 166422
3304 012412 012767 000100 166416
3305 012420 012767 010640 166412
3306 012426 012767 000000 166406
3307 012434 012767 032624 166170
3308 012442 012767 002300 166336
3309 012450 000167 177650

;LOGIC TEST 43: WRAP 3, PE, NORMAL, ODD*****
LT43: TST NRZOF ;SEE IF NRZ ONLY
      BEQ LT43A ;IF NOT: BR
      JMP TSCD2 ;RETURN TO SCHED
LT43A: MOV #4270,WCS1 ;SET EXPT CS1
      MOV #100,WCS2 ;SET EXPT CS2
      MOV #10640,WDS ;SET EXPT DS
      MOV #0,WER ;SET EXPT WER
      MOV #MSLT43,EMADDR ;SET HEADER
      MOV #2300,UDES ;SET PE, NORMAL, ODD
      JMP LT42A ;EXECUTE TEST SEQUENCE
```

```
3311  
3312  
3313 ;LOGIC TEST 44: WRAP 2, NRZ, NORMAL, ODD*****  
3314 012454 012767 004260 166352 LT44: MOV #4260,WCS1 ;SET EXPT CS1  
3315 012462 012767 000100 166346 MOV #100,WCS2 ;SET EXPT CS2  
3316 012470 012767 010600 166342 MOV #10600,WDS ;SET EXPT DS  
3317 012476 012767 000000 166336 MOV #0,WER ;SET EXPT WER  
3318 012504 012767 032672 166120 MOV #MSLT44,EMADDR ;SET HEADER  
3319 012512 012767 001700 166266 MOV #1700,UDES ;SET TO NRZ,NORMAL, ODD  
3320 012520 005067 166266 LT44A: CLR PATRN ;POINT TO PATTERN 0  
3321 012524 012767 012532 166170 MOV #LT44B,SCOLP ;SET SCOPE ADDRESS  
3322 012532 004767 003452 LT44B: JSR PC,WAM2 ;GO DO WRAP 2  
3323 012536 005267 166250 INC PATRN ;BUMP POINTER  
3324 012542 032767 000004 166242 BIT #4,PATRN ;SEE IF DONE  
3325 012550 001770 BEQ LT44B ;IF NOT: BR  
3326 012552 004767 011320 JSR PC,ITER ;GO SEE IF ITERATIONS  
3327 012556 000167 167670 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
3329
3330 ;LOGIC TEST 45: WRAP 2, PE, NORMAL, ODD*****
3331
3332 012562 005767 166104 LT45: TST NRZOF ;SEE IF NRZ ONLY
3333 012566 001402 BEQ LT45A ;IF NOT: BR
3334 012570 000167 167656 JMP TSCD2 ;RETURN TO SCHED
3335 012574 012767 004260 166232 LT45A: MOV #4260,WCS1 ;SET EXPT CS1
3336 012602 012767 000100 166226 MOV #100,WCS2 ;SET EXPT CS2
3337 012610 012767 010640 166222 MOV #10640,WDS ;SET EXPT DS
3338 012616 012767 000000 166216 MOV #0,WER ;SET EXPT WER
3339 012624 012767 032741 166000 MOV #MSLT45,EMADDR ;SET HEADER
3340 012632 012767 002300 166146 MOV #2300,UDES ;SET PE, NORMAL, ODD
3341 012640 000167 177654 JMP LT44A ;GO EXECUTE TEST SEQUENCES
```

```
3343  
3344 ;LOGIC TEST 46: WRAP 1, NRZ, NORMAL, ODD*****  
3345  
3346 012644 012767 004260 166162 LT46: MOV #4260,WCS1 ;SET EXPT CS1  
3347 012652 012767 000100 166156 MOV #100,WCS2 ;SET EXPT CS2  
3348 012660 012767 010600 166152 MOV #10600,WDS ;SET EXPT DS  
3349 012666 012767 000000 166146 MOV #0,WER ;SET EXPT WER  
3350 012674 012767 033007 165730 MOV #MSLT46,EMADDR ;SET HEADER  
3351 012702 012767 001700 166076 MOV #1700,UDES ;SET NRZ, NORMAL, ODD  
3352 012710 005067 166076 LT46A: CLR PATRN ;POINT TO PATTERN ZERO  
3353 012714 012767 012722 166000 MOV #LT46B,SCOLP ;SET SCOPE ADDRESS  
3354 012722 004767 003250 LT46B: JSR PC,WAM1 ;GO DO WRAP 1  
3355 012726 005267 166060 INC PATRN ;BUMP POINTER  
3356 012732 032767 000004 166052 BIT #4,PATRN ;SEE IF DONE  
3357 012740 001770 BEQ LT46B ;IF NOT: BR  
3358 012742 004767 011130 JSR PC,ITER ;GO SEE IF ITERATIONS  
3359 012746 000167 167500 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
3361  
3362  
3363  
3364 012752 005767 165714  
3365 012756 001402  
3366 012760 000167 167466  
3367 012764 004767 007206  
3368 012770 012767 004260 166036  
3369 012776 012767 000100 166032  
3370 013004 012767 010640 166026  
3371 013012 012767 000000 166022  
3372 013020 012767 033056 165604  
3373 013026 012767 002300 165752  
3374 013034 000167 177650
```

;LOGIC TEST 47: WRAP 1, PE, NORMAL, ODD*****

LT47:	TST	NRZOF		:SEE IF NRZ ONLY
	BEQ	LT47A		:IF NOT: BR
	JMP	TSCD2		:RETURN TO SCHED
LT47A:	JSR	PC,PPGEN		:GO GENERATE PRE/POSTAMBLE
	MOV	#4260,WCS1		:SET EXPT CS1
	MOV	#100,WCS2		:SET EXPT CS2
	MOV	#10640,WDS		:SET EXPT DS
	MOV	#0,WER		:SET EXPT WER
	MOV	#MSLT47,EMADDR		:SET HEADER
	MOV	#2300,UDES		:SET PE, NORMAL, ODD
	JMP	LT46A		:GO EXECUTE TEST SEQUENCE

```
3376
3377 ;LOGIC TEST 50: WRAP 0, NRZ,NORMAL, ODD*****
3378
3379 013040 012767 144260 165766 LT50: MOV #144260,WCS1 ;SET EXPT CS1
3380 013046 012767 000100 165762 MOV #100,WCS2 ;SET EXPT CS2
3381 013054 012767 150600 165756 MOV #150600,WDS ;SET EXPT DS
3382 013062 012767 000200 165752 MOV #200,WER ;SET EXPT ER
3383 013070 012767 033124 165534 MOV #MSLT50,EMADDR ;SET HEADER
3384 013076 012767 001700 165702 MCV #1700,UDES ;SET NRZ, NORMAL, ODD
3385 013104 005067 165702 LT50A: CLR PATRN ;POINT TO PATTERN 0
3386 013110 012767 013116 165604 MOV #LT50B,SCOLP ;SET SCOPE ADDRESS
3387 013116 004767 003010 LT50B: JSR PC,WAM0 ;GO DO WRAP 0
3388 013122 005267 165664 INC PATRN ;BUMP POINTER
3389 013126 032767 000004 165656 BIT #4,PATRN ;SEE IF DONE
3390 013134 001770 BEQ LT50B ;IF NOT: BR
3391 013136 004767 010734 JSR PC,ITER ;GO SEE IF ITERATIONS
3392 013142 000167 167304 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
3394  
3395 ;LOGIC TEST 51: WRAP 0, PE, NORMAL, ODD*****  
3396  
3397 013146 005767 165520 LT51: TST NRZOF ;SEE IF NRZ ONLY  
3398 013152 001402 BEQ LT51A ;IF NOT: BR  
3399 013154 000167 167272 JMP TSCD2 ;RETURN TI SCHED  
3400 013160 012767 004260 165646 LT51A: MOV #4260,WCS1 ;SET EXPT CS1  
3401 013166 012767 000100 165642 MOV #100,WCS2 ;SET EXPT CS2  
3402 013174 012767 010640 165636 MOV #10640,WDS ;SET EXPT DS  
3403 013202 012767 000000 165632 MOV #0,WER ;SET EXPT ER  
3404 013210 012767 033173 165414 MOV #MSLT51,EMADDR ;SET HEADER  
3405 013216 012767 002300 165562 MOV #2300,UDES ;SET PE, NORMAL, ODD  
3406 013224 000167 177654 JMP LT50A ;GO EXECUTE TEST SEQUENCE
```

```
3408
3409 ;LOGIC TEST 52: CORE DUMP WRITE, WAM2*****
3410
3411 013230 012767 004260 165576 LT52: MOV #4260,WCS1 ;SET EXPT CS1
3412 013236 012767 000100 165572 MOV #100,WCS2 ;SET EXPT CS2
3413 013244 012767 010600 165566 MOV #10600,WDS ;SET EXPT DS
3414 013252 012767 000000 165562 MOV #0,WER ;SET EXPT ER
3415 013260 012767 033241 165344 MOV #MSLT52,EMADDR ;SET HEADER
3416 013266 012767 001720 165512 MOV #1720,UDES ;SET NRZ, CORE DUMP, ODD
3417 013274 005067 165512 CLR PATRN ;POINT TO PATTERN 0
3418 013300 012767 013306 165414 MOV #LT52A,SCOLP ;SET SCOPE ADDRESS
3419 013306 004767 002676 LT52A: JSR PC,WAM2 ;GO DO WAM 2
3420 013312 022767 000002 165472 CMP #2,PATRN ;SEE IF DONE
3421 013320 001404 BEQ LT52X ;IF SO: BR
3422 013322 012767 000002 165462 MOV #2,PATRN ;SELECT PATTERN 2
3423 013330 000766 BR LT52A ;CONTINUE
3424 013332 004767 010540 LT52X: JSR PC,ITER ;GO SEE IF ITERATIONS
3425 013336 000167 167110 JMP TSCD2 ;RETURN TO SCHEDULES
3426
```



```
3428
3429
3430 ;LOGIC TEST 53: CORE DUMP READ, WAM 3*****
3431 013342 012767 004270 165464 LT53: MOV #4270,WCS1 ;SET EXPT CS1
3432 013350 012767 000100 165460 MOV #100,WCS2 ;SET EXPT CS2
3433 013356 012767 010600 165454 MOV #10600,WDS ;SET EXPT DS
3434 013364 012767 000000 165450 MOV #C,WER ;SET EXPT ER
3435 013372 012767 033312 165232 MOV #MSLT53,EMADDR ;SET HEADER
3436 013400 012767 001720 165400 MOV #1720,UDES ;SELECT NRE, CORE DUMP, ODD
3437 013406 005067 165400 CLR PATRN ;SELECT PATTERN 0
3438 013412 012767 013426 165302 MOV #LT53A,SCOLP ;SET SCOPE ADDRESS
3439 013420 012767 001070 165372 MOV #WCDP0,RCDP ;POINT TO PATTERN 0
3440 013426 004767 002622 LT53A: JSR PC,WAM3 ;GO DO WAM3
3441 013432 022767 000002 165352 CMP #2,PATRN ;SEE IF DONE
3442 013440 001407 BEQ LT53X ;IF SO: BR
3443 013442 012767 000002 165342 MOV #2,PATRN ;SELECT PATTERN 2
3444 013450 012767 001056 165342 MOV #WCDP2,RCDP ;POINT TO PATTERN 2
3445 013456 000763 BR LT53A ;CONTINUE
3446 013460 004767 010412 LT53X: JSR PC,ITER ;GO SEE IF ITERATION
3447 013464 000167 166762 JMP TSCD2 ;RETURN TO SCHEDULE
```

```
3449  
3450  
3451  
3452 013470 012767 004260 165336 LT54: MOV #4260,WCS1 :SET EXPT CS1  
3453 013476 012767 000100 165332 MOV #100,WCS2 :SET EXPT CS2  
3454 013504 012767 010600 165326 MOV #10600,WDS :SET EXPT DS  
3455 013512 012767 000000 165322 MOV #0,WER :SET EXPT ER  
3456 013520 012767 033362 165104 MOV #MSLT54,EMADDR :SET HEADER  
3457 013526 012767 001710 165252 MOV #1710,UDES :SET NRZ, NORMAL, EVEN  
3458 013534 000167 177150 JMP LT46A :GO EXECUTE WAM 1
```

```
3460  
3461  
3462  
3463 013540 012767 144260 165266 LT55: MOV #144260,WCS1 :SET EXPT CS1  
3464 013546 012767 000100 165262 MOV #100,WCS2 :SET EXPT CS2  
3465 013554 012767 150600 165256 MOV #150600,WDS :SET EXPT DS  
3466 013562 012767 000200 165252 MOV #200,WER :SET EXPT ER  
3467 013570 012767 033443 165034 MOV #MSLT55,EMADDR :SET HEADER  
3468 013576 012767 001710 165202 MOV #1710,UDES :SET NRZ, NORMAL, EVEN  
3469 013604 000167 177274 JMP LT50A :GO DO WAM 0
```

```
3471  
3472  
3473 ;LOGIC TEST 56: READ REVERSE: WAM 3(M8906)*****  
3474 013610 012767 004276 165216 LT56: MOV #4276,WCS1 :SET EXPT CS1  
3475 013616 012767 000100 165212 MOV #100,WCS2 :SET EXPT CS2  
3476 013624 012767 010640 165206 MOV #10640,WDS :SET EXPT DS  
3477 013632 012767 000000 165202 MOV #0,WER :SET EXPT ER  
3478 013640 012767 033522 164764 MOV #MSLT56,EMADDR :SET HEADER  
3479 013646 012767 002300 165132 MOV #2300,UDES :SELECT PE,NORMAL,ODD  
3480 013654 012767 000001 165134 MOV #1,RDRVF :SET READ REVERSE FLAG  
3481 013662 000167 176436 JMP LT42A :GO DO WAM 3, REVERSE  
3482
```

```

3484
3485 :THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT. EACH
3486 :TEST WILL PERFORM A TAPE WRITE WHILE IN A PARITICULAR MAINTENANCE
3487 :MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.
3488
3489 :LOGIC TEST 57: CYCLIC REDUNDANCY ERROR(CRC)*****
3490
3491 013666 004767 001746 LT57: JSR PC,STATIC ;GO SEE IF STATIC ONLY
3492 013672 012700 001000 MOV #1000,R0
3493 013676 005300 LT57PS: DEC R0
3494 013700 001376 BNE LT57PS ;PAUSE
3495 013702 012767 033567 164722 MOV #MSLT57,EMADDR
3496 013710 012767 013716 165004 MOV #LT57IT,SCOLP ;SET SCOPE ADDRESS
3497 013716 004767 010236 LT57IT: JSR PC,INIT1 ;INIT SELECT DRIVE+SLAVE
3498 013722 052777 001700 164612 BIS #1700,@TC ;SET NRZ + NORMAL FORMAT
3499 013730 012777 177770 164554 MOV #-10,@WC
3500 013736 012777 177760 164552 MOV #-20,@FC ;SET FC=20
3501 013744 012777 035054 164542 MOV #WDATA,@BA ;SET BUS ADDRESS
3502 013752 012777 000007 164554 MOV #7,@MR ;SET MM CODE
3503 013760 012777 000061 164522 MOV #61,@C1 ;LOAD WRITE+GO
3504 013766 005000 CLR R0
3505 013770 032777 000200 164524 LT57A: BIT #200,@DS ;SEE IF DRY=1
3506 013776 001002 BNE LT57B ;IF SO: BR
3507 014000 005300 DEC R0
3508 014002 001372 BNE LT57A ;DELAY
3509 014004 022777 000200 164512 LT57B: CMP #200,@ER ;SEE IF LRC ERROR ONLY
3510 014012 001007 BNE LT57B1 ;IF NOT: BR
3511 014014 017702 164510 MOV @CC,R2 ;GET CHECK CHAR
3512 014020 042702 177000 BIC #177000,R2 ;MASK CRC
3513 014024 022702 000777 CMP #777,R2 ;SEE IF SETUP CRC IS CORRECT
3514 014030 001410 BEQ LT57B2 ;IF SO: BR
3515 014032 004767 006406 LT57B1: JSR PC,LTGER3 ;ELSE PRINT ERROR SETUP
3516 014036 012704 030223 MOV #MSG55,R4
3517 014042 004767 010454 JSR PC,TTOUT ;PRINT SETUP ERROR MSG
3518 014046 000167 166400 JMP TSCD2 ;RETURN TO SCHED
3519 014052 004767 010102 LT57B2: JSR PC,INIT1 ;GO INIT
3520 014056 052777 000300 164456 BIS #300,@TC ;SET FORMAT+NRZ
3521 014064 012777 177770 164420 MOV #-10,@WC ;SET WC
3522 014072 012777 177760 164416 MOV #-20,@FC ;SET FC
3523 014100 012777 035054 164406 MOV #WDATA,@BA ;SET BA
3524 014106 012777 000021 164420 MOV #21,@MR ;SET MM
3525 014114 012777 000061 164366 MOV #61,@C1 ;LOAD WRITE+GO
3526 014122 005000 CLR R0
3527 014124 032777 000200 164370 LT57C: BIT #200,@DS ;SEE IF DRY
3528 014132 001002 BNE LT57D ;IF SO: BR
3529 014134 005300 DEC R0
3530 014136 001372 BNE LT57C ;AWAIT DRY
3531 014140 005777 164360 LT57D: TST @ER ;SEE IF CRC=1
3532 014144 100411 BMI LT57E ;IF SO: BR
3533 014146 012767 035022 164526 MOV #TMS36,ERADD ;SET ERROR CODE
3534 014154 012767 000001 164544 MOV #1,EXFL
3535 014162 004767 006270 JSR PC,LTGER0 ;GO PRINT ERROR
3536 014166 000410 BR LT57X
3537 014170 012701 100200 LT57E: MOV #100200,R1 ;SET EXPT ERROR BITS
3538 014174 017702 164324 MOV @ER,R2 ;GET ERROR REGISTER
3539 014200 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
  
```

CZTUGO TMO2/TU16 CTRL LGC
CZTUCG.P11 12-JUL-83 10:41

MACY11 30(1046) 12-JUL-83 10:47 J 8
PAGE 91-1

SEQ 0100

3540 014202 001402
3541 014204 004767 006234
3542 014210 004767 007662
3543 014214 004767 006714
3544 014220 000167 166226

LT57X: BEQ LT57X ;IF NOT: BR
JSR PC,LTGER3 ;ELSE PRINT ERROR
JSR PC,ITER ;DO ITERATIONS
JSR PC,DRVCLR
JMP TSCD2 ;RETURN TO SCHED

```

3546
3547 ;LOGIC TEST 60: LONGITUALINAL REDUNDANCY(LRC)*****
3548
3549 014224 004767 001410 LT60: JSR PC,STATIC ;GO SEE IF STATIC ONLY
3550 014230 012767 014244 164464 MOV #LT60IT,SCOLP ;SET SCOPE ADDRESS
3551 014236 012767 033623 164366 MOV #MSLT60,EMADDR
3552 014244 004767 007710 LT60IT: JSR PC,INIT1 ;INIT, SELECT DRIVE+SLAVE
3553 014250 052777 000300 164264 BIS #300,@TC ;SET FORMAT+NRZ
3554 014256 012777 000023 164250 MOV #23,@MR ;SET MM
3555 014264 012777 177770 164220 MOV #-10,@WC ;SET WC
3556 014272 012777 177760 164216 MOV #-20,@FC ;SET FC
3557 014300 012777 035054 164206 MOV #WDATA,@BA ;SET BA
3558 014306 012777 000061 164174 MOV #61,@C1 ;LOAD WRITE+GO
3559 014314 005000 CLR R0
3560 014316 032777 000200 164176 LT60C: BIT #200,@DS ;SEE IF DRY
3561 014324 001002 BNE LT60D ;IF SO: BR
3562 014326 005300 DEC R0
3563 014330 001372 BNE LT60C ;AWAIT DRY
3564 014332 032777 000200 164164 LT60D: BIT #200,@ER ;SEE IF LRC=1
3565 014340 001011 BNE LT60E ;IF SO: BR
3566 014342 012767 034677 164332 MOV #TMS26,ERADD ;SET ERROR CODE
3567 014350 012767 000001 164350 MOV #1,EXFL
3568 014356 004767 006074 JSR PC,LTGER0 ;GO PRINT
3569 014362 000425 BR LT60X
3570 014364 017702 164144 LT60E: MOV @MR,R2
3571 014370 042702 000177 BIC #177,R2 ;MASK LRC
3572 014374 012701 157600 MOV #157600,R1 ;SET EXPT LRC
3573 014400 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
3574 014402 001405 BEQ LT60F ;IF SO: BR
3575 014404 012767 030200 164270 MOV #MSG53,ERADD ;SET ERROR CODE
3576 014412 004767 007206 JSR PC,LTGER1 ;PRINT ERROR
3577 014416 017702 164102 LT60F: MOV @ER,R2 ;GET ERROR REGISTER
3578 014422 012701 000200 MOV #200,R1 ;SET EXPT ERROR BITS
3579 014426 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3580 014430 001402 BEQ LT60X ;IF NOT: BR
3581 014432 004767 006006 JSR PC,LTGER3 ;ELSE PRINT ERROR
3582 014436 004767 007434 LT60X: JSR PC,ITER
3583 014442 004767 006466 JSR PC,DRVCLR
3584 014446 000167 166000 JMP TSCD2 ;RETURN TO SCHED

```

