

Micro Fiche Scan

Name of device(s) tested:

TM02,TE16,TU16

Test description:

TM02-TE16/TU77 BSC FC

MAINDEC Number or Package Identifier (after SEP 1977):

CZTUBH0

Fiche Document Part Number:

AH-9452H-MC

Fiche preparation date unknown, using copyright year:

1983

Image resolution:

8-bit gray levels, max. quality for archiving

COPYRIGHT (C) 74,83 by d|i|g|i|t|a|l

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

.REM E

IDENTIFICATION

PRODUCT CODE: AC - 9451H - MC
PRODUCT TITLE: CZTUBHO TM02 - TU16/TE16 BSC FC
DATE CREATED: 15 AUGUST 1977
REVISED: 11 NOV 1977 BY CLEM WALSH
26 SEP 1983 BY B.T. 'EBLANC
MAINTAINER: DIAGNOSTIC ENGINEERING

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 EQUIPMENT CORPORATION

TABLE OF CONTENTS

36
37
38
39
40
41
42
43
44
45
46
47
48
49

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

51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92

1. ABSTRACT

THIS PROGRAM IS INTENDED TO TEST ALL OF THE BASIC FUNCTIONAL LEVEL OPERATIONS OF THE TM02-TU16/TE16 MAG TAPE SYSTEM. ALL FUNCTIONS; WRITE, READ, SPACE, ERASE, REWIND, ETC; WILL BE TESTED. IN ADDITION TO THE TM02-TU16/TE16 TESTS, THE RM WILL BE TESTED SEPARATELY IN SO FAR AS IT IS POSSIBLE TO SEPARATE THE RM FROM THE TM02-TU16/TE16 ITSELF.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT A HARDWARE SWITCH REGISTER
B. 8K OF CORE
C. CONSOLE TTY
D. TM02 MAGTAPE CONTROLLER
E. MASS BUS CONTROLLER
F. TU16 OR TE16 MAG TAPE TRANSPORT

3. LOADING PROCEEDURE

USE STANDARD BINARY LOADING PROCEEDURE

4. STARTING PROCEEDURE

****SOFTWARE SWITCH REGISTER IS LOCATED AT LOC. 176(REFER TO SECTION 5 FOR MORE THERE ARE TWO (2) STARTING ADDRESSES THAT MAYBE USED: 200(8) AND 210(8)

- A. 200(8): STARTING AT THIS ADDRESS WILL CAUSE THE PROGRAM IDENTIFICATION TO BE PRINTED FOLLOWED BY REQUESTS FOR THE VARIOUS PARAMETERS NEEDED BY THE PROGRAM.
B. 210(8): THIS ADDRESS IS INTENDED FOR USE AS A RESTART ONLY AND WILL USE THE CURRENT PARAMETER VALUES.

94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119

4.1 SAMPLE START AT 200(8): OPERATOR RESPONSES ARE IN PARENS.

***SWR=XXXXXX NEW= WILL BE PRINTED FIRST IF SOFTWARE SWITCH REGISTER IS SELEC
(REFER TO SECTION 5 FOR OPERATOR ACTION)

TM02-TU16/TE16 BASIC FUNCTION TEST
ENTER CONDITIONS IN OCTAL

REGISTER START: 172440 (CR)

VECTOR: 224 (CR)

DRIVE NUMBER: 0 (3)

SLAVE NUMBER: 0 (6) SERIAL NO: 200

RH11 OR RH70: (0)

RH ONLY: (0)

NRZ ONLY: (1) -NRZ (NON-RETURN-TO ZERO) IS THE METHOD OF RECORDING
ON MAGNETIC TAPE.

THIS EXAMPLE SLOWS THE PROGRAM START USING THE RH11
ADDRESS (CS1) OF 172440, AN INTERRUPT VECTOR OF 224,
DRIVE NUMBER 3, AND SLAVE NUMBER 6, NRZ ONLY.
NOTE THAT THE CURRENT VALUES FOR EACH PARAMETER IS
PRINTED AND MAY OR NOT BE CHANGED.

***IF THE SOFTWARE SWITCH REGISTER IS SELECTED THE FIRST TYPE OUT WILL BE
AS FOLLOWS: SWR=XXXXXX NEW=
THIS WILL BE TYPED OUT BEFORE THE HEADER MESSAGE (REFER TO SECTION 5 FOR
A MORE DETAIL DESCRIPTION FOR OPERATOR ACTION.)

121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
1555. CONSOLE SWITCH SETTING

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; I.E. SCOPE ROUTINE AND AFTER
- 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 CHANGED.
 - B) IF A CONTROL U <+U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 2.

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 EXCEPT 5-9 ARE USED AND THE NORMAL, OR DEFAULT,
RUN IS DONE WITH ALL SWITCHES SET TO ZERO (0).
ALL HARDWARE 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: RH TESTS ONLY)
0=CONTINUE
SW13(020000): 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
SW12(010000): 1=INHIBIT ITERATION
0=DO ALL ITERATIONS PER TEST
SW11(004000): 1=CONTINUOUS CYCLE
0=HALT AT END OF PASS
SW10(002000): 1=HALT AT END OF CURRENT TEST
0=CONTINUE
SW9-5: N/A
SW4-0: SELECT TEST NUMBER::00=ALL TESTS

THE USE OF SW0-4 IS TO ALLOW SELECTION AND CONTINUOUS
EXECUTION OF ANY TEST. THE TEST SELECTION MAY BE CHANGED AT
ANY TIME, HOWEVER IT IS ADVISABLE TO USE SW10 TO STOP THE
PROGRAM AT THE END OF THE CURRENT TEST BEFORE CHANGING NUMBER.

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

6. ERROR PRINTOUTS

THE ERROR PRINTOUTS FOR EACH TEST WILL APPEAR IN THE SAME GENERAL FORMAT. THE FIRST LINE WILL ALWAYS SHOW THE TEST NUMBER AND ITS TITLE. THE SECOND LINE WILL BE AN EXPLANATION OF THE ERROR. THE FOLLOWING LINES WILL SHOW THE APPROPRIATE REGISTER OR ADDRESS VALUES THAT ARE APPLICABLE TO THE INDIVIDUAL TEST

EXAMPLES:

1. THIS EXAMPLE SHOWS A TYPICAL ERROR PRINTOUT FOR THE WRITE READ TEST: A WRITE CRC ERROR OCCURRED ON SLAVE 6.

FT13: WRITE-READ TEST
WRITE ERROR NRZ

CS1	WC	BA	FC	CS2	DS	ER	TC
144260	000000	015650	000000	000103	150600	100000	101306

2. THIS EXAMPLE SHOWS A TYPICAL SPACE ERROR:
THE FC IS NOT ZERO AT THE END OF THE OPERATION.

FT14: SPACE TEST
SPACE REVERSE ERROR NRZ

CS1	WC	BA	FC	CS2	DS	ER	TC
144230	177700	017162	177740	000114	150600	001000	161700

3. THIS EXAMPLE SHOWS A SPACE OPERATION WHICH RESULTED IN INCORRECT POSITIONING. SHOULD BE AT RECORD 20, IS AT RECORD 22.

FT14: SPACE TEST
POSITION ERROR:
REVERSE ERROR EXPT:20 RCVD:22

219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259

7. OPERATION

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

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
4. ENTER APPROPRIATE RESPONSES TO THE TTY REQUESTS

ALL HARDWARE SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY
TIME. THE NORMAL, OR DEFAULT, OPERATING SEQUENCE IS ALL
SWITCHES DOWN (ZERO). THE END OF EACH PASS IS NOTED BY A
MESSAGE 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-SW4)

WHEN SW0-4 ARE SET TO ZERO (00) THE SCHEDULAR WILL
EXECUTE ALL OF THE TESTS IN SEQUENCE (1-24). IF SW0-4 IS
SET TO SOME SPECIFIC TEST NUMBER (1-24) THAT PARTICULAR TEST
WILL BE EXECUTED CONTINUOUSLY. ANY TEST MAY BE SINGLE
SELECTED IN ANY ORDER; HOWEVER, THE BEST WAY TO AFFECT
THE CHANGE IS TO USE SW10 TO HALT THE CURRENT TEST, THEN
CHANGE NUMBER AND PRESS CONTINUE.