```

3586
3587 ;LOGIC TEST 61: PE CORRECTABLE DATA (CORR)*****
3588
3589 014452 005767 154214 LT61: TST NRZOF ;SEE IF NRZ ONLY
3590 014456 001122 BNE LT61XX ;IF SO: BR
3591 014460 004767 001154 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3592 014464 012767 033657 164140 MOV #MSLT61,EMADDR
3593 014472 012767 014500 164222 MOV #LT61IT,SCOLP
3594 014500 004767 007454 LT61IT: JSR PC,INIT1 ;INIT, SELCT DRIVE+SLAVE
3595 014504 052777 002300 164030 BIS #2300,@TC ;SET PE,NORMAL
3596 014512 012777 177600 163772 MOV #-200,@WC ;SET WC=200
3597 014520 012777 177400 163770 MOV #-400,@FC ;SET FC=400
3598 014526 012777 035054 163760 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3599 014534 012777 000061 163746 MOV #61,@C1 ;LOAD WRITE+GO
3600 014542 005000 CLR R0
3601 014544 005777 163746 LT61A: TST @FC ;SEE IF FC=0
3602 014550 001402 BEQ LT61A1 ;IF SO: BR
3603 014552 005300 DEC R0
3604 014554 001373 BNE LT61A ;DELAY FOR FC=0
3605 014556 012777 000021 163750 LT61A1: MOV #21,@MR ;SET MAINT CODE
3606 014564 005000 CLR R0
3607 014566 032777 000200 163726 LT61B: BIT #200,@DS ;SEE IF DRY IS SET
3608 014574 001002 BNE LT61C ;IF SO: BR
3609 014576 005300 DEC R0
3610 014600 001372 BNE LT61B ;AWAIT DRY
3611 014602 005777 163716 LT61C: TST @ER ;SEE IF CORR=1
3612 014606 100410 BMI LT61D ;IF SO: BR
3613 014610 012767 035013 164064 MOV #TMS35,ERADD ;SET ERROR CODE
3614 014616 012767 000001 164102 MOV #1,EXFL
3615 014624 004767 005626 JSR PC,LTGERO ;GO PRINT ERROR
3616 014630 000240 LT61D: NOP
3617 014632 000240 LT61E: NOP
3618 014634 122777 000002 163666 CMPB #2,@CC ;SEE IF DEAD TRACK BIT 1
3619 014642 001414 BEQ LT61F ;IF SO: BR
3620 014644 117702 163660 MOVB @CC,R2 ;SAVE RCVD
3621 014650 042702 177000 BIC #177000,R2 ;MASK OUT CRC
3622 014654 112701 000002 MOVB #2,R1 ;SAVE EXPT
3623 014660 012767 027573 164014 MOV #MSG42,ERADD ;SET ERROR CODE
3624 014666 004767 006732 JSR PC,LTGER1 ;GO PRINT ERROR
3625 014672 000410 BR LT61X
3626 014674 017702 163624 LT61F: MOV @ER,R2 ;GET ERROR REGISTER
3627 014700 012701 100000 MOV #100000,R1 ;SET EXPT ERROR BITS
3628 014704 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3629 014706 001402 BEQ LT61X ;IF NOT: BR
3630 014710 004767 005530 JSR PC,LTGER3 ;ELSE PRINT ERROR
3631 014714 004767 007156 LT61X: JSR PC,ITER
3632 014720 004767 006210 JSR PC,DRVCLR
3633 014724 000167 165522 LT61XX: JMP TSCD2 ;RETURN TO SCHED
3634

```



```

3636
3637 ;LOGIC TEST 62: PE INCORRECTABLE DATA(INC,)*
3638
3639 014730 005767 163736 LT62: TST NRZOF ;SEE IF NRZ ONLY
3640 014734 001120 BNE LT62XX ;IF SO: BR
3641 014736 004767 000676 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3642 014742 012767 033737 163662 MOV #MSLT62,EMADDR
3643 014750 012767 014756 163744 MOV #LT62IT,SCOLP
3644 014756 004767 007176 LT62IT: JSR PC,INIT1 ;INIT SELECT DRIVE SLAVE
3645 014762 012777 177600 MOV #-200,@WC ;SET WC=200
3646 014770 012777 177400 163522 MOV #-400,@FC ;SET FC=400
3647 014776 012777 035054 163510 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3648 015004 052777 002300 163530 BIS #2300,@TC ;SET TO PE,NORMAL
3649 015012 012777 000061 163470 MOV #61,@C1 ;LOAD WRITE+GO
3650 015020 005000 CLR R0
3651 015022 005777 163470 LT62E: TST @FC ;AWAIT FC=0
3652 015026 001402 BEQ LT62E1
3653 015030 005300 DEC R0
3654 015032 001373 BNE LT62E ;AWAIT FC=0
3655 015034 012777 000023 163472 LT62E1: MOV #23,@MR ;SET MAINT CODE
3656 015042 005000 CLR R0
3657 015044 032777 000200 163450 LT62A: BIT #200,@DS ;SEE IF DRY IS SET
3658 015052 001002 BNE LT62B ;IF SO: BR
3659 015054 005300 DEC R0
3660 015056 001372 BNE LT62A ;AWAIT DRY
3661 015060 032777 000100 163436 LT62B: BIT #100,@ER ;SEE IF INC=1
3662 015066 001010 BNE LT62D ;IF SO:BR
3663 015070 012767 034655 163604 MOV #TMS23,ERADD ;SET ERROR CODE
3664 015076 012767 000001 163622 MOV #1,EXFL
3665 015104 004767 005346 JSR PC,LTGERO ;GO PRINT ERROR
3666 015110 017702 163414 LT62D: MOV @CC,R2 ;GET CHECK CHAR
3667 015114 042702 177000 BIC #177000,R2 ;MASK CHECK CHAR
3668 015120 012701 000046 MOV #46,R1 ;SET EXPT CK
3669 015124 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
3670 015126 001405 BEQ LT62F ;IF SO: BR
3671 015130 012767 030212 163544 MOV #MSG54,ERADD
3672 015136 004767 006462 JSR PC,LTGER1 ;ELSE GO PRINT ERROR
3673 015142 017702 163356 LT62F: MOV @ER,R2
3674 015146 042702 120600 BIC #120600,R2 ;MASK NSG ,OPI AND CORR
3675 015152 012701 000100 MOV #100,R1 ;SET EXPT ERROR BITS
3676 015156 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3677 015160 001402 BEQ LT62X ;IF NOT: BR
3678 015162 004767 005256 JSR PC,LTGER3 ;ELSE PRINT ERROR
3679 015166 004767 006704 LT62X: JSR PC,ITER
3680 015172 004767 005736 JSR PC,DRVCLR
3681 015176 000167 165250 LT62XX: JMP TSCD2 ;RETURN TO SCHED

```

```

3683
3684 ;LOGIC TEST 63: PE FORMAT ERROR(PEF,NSG)*****
3685
3686 015202 005767 163464 LT63: TST NRZOF ;SEE IF NRZ ONLY
3687 015206 001116 BNE LT63XX ;IF SO: BR
3688 015210 004767 000424 JSR PC,STATIC ;GO SEE IF STATIC ONLY
3689 015214 012767 034021 163410 MOV #MSLT63,EMADDR ;SET HEADER
3690 015222 012767 015230 163472 MOV #LT63IT,SCOLP ;SET SCOPE ADDRESS
3691 015230 004767 006724 LT63IT: JSR PC,INIT1 ;INITIALIZE
3692 015234 012777 177770 163250 MOV #-10,@WC ;SET WC=10
3693 015242 012777 177760 163246 MOV #-20,@FC ;SET FC=20
3694 015250 052777 002300 163264 BIS #2300,@TC ;SET TO PE,NORMAL
3695 015256 012777 035054 163230 MOV #WDATA,@BA ;SET BA=START OF WRITE BUFFER
3696 015264 012777 000061 163216 MOV #61,@C1 ;LOAD WRITE+GO
3697 015272 005777 163220 LT63A: TST @FC
3698 015276 001375 BNE LT63A ;AWAIT FC 0
3699 015300 032777 000100 163226 1$: BIT #100,@MR
3700 015306 001774 BEQ 1$ ;TIME DELAY
3701 015310 032777 000100 163216 2$: BIT #100,@MR
3702 015316 001374 BNE 2$
3703 015320 032777 000100 163206 3$: BIT #100,@MR
3704 015326 001774 BEQ 3$
3705 015330 012777 000027 163176 MOV #27,@MR ;SET MM CODE TO KILL PEF
3706 015336 012700 004000 MOV #4000,R0
3707 015342 032777 000200 163152 LT63B: BIT #200,@DS ;SEE IF DRY SET
3708 015350 001002 BNE LT63C ;IF SO: BR
3709 015352 005300 DE? R0
3710 015354 001372 BNE LT63B ;AWAIT DRY
3711 015356 032777 000200 163140 LT63C: BIT #200,@ER ;SEE IF PEF SET
3712 015364 001011 BNE LT63D ;IF SO: BR
3713 015366 012767 034671 163306 MOV #TMS25,ERADD ;SET ERROR TAG
3714 015374 012767 000001 163324 MOV #1,EXFL ;SET EXPT FLAG
3715 015402 004767 005050 JSR PC,LTGERO ;GO PRINT ERROR
3716 015406 000412 BR LT63X
3717 015410 017702 163110 LT63D: MOV @ER,R2 ;GET ERROR REGISTER
3718 015414 042702 120100 BIC #120100,R2 ;CLEAR KNOWN BITS
3719 015420 012701 000600 MOV #600,R1 ;SET EXPT ERROR BITS
3720 015424 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
3721 015426 001402 BEQ LT63X ;IF NOT: BR
3722 015430 004767 005010 JSR PC,LTGER3 ;ELSE PRINT ERROR
3723 015434 004767 006436 LT63X: JSR PC,ITER
3724 015440 004767 005470 JSR PC,DRVCLR
3725 015444 000167 165002 LT63XX: JMP TSCD2 ;RETURN TO SCHED
  
```

```
3727 ;LOGIC TEST 64: FRAME COUNT OVERFLOW(M8905)*****
3728
3729 015450 012767 034055 163154 LT64: MOV #MSLT64,EMADDR ;SET TEST HEADER
3730 015456 012767 015464 163236 MCV #LT64IT,SCOLP ;SET SCOPE ADDRESS
3731 015464 004767 006470 LT64IT: JSR PC,INIT1 ;GO INIT
3732 015470 012777 177770 163014 MOV #-10,@WC ;SET WC = 10
3733 015476 012777 177760 163012 MOV #-20,@FC ;SET FC = 20
3734 015504 052777 001700 163030 BIS #1700,@TC ;SET TO NRZ, NORMAL, ODD
3735 015512 012777 035054 162774 MOV #WDATA,@BA ;SET BUS ADDRESS
3736 015520 012777 000013 163006 MOV #13,@MR ;SET WRAP 2
3737 015526 012777 000061 162754 MOV #61,@C1 ;LOAD WRITE+GO
3738 015534 012700 040000 MOV #40000,R0
3739 015540 005777 162776 LT64A: TST @TC ;SEE IF ALPHA
3740 015544 10C002 BPL LT64B ;IF SO: BR
3741 015546 005300 DEC R0
3742 015550 001373 BNE LT64A ;AWAIT ALPHA
3743 015552 012700 000020 LT64B: MOV #20,R0 ;SET CLK CNT
3744 015556 052777 000040 162750 LT64C: BIS #40,@MR
3745 015564 042777 000040 162742 BIC #40,@MR ;CLOCK MR
3746 015572 005300 DEC R0
3747 015574 001370 BNE LT64C ;IF NOT DONE ALL: BR
3748 015576 017702 162714 MOV @FC,R2
3749 015602 005001 CLR R1 ;SET TEST WORD
3750 015604 020102 CMP R1,R2 ;SEE IF EXPT = RCVD
3751 015606 001410 BEQ LT64X ;IF SO: BR
3752 015610 012767 027046 163064 MOV #MSG19,ERADD ;SET ERROR CODE
3753 015616 012767 000001 163102 MOV #1,EXFL ;SET EXPT FLAG
3754 015624 004767 005774 JSR PC,LTGER1 ;GO PRINT ERROR
3755 015630 004767 006242 LT64X: JSR PC,ITER ;GO SEE IF ITERATIONS
3756 015634 000167 164612 JMP TSCD2 ;RETURN TO SCHEDULAR
3757
3758 ;STATIC TESTS ONLY SUBROUTINE*****
3759
3760 015640 005767 163104 STATIC: TST STFLG ;SEE IF SINGLE TEST ONLY
3761 015644 001006 BNE STATX ;IF SO: BR
3762 015646 005767 163150 TST STATC ;SEE IF STATIC ONLY
3763 015652 001403 BEQ STATX ;IF NOT: BR
3764 015654 005726 TST (SP)+ ;RESET STACK
3765 015656 000167 164570 JMP TSCD2 ;RETURN TO SCHEDULAR
3766 015662 000207 STATX: RTS PC ;RETURN TO TEST
3767
```

```
3769
3770
3771
3772
3773
3774
3775
3776
3777 015664 005767 163060 DSUP: TST STFLG ;SEE IF SINGLE TEST
3778 015670 001431 BEQ DSO ;IF NOT: BR
3779 015672 032777 000100 162672 BIT #100,@SWR ;SEE IF SELECT PATTERN
3780 015700 001425 BEQ DSO ;IF NOT: BR
3781 015702 012704 035472 MOV #WMSG3,R4
3782 015706 004767 006610 JSR PC,TOUT ;REQUEST PATTERN NUMBER
3783 015712 016703 163074 MOV PATRN,R3
3784 015716 004767 006742 JSR PC,OCTP ;PRINT PATTERN NUMBER
3785 015722 012705 001012 MOV #PATRN,R5 ;GET ADDRESS OF PATRN ENTRY
3786 015726 012701 000001 MOV #1,R1 ;SET SIZE OF ENTRY
3787 015732 012702 000003 MOV #3,R2 ;SET UPPER LIMIT
3788 015736 012703 000000 MOV #0,R3 ;SET LOWER LIMIT
3789 015742 004767 006316 JSR PC,TTR ;GO GET PATTERN NUMBER
3790 015746 012767 000001 163034 MOV #1,WPGFL ;SET FLAG
3791 015754 012703 036402 DS0: MOV #WBUF,R3 ;R3 = ADDR OF WRITE BUFFER
3792 015760 016701 163026 MOV PATRN,R1 ;R1 = PATTERN SELECTOR
3793 015764 062701 000001 ADD #1,R1 ;BUMP POINTER
3794 015770 000241 CLC
3795 015772 006101 ROL R1 ;MAKE PATTERN SELECTOR EVEN
3796 015774 000171 001044 JMP @DATBL(R1) ;GO GENERATE PATTERN
3797 016000 032777 010000 162530 DS1: BIT #10000,@DT ;SEE IF SEVEN TRACK
3798 016006 001410 BEQ DS3 ;IF NOT: BR
3799 016010 012702 000202 MOV #202,R2 ;SET BUFFER SIZE
3800 016014 012701 036402 MOV #WBUF,R1 ;SET START OF BUFFER
3801 016020 042721 140300 DS2: BIC #140300,(R1)+ ;MASK FOR 7 CH
3802 016024 005302 DEC R2 ;SEE IF DONE
3803 016026 001374 BNE DS2 ;IF NOT: BR
3804 016030 012702 000202 DS3: MOV #202,R2 ;R2=BUFFER SIZE +2
3805 016034 012701 037014 MOV #RBUF,R1 ;R1=READ DATA START
3806 016040 005021 DS4: CLR (R1)+ ;CLEAR BUFFER
3807 016042 005302 DEC R2 ;SEE IF DONE ALL
3808 016044 001375 BNE DS4 ;IF NOT: BR
3809 016046 000207 RTS PC ;EXIT
3810
3811 ;ALL ONES*****
3812
3813 016050 012701 177777 DAT1: MOV #-1,R1 ;R1=DATA
3814 016054 012702 000202 DAT1A: MOV #202,R2 ;R2=WORD COUNT +2
3815 016060 010123 DAT1B: MOV R1,(R3)+ ;LOAD BUFFER
3816 016062 005302 DEC R2 ;SEE IF DONE
3817 016064 001375 BNE DAT1B ;IF NOT: BR
3818 016066 000167 177706 JMP DS1 ;RETURN
3819
```

```
3821
3822                ;ALL ZEROS*****
3823
3824 016072 005001      DAT2: CLR    R1          ;R1=DATA
3825 016074 000167 177754  JMP    DAT1A        ;LOAD BUFFER
3826
3827                ;ONE/ZERO IN ALTERNATING CHARACTERS*****
3828
3829 016100 012701 125125  DAT3: MOV    #125125,R1 ;R1=DATA
3830 016104 000167 177744  JMP    DAT1A        ;LOAD BUFFER
3831
3832                ;ALL BITS 0-377*****
3833
3834 016110 005001      DAT4: CLR    R1          ;R1=STARTING DATA
3835 016112 012702 000404  MOV    #404,R2      ;R2=CHARACTER COUNT
3836 016116 110123      DAT4A: MOVB   R1,(R3)+ ;LOAD BUFFER
3837 016120 105201      INCB   R1          ;BUMP DATA
3838 016122 005302      DEC    R2          ;SEE IF DONE
3839 016124 001374      BNE    DAT4A       ;IF NOT: BR
3840 016126 000167 177646  JMP    DS1         ;RETURN
3841
```

```
3843
3844 ;WRAP AROUND MODE 0 GLOBAL*****
3845
3846 016132 012767 000006 162620 WAM0: MOV #6,WAM ;SET WAM NUMBER
3847 016140 012767 000060 162614 WAM01: MOV #60,FUN
3848 016146 005067 162612 CLR DATC
3849 016152 012767 036402 162610 MOV #WBUF,DATAD ;SET BUFFER ADDRESS
3850 016160 012767 037014 162604 MOV #RBUF,RDAD ;SET POINTER TO READ BUFFER
3851 016166 004767 000170 JSR PC,SETUP ;GO SET UP
3852 016172 000167 000516 JMP EXEC
3853
3854 ;WRAP AROUND MODE 1 WRITE BUFFER*****
3855
3856 016176 012767 000010 162554 WAM1: MOV #10,WAM
3857 016204 000167 177730 JMP WAM01
3858
3859 ;WRAP AROUND MODE 2 BIT FIDDLER WRITE*****
3860
3861 016210 012767 000012 162542 WAM2: MOV #12,WAM
3862 016216 012767 000060 162536 MOV #60,FUN
3863 016224 005067 162534 CLR DATC
3864 016230 012767 036402 162532 MOV #WBUF,DATAD
3865 016236 012767 037014 162526 MOV #RBUF,RDAD
3866 016244 004767 000112 WAM2A: JSR PC,SETUP
3867 016250 000167 000440 JMP EXEC
3868
3869 ;WRAP AROUND MODE 3 BIT FIDDLER READ*****
3870
3871 016254 012767 000014 162476 WAM3: MOV #14,WAM ;SET WAM NUMBER
3872 016262 012767 000070 162472 MOV #70,FUN ;SET FUNCTION
3873 016270 012767 037014 162472 MOV #RBUF,DATAD ;SET BUFFER ADDRESS
3874 016276 012767 036402 162462 MOV #WBUF,WTAD ;SET POINTER TO WRITE BUFFER
3875 016304 005767 162506 TST RDRVF
3876 016310 001411 BEQ WAM3A
3877 016312 062767 000376 162450 ADD #376,DATAD
3878 016320 062767 000377 162440 ADD #377,WTAD
3879 016326 012767 000076 162426 MOV #76,FUN ;SET READ REVERSE CODE
3880 016334 032767 000020 162444 WAM3A: BIT #20,UDES
3881 016342 001403 BEQ WAM3B
3882 016344 016767 162450 162414 WAM3B: MOV RCDP,WTAD
3883 016352 004767 000004 JSR PC,SETUP ;GO SET UP
3884 016356 000167 000332 JMP EXEC ;GO EXECUTE
3885
```

```

3887                                     ;REGISTER SETUP ROUTINE*****
3888
3889 016362 005767 162362          SETUP: TST      STFLG      ;SEE IF SINGLE TEST
3890 016366 001403                BEQ      SET0      ;IF NOT: BR
3891 016370 005767 162414                TST      WPGFL     ;SEE IF HAVE SELECTED PATTERN
3892 016374 001002                BNE     SET1      ;IF SO: BR
3893 016376 004767 177262          SET0: JSR      PC,DSUP ;GO DO DATA SETUP
3894 016402 004767 005552          SET1: JSR      PC,INIT1 ;GO INIT SELECT DRIVE, SLAVE
3895 016406 004767 000656                JSR      PC,BOTT   ;GO ASSURE NOT AT BOT
3896 016412 012777 177400 162076    MOV      #-400,@FC ;SET FC=WCX2
3897 016420 032767 000020 162360    BIT      #20,UDES  ;SEE IF CORE DUMP
3898 016426 001403                BEQ      SET2      ;IF NOT: BR
3899 016430 012777 177000 162060    MOV      #-1000,@FC ;SET FC=WCX4
3900 016436 012777 177600 162046    SET2: MOV      #-200,@WC ;SET WC
3901 016444 016777 162320 162042    MOV      DATAD,@BA ;SET BUS ADDRESS
3902 016452 032777 010000 162040    BIT      #10000,@CS ;ASSURE DRIVE THERE
3903 016460 001420                BEQ      SP1       ;IF SO: BR
3904 016462 032777 020000 162102    BIT      #20000,@SWR ;SEE IF PRINT ERRORS
3905 016470 001004                BNE     SP01      ;IF NOT: BR
3906 016472 012704 035513                MOV      #WMSG4,R4
3907 016476 004767 006020                JSR      PC,TTOUT  ;PRINT NON-EXISTANT DRIVE
3908 016502 032777 100000 162062    SP01: BIT      #100000,@SWR ;SEE IF HALT ON ERROR
3909 016510 001402                BEQ      SP0       ;IF NOT: BR
3910 016512 004767 007122                JSR      PC,STOP
3911 016516 000167 177660                JMP      SET1      ;RESETUP
3912 016522 022767 000014 162230    SP1:  CMP      #14,WAM ;SEE IF WAM 3
3913 016530 001026                BNE     SP1B      ;IF NOT: BR
3914 016532 117767 162230 162224    MOVB     @WTAD,DATC ;GET FIRST CHAR
3915 016540 042767 177400 162216    BIC      #177400,DATC
3916 016546 000367 162212                SWAB     DATC
3917 016552 005767 162240                TST      RDRVF     ;SEE IF READ REVERSE
3918 016556 001403                BEQ      SP1A     ;IF NOT: BR
3919 016560 005367 162202                DEC      WTAD      ;DECREMENT POINTER
3920 016564 000410                BR       SP1B
3921 016566 005267 162174                SP1A: INC      WTAD ;BUMP POINTER
3922 016572 032767 000020 162206    BIT      #20,UDES  ;SEE IF CORE DUMP
3923 016600 001402                BEQ      SP1B     ;IF NOT: BR
3924 016602 005267 162160                INC      WTAD      ;BUMP POINTER AGAIN
3925 016606 056777 162174 161726    SP1B: BIS      UDES,@TC ;SET UNIT DESCRIPTION (DEN,PAR,FMT)
3926 016614 052777 000001 161712    BIS      #1,@MR   ;SET MAINT MODE
3927 016622 056777 162132 161704    BIS      WAM,@MR  ;SET WAM
3928 016630 056777 162130 161676    BIS      DATC,@MR ;SET DATA
3929 016636 016777 162120 161644    MOV      FUN,@C1  ;SET FUNCTION
3930 016644 032777 040000 161650    BIT      #40000,@DS ;ASSURE NO ERROR
3931 016652 001001                BNE     SP3       ;IF NOT: BR
3932 016654 000207                RTS      PC        ;RETURN
3933 016656 032777 020000 161706    SP3:  BIT      #20000,@SWR ;SEE IF PRINT ERRORS
3934 016664 001004                BNE     SP4       ;IF NOT: BR
3935 016666 012704 035454                MOV      #WMSG2,R4
3936 016672 004767 005624                JSR      PC,TTOUT  ;PRINT SETUP ERROR
3937 016676 032777 100000 161666    SP4:  BIT      #100000,@SWR ;SEE IF HALT ON ERROR
3938 016704 001402                BEQ      SP5      ;IF NOT: BR
3939 016706 004767 006726                JSR      PC,STOP
3940 016712 000207                SP5:  RTS      PC        ;RETURN

```

```

3942                                     ;EXECUTE WAM ROUTINE*****
3943
3944 016714 032777 000040 161612 EXEC: BIT #40,@MR
3945 016722 001403 BEQ EX0 ;ASSURE MAINT CLOCK IS ZERO
3946 016724 042777 000040 161602 BIC #40,@MR ;IF NOT: CLEAR IT
3947 016732 022767 000010 162020 EX0: CMP #10,WAM ;SEE IF WAM 1 OR 2 OR 3
3948 016740 003402 BLE EX1 ;IF SO: BR
3949 016742 000167 000364 JMP EXW2 ;GO DO WAM 0
3950 016746 052777 000001 161534 EX1: BIS #1,@C1 ;SET GO BIT
3951 016754 005000 CLR R0
3952 016756 012701 000002 MOV #2,R1 ;SET DELAY
3953 016762 032777 100000 161552 EX1A: BIT #100000,@TC ;SEE IF ALPHA
3954 016770 001404 BEQ EX2 ;IF SO: BR
3955 016772 005300 DEC R0
3956 016774 001372 BNE EX1A ;AWAIT ALPHA
3957 016776 005301 DEC R1
3958 017000 001370 BNE EX1A
3959 017002 005077 161562 EX2: CLR @PSW
3960 017006 012701 000400 MOV #400,R1 ;SET NUMBER OF CLKS
3961 017012 032767 000020 161766 BIT #20,UDES ;SEE IF CORE DUMP
3962 017020 001402 BEQ EX3 ;IF NOT: BR
3963 017022 012701 001000 MOV #1000,R1 ;SET CLOCKS LWCX4
3964 017026 022767 000014 161724 EX3: CMP #14,WAM ;SEE IF WAM 3
3965 017034 001413 BEQ EX5A ;IF SO: BR
3966 017036 032767 002000 161742 BIT #2000,UDES ;SEE IF PE
3967 017044 001405 BEQ EX5 ;IF NOT PE: BR
3968 017046 000241 CLC
3969 017050 006101 ROL R1
3970 017052 062701 000246 ADD #246,R1 ;SET TO ALLOW FOR PRE/POSTAMBLE
3971 017056 000402 BR EX5A
3972 017060 062701 000010 EX5: ADD #10,R1 ;ADD CLOCKS FOR CRC AND LRC
3973 017064 022767 000014 161666 EX5A: CMP #14,WAM ;SEE IF WAM 3
3974 017072 001044 BNE EX5C ;IF NOT: BR
3975 017074 117700 161666 MOV @WTAD,R0
3976 017100 042700 177400 BIC #177400,R0
3977 017104 005767 161706 TST RDRVF ;SEE IF REVERSE
3978 017110 001403 BEQ EX5A1 ;IF NOT: BR
3979 017112 005367 161650 DEC WTAD ;DEC POINTER
3980 017116 000416 BR EX5B
3981 017120 005267 161642 EX5A1: INC WTAD
3982 017124 032767 000020 161654 BIT #20,UDES ;SEE IF CORE DUMP
3983 017132 001410 BEQ EX5B ;IF NOT: BR
3984 017134 005267 161626 INC WTAD ;BUMP POINTER
3985 017140 005777 161622 TST @WTAD ;SEE IF END
3986 017144 001003 BNE EX5B ;IF NOT: BR
3987 017146 162767 000010 161612 SUB #10,WTAD ;RESTORE POINTER
3988 017154 052777 000040 161352 EX5B: BIS #40,@MR ;CLOCK UP
3989 017162 017702 161346 MOV @MR,R2 ;READ MR
3990 017166 042702 177400 BIC #177400,R2 ;MASK OUT DATA
3991 017172 000300 SWAB R0 ;POSITION DATA
3992 017174 050002 BIS R0,R2 ;LOAD NEW DATA
3993 017176 010277 161332 MOV R2,@MR ;CLOCK DOWN AND LOAD NEW DATA
3994 017202 000426 BR EX5D
3995 017204 052777 000040 161322 EX5C: BIS #40,@MR ;CLOCK UP
3996 017212 042777 000040 161314 BIC #40,@MR ;CLOCK DOWN
3997 017220 017700 161310 MOV @MR,R0 ;GET MR

```



```
3998 017224 000300          SWAB  R0
3999 017226 032767 000010 161552  BIT  #10,UDES      ;SEE IF EVEN PAR
4000 017234 001405          BEQ  EX5C0        ;IF NOT: BR
4001 017236 010077 161530  MOV  R0,@RDAD
4002 017242 005267 161524  INC  RDAD
4003 017246 000402          BR   EX5C1
4004 017250 110077 161516  EX5C0: MOVB R0,@RDAD ;PUT CHAR IN CORE
4005 017254 005267 161512  EX5C1: INC  RDAD
4006 017260 005301          EX5D: DEC  R1      ;SEE IF DONE CLKS
4007 017262 001300          BNE  EX5A        ;IF NOT: BR
4008 017264 000167 002762  JMP  EORP        ;GO DO EOR
4009
4010          ;ASSURE NOT AT BOT FOR WRAP TESTS*****
4011
4012 017270 032777 000002 161224  BOTT: BIT  #2,ADS   ;SEE IF BOT
4013 017276 001414          BEQ  BOTTX        ;IF NOT: BR
4014 017300 052777 001700 161234  BIS  #1700,@TC   ;SET NRZ
4015 017306 012777 000025 161174  MOV  #25,@C1     ;DO ERASE
4016 017314 032777 000200 161200  BOTTA: BIT  #200,ADS
4017 017322 001774          BEQ  BOTTA
4018 017324 004767 004630          JSR  PC,INIT1   ;AWAIT DRY
4019 017330 000207          BOTTX: RTS  PC   ;INIT
4020          ;RETURN
```

```

4022                                     ;EXECUTE WAM 0*****
4023
4024 017332 012767 017504 161340 EXW2:  MOV    #EXW2H,RTRN    ;SET INTERRUPT RETURN ADDRESS
4025 017340 012701 000200                MOV    #200,R1      ;SET NUMBER OF CLOCKS = FC/2
4026 017344 032767 002000 161434        BIT    #2000,UDES   ;SEE IF PE
4027 017352 001402                BEQ    EXW2A        ;IF NOT: BR
4028 017354 012701 000100                MOV    #100,R1     ;ELSE SET CLKS = FC/4
4029 017360 016702 161406                EXW2A: MOV    RDAD,R2 ;SET BUFFER ADDRESS
4030 017364 012777 000161 161116        MOV    #161,@C1    ;SET WRITE+GO
4031 017372 005077 161172                CLR    @PSW        ;ALLOW INTERRUPT
4032 017376 032777 000040 161130 EXW2B:  BIT    #40,@MR      ;AWAIT CLOCK UP
4033 017404 001774                BEQ    EXW2B        ;GET DATA
4034 017406 017722 161122                MOV    @MR,(R2)+
4035 017412 032777 000040 161114 EXW2C:  BIT    #40,@MR      ;AWAIT CLOCK DOWN
4036 017420 001374                BNE    EXW2C        ;GET DATA
4037 017422 017722 161106                MOV    @MR,(R2)+  ;SEE IF DONE ALL
4038 017426 005301                DEC    R1           ;IF NOT: BR
4039 017430 001362                BNE    EXW2B
4040 017432 012701 000003                EXW2E: MOV    #3,R1
4041 017436 005000                CLR    R0           ;SET DELAY
4042 017440 005300                EXW2F:  DEC    R0
4043 017442 001376                BNE    EXW2F
4044 017444 005301                DEC    R1
4045 017446 001374                BNE    EXW2F        ;DELAY
4046 017450 032777 020000 161114        BIT    #20000,@SWR ;SEE IF ERROR PRINT
4047 017456 001004                BNE    EXW2G        ;IF NOT: BR
4048 017460 012704 035716                MOV    #WMSG24,R4
4049 017464 004767 005032                JSR    PC,TTOUT    ;PRINT NO INTERUPT
4050 017470 032777 100000 161074 EXW2G:  BIT    #100000,@SWR ;SEE IF HALT ON ERROR
4051 017476 001402                BEQ    EXW2H        ;IF NOT: BR
4052 017500 004767 006134                JSR    PC,STOP
4053 017504 012701 037014                EXW2H:  MOV    #RBUF,R1 ;GET START OF READ BUFFER
4054 017510 012700 000400                MOV    #400,R0    ;SET SIZE
4055 017514 010102                MOV    R1,R2
4056 017516 012203                EXW2J:  MOV    (R2)+,R3
4057 017520 000303                SWAB   R3
4058 017522 032767 000010 161256        BIT    #10,UDES    ;SEE IF EVEN PAR
4059 017530 001402                BEQ    EXW2J0       ;IF NOT: BR
4060 017532 010321                MOV    R3,(R1)+    ;SAVE PAR + DATA
4061 017534 000401                BR     EXW2J1
4062 017536 110321                EXW2J0: MOVB   R3,(R1)+ ;ASSEMBLE DATA IN BYTES
4063 017540 005300                EXW2J1: DEC    R0
4064 017542 001365                BNE    EXW2J        ;CONTINUE FOR ALL
4065 017544 032777 000200 161020        BIT    #200,@SWR   ;SEE IF STATUS CHECK
4066 017552 001002                BNE    EXW2K        ;IF NOT: BR
4067 017554 004767 000016                JSR    PC,WSTCK    ;ELSE GO CHECK STATUS
4068 017560 032777 000400 161004 EXW2K:  BIT    #400,@SWR ;SEE IF DATA CHECK
4069 017566 001002                BNE    EXW2X        ;IF NOT: BR
4070 017570 004767 000272                JSR    PC,DCHK     ;ELSE GO CHECK DATA
4071 017574 000207                EXW2X:  RTS     PC   ;EXIT
  
```

```
4073
4074
4075
4076 017576 005067 161200
4077 017602 005067 161022
4078 017606 012767 035540 161066
4079 017614 017702 160670
4080 017620 016705 161210
4081 017624 004767 000114
4082 017630 012767 035565 161044
4083 017636 017702 160656
4084 017642 016705 161170
4085 017646 056705 160762
4086 017652 004767 000066
4087 017656 012767 035573 161016
4088 017664 017702 160632
4089 017670 016705 161144
4090 017674 004767 000044
4091 017700 012767 035600 160774
4092 017706 017702 160612
4093 017712 016705 161124
4094 017716 004767 000022
4095 017722 005767 161054
4096 017726 001456
4097 017730 005777 160636
4098 017734 100053
4099 017736 004767 005676
4100 017742 000450
4101 017744 020205
4102 017746 001446
4103 017750 005267 161026
4104 017754 032777 020000 160610
4105 017762 001040
4106 017764 005767 160640
4107 017770 001010
4108 017772 016704 160634
4109 017776 004767 004520
4110 020002 012704 035702
4111 020006 004767 004510
4112 020012 012767 000001 160610
4113 020020 016704 160656
4114 020024 004767 004472
4115 020030 012704 026732
4116 020034 004767 004462
4117 020040 010503
4118 020042 004767 004616
4119 020046 012704 026741
4120 020052 004767 004444
4121 020056 010203
4122 020060 004767 004600
4123 020064 000207
4124
```

```

;WRAP AROUND STATUS CHECK ROUTINE*****
WSTCK: CLR SERFL ;CLEAR ERROR FLAG
CLR HDRFL ;CLEAR HEADER FLAG
MOV #WMSG6,ERADD ;SET CODE=CS1
MOV @C1,R2 ;GET RCVD CS1
MOV WCS1,R5 ;GET EXPT CS1
JSR PC,WSTG ;GO CHK
MOV #WMSG6D,ERADD ;SET CODE=CS2
MOV @CS,R2 ;SET RCVD CS2
MOV WCS2,R5 ;GET EXPT CS2
BIS DRVN,R5 ;SET DRIVE NUMBER IN EXPT CS2
JSR PC,WSTG ;GO CHK
MOV #WMSG6E,ERADD ;SET CODE=DS
MOV @DS,R2 ;SET RCVD DS
MOV WDS,R5 ;GET EXPT DS
JSR PC,WSTG ;GO CHK
MOV #WMSG6F,ERADD ;SET CODE=ER
MOV @ER,R2 ;GET RCVD ER
MOV WER,R5 ;GET EXPT ER
JSR PC,WSTG ;GO CHK
TST SERFL ;SEE IF ANY ERRORS
BEQ WSTX ;IF NOT: BR
TST @SWR ;SEE IF HALT ON ERROR
BPL WSTX ;IF NOT: BR
JSR PC,STOP
BR WSTX ;CONTINUE
WSTG: CMP R2,R5 ;SEE IF EXPT=RCVD
BEQ WSTX ;IF SO: BR
INC SERFL ;SET ERROR FLAG
BIT #20000,@SWR ;SEE IF PRINT ERRORS
BNE WSTX ;IF NOT: BR
TST HDRFL ;SEE IF DONE HEADER
BNE WSTG ;IF SO: BR
MOV EMADDR,R4 ;PRINT TEST HEADER
JSR PC,TTOUT
MOV #WMSG23,R4
JSR PC,TTOUT ;PRINT STATUS TAG
WSTG0: MOV #1,HDRFL ;SET HEADER FLAG
MOV ERADD,R4 ;PRINT CODE
JSR PC,TTOUT
MOV #MSG12,R4 ;PRINT EXPT TAG
JSR PC,TTOUT
MOV R5,R3 ;PRINT EXPT STATUS
JSR PC,OCTP
MOV #MSG13,R4 ;PRINT RCVD TAG
JSR PC,TTOUT
MOV R2,R3 ;PRINT RCVD S.ATUS
JSR PC,OCTP ;RETURN
WSTX: RTS PC
```

```

4126
4127
4128
4129 020066 005067 160536          DCHK: CLR      HDRFL      :CLEAR HEADER FLAG
4130 020072 005067 160700          CLR      DERFL      :CLEAR DATA ERROR FLAG
4131 020076 005067 160702          CLR      CRCNT      :CLEAR CHAR CNTR
4132 020102 032767 000010 160676  BIT      #10,UDES    :SEE IF EVEN PARITY
4133 020110 001402                    BEQ      DCHKA0      :IF NOT: BR
4134 020112 000167 000602          JMP      DCHKE      :ELSE GO CHECK EVEN
4135 020116 022767 000010 160634  DCHKA0: CMP      #10,WAM    :SEE IF WAM 1
4136 020124 001006                    BNE      DCHKA      :IF NOT: BR
4137 020126 032767 002000 160652  BIT      #2000,UDES  :SEE IF PE
4138 020134 001402                    BEQ      DCHKA      :IF NOT: BR
4139 020136 000167 001202          JMP      PRCHK      :GO CHK DATA
4140 020142 012700 177400          DCHKA: MOV      #-400,R0  :SET NUMBER OF CHARACTERS
4141 020146 022767 000012 160604  CMP      #12,WAM
4142 020154 001006                    BNE      DCHKA1     :IF NOT WRAP 2: BR
4143 020156 032767 000020 160622  BIT      #20,UDES
4144 020164 001402                    BEQ      DCHKA1     :IF NOT CORE DUMP: BR
4145 020166 012700 177000          MOV      #-1000,R0
4146 020172 022767 000006 160560  DCHKA1: CMP      #6,WAM
4147 020200 001007                    BNE      DCHKA2
4148 020202 032767 002000 160576  BIT      #2000,UDES  :SEE IF PE MODE
4149 020210 001430                    BEQ      DCHKB      :IF NOT: BR
4150 020212 012700 177600          MOV      #-200,R0   :SET CHAR CNTR TO FC/2 FOR PE
4151 020216 000425                    BR       DCHKB
4152 020220 022767 000012 160532  DCHKA2: CMP      #12,WAM    :SEE IF WRAP 2
4153 020226 001021                    BNE      DCHKB      :IF NOT: BR
4154 020230 032767 002000 160550  BIT      #2000,UDES  :SEE IF PE
4155 020236 001415                    BEQ      DCHKB      :IF NOT: BR
4156 020240 012700 177653          MOV      #-125,R0   :POINT TO START OF DATA
4157 020244 012767 000001 160522  MOV      #1,W2FLG    :SET WRAP 2 FLAG
4158 020252 004767 000014          JSR      PC,DCHKB   :GO CHECK DATA
4159 020256 004767 001362          JSR      PC,W1DCHK  :GO CHECK WRAP 1 DATA
4160 020262 005067 160506          CLR      W2FLG
4161 020266 000167 000362          JMP      DCHKX
4162 020272 005067 160500          DCHKB: CLR      DERFL
4163 020276 012701 036402          MOV      #WBUF,R1   :SET GOOD POINTER
4164 020302 012702 037014          MOV      #RBUF,R2   :SET READ POINTER
4165 020306 032767 000020 160472  BIT      #20,UDES    :SEE IF CORE DUMP
4166 020314 001416                    BEQ      DCHK0      :IF NOT: BR
4167 020316 022767 000012 160434  CMP      #12,WAM    :SEE IF WAM 2
4168 020324 001011                    BNE      DCHKD      :IF NOT: BR
4169 020326 005767 160460          TST      PATRN      :SEE IF PATTERN 0
4170 020332 001003                    BNE      DCHKC      :IF NOT: BR
4171 020334 012701 001070          MOV      #WCDP0,R1  :SET CORE DUMP PATTERN 0
4172 020340 000404                    BR       DCHK0      :GO CHECK DATA
4173 020342 012701 001056          DCHKC: MOV      #WCDP2,R1 :SET CORE DUMP WRITE PATTERN 2
4174 020346 000401                    BR       DCHK0      :GO CHECK DATA
4175 020350 000240          DCHKD: NOP
4176 020352 121112          DCHK0: CMPB     (R1),(R2) :SEE IF DATA OK
4177 020354 001466                    BEQ      DCHK2      :IF SO: BR
4178 020356 032777 020000 160206  BIT      #20000,@SWR :SEE IF PRINT ERRORS
4179 020364 001062                    BNE      DCHK2      :IF NOT: BR
4180 020366 005767 160236          TST      HDRFL      :SEE IF DONE HEADER
4181 020372 001004                    BNE      DCHK1      :IF SO: BR

```

4182	020374	016704	160232		MOV	EMADDR,R4	
4183	020400	004767	004116		JSR	PC,TTOUT	:PRINT HEADER
4184	020404	005767	160366		DCHK1: TST	DERFL	:SEE IF FIRST ERROR
4185	020410	001014			BNE	DCHK1A	:IF NOT: BR
4186	020412	012704	035650		MOV	#WMSG16,R4	
4187	020416	004767	004100		JSR	PC,TTOUT	:PRINT DATA ERROR TAG
4188	020422	012704	036075		MOV	#WMSG32,R4	
4189	020426	004767	004070		JSR	PC,TTOUT	:PRINT PATRN TAG
4190	020432	016703	160354		MOV	PATRN,R3	
4191	020436	004767	004222		JSR	PC,OCTP	:PRINT PATTERN NUMBER
4192	020442	012767	000001	160160	DCHK1A: MOV	#1,HDRFL	:SET HEADER FLAG
4193	020450	012767	000001	160320	MOV	#1,DERFL	:SET DATA ERROR FLAG
4194	020456	012704	035674		MOV	#WMSG21,R4	
4195	020462	004767	004034		JSR	PC,TTOUT	:PRINT CHARACTER NUMBER TAG
4196	020466	016703	160312		MOV	CRCNT,R3	
4197	020472	004767	004166		JSR	PC,OCTP	:PRINT CHARACTER NUMBER
4198	020476	012704	035662		MOV	#WMSG17,R4	
4199	020502	004767	004014		JSR	PC,TTOUT	:PRINT GOOD TAG
4200	020506	111103			MOVB	(R1),R3	
4201	020510	004767	004714		JSR	PC,DOUT	:PRINT GOOD DATA
4202	020514	012704	035667		MOV	#WMSG20,R4	
4203	020520	004767	003776		JSR	PC,TTOUT	:PRINT BAD TAG
4204	020524	111203			MOVB	(R2),R3	
4205	020526	004767	004676		JSR	PC,DOUT	:PRINT BAD DATA
4206	020532	005767	160236		DCHK2: TST	W2FLG	
4207	020536	001020			BNE	DCHK2B	
4208	020540	005201			INC	R1	:BUMP POINTER
4209	020542	032767	000020	160236	BIT	#20,UDES	:SEE IF CORE DUMP
4210	020550	001413			BEQ	DCHK2B	:IF NOT: BR
4211	020552	022767	000012	160200	CMP	#12,WAM	:SEE IF WAM 2
4212	020560	001006			BNE	DCHK2A	:IF NOT: BR
4213	020562	005201			INC	R1	:BUMP POINTER
4214	020564	005711			TST	(R1)	:SEE IF END OF PATTERN
4215	020566	001004			BNE	DCHK2B	:IF NOT: BR
4216	020570	162701	000010		SUB	#10,R1	:RESET POINTER TO START OF PATTERN
4217	020574	000401			BR	DCHK2B	:CONTINUE CHECK
4218	020576	000240			DCHK2A: NOP		
4219	020600	005202			DCHK2B: INC	R2	
4220	020602	022767	000006	160150	CMP	#6,WAM	:SEE IF WAM 0
4221	020610	001005			BNE	DCHK3	
4222	020612	032767	002000	160166	BIT	#2000,UDES	:SEE IF PE
4223	020620	001401			BEQ	DCHK3	:IF NOT PE: BR
4224	020622	005201			INC	R1	:BUMP WRITE DATA ADDRESS
4225	020624	005267	160154		DCHK3: INC	CRCNT	:BUMP CHAR CNTR
4226	020630	032777	000100	157734	BIT	#400,@SWR	:SEE IF CONT DATA CHK
4227	020636	001006			BNE	DCHKX	:IF NOT: BR
4228	020640	005200			INC	R0	:SEE IF DONE
4229	020642	001243			BNE	DCHK0	:IF NOT: BR
4230	020644	005767	160124		TST	W2FLG	
4231	020650	001401			BEQ	DCHKX	
4232	020652	000207			RTS	PC	
4233	020654	032777	100000	157710	DCHKX: BIT	#100000,@SWR	:SEE IF HALT ON ERROR
4234	020662	001405			BEQ	DCHKX1	:IF NOT: BR
4235	020664	005767	160106		TST	DERFL	:SEE IF DATA ERROR OCCURED
4236	020670	001402			BEQ	DCHKX1	:IF NOT: BR
4237	020672	004767	004742		JSR	PC,STOP	

CZTUGO TM02/TU16 CTRL LGC
CZTUCG.P11 12-JUL-83 10:41

MACY11 30(1046) 12-JUL-83 10:47 M 9
PAGE 104-2

SEQ 0116

4238	020676	005067	160102
4239	020702	005067	157722
4240	020706	005067	160064
4241	020712	005067	160062
4242	020716	000207	

DCHKX1:	CLR	CRCNT	:CLEAR CHAR CNTR
	CLR	HDRFL	:CLEAR HEADER FLAG
	CLR	DERFL	:CLEAR DATA ERROR FLAG
	CLR	PREFL	:CLEAR PREAMBLE FLAG
	RTS	PC	:RETURN