RH11 OR RH70 OPTION:

A ONE RESPONSE IS FOR THE RH70;
A ZERO RESPONSE IS FOR THE RH11.

RH ONLY OPTION:

BY RESPONDING TO THE REQUEST (RH ONLY:) WITH
A ONE (1), ONLY THE TESTS WHICH ARE POINTED TO
THE RH (TESTS 1 - 10) WILL BE EXECUTED IN
EACH PASS.

261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306

8. TEST DESCRIPTION

THE FOLLOWING IS A LIST OF ALL TESTS IN THEIR PROPER SEQUENCE. A BASIC DESCRIPTION OF EACH TEST IS PROVIDED TO AID IN UNDERSTANDING OF THE ERROR MESSAGES ASSOCIATED WITH EACH ONE.

A. RH TESTS: THE FIRST TEN (10) TESTS WILL PERFORM BASIC RH OPERATIONS AS FAR AS IS POSSIBLE WITHOUT REQUIRING THE TM02-TU16/TE16 ITSELF. (SEE RH ONLY OPTION; PAR 7)

FT1: RH ADDRESSING: THIS TEST WILL ASSURE THAT THE RH WILL RESPOND WITHOUT CAUSING A BUS TRAP TO ALL TM02 REGISTER ADDRESS IN SEQUENCE STARTING AT THE ADDRESS OF CS1 ENTERED BY THE OPERATOR.

FT2: RH REGISTER BITS READ/WRITE: THIS TEST WILL ASSURE THAT ALL BITS OF THE RH WRITE/READ REGISTERS CAN BE SET AND RESET.

FT3: RH INITIALIZE: THIS TEST WILL ASSURE THAT A RH INITIALIZE (BIT 5 OF CS2-1) WILL INDEED CLEAR THE RH ERRORS.

* FT4: SILO TEST 1: THIS TEST WILL ASSURE THAT A READ FROM AN EMPTY SILO WILL CAUSE DLT TO SET.

* FT5: SILO TEST 2: THIS TEST WILL ASSURE THAT BOTH THE IR AND OR BITS WILL CORRECTLY RESPOND TO LOADING OF THE SILO WITH ALL ZEROS AND THEN A WORD OF ALL ONES.

* FT6: SILO TEST 3: THIS TEST WILL WRITE AND THEN READ THE ENTIRE SILO TO ASSURE THAT DATA CAN BE PROPERLY FILLED AND READ. ALSO THE PROPER STATUS OF IR AND OR ARE CHECKED.

* FT7: SILO TEST 4: THIS TEST WILL ASSURE PROPER RH11 RESPONSE TO SILO OVERFLOW.

* FT10: SILO TEST 5: THIS TEST WILL ASSURE SILO RESET BY RH11 INITIALIZE.

**** NOTE: SILO TESTS (FT4-FT10) ARE FOR THE RH11 ONLY. ****

308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363

B. TM02-TU16/TE16 BASIC FUNCTIONS: THE FOLLOWING FOURTEEN (14) TESTS WILL ASSURE OPERATION OF THE MAG TAPE BASIC FUNCTIONS.

FT11: NOP TEST: THIS TEST WILL ASSURE THAT THE NOP FUNCTION EXECUTES WITH NO ERROR.

FT12: REWIND TEST: THIS TEST WILL ASSURE THAT THE REWIND FUNCTION WILL POSITION THE TAPE TO BOT WITH NO ERROR.

1. ISSUE A REWIND COMMAND
2. AWAIT PIP RESET (MOTION STOPPED)
3. ASSURE THAT NO ERROR OCCURED
4. END

FT13: WRITE/READ TEST: THIS TEST WILL ASSURE THAT THE UNIT UNDER TEST CAN WRITE AND READ IN ALL DENSITIES (FOR BOTH PE AND NRZ).

1. REWIND TO BOT
2. WRITE 100 RECORDS
 - A. ALL ONES DATA
 - B. 200 FRAMES
 - C. 200 BPI; ODD
3. CHECK FOR ERRORS ON EACH RECORD
4. READ REVERSE THEN FORWARD ALL 100 RECORDS
5. CHECK FOR ERRORS ON EACH RECORD
6. REPEAT STEPS 2 THRU 5 FOR 556,800,1600 BPI
7. END.