```

4244
4245 ;EVEN PARITY DATA CHECK*****
4246
4247 020720 012700 177400 DCHKE: MOV #-400,R0 ;SET NUMBER OF CHARACTERS
4248 020724 012701 036402 MOV #WBUF,R1 ;R1=START OF WRITE BUFFER
4249 020730 012702 037014 MOV #RBUF,R2 ;R2=START OF READ BUFFER
4250 020734 111105 DCKE0: MOV (R1),R5 ;GET EXPT DATA
4251 020736 005003 CLR R3
4252 020740 012704 000010 DCKE1: MOV #10,R4 ;SET NUMBER OF BITS
4253 020744 032705 000001 BIT #1,R5 ;SEE IF ONE BIT
4254 020750 001401 BEQ DCKE2 ;IF NOT: BR
4255 020752 005203 INC R3 ;COUNT ONE BITS FOR PARITY CHECK
4256 020754 005304 DCKE2: DEC R4 ;SEE IF DONE
4257 020756 001402 BEQ DCKE3 ;IF SO: BR
4258 020760 006005 ROR R5 ;POINT TO NEXT BIT
4259 020762 000770 BR DCKE1
4260 020764 111105 DCKE3: MOV (R1),R5 ;GET EXPT DATA
4261 020766 042705 177400 BIC #177400,R5 ;MASK DATA FIELD
4262 020772 005703 TST R3
4263 020774 001003 BNE DCKE4 ;IF NO ONE BITS SET: BR
4264 020776 012705 100020 MOV #100020,R5
4265 021002 000405 BR DCKE5
4266 021004 032703 000001 DCKE4: BIT #1,R3 ;SEE IF ODD NUMBER OF ONE BITS
4267 021010 001402 BEQ DCKE5 ;IF NOT: BR
4268 021012 052705 100000 BIS #100000,R5 ;SET EVEN PARITY BIT=1
4269 021016 042712 077400 DCKE5: BIC #77400,(R2) ;MASK DATA FIELD
4270 021022 020512 CMP R5,(R2) ;SEE IF DATA + PARITY GOOD
4271 021024 001474 BEQ DCKE10 ;IF SO: BR
4272 021026 032777 020000 157536 BIT #20000,@SWR ;SEE IF ERROR PRINT
4273 021034 001070 BNE DCKE10 ;IF NOT: BR
4274 021036 005767 157566 TST HDRFL ;SEE IF DONE HEADER
4275 021042 001004 BNE DCKE6 ;IF SO: BR
4276 021044 016704 157562 MOV EMADDR,R4
4277 021050 004767 003446 JSR PC,TTOUT ;PRINT HEADER
4278 021054 005767 157716 DCKE6: TST DERFL ;SEE IF FIRST BAD CHAR
4279 021060 001014 BNE DCKE7 ;IF NOT: BR
4280 021062 012704 035650 MOV #WMSG16,R4
4281 021066 004767 003430 JSR PC,TTOUT ;PRINT BAD DATA TAG
4282 021072 012704 036075 MOV #WMSG32,R4
4283 021076 004767 003420 JSR PC,TTOUT ;PRINT PATTERN TAG
4284 021102 016703 157704 MOV PATRN,R3
4285 021106 004767 003552 JSR PC,OCIP ;PRINT PATTERN NUMBER
4286 021112 012767 000001 157656 DCKE7: MOV #1,DERFL ;SET DATA ERROR FLAG
4287 021120 012767 000001 157502 MOV #1,HDRFL ;SET HEADER FLAG
4288 021126 012704 035674 MOV #WMSG21,R4
4289 021132 004767 003364 JSR PC,TTOUT ;PRINT CHAR NUMBER TAG
4290 021136 016703 157642 MOV CRCNT,R3
4291 021142 004767 003516 JSR PC,OCIP ;ORINT CHAR NUMBER
4292 021146 012704 035662 MOV #WMSG17,R4
4293 021152 004767 003344 JSR PC,TTOUT ;PRINT GOOD DATA TAG
4294 021156 110503 MOV R5,R3
4295 021160 004767 004244 JSR PC,DCOUT ;PRINT EXPT DATA
4296 021164 010503 MOV R5,R3
4297 021166 004767 000064 JSR PC,DCKEP ;GO PRINT PARITY BIT
4298 021172 012704 035667 MOV #WMSG20,R4
4299 021176 004767 003320 JSR PC,TTOUT ;PRINT BAD TAG

```

4300	021202	111203			MOV8	(R2),R3	
4301	021204	004767	004220		JSP	PC,DOUT	:PRINT BAD DATA
4302	021210	011203			MOV	(R2),R3	
4303	021212	004767	000040		JSR	PC,DCKEP	:GO PRINT PARITY BIT
4304	021216	005201			INC	R1	
4305	021220	005722			TST	(R2)+	:BUMP POINTERS
4306	021222	005267	157556		INC	CRCNT	:BUMP CHAR CNTR
4307	021226	032777	000400	157336	BIT	#400,@SWR	:SEE IF CONTINUE DATA CHECK
4308	021234	001402			BEQ	DCKE11	:IF SO: BR
4309	021236	000167	177412		JMP	DCHKX	:GO TO END OF DATA CHECK
4310	021242	005200			INC	R0	:SEE IF DONE
4311	021244	001402			BEQ	DCKE12	:IF SO: BR
4312	021246	000167	177462		JMP	DCKE0	:ELSE CONTINUE
4313	021252	000167	177376		DCKE12: JMP	DCHKX	:GO TO END OF DATA CHECK
4314	021256	012767	000240	157340	DCKEP: MOV	#240,TOB	
4315	021264	004767	003346		JSR	PC,TOG	:SPACE
4316	021270	012767	000260	157326	MOV	#260,TOB	:SET PAR=0
4317	021276	005703			TST	R3	:SEE IF PARITY REALLY=0
4318	021300	100002			BPL	DCKEP0	:IF SO: BR
4319	021302	005267	157316		INC	TOB	:ELSE SET TO 1
4320	021306	004767	003324		DCKEP0: JSR	PC,TOG	:PRINT PARITY BIT
4321	021312	000207			RTS	PC	:RETURN
4322							


```

4324
4325                                     ;PREAMBLE/POSTAMPLE CHECK*****
4326
4327 021314 012700 000051      PSCHK: MOV      #51,R0      ;SET SIZE OF POSTAMPLE
4328 021320 012701 036260      MOV      #POST,R1     ;SET POINTER TO POSTAMPLE
4329 021324 005067 157300      CLR      HDRFL        ;CLEAR HEADER FLAG
4330 021330 005067 157450      CLR      CRCNT        ;CLEAR CHAR CNTR
4331 021334 005067 157436      CLR      DERFL        ;CLEAR DATA ERROR FLAG
4332 021340 000167 000016      JMP      PDO          ;GO CHECK POSTAMPLE
4333
4334 021344 012700 000051      PRCHK: MOV      #51,R0      ;SET SIZE OF PREAMBLE
4335 021350 012701 036136      MOV      #PRE,R1     ;SET POINTER TO PREAMBLE
4336 021354 012702 037014      MOV      #RBUF,R2     ;SET POINTER TO START OF READ BUFFER
4337 021360 022122              CMP      (R1)+,(R2)+  ;BUMP ADDRESS POINTERS
4338 021362 121112              PDO:  CMPB     (R1),(R2)  ;CHECK DATA
4339 021364 001004              BNE     PD1          ;IF NOT GOOD: BR
4340 021366 126162 000001 000001  CMPB     1(R1),1(R2)  ;COMPARE COMPLIMENT BYTE
4341 021374 001477              BEQ     PD5          ;IF GOOD: BR
4342 021376 032777 020000 157166 PD1:  BIT      #20000,@SWR  ;SEE IF PRINT INHIBIT
4343 021404 001073              BNE     PD5          ;IF SO: BR
4344 021406 005767 157216      TST     HDRFL        ;SEE IF DONE HEADER
4345 021412 001020              BNE     PD4          ;IF SO: BR
4346 021414 016704 157212      MOV     EMADDR,R4
4347 021420 004767 003076      JSR     PC,TTOUT     ;PRINT TEST HEADER
4348 021424 005767 157350      TST     PREFL        ;SEE IF PREAMBLE CHECK
4349 021430 001403              BEQ     PD2          ;IF NOT: BR
4350 021432 012704 036021      MOV     #WMSG29,R4   ;SET POSTAMPLE HEADER
4351 021436 000402              BR      PD3
4352 021440 012704 036003      PD2:  MOV     #WMSG28,R4 ;SET PREAMBLE HEADER
4353 021444 004767 003052      PD3:  JSR     PC,TTOUT ;PRINT HEADER
4354 021450 005267 157154      INC     HDRFL
4355 021454 012704 035674      PD4:  MOV     #WMSG21,R4
4356 021460 004767 003036      JSR     PC,TTOUT     ;PRINT CHAR NUMBER TAG
4357 021464 016703 157314      MOV     CRCNT,R3
4358 021470 004767 003170      JSR     PC,OCTP      ;PRINT CHAR NUMBER
4359 021474 012704 035662      MOV     #WMSG17,R4
4360 021500 004767 003016      JSR     PC,TTOUT     ;PRINT GOOD TAG
4361 021504 116103 000001      MOVB    1(R1),R3
4362 021510 004767 003714      JSR     PC,DOUT      ;PRINT GOOD CHAR
4363 021514 012767 000240 157102  MOV     #240,TOB
4364 021522 004767 003110      JSR     PC,TOG
4365 021526 111103      MOVB    (R1),R3
4366 021530 004767 003674      JSR     PC,DOUT      ;PRINT COMPLIMENT
4367 021534 012704 035667      MOV     #WMSG20,R4
4368 021540 004767 002756      JSR     PC,TTOUT     ;PRINT BAD TAG
4369 021544 116203 000001      MOVB    1(R2),R3
4370 021550 004767 003654      JSR     PC,DOUT      ;PRINT BAD CHAR
4371 021554 012767 000240 157042  MOV     #240,TOB
4372 021562 004767 003050      JSR     PC,TOG
4373 021566 111203      MOVB    (R2),R3
4374 021570 004767 003634      JSR     PC,DOUT      ;PRINT COMPLIMENT
4375 021574 022122      PD5:  CMP      (R1)+,(R2)+  ;BUMP ADDRESS POINTERS
4376 021576 005267 157202      INC     CRCNT        ;BUMP CHAR NUMBER
4377 021602 005300      DEC     R0           ;SEE IF DONE
4378 021604 001266      BNE     PDO          ;IF NOT: BR
4379 021606 005767 157166      TST     PREFL        ;SEE IF PREAMBLE

```

CZTUGO TMO2/TU16 CTRL LGC
CZTUCG.P11 12-JUL-83 10:41

MACY11 30(1046) 12-JUL-83 10:47 D 10
PAGE 106-1

SEQ 0120

4380 021612 001402
4381 021614 000167 177034
4382 021620 005267 157154
4383 021624 005067 157000
4384 021630 005067 157150
4385 021634 005067 157136
4386 021640 000167 000000
4387

PD6:

BEQ PD6
JMP DCHKX
INC PREFL
CLR HDRFL
CLR CRCNT
CLR DERFL
JMP WIDCHK

:IF SO: BR
:GO TO EXIT ROUTINE
:SET PREAMBLE FLAG
:CLEAR HEADER FLAG
:CLEAR CHAR CNTR
:CLEAR DATA ERROR FLAG
:GO CHECK WRAP 1 DATA

```
4389
4390
4391
4392 021644 012700 177400
4393 021650 012701 036402
4394 021654 012702 037014
4395 021660 062702 000124
4396 021664 005767 157104
4397 021670 001401
4398 021672 005302
4399 021674 111105
4400 021676 120512
4401 021700 001007
4402 021702 005767 157066
4403 021706 001001
4404 021710 105105
4405 021712 120562 000001
4406 021716 001510
4407 021720 032777 020000 156644
4408 021726 001104
4409 021730 005767 156674
4410 021734 001020
4411 021736 016704 156670
4412 021742 004767 002554
4413 021746 012704 035650
4414 021752 004767 002544
4415 021756 012704 036075
4416 021762 004767 002534
4417 021766 016703 157020
4418 021772 004767 002666
4419 021776 012767 000001 156624
4420 022004 012704 035674
4421 022010 004767 002506
4422 022014 016703 156764
4423 022020 004767 002640
4424 022024 012704 035662
4425 022030 004767 002466
4426 022034 111105
4427 022036 110503
4428 022040 005767 156730
4429 022044 001001
4430 022046 105103
4431 022050 004767 003354
4432 022054 012767 000240 156542
4433 022062 004767 002550
4434 022066 110503
4435 022070 004767 003334
4436 022074 012704 035667
4437 022100 004767 002416
4438 022104 116203 000001
4439 022110 004767 003314
4440 022114 012767 000240 156502
4441 022122 004767 002510
4442 022126 111203
4443 022130 004767 003274
4444 022134 005267 156636
```

;WAM 1 PE DATA CHECK*****

```
W1DCHK: MOV #-400,R0 ;SET NUMBER OF CHAR TO CHECK
MOV #WBUF,R1 ;SET WRITE DATA POINTER
MOV #RBUF,R2 ;SET READ DATA POINTER
ADD #124,R2 ;POINT TO START OF DATA
TST W2FLG ;SEE IF WRAP 2
BEQ W1D0 ;IF NOT WAM 2: BR
DEC R2 ;RESET POINTER

W1D0: MOVB (R1),R5 ;CHECK DATA
CMPB R5,(R2) ;IF NOT GOOD:BR
BNE W1D1 ;SEE IF WRAP 2
TST W2FLG ;IF SO: BR
BNE W1D0A ;COMPLIMENT EXPT DATA

W1D0A: CMPB R5,1(R2) ;CHECK COMPLIMENT DATA
BEQ W1D3 ;IF GOOD: BR
BIT #20000,@SWR ;SEE IF PRINT INHIBIT
BNE W1D3 ;IF SO: BR
TST HDRFL ;SEE IF DONE HEADER
BNE W1D2 ;IF SO: BR

MOV EMADDR,R4
JSR PC,TTOUT ;PRINT TEST HEADER
MOV #WMSG16,R4
JSR PC,TTOUT ;PRINT BAD DATA TAG
MOV #WMSG32,R4
JSR PC,TTOUT ;PRINT PATRN TAG
MOV PATRN,R3
JSR PC,OCIP ;PRINT PATTERN NUMBER

W1D2: MOV #1,HDRFL ;SET HEADER FLAG
MOV #WMSG21,R4
JSR PC,TTOUT ;PRINT CHAR NUMBER TAG
MOV CRCNT,R3
JSR PC,OCIP ;PRINT CHAR NUMBER
MOV #WMSG17,R4
JSR PC,TTOUT ;PRNT GOOD TAG
MOVB (R1),R5
MOVB R5,R3 ;GET GOOD CHAR
TST W2FLG ;SEE IF WRAP 2
BNE W1D2A ;IF SO: BR
COMB R3 ;ELSE COMPLIMENT CHAR

W1D2A: JSR PC,DOUT ;PRINT CHARACTER
MOV #240,TOB
JSR PC,TOG ;SPACE
MOVB R5,R3
JSR PC,DOUT ;PRINT CHAR
MOV #WMSG20,R4
JSR PC,TTOUT ;PRINT BAD TAG
MOVB 1(R2),R3
JSR PC,DOUT ;PRINT BAD CHAR
MOV #240,TOB
JSR PC,TOG ;SPACE
MOVB (R2),R3
JSR PC,DOUT ;PRINT CHAR
INC DERFL ;SET DATA ERROR FLAG
```

4445	022140	122122		W1D3:	CMPB	(R1)+,(R2)+	:BUMP ADDRESS
4446	022142	105722			TSTB	(R2)+	:BUMP ADDRESS
4447	022144	005267	156634		INC	CRCNT	:BUMP CHAR CNTR
4448	022150	000406			BR	W1D5	
4449	022152	005767	156616	W1D4:	TST	W2FLG	:SEE IF WRAP 2
4450	022156	001401			BEQ	W1D4A	:IF NOT: BR
4451	022160	000207			RTS	PC	:ELSE RETURN
4452	022162	000167	177126	W1D4A:	JMP	PSCHK	:GO CHECK POSTAMBLE
4453	022166	005200		W1D5:	INC	R0	
4454	022170	001770			BEQ	W1D4	
4455	022172	000167	177476		JMP	W1D0	

4456
4457 ;PREAMBLE/POSTAMBLE GENERATE SUBROUTINE*****
4458

4459	022176	012700	000050	PPGEN:	MOV	#50,R0	:SET SIZE OF PREAMBLE
4460	022202	012701	036136		MOV	#PRE,R1	
4461	022206	005721			TST	(R1)+	:SET ADDRESS OF PRE
4462	022210	012721	177400	PPG0:	MOV	#177400,(R1)+	:FILL TABLE
4463	022214	005300			DEC	R0	:SEE IF DONE
4464	022216	001374			BNE	PPG0	:IF NOT: BR
4465	022220	012701	036260		MOV	#POST,R1	:SET ADDRESS OF POST
4466	022224	012700	000050		MOV	#50,R0	:SET SIZE OF POST
4467	022230	012721	000377		MOV	#377,(R1)+	:SET SYNC CHAR
4468	022234	012721	177400	PPG1:	MOV	#177400,(R1)+	:FILL TABLE
4469	022240	005300			DEC	R0	:SEE IF DONE
4470	022242	001374			BNE	PPG1	:IF NOT: BR
4471	022244	000207			RTS	PC	:RETURN

```
4473
4474
4475
4476 022246 005267 156434 EORPA: INC TEMP2 ;SET WRAP FLAG
4477 022252 017700 156256 EORP: MOV @MR,R0 ;GET MAINT REG
4478 022256 042700 000036 BIC #36,R0 ;CLEAR CURRENT OP CODE
4479 022262 052700 000024 BIS #24,R0 ;SET EOR CLEAR OP CODE
4480 022266 010077 156242 MOV R0,@MR ;DO EOR
4481 022272 042777 000037 156234 BIC #37,@MR ;CLEAR EOR AND MM
4482 022300 005000 CLR R0
4483 022302 012701 000002 MOV #2,R1
4484 022306 032777 000001 156174 EORP1: BIT #1,@C1 ;SEE IF GO GONE
4485 022314 001431 BEQ EORP2 ;IF SO: BR
4486 022316 005300 DEC R0
4487 022320 001372 BNE EORP1 ;AWAIT GO RESET
4488 022322 005301 DEC R1
4489 022324 001370 BNE EORP1
4490 022326 032777 020000 156236 BIT #20000,@SWR ;SEE IF ERROR PRINT INHIBIT
4491 022334 001021 BNE EORP2 ;IF SO: BR
4492 022336 005767 156266 TST HDRFL ;SEE IF DONE HEADER
4493 022342 001004 BNE EORP1A ;IF SO: BR
4494 022344 016704 156262 MOV EMADDR,R4
4495 022350 004767 002146 JSR PC,TTOUT ;PRINT HEADER
4496 022354 012704 036040 EORP1A: MOV #WMSG31,R4
4497 022360 004767 002136 JSR PC,TTOUT ;PRINT EOR GO BIT ERROR
4498 022364 032777 100000 156200 BIT #100000,@SWR ;SEE IF HALT ON ERROR
4499 022372 001402 BEQ EORP2 ;IF NOT: BR
4500 022374 004767 003240 JSR PC,STOP
4501 022400 005767 156302 EORP2: TST TEMP2 ;SEE IF WAM
4502 022404 001014 BNE EORPX ;IF NOT: BR
4503 022406 032777 000200 156156 BIT #200,@SWR ;SEE IF STATUS CHECK
4504 022414 001002 BNE EORP3 ;IF NOT: BR
4505 022416 004767 175154 JSR PC,WSTCK ;ELSE GO CHECK STATUS
4506 022422 032777 000400 156142 EORP3: BIT #400,@SWR ;SEE IF DATA CHECK
4507 022430 001002 BNE EORPX ;IF NOT: BR
4508 022432 004767 175430 JSR PC,DCHK ;ELSE GO CHECK DATA
4509 022436 005067 156244 EORPX: CLR TEMP2 ;CLEAR FLAG
4510 022442 000207 RTS PC ;RETURN
4511
```

```

4513                                     ;LOGIC TEST ADDRESSING ERROR SUBROUTINE*****
4514
4515 022444 005067 156256          LTGER3: CLR      EXFL
4516 022450 012767 030135 156224  MOV      #MSG51,ERADD
4517 022456 012767 000001 156272  LTGER0: MOV      #1,ADDFL      ;SET NO ADDRESS FLAG
4518 022464 005067 156206          LTGER:  CLR      PFLG        ;CLEAR PRINT FLAG
4519 022470 032777 020000 156074  BIT      #20000,@SWR      ;SEE IF SHOULD PRINT
4520 022476 001402          BEQ      LTGA          ;IF SO: BR
4521 022500 000167 000224          JMP      LTGX          ;ELSE GO TO EXIT
4522 022504 005767 156120          LTGA:  TST      HDRFL      ;SEE IF PRINTED HEADER
4523 022510 001004          BNE      LTGA1         ;IF SO: BR
4524 022512 016704 156114          MOV      EMADDR,R4
4525 022516 004767 002000          JSR      PC,TTOUT      ;PRINT TEST HEADER
4526 022522 012767 000001 156100  LTGA1: MOV      #1,HDRFL      ;SET HEADER FLAG
4527 022530 016704 156146          MOV      ERADD,R4
4528 022534 004767 001762          JSR      PC,TTOUT      ;PRINT CONDITION ERROR
4529 022540 005767 156212          TST      ADDFL
4530 022544 001003          BNE      LTGA2
4531 022546 010103          MOV      R1,R3
4532 022550 004767 002110          JSR      PC,OCTP      ;PRINT ADDRESS
4533 022554 005767 156146          LTGA2: TST      EXFL
4534 022560 001412          BEQ      LTGC          ;IF NO STATUS: BR
4535 022562 012704 026434          MOV      #MSG6,R4
4536 022566 022767 000001 156132  CMP      #1,EXFL      ;EXPT-NOT RCVD
4537 022574 001402          BEQ      LTGB
4538 022576 012704 026453          MOV      #MSG7,R4      ;RCVD-NOT EXPT
4539 022602 004767 001714          LTGB:  JSR      PC,TTOUT  ;PRINT STATUS
4540 022606 005267 156064          LTGC:  INC      PFLG
4541 022612 005767 156140          TST      ADDFL      ;SEE IF ADD TST
4542 022616 001430          BEQ      LTGD          ;IF SO: BR
4543 022620 005767 156130          TST      T24FL      ;SEE IF TEST 24
4544 022624 001423          BEQ      LTGCO        ;IF NOT: BR
4545 022626 012704 035770          MOV      #WMSG27,R4
4546 022632 004767 001664          JSR      PC,TTOUT      ;PRINT DATA TAG
4547 022636 012704 026732          MOV      #MSG12,R4
4548 022642 004767 001654          JSR      PC,TTOUT      ;PRINT EXPT TAG
4549 022646 012703 177777          MOV      #-1,R3
4550 022652 004767 001776          JSR      PC,OCTPE      ;PRINT EXPT
4551 022656 012704 026741          MOV      #MSG13,R4
4552 022662 004767 001634          JSR      PC,TTOUT      ;PRINT RCVD TAG
4553 022666 010103          MOV      R1,R3      ;GET RCVD
4554 022670 004767 001760          JSR      PC,OCTPE      ;PRINT RCVD
4555 022674 004767 000102          LTGCO: JSR      PC,REGP      ;PRINT REGISTERS
4556 022700 032777 010000 155664  LTGD:  BIT      #10000,@SWR
4557 022706 001010          BNE      LTGX
4558 022710 012704 027003          MOV      #MSG16,R4
4559 022714 004767 001602          JSR      PC,TTOUT
4560 022720 016703 155766          MOV      ITCNT,R3      ;PRINT ITERATION
4561 022724 004767 001734          JSR      PC,OCTP
4562 022730 005777 155636          LTGX:  TST      @SWR
4563 022734 100002          BPL      LTGXA
4564 022736 004767 002676          JSR      PC,STOP
4565 022742 005767 155730          LTGXA: TST      PFLG
4566 022746 001006          BNE      LTGXX
4567 022750 032777 020000 155614  BIT      #20000,@SWR
4568 022756 001002          BNE      LTGXX      ;IF STILL NO PRINT: BR

```

4569 022760 000167 177520
4570 022764 005067 155766
4571 022770 005067 155732
4572 022774 000167 001044
4573 023000 000207
4574
4575
4576
4577 023002 012704 027760
4578 023006 004767 001510
4579 023012 017703 155472
4580 023016 004767 001632
4581 023022 017703 155464
4582 023026 004767 001622
4583 023032 017703 155456
4584 023036 004767 001612
4585 023042 017703 155450
4586 023046 004767 001602
4587 023052 017703 155442
4588 023056 004767 001572
4589 023062 017703 155434
4590 023066 004767 001562
4591 023072 017703 155426
4592 023076 004767 001552
4593 023102 017703 155420
4594 023106 004767 001542
4595 023112 017703 155416
4596 023116 004767 001532
4597 023122 017703 155414
4598 023126 004767 001522
4599 023132 000207
4600
4601

LTGXX: JMP LTGA ;ELSE GO PRINT ERROR
CLR ADDFL ;CLEAR ADDRESS FLAG
CLR EXFL
JMP SCOPE
RTS PC ;EXIT

;SUBROUTINE TO PRINT MAJOR REGISTERS*****
REGP: MOV #MSG46,R4
JSR PC,TTOUT ;PRINT REGISTER HEADER
MOV @C1,R3
JSR PC,OCTPE
MOV @WC,R3
JSR PC,OCTPE
MOV @BA,R3
JSR PC,OCTPE
MOV @FC,R3
JSR PC,OCTPE
MOV @CS,R3
JSR PC,OCTPE
MOV @DS,R3 ;PRINT REGISTERS
JSR PC,OCTPE
MOV @ER,R3
JSR PC,OCTPE
MOV @AS,R3
JSR PC,OCTPE
MOV @MR,R3
JSR PC,OCTPE
MOV @TC,R3
JSR PC,OCTPE
RTS PC

```
4603 ;DRIVE CLEAR SUBROUTINE*****
4604
4605 023134 012704 040000 DRVCLR: MOV #40000,R4
4606 023140 005304 DCD: DEC R4
4607 023142 001376 BNE DCD ;DELAY
4608 023144 005067 155526 CLR PFLG
4609 023150 004767 000222 JSR PC,ATTN ;GO SEE OF ATTN SET
4610 023154 012777 000011 155326 MOV #11,@C1 ;ISSUE DRIVE CLEAR
4611 023162 005000 CLR R0
4612 023164 032777 000200 155330 DCA: BIT #200,@DS ;SEE IF DRY
4613 023172 001002 BNE DCA0
4614 023174 005300 DEC R0
4615 023176 001372 BNE DCA ;WAIT FOR DRY
4616 023200 032777 040000 155314 DCA0: BIT #40000,@DS ;SEE IF ERR RESET
4617 023206 001024 BNE DCE ;IF NOT: BR
4618 023210 005777 155310 TST @ER ;SEE IF ERROR REGISTER RESET
4619 023214 001021 BNE DCE ;IF NOT: BR
4620 023216 005777 155300 TST @DS ;SEE IF ATA RESET
4621 023222 100416 BMI DCE ;IF NOT: BR
4622 023224 012703 000001 MOV #1,R3 ;SET TEST BIT
4623 023230 016704 155400 MOV DRVN,R4 ;GET DRIVE NUMBER
4624 023234 005704 TST R4 ;SEE IF DRIVE 0
4625 023236 001404 BEQ DCC ;IF SO: BR
4626 023240 000241 DCB: CLC
4627 023242 006103 ROL R3 ;POSITION TEST BIT PER DRIVE NUMBER
4628 023244 005304 DEC R4 ;SEE IF DONE
4629 023246 001374 BNE DCB ;IF NOT: BR
4630 023250 030377 155252 DCC: BIT R3,@AS ;SEE IF ATTN IS RESET
4631 023254 001001 BNE DCE ;IF NOT: BR
4632 023256 000207 RTS PC ;RETURN
4633 023260 032777 020000 155304 DCE: BIT #20000,@SWR ;SEE IF ERROR PRINT INHIBIT
4634 023266 001017 BNE DCEX ;IF SO: BR
4635 023270 005767 155334 TST HDRFL ;SEE IF PRINT HEADER
4636 023274 001004 BNE DCEA ;IF NOT: BR
4637 023276 016704 155330 MOV EMADDR,R4
4638 023302 004767 001214 JSR PC,TTOUT ;PRINT HEADER
4639 023305 012704 030064 DCEA: MOV #MSG47,R4
4640 023312 004767 001204 JSR PC,TTOUT ;PRINT DRIVE CLEAR ERROR
4641 023316 004767 177460 JSR PC,REGP ;PRINT REGISTERS
4642 023322 005267 155350 INC PFLG ;SET PRINTED FLAG
4643 023326 005777 155240 DCEX: TST @SWR ;SEE IF HALT ON ERROR
4644 023332 100002 BPL DCEXA ;IF NOT: BR
4645 023334 004767 002300 JSR PC,STOP
4646 023340 005767 155332 DCEXA: TST PFLG ;SEE IF HAVE PRINTED
4647 023344 001006 BNE DCEXX ;IF SO: BR
4648 023346 032777 020000 155216 BIT #20000,@SWR ;SEE IF SHOULD PRINT
4649 023354 001002 BNE DCEXX ;IF NOT: BR
4650 023356 000167 177676 JMP DCE ;ELSE PRINT THIS ERROR
4651 023362 012767 023134 155332 DCEXX: MOV #DRVCLR,SCOLP ;SET SCOPE LOOP ADDRESS
4652 023370 000167 000450 JMP SCOPE ;GO DO SCOPE LOOP
4653 023374 000207 RTS PC ;RETURN
```



```

4655                                     ;COMPOSITE ERROR CHECK SUBROUTINE*****
4656
4657 023376 005777 155120          ATTN: TST      @DS          ;SEE IF ATA SET
4658 023402 001005                BNE      ATTA          ;IF SO: BR
4659 023404 012767 027422 155276  MOV      #MSG32,TEMP3
4660 023412 000167 000064                JMP      ATTP          ;ELSE PRINT ERROR
4661 023416 032777 040000 155076  ATTA: BIT      #40000,@DS ;SEE IF COMPOSITE ERROR SET
4662 023424 001005                BNE      ATTB          ;IF SO: BR
4663 023426 012767 027404 155254  MOV      #MSG31,TEMP3
4664 023434 000167 000042                JMP      ATTP          ;ELSE PRINT ERROR
4665 023440 012703 000001                ATTB: MOV      #1,R3    ;SET TEST BIT
4666 023444 012767 027440 155236  MOV      #MSG33,TEMP3
4667 023452 016704 155156                MOV      DRVN,R4      ;GET DRIVE NUMBER
4668 023456 005704                TST      R4           ;SEE IF DRIVE 0
4669 023460 001404                BEQ      ATTD          ;IF SO: BR
4670 023462 000241                ATTC: CLC
4671 023464 006103                ROL      R3           ;POSITION TEST BIT
4672 023466 005304                DEC      R4           ;SEE IF DONE
4673 023470 001374                BNE      ATTC          ;IF NOT: BR
4674 023472 030377 155030                ATTD: BIT      R3,@AS  ;SEE IF ATTN SUMMARY SET
4675 023476 001401                BEQ      ATTP          ;IF NOT: BR
4676 023500 000207                RTS      PC           ;ELSE RETURN
4677 023502 032777 020000 155062  ATTP: BIT      #20000,@SWR ;SEE IF PRINT INHIBIT
4678 023510 001021                BNE      ATTX          ;IF SO: BR
4679 023512 005767 155112                TST      HDRFL        ;SEE IF DONE HEADER
4680 023516 001004                BNE      ATTPA         ;IF SO: BR
4681 023520 016704 155106                MOV      EMADDR,R4
4682 023524 004767 000772                JSR      PC,TTOUT     ;PRINT HEADER
4683 023530 016704 155154                ATTPA: MOV      TEMP3,R4
4684 023534 004767 000762                JSR      PC,TTOUT     ;PRINT ERROR TYPE
4685 023540 004767 177236                JSR      PC,REGP      ;PRINT REGISTERS
4686 023544 005267 155126                INC      PFLG         ;SET PRINT FLAG
4687 023550 005267 155054                INC      HDRFL        ;SET HEADER FLAG
4688 023554 005777 155012                ATTX: TST      @SWR    ;SEE IF HALT ON ERROR
4689 023560 100002                BPL      ATTXA         ;IF NOT: BR
4690 023562 004767 002052                JSR      PC,STOP
4691 023566 005767 155104                ATTXA: TST      PFLG   ;SEE IF DONE PRINT
4692 023572 001006                BNE      ATTXX         ;IF SO: BR
4693 023574 032777 020000 154770  BIT      #20000,@SWP  ;SEE IF SHOULD PRINT
4694 023602 001002                BNE      ATTXX         ;IF NOT: BR
4695 023604 000167 177672                JMP      ATTP          ;ELSE PRINT ERROR
4696 023610 005067 155062                ATTXX: CLR      PFLG   ;CLEAR PRINT FLAG
4697 023614 000207                RTS      PC           ;RETURN

```

```
4699 ;LOGIC TEST REGISTER BIT ERROR SUBROUTINE*****
4700
4701 023616 012767 000001 155122 LTGER2: MOV #1,PEXFL ;SET FLAG
4702 023624 005067 155046 LTGER1: CLR PFLG ;CLEAR PRINT FLAG
4703 023630 032777 020000 154734 BIT #20000,@SWR ;SEE IF PRINT ERRORS
4704 023636 001402 BEQ LTG1A ;IF SO: BR
4705 023640 000167 000132 JMP LTG1X ;ELSE GO TO EXIT
4706 023644 005767 154760 LTG1A: TST HDRFL ;SEE IF PRINT HEADER
4707 023650 001004 BNE LTG1B ;IF NOT: BR
4708 023652 016704 154754 MOV EMADDR,R4
4709 023656 004767 000640 JSR PC,TTOUT ;PRINT HEADER
4710 023662 012767 000001 154740 LTG1B: MOV #1,HDRFL ;SET FLAG
4711 023670 016704 155006 MOV ERADD,R4
4712 023674 004767 000622 JSR PC,TTOUT ;PRINT ERROR CODE
4713 023700 005767 155042 TST PEXFL ;SEE IF PRINT EXPT-RCVD
4714 023704 001016 BNE LTG1T ;IF NOT: BR
4715 023706 012704 026732 MOV #MSG12,R4
4716 023712 004767 000604 JSR PC,TTOUT ;PRINT EXPT TAG
4717 023716 010103 MOV R1,R3
4718 023720 004767 000740 JSR PC,OCTP ;PRINT EXPT
4719 023724 012704 026741 MOV #MSG13,R4
4720 023730 004767 000566 JSR PC,TTOUT ;PRINT RCVD TAG
4721 023734 010203 MOV R2,R3
4722 023736 004767 000722 JSR PC,OCTP ;PRINT RCVD
4723 023742 032777 010000 154622 LTG1T: BIT #10000,@SWR
4724 023750 001010 BNE LTG1C
4725 023752 012704 027003 MOV #MSG16,R4
4726 023756 004767 000540 JSR PC,TTOUT
4727 023762 016703 154724 MOV ITCNT,R3
4728 023766 004767 000672 JSR PC,OCTP ;PRINT ITERATION
4729 023772 005267 154700 LTG1C: INC PFLG
4730 023776 005777 154570 LTG1X: TST @SWR
4731 024002 100002 BPL LTG1X1 ;IF NOT STOP ON ERROR: BR
4732 024004 004767 001630 JSR PC,STOP
4733 024010 005767 154662 LTG1X1: TST PFLG
4734 024014 001006 BNE LTG1XX ;IF HAVE PRINTED: BR
4735 024016 032777 020000 154546 BIT #20000,@SWR
4736 024024 001002 BNE LTG1XX ;IF STILL NO PRINT: BR
4737 024026 000167 177612 JMP LTG1A ;ELSE PRINT ERROR
4738 024032 005067 154710 LTG1XX: CLR PEXFL ;CLEAR EXPT-RCVD FLAG
4739 024036 000167 000002 JMP SCOPE ;GO TO SCOPE
4740 024042 000207 RTS PC ;RETURN
4741
```

```
4743
4744 ;SCOPE LOOP ON ERROR SUBROUTINE*****
4745
4746 024044 004767 001050 SCOPE: JSR PC,CKSWR ;CHECK FOR CONTROL G
4747 024050 032777 040000 154514 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
4748 024056 001001 BNE SCOPE1 ;IF SO: BR
4749 024060 000207 RTS PC ;ELSE EXIT
4750 024062 005726 SCOPE1: TST (SP)+ ;RESET STACK
4751 024064 000177 154632 JMP @SCOLP ;LOOP ON ERROR
4752
4753 ;TEST ITERATION SUBROUTINE*****
4754
4755 : *****
4756 : CHECK FOR END OF PASS INDICATOR
4757 : TO INSURE QUICK VERIFY
4758
4759 024070 005767 154734 TST PAFLG ;PASS FLAG INDICATOR SET?
4760 024074 001404 BEQ ITER0 ;BRANCH - IF NO
4761
4762 : *****
4763
4764 024076 032777 010000 154466 ITER: BIT #10000,@SWR ;SEE IF ITERATIONS
4765 024104 001403 BEQ ITER1 ;IF SO: BR
4766 024106 005067 154600 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
4767 024112 000207 RTS PC ;ELSE EXIT
4768 024114 005267 154572 ITER1: INC ITCNT ;BUMP COUNTER
4769 024120 026767 154566 154462 CMP ITCNT,ITAMT ;SEE IF DONE ALL
4770 024126 001767 BEQ ITER0 ;IF SO: BR
4771 024130 005726 TST (SP)+ ;RESET STACK
4772 024132 017700 154566 MOV @ITRLP,R0 ;SET ITERATION POINTER
4773 024136 000110 JMP (R0) ;GO ITERATE
4774
4775 ;MANUAL INTERVENTION INHIBIT*****
4776
4777 024140 012704 027610 INMT: MOV #MSG43,R4
4778 024144 004767 000352 JSR PC,TTOUT ;GO PRINT INHIB MSG
4779 024150 004767 001464 JSR PC,STOP
4780 024154 000167 156272 JMP TSCD2 ;RETURN TO SCHED
4781
```

```
4783
4784 ;INITIALIZE SUBROUTINE*****
4785
4786 024160 012777 000040 154332 INIT1: MOV #40,@CS ;INIT
4787 024166 016777 154442 154324 INIT2: MOV DRVN,@CS ;SELECT DRIVE
4788 024174 016777 154474 154340 MOV SLVN,@TC ;SELECT SLAVE
4789 024202 000207 RTS PC ;RETURN
4790
4791 ;MANUAL INSTRUCTION SUBROUTINE*****
4792
4793 024204 004767 000312 INST: JSR PC,TTOUT ;PRINT INSTRUCTION
4794 024210 012704 034122 MOV #MMSG0,R4
4795 024214 004767 000302 JSR PC,TTOUT ;PRINT REPLY
4796 024220 012705 000710 MOV #TEMP3,R5
4797 024224 012701 000001 MOV #1,R1
4798 024230 012702 177777 MOV #-1,k2
4799 024234 012703 000000 MOV #0,R3
4800 024240 004767 000020 JSR PC,TTR ;AWAIT REPLY
4801 024244 000207 RTS PC ;EXIT
4802
4803 ;MAG TAPE INTERRUPT HANDLER*****
4804
4805 024246 022626 MTINT: CMP (SP)+,(SP)+ ;RESET STACK POINTER
4806 024250 000240 NOP
4807 024252 000240 NOP
4808 024254 000177 154420 JMF @RTRN ;RETURN TO CALLER
4809
4810 ;TTY INTERRUPT HANDLER*****
4811
4812 024260 TTINT:
4813 024260 000240 NOP
4814 024262 000002 RTI
4815
```

```
4817 :*****  
4818 :TTY ENTRY SUBROUTINE:  
4819 :  
4820 :THIS SUBROUTINE IS USED BY THE TEST CONDITION  
4821 :ENTRY ROUTINE TO READ THE RESPONSE ENTERED  
4822 :AT THE TTY AND CHECK THEM FOR LEGALITY AND  
4823 :LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL  
4824 :(0-7) AND MUST FALL WITHIN THE LIMITS SET BY  
4825 :THE CALLING ROUTINE.  
4826 :IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,  
4827 :A QUESTION MARK IS TYPED (?) AND THE RESPONSE  
4828 :MAY BE REENTERED.  
4829 :ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND  
4830 :MAY BE TERMINATED AT LESS THAN SIX BY TYPING A  
4831 :CARRIAGE RETURN  
4832 :*****  
4833  
4834 024264 005067 154414 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
4835 024270 005000 CLR RO  
4836 024272 004767 000152 TTR0: JSR PC,TIN ;GO READ CHARACTER  
4837 024276 122767 000215 154322 CMPB #215,TIB ;SEE IF CR  
4838 024304 001005 BNE TTR1 ;IF NOT: BR  
4839 024306 005767 154372 TST TEMP1 ;SEE IF FIRST CHARACTER  
4840 024312 001446 BEQ TTR5 ;IF SO: BR  
4841 024314 000167 000066 JMP TTR2 ;ELSE GO LOAD VALUE  
4842 024320 122767 000260 154300 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0  
4843 024326 101402 BLOS TTR1A ;IF NOT: BR  
4844 024330 000167 000076 JMP TINER ;ELSE GO TO ERROR  
4845 024334 122767 000270 154264 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7  
4846 024342 101002 BHI TTR1B ;IF NOT: BR  
4847 024344 000167 000062 JMP TINER ;ELSE GO TO ERROR  
4848 024350 005267 154330 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG  
4849 024354 000241 CLC  
4850 024356 006100 ROL RO  
4851 024360 000241 CLC  
4852 024362 006100 ROL RO ;SHIFT 3 LEFT  
4853 024364 000241 CLC  
4854 024366 006100 ROL RO  
4855 024370 042767 177770 154230 BIC #177770,TIB ;STRIP ASCII  
4856 024376 056700 154224 BIS TIB,RO ;LOAD CHARACTER  
4857 024402 005301 DEC R1 ;SEE IF DONE  
4858 024404 001332 BNE TTR0 ;IF NOT: BR  
4859 024406 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT  
4860 024410 101402 BLOS TTR3 ;IF NOT: BR  
4861 024412 000167 000014 JMP TINER ;ELSE GO TO ERROR  
4862 024416 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT  
4863 024420 101402 BLOS TTR4 ;IF NOT: BR  
4864 024422 000167 000004 JMP TINER ;ELSE GO TO ERROR  
4865 024426 010015 TTR4: MOV RO,(P5) ;LOAD VALUE  
4866 024430 000207 TTR5: RTS PC ;EXIT  
4867
```

```
4869
4870 ;TTY ENTRY ERROR SUBROUTINE*****
4871
4872 024432 012704 027550 T1NER: MOV #MSG40,R4
4873 024436 004767 000060 JSR PC,TTOUT ;PRINT?
4874 024442 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
4875 024446 000207 RTS PC ;REDO VALUE ENTRY
4876
4877 ;TTY READ SUBROUTINE*****
4878
4879 024450 005077 154120 TTIN: CLR @TKS
4880 024454 005077 154116 CLR @TKB
4881 024460 005067 154142 CLR TIB
4882 024464 005277 154104 INC @TKS
4883 024470 105777 154100 TTIN1: TSTB @TKS
4884 024474 100375 BPL TTIN1
4885 024476 017767 154074 154122 MOV @TKB,TIB
4886 024504 105777 154070 TTIN2: TSTB @TPS
4887 024510 100375 BPL TTIN2
4888 024512 116777 154110 154062 MOVB TIB,@TPB
4889 024520 000207 RTS PC
4890
4891 ;TTY OUTPUT SUBROUTINE*****
4892
4893 024522 112467 154076 TTOUT: MOVB (R4)+,TOB
4894 024526 122767 000043 154070 CMPB #43,TOB
4895 024534 001446 BEQ TEX
4896 024536 122767 000045 154060 CMPB #45,TOB
4897 024544 001403 BEQ TCRLF
4898 024546 004767 000064 JSR PC,TOG
4899 024552 000763 BR TTOUT
4900 024554 112767 000015 154042 TCRLF: MOVB #15,TOB
4901 024562 004767 000050 JSR PC,TOG
4902 024566 012703 000004 MOV #4,R3
4903 024572 005067 154026 TCRLFA: CLR TOB
4904 024576 004767 000034 JSR PC,TOG
4905 024602 005303 DEC R3
4906 024604 001372 BNE TCRLFA ;DO FILLERS
4907 024606 112767 000012 154010 MOVB #12,TOB
4908 024614 004767 000016 JSR PC,TOG
4909 024620 105767 000272 TSTB RDSW
4910 024624 100401 BMI 1$
4911 024626 000735 BR TTOUT
4912 024630 005067 000262 1$: CLR RDSW
4913 024634 000406 BR TEX
4914 024636 105777 153736 TOG: TSTB @TPS
4915 024642 100375 BPL TOG
4916 024644 116777 153754 153730 MOVB TOB,@TPB
4917 024652 000207 TEX: RTS PC
4918
4919 ;OCTAL OUTPUT SUBROUTINE*****
4920
4921
4922 024654 012767 000001 000226 OCTPE: MOV #1,OFL
4923 024662 000402 BR OCTPE1
4924 024664 005067 000220 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
```

```
4925 024670 010304          OCTPE1: MOV      R3,R4          ;SEE IF NUMBER IS ZERO
4926 024672 001007          BNE      OCTP0          ;IF NOT ZERO: BR
4927 024674 005767 000210   TST      OFL           ;SEE IF PRINT ALL 0
4928 024700 001004          BNE      OCTP0          ;IF SO: BR
4929 024702 004767 000162   JSR      PC,OCTPG1     ;ELSE PRINT ZERO
4930 024706 000167 000120   JMP      OCTP3         ;SPACE AND EXIT
4931 024712 032704 100000   OCTP0:  BIT      #100000,R4 ;SEE IF MSD = 1
4932 024716 001406          BEQ      OCTP1         ;IF NOT: BR
4933 024720 012704 000001   MOV      #1,R4
4934 024724 004767 000116   JSR      PC,OCTPG     ;PRINT 1
4935 024730 000167 000006   JMP      OCTP2
4936 024734 005004          OCTP1:  CLR      R4
4937 024736 004767 000104   JSR      PC,OCTPG     ;PRINT 0
4938 024742 010304          OCTP2:  MOV      R3,R4
4939 024744 006004          ROR      R4
4940 024746 006004          ROR      R4
4941 024750 006004          ROR      R4          ;POSITION DIGIT
4942 024752 006004          ROR      R4
4943 024754 000304          SWAB     R4
4944 024756 004767 000064   JSR      PC,OCTPG     ;PRINT DIGIT 2
4945 024762 010304          MOV      R3,R4
4946 024764 006004          ROR      R4
4947 024766 000304          SWAB     R4
4948 024770 004767 000052   JSR      PC,OCTPG     ;PRINT DIGIT 3
4949 024774 010304          MOV      R3,R4
4950 024776 006104          ROL      R4
4951 025000 006104          ROL      R4
4952 025002 000304          SWAB     R4
4953 025004 004767 000036   JSR      PC,OCTPG     ;PRINT DIGIT 4
4954 025010 010304          MOV      R3,R4
4955 025012 006004          ROR      R4
4956 025014 006004          ROR      R4
4957 025016 006004          ROR      R4
4958 025020 004767 000022   JSR      PC,OCTPG
4959 025024 010304          MOV      R3,R4
4960 025026 004767 000014   JSR      PC,OCTPG     ;PRINT DIGIT 5
4961 025032 012767 000240 153564 OCTP3:  MOV      #240,TOB
4962 025040 004767 177572   JSR      PC,TOG
4963 025044 000207          RTS      PC           ;PRINT SPACE
4964 025046 042704 177770   OCTPG:  BIC      #177770,R4 ;EXIT
4965 025052 001004          BNE      OCTPG0
4966 025054 005767 000030   TST      OFL
4967 025060 001001          BNE      OCTPG0
4968 025062 000207          RTS      PC
4969 025064 005267 000020   OCTPG0: INC      OFL
4970 025070 052704 000260   OCTPG1: BIS      #260,R4
4971 025074 010467 153524   MOV      R4,TOB
4972 025100 004767 177532   JSR      PC,TOG
4973 025104 010304          MOV      R3,R4
4974 025106 000207          RTS      PC
4975 025110 000000          OFL:    0             ;FIRST CHAR FLAG
4976
4977
4978          ;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR ^G TO ALLOW CHANGING
4979          ;OF LOC.176.
4980          ;CALL IS BY WAY OF JSR PC,CKSWR
```

```

4981                                     ;LOCATIONS USED:
4982 025112 000000                      TEMPST: .WORD 0
4983 025114 000000                      COUNT: .WORD 0
4984 025116 000000                      RDSW: .WORD 0
4985 025120 022767 000176 153444 CKSWR: CMP #SWREG,SWR ;SOFTWARE SWITCH REG PRESENT
4986 025126 001037                      BNE OUT ;NO, GET OUT
4987 025130 105777 153440              TSTB @TKS ;YES, WAIT FOR
4988 025134 100034                      BPL OUT ;READY, GET CHARACTER
4989 025136 017767 153434 153462      MOV @TKB,TIB ;AND STRIP OFF
4990 025144 042767 177600 153454      BIC #177600,TIB ;THE GARBAGE
4991 025152 022767 000007 153446      CMP #7,TIB ;IS IT A <^G>
4992 025160 001022                      BNE OUT
4993 025162 012704 036105              MOV #SCNTG,R4
4994 025166 004767 177330              JSR PC,TTOUT
4995 025172 012704 036111              CNTLU: MOV #SMSWR,R4
4996 025176 004767 177320              JSR PC,TTOUT
4997 025202 017703 153364              MOV @SWR,R3
4998 025206 004767 177442              JSR PC,OCTPE
4999 025212 012704 036120              MOV #SMNEW,R4
5000 025216 004767 177300              JSR PC,TTOUT
5001 025222 004767 000002              JSR PC,$READ ;GO READ A LINE
5002 025226 000207                      OUT: RTS ;RETURN TO MAIN BODY OF PROGRAM
5003
5004 025230 005067 177656              $READ: CLR TEMPST
5005 025234 012767 000007 177652      MOV #7,COUNT
5006 025242 004767 177202              1$: JSR PC,TTIN ;GO READ A CHARACTER
5007 025246 042767 177600 153352      BIC #177600,TIB ;STRIP OFF GARBAGE
5008 025254 122767 000025 153344      CMPB #25,TIB ;IS IT A ^U?
5009 025262 001002                      BNE 2$ ;BRANCH IF NOT
5010 025264 005726                      3$: TST (SP)+ ;POP THE STACK
5011 025266 000741                      BR CNTLU ;START OVER
5012 025270 122767 000015 153330      2$: CMPB #15,TIB ;IS IT A <CR>?
5013 025276 001013                      BNE 4$ ;BRANCH IF NOT
5014 025300 012767 000200 177610      MOV #200,RDSW
5015 025306 004767 177242              JSR PC,TCRLF ;ECHO IT WITH <LF>
5016 025312 022767 000007 177574      CMP #7,COUNT ;WAS IT FIRST CHARACTER
5017 025320 001037                      BNE 7$ ;CHANGE SWR IF NOT FIRST ONE
5018 025322 005726                      8$: TST (SP)+ ;POP THE STACK
5019 025324 000740                      BR OUT ;GET OUT
5020 025326 122767 000060 153272      4$: CMPB #60,TIB
5021 025334 003004                      BGT 5$
5022 025336 122767 000067 153262      CMPB #67,TIB
5023 025344 002005                      BGE 6$
5024 025346 012704 036130              5$: MOV #SQUEST,R4
5025 025352 004767 177144              JSR PC,TTOUT
5026 025356 000742                      BR 3$ ;START OVER IF NOT LEGAL CHARACTER
5027 025360 006367 177526              6$: ASL TEMPST
5028 025364 006367 177522              ASL TEMPST
5029 025370 006367 177516              ASL TEMPST
5030 025374 142767 000060 153224      BICB #60,TIB ;GET NITTY-GRITTY
5031 025402 156767 153220 177502      BISB TIB,TEMPST
5032 025410 005367 177500              DEC COUNT ;ONLY WANT 6 DIGITS
5033 025414 001754                      BEQ 5$
5034 025416 000711                      BR 1$
5035 025420 016777 177466 153144      7$: MOV TEMPST,@SWR ;CHANGE SWITCH REGISTER CONTENTS
5036 025426 000735                      BR 8$

```



```
5038  
5039 ;DATA CHARACTER OUTPUT SUBROUTINE*****  
5040  
5041 025430 005067 153170 DOUT: CLR TOB  
5042 025434 012704 000010 MOV #10,R4 ;SET NUMBER TO PRINT  
5043 025440 110367 153160 MOVB R3,TOB  
5044 025444 105777 153130 DOUT1: TSTB @TPS  
5045 025450 100375 BPL DOUT1  
5046 025452 132767 000200 153144 BITB #200,TOB  
5047 025460 001404 BEQ DOUT2  
5048 025462 012777 000061 153112 MOV #061,@TPB  
5049 025470 000403 BR DOUT3  
5050 025472 012777 000060 153102 DOUT2: MOV #060,@TPB  
5051 025500 006167 153120 DOUT3: ROL TOB  
5052 025504 005304 DEC R4  
5053 025506 001356 BNE DOUT1  
5054 025510 000207 RTS PC  
5055 025512 016703 153172 DOUTD: MOV TEMP3,R3  
5056 025516 000303 SWAB R3  
5057 025520 004767 177704 JSR PC,DOUT  
5058 025524 016703 153160 MOV TEMP3,R3  
5059 025530 004767 177674 JSR PC,DOUT  
5060 025534 000207 RTS PC
```

```
5061  
5062 ;TU16 SERIAL NUMBER PRINT SUBROUTINE*****  
5063  
5064 025536 010304 SNPT: MOV R3,R4  
5065 025540 000304 SWAB R4  
5066 025542 006004 ROR R4  
5067 025544 006004 ROR R4  
5068 025546 006004 ROR R4  
5069 025550 006004 ROR R4 ;GET FIRST DIGIT  
5070 025552 004767 000036 JSR PC,SNPG ;PRINT  
5071 025556 010304 MOV R3,R4  
5072 025560 000304 SWAB R4 ;GET SECOND DIGIT  
5073 025562 004767 000026 JSR PC,SNPG ;PRINT  
5074 025566 010304 MOV R3,R4  
5075 025570 006004 ROR R4  
5076 025572 006004 ROR R4  
5077 025574 006004 ROR R4  
5078 025576 006004 ROR R4  
5079 025600 004767 000010 JSR PC,SNPG ;PRINT THIRD DIGIT  
5080 025604 010304 MOV R3,R4  
5081 025606 004767 000002 JSR PC,SNPG ;PRINT FOURTH DIGIT  
5082 025612 000207 RTS PC ;EXIT  
5083 025614 012767 000260 153002 SNPG: MOV #260,TOB ;SET BASE = 0  
5084 025622 042704 177760 BIC #177760,R4 ;MASK DIGIT  
5085 025626 050467 152772 BIS R4,TOB ;SET ASCII  
5086 025632 004767 177000 JSR PC,TCG ;TYPE DIGIT  
5087 025636 000207 RTS PC ;RETURN
```

```
5088  
5089 ;HALT HANDLEN*****  
5090  
5091 025640 000000 STOP: HALT  
5092 025642 004767 177252 JSR PC,CKSWR ;CHECK FOR CONTROL G  
5093 025646 000207 RTS PC
```

5095
5096
5097
5098
5099
5100
5101
5102
5103
5104
5105
5106
5107
5108
5109
5110
5111
5112
5113
5114
5115
5116
5117
5118
5119

```
*****  
: CHECK FOR DUMP MODE OR AUTOMATIC MODE  
:*****  
CKMODE: CLR AUTOM :INIT AUTO MODE  
TST @#42 :AUTOMATIC MODE?  
BEQ 2$ :BRANCH - IF NOT  
INC AUTOM :SET AUTO MODE INDICATOR  
CMP @#42,@#46 :ACT11 MODE?  
BEQ 1$ :BRANCH - IF YES  
INCB XXDPM :INDICATE XXDP AUTO MODE  
BR 5$ :AND EXIT  
1$: INCB ACT11M :INDICATE ACT11 AUTO MODE  
BIS #104000,@SWR :SET FOR CON:CYCLE & HALT ON ERROR  
BR 5$ :AND EXIT  
2$: TSTB @#41 :MAN:MODE VIA ACT11/PAPER TAPE?  
BNE 3$ :BRANCH IF NOT  
INCB ADUMPM :INDICATE MAN: MODE VIA ACT11/PPAPER TAPE  
BR 5$ :AND EXIT THRU M.I  
3$: INCB XDUMPM :INDICATE MANUAL MODE VIA XXDP  
5$: RTS PC :RETURN  
:*****
```

5121
5122
5123
5124
5125
5126
5127 025742
5128 025742 000005
5129 025744 012704 026024
5130 025750 004767 176546
5131 025754 105767 152641
5132 025760 001405
5133 025762 013700 000042
5134 025766 005037 000042
5135 025772 004710
5136 025774 000777
5137
5138
5139

```
: *****  
: DISCONTINUE TESTING FOR  
: ILLEGAL CONDITIONS  
: *****  
ABORT: RESET ;INITIALIZE THE WORLD  
MOV #MSGD,R4 ;GET ABORT MESSAGE  
JSR PC,TTOUT ;TYPE ABORT MESSAGE  
TSTB XXDPM ;XXDP AUTO MODE  
BEQ 1$ ;BRANCH - IF NOT  
MOV @#42,R0 ;GET MONITOR EXIT ADDRESS  
CLR @#42 ;USE AS ABORT FLAG  
JSR PC,(R0) ;EXIT TO XXDP MONITOR  
1$: BR . ;AND HANG  
: *****
```

```
5141 ;MESSAGE TABLE*****
5142
5143
5144 025776 050045 047522 051107 MSGC: .ASCII /%PROGRAM IS DISABLED%/
      026004 046501 044440 020123
      026012 044504 040523 046102
      026020 042105 021445
5145 026024 050045 047522 051107 MSGD: .ASCII /%PROGRAM IS ABORTED%/
      026032 046501 044440 020123
      026040 041101 051117 042524
      026046 022504 043
5146 026051 045 051105 047522 MSGE: .ASCII /%ERROR: DRIVE NOT TE16 NOR TU16%/
      026056 035122 042011 044522
      026064 042526 047040 052117
      026072 052040 030505 020066
      026100 047516 020122 052524
      026106 033061 021445
5147 ;
5148
5149 026112 022445 046524 031060 MSG1: .ASCII /%TM02 - TU16 CTRL LGC (CZTUCGO)%/
      026120 026440 052040 030525
      026126 020066 052103 046122
      026134 046040 041507 024040
      026142 055103 052524 043503
      026150 024460 043
5150 026153 105 052116 051105 .ASCII /ENTER CONDITIONS IN OCTAL%/
      026160 041440 047117 044504
      026166 044524 047117 020123
      026174 047111 047440 052103
      026202 046101 021445
5151 026206 042045 044522 042526 MSG2: .ASCII /%DRIVE NUMBER #/
      026214 047040 046525 042502
      026222 020122 043
5152 026225 045 043045 051117 MSG2A: .ASCII /%%FOR DRIVE ADDRESS TEST;/
      026232 042040 044522 042526
      026240 040440 042104 042522
      026246 051523 052040 051505
      026254 035524
5153 026256 020045 047105 042524 .ASCII /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
      026264 020122 054105 052120
      026272 042040 044522 042526
      026300 047040 046525 042502
      026306 026122 040440 046114
      026314 047440 044124 051105
      026322 020123 044123 052517
      026330 042114 041040 020105
      026336 047516 026516 054105
      026344 051511 040524 052116
      026352 021456
5154 026354 047045 047117 042455 MSG3: .ASCII /%NON-EXIST DRIVE #/
      026362 044530 052123 042040
      026370 044522 042526 021440
5155 026376 051045 020110 042504 MSG4: .ASCII /%RH DETECTED #/
      026404 042524 052103 042105
      026412 021440
5156 026414 052045 030115 020062 MSG5: .ASCII /%TM02 DETECTED #/
```

	026422	042504	042524	052103		
	026430	042105	021440			
5157	026434	054105	052120	047055	MSG6:	.ASCII /EXPT-NOT RECVD#/
	026442	052117	051040	041505		
	026450	042126	043			
5158	026453	122	053103	026504	MSG7:	.ASCII /RCVD-NOT EXPT#/
	026460	047516	020124	054105		
	026466	052120	043			
5159	026471	045	046123	053101	MSG8:	.ASCII /%SLAVE NUMBER #/
	026476	020105	052516	041115		
	026504	051105	021440			
5160	026510	022445	047506	020122	MSG8A:	.ASCII /%%FOR SLAVE ADDRESS TEST:/
	026516	046123	053101	020105		
	026524	042101	051104	051505		
	026532	020123	042524	052123		
	026540	073				
5161	026541	045	042440	052116		.ASCII /% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
	026546	051105	042440	050130		
	026554	020124	046123	053101		
	026562	020105	052516	041115		
	026570	051105	020054	046101		
	026576	020114	052117	042510		
	026604	051522	051440	047510		
	026612	046125	020104	042502		
	026620	047040	047117	042455		
	026626	044530	052123	047101		
	026634	027124	043			
5162	026637	045	047516	026516	MSG9:	.ASCII /%NON-EXIST SLAVE #/
	026644	054105	051511	020124		
	026652	046123	053101	020105		
	026660	043				
5163	026661	045	042522	042101	MSG10:	.ASCII /%READ CONT BUS PAR #/
	026666	041440	047117	020124		
	026674	052502	020123	040520		
	026702	020122	043			
5164	026705	045	051127	052111	MSG11:	.ASCII /%WRITE CONT BUS PAR #/
	026712	020105	047503	052116		
	026720	041040	051525	050040		
	026726	051101	021440			
5165	026732	042440	050130	020124	MSG12:	.ASCII / EXPT #/
	026740	043				
5166	026741	040	041522	042126	MSG13:	.ASCII / RCVD #/
	026746	021440				
5167	026750	046445	020122	044502	MSG14:	.ASCII /%MR BITS 4-G#/
	026756	051524	032040	030055		
	026764	043				
5168	026765	045	051115	041040	MSG15:	.ASCII /%MR BITS 15-7#/
	026772	052111	020123	032461		
	027000	033455	043			
5169	027003	045	052111	051105	MSG16:	.ASCII /%ITER: #/
	027010	020072	043			
5170	027013	045	041524	041040	MSG17:	.ASCII /%TC BIT 13 #/
	027020	052111	030440	020063		
	027026	043				
5171	027027	045	041524	041040	MSG18:	.ASCII /%TC BITS 12-0 #/
	027034	052111	020123	031061		

5172	027042	030055	021440						
	027046	043045	020103	044502	MSG19:	.ASCII	/XFC BITS 15-0 #/		
	027054	051524	030440	026465					
	027062	020060	043						
5173	027065	045	052506	020116	MSG20:	.ASCII	/XFUN CODE BITS 5-1 OF C1 #/		
	027072	047503	042504	041040					
	027100	052111	020123	026465					
	027106	020061	043117	041440					
	027114	020061	043						
5174	027117	045	047507	041040	MSG21:	.ASCII	/XGO BIT NOT CORRECT AT START #/		
	027124	052111	047040	052117					
	027132	041440	051117	042522					
	027140	052103	040440	020124					
	027146	052123	051101	020124					
	027154	043							
5175	027155	045	047507	041040	MSG22:	.ASCII	/XGO BIT NOT SET #/		
	027162	052111	047040	052117					
	027170	051440	052105	021440					
5176	027176	043445	020117	044502	MSG23:	.ASCII	/XGO BIT NOT RESET BY INIT #/		
	027204	020124	047516	020124					
	027212	042522	042523	020124					
	027220	054502	044440	044516					
	027226	020124	043						
5177	027231	045	051104	020131	MSG24:	.ASCII	/XDRY NOT SET BY INIT #/		
	027236	047516	020124	042523					
	027244	020124	054502	044440					
	027252	044516	020124	043					
5178	027257	045	051104	020131	MSG25:	.ASCII	/XDRY NOT RESET BY GO=1#/		
	027264	047516	020124	042522					
	027272	042523	020124	054502					
	027300	043440	036517	021461					
5179	027306	042045	054522	047040	MSG25A:	.ASCII	/XDRY NOT SET BY GO=0#/		
	027314	052117	051440	052105					
	027322	041040	020131	047507					
	027330	030075	043						
5180	027333	045	047516	044440	MSG26:	.ASCII	/XNO INTERRUPT RETURNED#/		
	027340	052116	051105	052522					
	027346	052120	051040	052105					
	027354	051125	042516	021504					
5181	027362	041045	042101	051440	MSG27:	.ASCII	/XBAD STATUS#/		
	027370	040524	052524	021523					
5182	027376	051440	035116	021440	MSG30:	.ASCII	/ SN: #/		
5183	027404	042445	051122	047040	MSG31:	.ASCII	/XERR NOT SET #/		
	027412	052117	051440	052105					
	027420	021440							
5184	027422	040445	040524	047040	MSG32:	.ASCII	/XATA NOT SET #/		
	027430	052117	051440	052105					
	027436	021440							
5185	027440	040445	020123	044502	MSG33:	.ASCII	/XAS BIT NOT SET #/		
	027446	020124	047516	020124					
	027454	042523	020124	043					
5186	027461	045	041523	047040	MSG34:	.ASCII	/XSC NOT SET #/		
	027466	052117	051440	052105					
	027474	021440							
5187	027476	052045	042522	047040	MSG35:	.ASCII	/XTRE NOT SET #/		
	027504	052117	051440	052105					

CZTUGO TMO2/TU16 CTRL LGC
CZTUCG.P11 12-JUL-83 10:41

MACY11 30(1046) 12-JUL-83 10:47 M 11
PAGE 118-4

SEQ 0142

5202	030164	047045	055122	047440	MSG52: .ASCII /%NRZ ONLY: #/
	030172	046116	035131	021440	
5203	030200	041045	042101	046040	MSG53: .ASCII /%BAD LRC #/
	030206	041522	021440		
5204	030212	041045	042101	041440	MSG54: .ASCII /%BAD CK #/
	030220	020113	043		
5205	030223	045	042523	052524	MSG55: .ASCII /%SETUP ERROR: CHECK WRAP 0 WITH TEST 50#/
	030230	020120	051105	047522	
	030236	035122	041440	042510	
	030244	045503	053440	040522	
	030252	020120	020060	044527	
	030260	044124	052040	051505	
	030266	020124	030065	043	
5206	030273	045	052123	052101	MSG56: .ASCII /%STATIC TESTS ONLY: #/
	030300	041511	052040	051505	
	030306	051524	047440	046116	
	030314	035131	021440		


```
5208 ;TEST HEADER*****
5209
5210 030320 022445 047514 044507 MSLT1: .ASCII /%%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)#/
      030326 020103 042524 052123
      030334 030440 020072 051104
      030342 053111 020105 042101
      030350 051104 051505 044523
      030356 043516 024040 034115
      030364 030071 020071 044122
      030372 021451
5211 030374 022445 047514 044507 MSLT2: .ASCII /%%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)#/
      030402 020103 042524 052123
      030410 031040 020072 042522
      030416 044507 052123 051105
      030424 040440 042104 042522
      030432 051523 047111 020107
      030440 046450 034470 034460
      030446 051040 024510 043
5212 030453 045 046045 043517 MSLT3: .ASCII /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905 M8909)#/
      030460 041511 052040 051505
      030466 020124 035063 041440
      030474 047117 051124 046117
      030502 041040 051525 052040
      030510 051505 020124 051050
      030516 020110 034115 030071
      030524 020065 034115 030071
      030532 024471 043
5213 030535 045 046045 043517 MSLT4: .ASCII /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905 M8903)#/
      030542 041511 052040 051505
      030550 020124 035064 051440
      030556 040514 042526 040440
      030564 042104 042522 051523
      030572 047111 020107 046450
      030600 034470 032460 046440
      030606 034470 031460 021451
5214 030614 022445 047514 044507 MSLT5: .ASCII /%%LOGIC TEST 5: MR BIT TEST (M8905)#/
      030622 020103 042524 052123
      030630 032440 020072 051115
      030636 041040 052111 052040
      030644 051505 020124 046450
      030652 034470 032460 021451
5215 030660 022445 047514 044507 MSLT6: .ASCII /%%LOGIC TEST 6: TC BIT TEST (M8905)#/
      030666 020103 042524 052123
      030674 033040 020072 041524
      030702 041040 052111 052040
      030710 051505 020124 046450
      030716 034470 032460 021451
5216 030724 022445 047514 044507 MSLT7: .ASCII /%%LOGIC TEST 7: FC BIT TEST (M8905)#/
      030732 020103 042524 052123
      030740 033440 020072 041506
      030746 041040 052111 052040
      030754 051505 020124 046450
      030762 034470 032460 021451
5217 030770 022445 047514 044507 MSLT10: .ASCII /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905)#/
      030776 020103 042524 052123
      031004 030440 035060 043040
```

	031012	047125	052103	047511	
	031020	020116	044502	020124	
	031026	042524	052123	024040	
	031034	034115	030071	024465	
5218	031042	043			
	031043	045	046045	043517	MSLT11: .ASCII /%%LOGIC TEST 11: GO BIT TEST (M8909)#/
	031050	041511	052040	051505	
	031056	020124	030461	020072	
	031064	047507	041040	052111	
	031072	052040	051505	020124	
	031100	046450	034470	034460	
5219	031106	021451			
	031110	022445	047514	044507	MSLT12: .ASCII /%%LOGIC TEST 12: DRIVE READY BIT (M8909)#/
	031116	020103	042524	052123	
	031124	030440	035062	042040	
	031132	044522	042526	051040	
	031140	040505	054504	041040	
	031146	052111	024040	034115	
5220	031154	030071	024471	043	
	031161	045	046045	043517	MSLT13: .ASCII /%%LOGIC TEST 13: INTERRUPT TEST (RH)#/
	031166	041511	052040	051505	
	031174	020124	031461	020072	
	031202	047111	042524	051122	
	031210	050125	020124	042524	
	031216	052123	024040	044122	
5221	031224	021451			
	031226	022445	047514	044507	MSLT14: .ASCII /%%LOGIC TEST 14: MANUAL STATUS TEST 1#/
	031234	020103	042524	052123	
	031242	030440	035064	046440	
	031250	047101	040525	020114	
	031256	052123	052101	051525	
	031264	052040	051505	020124	
5222	031272	021461			
	031274	022445	047514	044507	MSLT15: .ASCII /%%LOGIC TEST 15: MANUAL STATUS TEST 2#/
	031302	020103	042524	052123	
	031310	030440	035065	046440	
	031316	047101	040525	020114	
	031324	052123	052101	051525	
	031332	052040	051505	020124	
5223	031340	021462			
	031342	022445	047514	044507	MSLT16: .ASCII /%%LOGIC TEST 16: MANUAL STATUS TEST 3#/
	031350	020103	042524	052123	
	031356	030440	035066	046440	
	031364	047101	040525	020114	
	031372	052123	052101	051525	
	031400	052040	051505	020124	
5224	031406	021463			
	031410	022445	047514	044507	MSLT17: .ASCII /%%LOGIC TEST 17: MANUAL STATUS TEST 4#/
	031416	020103	042524	052123	
	031424	030440	035067	046440	
	031432	047101	040525	020114	
	031440	052123	052101	051525	
	031446	052040	051505	020124	
5225	031454	021464			
	031456	022445	047514	044507	MSLT20: .ASCII /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)#/
	031464	020103	042524	052123	

	031472	031040	035060	044440	
	031500	046114	043505	046101	
	031506	043040	047125	052103	
	031514	047511	020116	042524	
	031522	052123	024040	034115	
	031530	030071	024471	043	
5226	031535	045	046045	043517	MSLT21: .ASCII /%%LOGIC TEST 21: RMR(M8909)#/
	031542	041511	052040	051505	
	031550	020124	030462	020072	
	031556	046522	024122	034115	
	031564	030071	024471	043	
5227	031571	045	046045	043517	MSLT22: .ASCII /%%LOGIC TEST 22: CPAR(M8909)#/
	031576	041511	052040	051505	
	031604	020124	031062	020072	
	031612	050103	051101	046450	
5228	031620	034470	034460	021451	MSLT23: .ASCII /%%LOGIC TEST 23: FMT(M8905 M8906)#/
	031626	022445	047514	044507	
	031634	020103	042524	052123	
	031642	031040	035063	043040	
	031650	052115	046450	034470	
	031656	032460	046440	034470	
5229	031664	033060	021451		MSLT24: .ASCII /%%LOGIC TEST 24: DPAR(M8906 RH)#/
	031670	022445	047514	044507	
	031676	020103	042524	052123	
	031704	031040	035064	042040	
	031712	040520	024122	034115	
	031720	030071	020066	044122	
	031726	021451			
5230	031730	022445	047514	044507	MSLT25: .ASCII /%%LOGIC TEST 25: NEF(M8909)#/
	031736	020103	042524	052123	
	031744	031040	035065	047040	
	031752	043105	046450	034470	
	031760	034460	021451		
5231	031764	022445	047514	044507	MSLT26: .ASCII /%%LOGIC TEST 26: FCE(M8909)#/
	031772	020103	042524	052123	
	032000	031040	035066	043040	
	032006	042503	046450	034470	
	032014	034460	021451		
5232	032020	022445	047514	044507	MSLT27: .ASCII /%%LOGIC TEST 27: ILR(M8909)#/
	032026	020103	042524	052123	
	032034	031040	035067	044440	
	032042	051114	046450	034470	
	032050	034460	021451		
5233	032054	022445	047514	044507	MSLT30: .ASCII /%%LOGIC TEST 30: DTE(M8906 RH)#/
	032062	020103	042524	052123	
	032070	031440	035060	052104	
	032076	024105	034115	030071	
	032104	020066	044122	021451	
5234	032112	022445	047514	044507	MSLT31: .ASCII /%%LOGIC TEST 31: OPI(M8903)#/
	032120	020103	042524	052123	
	032126	031440	035061	047440	
	032134	044520	046450	034470	
	032142	031460	021451		
5235	032146	022445	047514	044507	MSLT32: .ASCII /%%LOGIC TEST 32: UNS(M8909)#/
	032154	020103	042524	052123	
	032162	031440	035062	052440	

	032170	051516	046450	034470	
	032176	034460	021451		
5236	032202	022445	047514	044507	MSLT33: .ASCII /%%LOGIC TEST 33: PIP(M8909)#/
	032210	020103	042524	052123	
	032216	031440	035063	050040	
	032224	050111	046450	034470	
	032232	034460	021451		
5237	032236	022445	047514	044507	MSLT34: .ASCII /%%LOGIC TEST 34: PES(M8911)#/
	032244	020103	042524	052123	
	032252	031440	035064	050040	
	032260	051505	046450	034470	
	032266	030461	021451		
5238	032272	022445	047514	044507	MSLT35: .ASCII /%%LOGIC TEST 35: TCW(M8903 M8905)#/
	032300	020103	042524	052123	
	032306	031440	035065	052040	
	032314	053503	046450	034470	
	032322	031460	046440	034470	
	032330	032460	021451		
5239	032334	022445	047514	044507	MSLT36: .ASCII /%%LOGIC TEST 36: FCS(M8903 M8905)#/
	032342	020103	042524	052123	
	032350	031440	035066	043040	
	032356	051503	046450	034470	
	032364	031460	046440	034470	
	032372	032460	021451		
5240	032376	022445	047514	044507	MSLT37: .ASCII /%%LOGIC TEST 37: ACCL(M8903 M8905)#/
	032404	020103	042524	052123	
	032412	031440	035067	040440	
	032420	041503	024114	034115	
	032426	030071	020063	034115	
	032434	030071	024465	043	
5241	032441	045	046045	043517	MSLT40: .ASCII /%%LOGIC TEST 40: PE TAPE MARK(M8902)#/
	032446	041511	052040	051505	
	032454	020124	030064	020072	
	032462	042520	052040	050101	
	032470	020105	040515	045522	
	032476	046450	034470	031060	
	032504	021451			
5242	032506	022445	047514	044507	MSLT41: .ASCII /%%LOGIC TEST 41: NRZ TAPE MARK (M8904)#/
	032514	020103	042524	052123	
	032522	032040	035061	047040	
	032530	055122	052040	050101	
	032536	020105	040515	045522	
	032544	024040	034115	030071	
	032552	024464	043		
5243	032555	045	046045	043517	MSLT42: .ASCII /%%LOGIC TEST 42: WRAP 3,NRZ,NORMAL,ODD#/
	032562	041511	052040	051505	
	032570	020124	031064	020072	
	032576	051127	050101	031440	
	032604	047054	055122	047054	
	032612	051117	040515	026114	
	032620	042117	021504		
5244	032624	022445	047514	044507	MSLT43: .ASCII /%%LOGIC TEST 43: WRAP 3,PE,NORMAL,ODD#/
	032632	020103	042524	052123	
	032640	032040	035063	053440	
	032646	040522	020120	026063	
	032654	042520	047054	051117	

	032662	040515	026114	042117	
	032670	021504			
5245	032672	022445	047514	044507	MSLT44: .ASCII /%%LOGIC TEST 44: WRAP 2,NRZ,NORMAL,ODD#/ 032700 020103 042524 052123 032706 032040 035064 053440 032714 040522 020120 026062 032722 051116 026132 047516 032730 046522 046101 047454 032736 042104 043
5246	032741	045	046045	043517	MSLT45: .ASCII /%%LOGIC TEST 45: WRAP 2,PE,NORMAL,ODD#/ 032746 041511 052040 051505 032754 020124 032464 020072 032762 051127 050101 031040 032770 050054 026105 047516 032776 046522 046101 047454 033004 042104 043
5247	033007	045	046045	043517	MSLT46: .ASCII /%%LOGIC TEST 46: WRAP 1,NRZ,NORMAL,ODD#/ 033014 041511 052040 051505 033022 020124 033064 020072 033030 051127 050101 030440 033036 047054 055122 047054 033044 051117 040515 026114 033052 042117 021504
5248	033056	022445	047514	044507	MSLT47: .ASCII /%%LOGIC TEST 47: WRAP 1,PE,NORMAL,ODD#/ 033064 020103 042524 052123 033072 032040 035067 053440 033100 040522 020120 026061 033106 042520 047054 051117 033114 040515 026114 042117 033122 021504
5249	033124	022445	047514	044507	MSLT50: .ASCII /%%LOGIC TEST 50: WRAP 0,NRZ,NORMAL,ODD#/ 033132 020103 042524 052123 033140 032440 035060 053440 033146 040522 020120 026060 033154 051116 026132 047516 033162 046522 046101 047454 033170 042104 043
5250	033173	045	046045	043517	MSLT51: .ASCII /%%LOGIC TEST 51: WRAP 0,PE,NORMAL,ODD#/ 033200 041511 052040 051505 033206 020124 030465 020072 033214 051127 050101 030040 033222 050054 026105 047516 033230 046522 046101 047454 033236 042104 043
5251	033241	045	046045	043517	MSLT52: .ASCII /%%LOGIC TEST 52: CORE DUMP WRITE (M8906)#/ 033246 041511 052040 051505 033254 020124 031065 020072 033262 047503 042522 042040 033270 046525 020120 051127 033276 052111 020105 046450 033304 034470 033060 021451
5252	033312	022445	047514	044507	MSLT53: .ASCII /%%LOGIC TEST 53: CORE DUMP READ (M8906)#/ 033320 020103 042524 052123 033326 032440 035063 041440 033334 051117 020105 052504 033342 050115 051040 040505

	033350	020104	046450	034470	
	033356	033060	021451		
5253	033362	022445	047514	044507	MSLT54: .ASCII /%%LOGIC TEST 54: EVEN PARITY WRITE (M8903 M8904)#/
	033370	020103	042524	052123	
	033376	032440	035064	042440	
	033404	042526	020116	040520	
	033412	044522	054524	053440	
	033420	044522	042524	024040	
	033426	034115	030071	020063	
	033434	034115	030071	024464	
	033442	043			
5254	033443	045	046045	043517	MSLT55: .ASCII /%%LOGIC TEST 55: EVEN PARITY READ(M8903 M8904)#/
	033450	041511	052040	051505	
	033456	020124	032465	020072	
	033464	053105	047105	050040	
	033472	051101	052111	020131	
	033500	042522	042101	046450	
	033506	034470	031460	046440	
	033514	034470	032060	021451	
5255	033522	022445	047514	044507	MSLT56: .ASCII /%%LOGIC TEST 56: READ REVERSE(M8906)#/
	033530	020103	042524	052123	
	033536	032440	035066	051040	
	033544	040505	020104	042522	
	033552	042526	051522	024105	
	033560	034115	030071	024466	
	033566	043			
5256	033567	045	046045	043517	MSLT57: .ASCII /%%LOGIC TEST 57: CRC(M8904)#/
	033574	041511	052040	051505	
	033602	020124	033465	020072	
	033610	051103	024103	034115	
	033616	030071	024464	043	
5257	033623	045	046045	043517	MSLT60: .ASCII /%%LOGIC TEST 60: LRC(M8904)#/
	033630	041511	052040	051505	
	033636	020124	030066	020072	
	033644	051114	024103	034115	
	033652	030071	024464	043	
5258	033657	045	046045	043517	MSLT61: .ASCII /%%LOGIC TEST 61: CORRECTABLE DATA (M8902 M8901)#/
	033664	041511	052040	051505	
	033672	020124	030466	020072	
	033700	047503	051122	041505	
	033706	040524	046102	020105	
	033714	040504	040524	024040	
	033722	034115	030071	020062	
	033730	034115	030071	024461	
	033736	043			
5259	033737	045	046045	043517	MSLT62: .ASCII /%%LOGIC TEST 62: INCORRECTABLE DATA (M8902 M8904)#/
	033744	041511	052040	051505	
	033752	020124	031066	020072	
	033760	047111	047503	051122	
	033766	041505	040524	046102	
	033774	020105	040504	040524	
	034002	024040	034115	030071	
	034010	020062	034115	030071	
	034016	024464	043		
5260	034021	045	046045	043517	MSLT63: .ASCII /%%LOGIC TEST 63: PEF(M8902)#/
	034026	041511	052040	051505	

CZTUGO TMO2/TU16 CTRL LGC
CZTUCG.P11 12-JUL-83 10:41

MACY11 30(1046) 12-JUL-83 10:47 G 12
PAGE 119-6

SEQ

5261 034034 020124 031466 020072
034042 042520 024106 034115
034050 030071 024462 043
034055 045 046045 043517
034062 041511 052040 051505
034070 020124 032066 020072
034076 041506 047440 042526
034104 043122 047514 020127
034112 046450 034470 032460
034120 021451

MSLT64: .ASCII /%%LOGIC TEST 64: FC OVERFLOW (M8905)*/

```
5263
5264 ;MANUAL INSTRUCTION*****
5265
5266 034122 052045 050131 020105 MMSG0: .ASCII /%TYPE CR WHEN READY:#/
      034130 051103 053440 042510
      034136 020116 042522 042101
      034144 035531 043
5267 034147 045 046445 052517 MMSG1: .ASCII /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
      034154 052116 052040 050101
      034162 020105 044527 044124
      034170 047040 020117 051127
      034176 052111 020105 044522
      034204 043516 020054 047514
      034212 042101 052040 020117
      034220 047502 026124 051440
      034226 052105 052040 020117
      034234 047117 046040 047111
      034242 035105 043
5268 034245 045 042523 020124 MMSG2: ASCII /%SET TO OFFLINE:#/
      034252 047524 047440 043106
      034260 044514 042516 021472
5269 034266 046445 053117 020105 MMSG3: .ASCII /%MOVE FORWARD TO EOT, ONLINE:#/
      034274 047506 053522 051101
      034302 020104 047524 042440
      034310 052117 020054 047117
      034316 044514 042516 021472
5270 034324 047445 043106 046040 MMSG4: .ASCII /%OFF LINE REVERSE PAST EOT INSERT WRITE RING, ON LINE#/
      034332 047111 020105 042522
      034340 042526 051522 020105
      034346 040520 052123 042440
      034354 052117 020054 047111
      034362 042523 052122 053440
      034370 044522 042524 051040
      034376 047111 026107 047440
      034404 020116 044514 042516
      034412 043
5271 034413 045 046445 053117 MMSG5: .ASCII /%%MOVE TAPE TO BOT; ON LINE#/
      034420 020105 040524 042520
      034426 052040 020117 047502
      034434 035524 047440 020116
      034442 044514 042516 043
```


5273
5274 :TAG MESSAGE
5275
5276 034447 045 046123 020101 TMS1: .ASCII /%SLA #/
034454 043
5277 034455 045 047502 020124 TMS2: .ASCII /%BOT #/
034462 043
5278 034463 045 046524 021440 TMS3: .ASCII /%TM #/
5279 034470 044445 041104 021440 TMS4: .ASCII /%IDB #/
5280 034476 051445 053504 020116 TMS5: .ASCII /%SDWN #/
034504 043
5281 034505 045 042520 020123 TMS6: .ASCII /%PES #/
034512 043
5282 034513 045 051523 020103 TMS7: .ASCII /%SSC #/
034520 043
5283 034521 045 051104 020131 TMS8: .ASCII /%DRY #/
034526 043
5284 034527 045 050104 020122 TMS9: .ASCII /%DPR #/
034534 043
5285 034535 045 052116 020114 TMS10: .ASCII /%NTL #/
034542 043
5286 034543 045 047505 020124 TMS11: .ASCII /%EOT #/
034550 043
5287 034551 045 051127 020114 TMS12: .ASCII /%WRL #/
034556 043
5288 034557 045 047515 020114 TMS13: .ASCII /%MOL #/
034564 043
5289 034565 045 044520 020120 TMS14: .ASCII /%PIP #/
034572 043
5290 034573 045 051105 020122 TMS15: .ASCII /%ERR #/
034600 043
5291 034601 045 052101 020101 TMS16: .ASCII /%ATA #/
034606 043
5292 034607 045 046111 020106 TMS17: .ASCII /%ILF #/
034614 043
5293 034615 045 046111 020122 TMS18: .ASCII /%ILR #/
034622 043
5294 034623 045 046522 020122 TMS19: .ASCII /%RMR #/
034630 043
5295 034631 045 050103 051101 TMS20: .ASCII /%CPAR #/
034636 021440
5296 034640 043045 052115 021440 TMS21: .ASCII /%FMT #/
5297 034646 042045 040520 020122 TMS22: .ASCII /%DPAR #/
034654 043
5298 034655 045 047111 020103 TMS23: .ASCII /%INC #/
034662 043
5299 034663 045 050126 020105 TMS24: .ASCII /%VPE #/
034670 043
5300 034671 045 042520 020106 TMS25: .ASCII /%PEF #/
034676 043
5301 034677 045 051114 020103 TMS26: .ASCII /%LRC #/
034704 043
5302 034705 045 051516 020107 TMS27: .ASCII /%NSG #/
034712 043
5303 034713 045 041506 020105 TMS28: .ASCII /%FCE #/
034720 043

5304	034721	045	051503	021440	TMS29:	.ASCII	/XCS #/
5305	034726	044445	046524	021440	TMS30:	.ASCII	/XITM #/
5306	034734	047045	043105	021440	TMS31:	.ASCII	/XNEF #/
5307	034742	042045	042524	021440	TMS32:	.ASCII	/XDTE #/
5308	034750	047445	044520	021440	TMS33:	.ASCII	/XOPI #/
5309	034756	053445	044522	042524	TMS33A:	.ASCII	/XWRITE OPI #/
	034764	047440	044520	021440			
5310	034772	051045	040505	020104	TMS33B:	.ASCII	/XREAD OPI #/
	035000	050117	020111	043			
5311	035005	045	047125	020123	TMS34:	.ASCII	/XUNS #/
	035012	043					
5312	035013	045	047503	051122	TMS35:	.ASCII	/XCORR #/
	035020	021440					
5313	035022	041445	041522	021440	TMS36:	.ASCII	/XCRC #/
5314	035030	052045	053503	021440	TMS37:	.ASCII	/XTCW #/
5315	035036	043045	051503	021440	TMS38:	.ASCII	/XFCS #/
5316	035044	040445	041503	020114	TMS39:	.ASCII	/XACCL #/
	035052	043					
5317							
5318		035054			.EVEN		
5319					.WRITE	BUFFER	
5320							
5321	035054	000100			WDATA:	.REPT	100
5322						-1	
5323						.ENDR	
(1)	035054	177777				-1	
(1)	035056	177777				-1	
(1)	035060	177777				-1	
(1)	035062	177777				-1	
(1)	035064	177777				-1	
(1)	035066	177777				-1	
(1)	035070	177777				-1	
(1)	035072	177777				-1	
(1)	035074	177777				-1	
(1)	035076	177777				-1	
(1)	035100	177777				-1	
(1)	035102	177777				-1	
(1)	035104	177777				-1	
(1)	035106	177777				-1	
(1)	035110	177777				-1	
(1)	035112	177777				-1	
(1)	035114	177777				-1	
(1)	035116	177777				-1	
(1)	035120	177777				-1	
(1)	035122	177777				-1	
(1)	035124	177777				-1	
(1)	035126	177777				-1	
(1)	035130	177777				-1	
(1)	035132	177777				-1	
(1)	035134	177777				-1	
(1)	035136	177777				-1	
(1)	035140	177777				-1	
(1)	035142	177777				-1	
(1)	035144	177777				-1	
(1)	035146	177777				-1	
(1)	035150	177777				-1	

(1)	035152	177777	-1
(1)	035154	177777	-1
(1)	035156	177777	-1
(1)	035160	177777	-1
(1)	035162	177777	-1
(1)	035164	177777	-1
(1)	035166	177777	-1
(1)	035170	177777	-1
(1)	035172	177777	-1
(1)	035174	177777	-1
(1)	035176	177777	-1
(1)	035200	177777	-1
(1)	035202	177777	-1
(1)	035204	177777	-1
(1)	035206	177777	-1
(1)	035210	177777	-1
(1)	035212	177777	-1
(1)	035214	177777	-1
(1)	035216	177777	-1
(1)	035220	177777	-1
(1)	035222	177777	-1
(1)	035224	177777	-1
(1)	035226	177777	-1
(1)	035230	177777	-1
(1)	035232	177777	-1
(1)	035234	177777	-1
(1)	035236	177777	-1
(1)	035240	177777	-1
(1)	035242	177777	-1
(1)	035244	177777	-1
(1)	035246	177777	-1
(1)	035250	177777	-1
(1)	035252	177777	-1

5324
5325
5326
5327
5328
5329
5330

:READ BUFFER
RDATA: .REPT 100
0
.ENDR

(1)	035254	000100	0
(1)	035254	000000	0
(1)	035256	000000	0
(1)	035260	000000	0
(1)	035262	000000	0
(1)	035264	000000	0
(1)	035266	000000	0
(1)	035270	000000	0
(1)	035272	000000	0
(1)	035274	000000	0
(1)	035276	000000	0
(1)	035300	000000	0
(1)	035302	000000	0
(1)	035304	000000	0
(1)	035306	000000	0
(1)	035310	000000	0
(1)	035312	000000	0

5336	035506	020122	020075	043					
	035513	045	047516	026516	WMSG4:	.ASCII	/NON-EXISTANT DRIVE#/		
	035520	054105	051511	040524					
	035526	052116	042040	044522					
	035534	042526	021445						
5337	035540	041445	030523	021440	WMSG6:	.ASCII	/CS1 #/		
5338	035546	053445	020103	043	WMSG6A:	.ASCII	/WC #/		
5339	035553	045	040502	021440	WMSG6B:	.ASCII	/BA #/		
5340	035560	043045	020103	043	WMSG6C:	.ASCII	/FC #/		
5341	035565	045	051503	020062	WMSG6D:	.ASCII	/CS2 #/		
	035572	043							
5342	035573	045	051504	021440	WMSG6E:	.ASCII	/DS #/		
5343	035600	042445	020122	043	WMSG6F:	.ASCII	/ER #/		
5344	035605	045	051501	021440	WMSG6G:	.ASCII	/AS #/		
5345	035612	041445	020103	043	WMSG6H:	.ASCII	/CC #/		
5346	035617	045	041104	021440	WMSG6I:	.ASCII	/DB #/		
5347	035624	046445	020122	043	WMSG6J:	.ASCII	/MR #/		
5348	035631	045	052104	021440	WMSG6K:	.ASCII	/DT #/		
5349	035636	052045	020103	043	WMSG6L:	.ASCII	/TC #/		
5350	035643	045	047123	021440	WMSG6M:	.ASCII	/SN #/		
5351	035650	041045	042101	042040	WMSG16:	.ASCII	/BAD DATA#/		
	035656	052101	021501						
5352	035662	043445	020072	043	WMSG17:	.ASCII	/G: #/		
5353	035667	045	035102	021440	WMSG20:	.ASCII	/B: #/		
5354	035674	041445	035116	021440	WMSG21:	.ASCII	/CN: #/		
5355	035702	041045	042101	051440	WMSG23:	.ASCII	/BAD STATUS#/		
	035710	040524	052524	021523					
5356	035716	047045	020117	047111	WMSG24:	.ASCII	/NO INTERRUPT#/		
	035724	042524	051122	050125					
	035732	021524							
5357	035734	047045	020117	046103	WMSG25:	.ASCII	/NO CLOCK UP#/		
	035742	041517	020113	050125					
	035750	043							
5358	035751	045	047516	041440	WMSG26:	.ASCII	/NO CLCCK DOWN#/		
	035756	047514	045503	042040					
	035764	053517	021513						
5359	035770	042045	052101	020101	WMSG27:	.ASCII	/DATA PAT: #/		
	035776	040520	035124	043					
5360	036003	045	040502	020104	WMSG28:	.ASCII	/BAD PREAMBLE#/		
	036010	051120	040505	041115					
	036016	042514	043						
5361	036021	045	040502	020104	WMSG29:	.ASCII	/BAD POSTAMBLE#/		
	036026	047520	052123	046501					
	036034	046102	021505						
5362	036040	042445	051117	041440	WMSG31:	.ASCII	/EOR CLEAR DID NOT CLEAR GO#/		
	036046	042514	051101	042040					
	036054	042111	047040	052117					
	036062	041440	042514	051101					
	036070	043440	022517	043					
5363	036075	040	040520	051124	WMSG32:	.ASCII	/ PATRN #/		
	036102	020116	043						
5364									
5365	036105	045	043536	043	\$CNTG:	.ASCII	/^G#/		
5366	036111	045	053523	036522	\$MSWR:	.ASCII	/SWR= #/		
	036116	021440							
5367	036120	020040	042516	036527	\$MNEW:	.ASCII	/ NEW= #/		

5368	036126	021440			
5369	036130	037440	022440	043	\$QUEST: .ASCII / ? %#/
5370		036136			
5371	036136	000000		PRE:	0 .EVEN
5372		000050			0 .REPT 50
5373					0 .ENDR
5374					0
(1)	036140	000000			0
(1)	036142	000000			0
(1)	036144	000000			0
(1)	036146	000000			0
(1)	036150	000000			0
(1)	036152	000000			0
(1)	036154	000000			0
(1)	036156	000000			0
(1)	036160	000000			0
(1)	036162	000000			0
(1)	036164	000000			0
(1)	036166	000000			0
(1)	036170	000000			0
(1)	036172	000000			0
(1)	036174	000000			0
(1)	036176	000000			0
(1)	036200	000000			0
(1)	036202	000000			0
(1)	036204	000000			0
(1)	036206	000000			0
(1)	036210	000000			0
(1)	036212	000000			0
(1)	036214	000000			0
(1)	036216	000000			0
(1)	036220	000000			0
(1)	036222	000000			0
(1)	036224	000000			0
(1)	036226	000000			0
(1)	036230	000000			0
(1)	036232	000000			0
(1)	036234	000000			0
(1)	036236	000000			0
(1)	036240	000000			0
(1)	036242	000000			0
(1)	036244	000000			0
(1)	036246	000000			0
(1)	036250	000000			0
(1)	036252	000000			0
(1)	036254	000000			0
(1)	036256	000000			0
5375	036260	000000		POST:	0 .REPT 50
5376		000050			0 .ENDR
5377					0
5378					0
(1)	036262	000000			0
(1)	036264	000000			0
(1)	036266	000000			0
(1)	036270	000000			0

LT1G0	002740	2159	2162#		
LT1X	003150	2161	2173	2198#	
LT10	004564	1854	2433#		
LT10A	004602	2437#	2450	2452	
LT10A1	004600	2436#	2448		
LT10B	004642	2443	2445#	2454	
LT10E1	004664	2444	2451#		
LT10IT	004566	1855	2434#		
LT10X	004710	2446	2455#		
LT11	004720	1856	2461#		
LT11B	004752	2466	2468#	2486	2490
LT11C	005016	2474	2476#	2492	2496
LT11E1	005040	2467	2481#		
LT11E2	005072	2475	2487#		
LT11E3	005124	2480	2493#		
LT11IT	004722	1857	2462#	2484	
LT11X	005152	2479	2498#		
LT12	005162	1858	2504#		
LT12B	005212	2508	2510#	2524	2526
LT12C	005254	2515	2517#	2528	2530
LT12E1	005274	2509	2521#		
LT12E2	005316	2516	2525#		
LT12E3	005340	2520	2529#		
LT12IT	005164	1859	2505#	2522	
LT12X	005360	2519	2532#		
LT13	005370	1860	2538#		
LT13A	005430	2545#	2546		
LT13E1	005434	2547#			
LT13IT	005400	1861	2540#	2549	
LT13X	005462	2541	2551#		
LT14	005472	1862	2566#		
LT14A	005522	2572	2576#		
LT14IT	005540	1863	2579#	2584	
LT14X	005600	2583	2587#		
LT14XX	005604	2567	2574	2588#	
LT15	005610	1864	2596#		
LT15A	005640	2603	2607#		
LT15IT	005656	1865	2610#	2615	
LT15X	005716	2614	2618#		
LT15XX	005722	2598	2605	2619#	
LT16	005726	1866	2626#		
LT16A	005756	2633	2637#		
LT16IT	005774	1867	2640#	2645	
LT16X	006034	2644	2648#		
LT16XX	006040	2628	2635	2649#	
LT17	006044	1868	2657#		
LT17A	006074	2664	2668#		
LT17IT	006112	1869	2671#	2676	
LT17X	006152	2675	2679#		
LT17XX	006156	2659	2666	2680#	
LT2	003154	1842	1843	2203#	
LT2A	003214	2210#	2220	2234	
LT2B	003234	2213	2215#		
LT2C	003250	2216	2218#		
LT2ERG	003330	2224	2227	2230#	
LT2ER1	003260	2214	2222#		

LT2ER2	003276	2217	2225#		
LT2ER3	003314	2228#			
LT2IT	003156	2204#			
LT2LP	003344	2230	2233#		
LT2X	003352	2221	2235#		
LT20	006162	1870	2690#		
LT20A	006210	2691	2694#	2711	
LT20B	006266	2700	2705#		
LT20C	006276	2704	2706	2708#	
LT20IT	006176	1871	2692#		
LT20X	006312	2709	2712#		
LT21	006326	1872	2719#		
LT21A	006432	2729	2734#		
LT21B	006442	2733	2735	2737#	
LT21IT	006342	1873	2720	2721#	
LT21XA	006452	2739#	2740		
LT22	006472	1874	2748#		
LT22A	006572	2757	2762#		
LT22IT	006506	1875	2749	2750#	
LT22X	006602	2761	2763	2765#	
LT23	006616	1876	2772#		
LT23A	006716	2781	2786#		
LT23IT	006632	1877	2773	2774#	
LT23X	006726	2785	2787	2789#	
LT24	006746	1878	2796#		
LT24A	007060	2810#	2811	2813	
LT24B	007074	2815#	2821		
LT24B0	007116	2818	2820#		
LT24C	007130	2824#	2825		
LT24D	007212	2827	2838#		
LT24IT	006762	1879	2797	2798#	
LT24X	007236	2837	2842	2844#	
LT25	007272	1880	2854#		
LT25A	007400	2864	2869#		
LT25IT	007306	1881	2855	2856#	
LT25X	007410	2868	2870	2872#	
LT26	007424	1882	2879#		
LT26A	007510	2891#	2892		
LT26B	007534	2897#	2900	2902	
LT26C	007610	2898	2909#		
LT26IT	007432	1883	2880#	2904	
LT26W	007440	2882#	2883		
LT26X	007624	2908	2911	2913#	
LT27	007640	1884	2920#		
LT27A	007674	2922	2926#	2936	
LT27B	007732	2929	2933#		
LT27IT	007664	1885	2924#		
LT27X	007742	2934	2937#		
LT27XX	007752	2921	2939#		
LT3	003362	1844	2240#		
LT3A	003402	2244#	2255	2257	2269
LT3B	003422	2248#	2260	2262	
LT3C	003436	2251#	2264		
LT3ER1	003450	2247	2256#		
LT3ER2	003476	2250	2261#		
LT3IT	003364	1845	2241#		

LT3X	003520	2252	2265#			
LT3XX	003536	2266	2270#			
LT30	007756	1886	2944#			
LT30A	010060	2954	2957#			
LT30B	010122	2964#	2967			
LT30C	010162	2965	2974#			
LT30D	010200	2977#	2978			
LT30E	010230	2980	2984#			
LT30IT	010000	1887	2946	2947#		
LT30X	010254	2973	2983	2988	2990#	
LT31	010274	1888	2999#			
LT31A	010370	3013#	3016	3018		
LT31B	010432	3014	3023#			
LT31C	010450	3025	3028#	3044		
LT31D	010516	3038#	3041	3043		
LT31E	010566	3039	3049#			
LT31IT	010310	1889	3000	3001#		
LT31W	010324	3004#	3005			
LT31W1	010456	3030#	3031			
LT31X	010602	3022	3027	3048	3051	3053#
LT32	010624	1890	3061#			
LT32A	010732	3073	3078#			
LT32IT	010640	1891	3062	3063#		
LT32X	010742	3077	3079	3081#		
LT33	010756	1892	3093#			
LT33IT	010772	1893	3094	3095#		
LT33X	011050	3100	3104#			
LT34	011060	1894	3110#			
LT34A	011120	3117#	3124			
LT34A1	011106	3115#	3120			
LT34B	011150	3118	3122#			
LT34C	011162	3125#	3127			
LT34IT	011102	1895	3114#			
LT34X	011212	3126	3130#			
LT34XX	011216	3111	3131#			
LT35	011222	1896	3136#			
LT35A	011324	3145	3149#	3153		
LT35IT	011236	1897	3138#	3147		
LT35X	011364	3152	3156#			
LT36	011374	1898	3162#			
LT36A	011444	3166	3170#	3174		
LT36IT	011410	1899	3164#	3167		
LT36X	011504	3173	3177#			
LT37	011514	1900	3183#			
LT37A	011570	3188	3192#	3201		
LT37B	011622	3197#	3200			
LT37IT	011530	1901	3185#	3190		
LT37X	011654	3198	3204#			
LT4	003546	1846	1847	2276#		
LT4A	003642	2293#	2312	2318		
LT4B	003706	2298	2302#			
LT4C	003720	2303	2305#			
LT4D	003740	2300	2309#	2320		
LT4ERG	003770	2314	2316#			
LT4ER1	003752	2301	2313#			
LT4ER2	003762	2304	2315#			

LT4G	003572	2282#	2310	
LT4GO	003562	2277	2280#	
LT4X	004020	2279	2290	2321#
LT40	011664	1902	3210#	
LT40A	011746	3222#	3225	
LT40IT	011706	1903	3212	3214#
LT40W	011714	3216#	3217	
LT40X	012002	3223	3229#	
LT40XX	012006	3211	3230#	
LT41	012012	1904	3235#	
LT41A	012070	3244#	3247	
LT41B	012124	3245	3251#	
LT41C	012154	3252	3256#	
LT41D	012206	3257	3262#	
LT41IT	012026	1905	3235	3237#
LT41X	012226	3261	3265	3267#
LT42	012260	1906	1907	3281#
LT42A	012324	3287#	3309	3481
LT42B	012336	3288	3289#	3292
LT43	012372	1908	1909	3300#
LT43A	012404	3301	3303#	
LT44	012454	1910	1911	3314#
LT44A	012520	3320#	3341	
LT44B	012532	3321	3322#	3325
LT45	012562	1912	1913	3332#
LT45A	012574	3333	3335#	
LT46	012644	1914	1915	3346#
LT46A	012710	3352#	3374	3458
LT46B	012722	3353	3354#	3357
LT47	012752	1916	1917	3364#
LT47A	012764	3365	3367#	
LT5	004024	1848	2326#	
LT5A	004044	2330#	2339	2353
LT5B	004070	2331	2336#	2355
LT5C	004100	2337	2340#	
LT5D	004110	2342#	2351	2357
LT5E	004134	2346	2348#	2359
LT5ER1	004146	2335	2352#	
LT5ER2	004172	2347	2356#	
LT5IT	004032	1849	2327#	
LT5X	004216	2349	2360#	
LT50	013040	1918	1919	3379#
LT50A	013104	3385#	3406	3469
LT50B	013116	3386	3387#	3390
LT51	013146	1920	1921	3397#
LT51A	013160	3398	3400#	
LT52	013230	1922	1923	3411#
LT52A	013306	3418	3419#	3423
LT52X	013332	3421	3424#	
LT53	013342	1924	1925	3431#
LT53A	013426	3438	3440#	3445
LT53X	013460	3442	3446#	
LT54	013470	1926	1927	3452#
LT55	013540	1928	1929	3463#
LT56	013610	1930	1931	3474#
LT57	013666	1932	3491#	

LT57A	013770	3505#	3508		
LT57B	014004	3506	3509#		
LT57B1	014032	3510	3515#		
LT57B2	014052	3514	3519#		
LT57C	014124	3527#	3530		
LT57D	014140	3528	3531#		
LT57E	014170	3532	3537#		
LT57IT	013716	1933	3496	3497#	
LT57PS	013676	3493#	3494		
LT57X	014210	3536	3540	3542#	
LT6	004226	1850	2366#		
LT6A	004244	2370#	2385		
LT6A1	004242	2369#	2383		
LT6B	004250	2371#	2387		
LT6C	004274	2375	2377#	2389	2399
LT6D	004310	2378	2380#	2397	
LT6ER1	004332	2376	2386#		
LT6ER2	004356	2379	2390#		
LT6ER4	004420	2394	2398#		
LT6IT	004230	1851	2367#		
LT6X	004430	2381	2400#		
LT60	014224	1934	3549#		
LT60C	014316	3560#	3563		
LT60D	014332	3561	3564#		
LT60E	014364	3565	3570#		
LT60F	014416	3574	3577#		
LT60IT	014244	1935	3550	3552#	
LT60X	014436	3569	3580	3582#	
LT61	014452	1936	3589#		
LT61A	014544	3601#	3604		
LT61A1	014556	3602	3605#		
LT61B	014566	3607#	3610		
LT61C	014602	3608	3611#		
LT61D	014630	3612	3616#		
LT61E	014632	3617#			
LT61F	014674	3619	3626#		
LT61IT	014500	1937	3593	3594#	
LT61X	014714	3625	3629	3631#	
LT61XX	014724	3590	3633#		
LT62	014730	1938	3639#		
LT62A	015044	3657#	3660		
LT62B	015060	3658	3661#		
LT62D	015110	3662	3666#		
LT62E	015022	3651#	3654		
LT62E1	015034	3652	3655#		
LT62F	015142	3670	3673#		
LT62IT	014756	1939	3643	3644#	
LT62X	015166	3677	3679#		
LT62XX	015176	3640	3681#		
LT63	015202	1940	3686#		
LT63A	015272	3697#	3698		
LT63B	015342	3707#	3710		
LT63C	015356	3708	3711#		
LT63D	015410	3712	3717#		
LT63IT	015230	1941	3690	3691#	
LT63X	015434	3716	3721	3723#	

MSG34	027461	5186#		
MSG35	027476	5187#		
MSG36	027514	5188#		
MSG37	027532	5189#		
MSG4	026376	2223	2229	5155#
MSG40	027550	4872	5190#	
MSG41	027554	2127	5191#	
MSG42	027573	3623	5192#	
MSG43	027610	4777	5193#	
MSG44	027714	1998	5195#	
MSG45	027736	2007	5196#	
MSG46	027760	4577	5197#	
MSG47	030064	4639	5199#	
MSG5	026414	2226	5156#	
MSG50	030116	2970	5200#	
MSG51	030135	4516	5201#	
MSG52	030164	2031	5202#	
MSG53	030200	3575	5203#	
MSG54	030212	3671	5204#	
MSG55	030223	3516	5205#	
MSG56	030273	2038	5206#	
MSG6	026434	4535	5157#	
MSG7	026453	4538	5158#	
MSG8	026471	2282	5159#	
MSG8A	026510	2280	5160#	
MSG9	026637	2317	5162#	
MSLT1	030320	2164	5210#	
MSLT10	030770	2434	5217#	
MSLT11	031043	2462	5218#	
MSLT12	031110	2505	5219#	
MSLT13	031161	2539	5220#	
MSLT14	031226	2576	5221#	
MSLT15	031274	2607	5222#	
MSLT16	031342	2637	5223#	
MSLT17	031410	2668	5224#	
MSLT2	030374	2206	5211#	
MSLT20	031456	2690	5225#	
MSLT21	031535	2719	5226#	
MSLT22	031571	2748	5227#	
MSLT23	031626	2772	5228#	
MSLT24	031670	2796	5229#	
MSLT25	031700	2854	5230#	
MSLT26	031764	2879	5231#	
MSLT27	032020	2923	5232#	
MSLT28	030453	2241	5212#	
MSLT30	032054	2945	5233#	
MSLT31	032112	2999	5234#	
MSLT32	032146	3061	5235#	
MSLT33	032202	3093	5236#	
MSLT34	032236	3113	5237#	
MSLT35	032272	3137	5238#	
MSLT36	032334	3162	5239#	
MSLT37	032376	3183	5240#	
MSLT4	030535	2316	5213#	
MSLT40	032441	3213	5241#	
MSLT41	032506	3236	5242#	

CZTUGO TM02/TU16 CTRL LGC
CZTUCG.P11 12-JUL-83 10:41

MACY11 30(1046) 12-JUL-83 10:47 PAGE 123
CROSS REFERENCE TABLE -- MACRO NAMES

I 14

SEQ 0177

\$CHNMO 1613# 2077

. ABS. 037016 000

ERRORS DETECTED: 0

CZTUCG,CZTUCG/CRF/NL:TOC=CZTUCG.P11
RUN-TIME: 5 11 2 SECONDS
RUN-TIME RATIO: 24/18=1.3
CORE USED: 17K (34 PAGES)