DATA RELATED ERRORS (PARITY ERROR, CRC ERROR, ETC) ARE IGNORED. T DATA READ IS NOT CHECKED; ONLY THE FUNCTION IS TESTED, NOT THE M

FT14: SPACE TEST: THIS TEST WILL ASSURE THAT PROPER POSITIONING IS MAINTAINED BY BOTH SPACE FORWARD AND REVERSE.

1. REWIND TO BOT
2. WRITE 100 RECORDS
 - A. EACH RECORD IS ONE FRAME LARGER THAN THE LAST THIS WILL ALLOW FOR POSITION CHECKING BY RECO
3. EACH RECORD IS ERROR CHECKED.
4. DATA RELATED ERRORS ARE IGNORED.
5. NOW SPACE REVERSE 77 RECORDS AND READ REVERSE 1, THE FRAME COUNT SHOULD BE 100.
THIS IS THE SIZE OF THE FIRST RECORD.
6. NOW SPACE FORWARD 76 RECORDS AND READ FORWARD 1, THE FRAME COUNT SHOULD BE 177.
THIS IS THE SIZE OF THE NEXT TO LAST RECORD.
7. CONTINUE THE SPACE AND READ (DECREMENTING THE RECORD UNTIL ALL POSITIONS HAVE BEEN CHECKED. IF POSITION IS
8. REPEAT STEPS 1 THRU 7 FOR PE.
9. END

365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420

FT15: ERASE TEST:

THIS TEST WILL ASSURE THAT THE ERASE FUNCTION WILL INDEED ERASE TAPES.

1. REWIND TO BOT
2. ISSUE 200 ERRASE COMMANDS.
3. ASSURE NO ERRORS FOR EACH COMMAND.
4. REWIND TO BOT.
5. ISSUE A READ FORWARD COMMAND.
6. THE TAPE SHOULD MOVE FORWARD UNTIL STOPPED BY OPI (APPROX 25 FT).
7. ASSURE NO ERRORS OTHER THAN OPI.
8. END

FT16: TAPE MARK WRITE/READ: THIS TEST WILL ASSURE THAT A TAPE MARK CAN BE WRITTEN AND READ IN BOTH PE AND NRZ.

1. REWIND TO BOT.
2. ISSUE A WRITE TAPE MARK COMMAND.
3. ASSURE NO ERRORS.
4. ASSURE THAT TAPE MARK STATUS IS SET IN DRIVE STATUS (BIT 2).
5. READ REVERSE.
6. ASSURE THAT TAPE MARK IS SET.
7. ASSURE THAT NO ERRORS OTHER THAN FCE OCCURED.
8. READ FORWARD.
9. REPEAT STEPS 6 AND 7
10. REPEAT STEPS 1 THRU 9 FOR PE.
11. END

FT17: TAPE MARK SPACE TEST: THIS TEST WILL ASSURE THAT SPACING WILL BE TERMINATED BY RECOGNITION OF TAPE MARK BOTH IN PE AND NRZ.

1. REWIND TO BOT.
2. WRITE THE FOLLOWING PATTERN OF TAPE MARKS AND DATA RECORDS:

TM:20 RECS:TM:40 RECS:TM:60 RECS:TM:100 RECS:TM:

3. ASSURE NO ERRORS.
4. ASSURE THAT TAPE MARK STATUS IS SET FOR TM WRITES.
5. NOW SPACE REVERSE 200 RECORDS.
6. THE SPACE OPERATION SHOULD STOP ON EACH TAPE MARK IT FINDS. THEREFOR 5 SPACE COMMANDS ARE ISSUED TO COVER THE ENTIRE PATTERN WRITTEN ON TAPE. BOT SHOULD NEVER BE REACHED AND THE FRAME COUNT WILL REFELCT THE NUMBER OF RECORDS BETWEEN TAPE MARKS.
7. REPEAT STEP 6 IN THE FORWARD DIRECTION.
8. ASSURE NO ERRORS OTHER THAN FCE.
9. REPEAT STEPS 1 THRU 8 FOR PE
10. END

422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476

FT20: WRITE CHECK TEST: BOTH WRITE CHECK FORWARD AND REVERSE ARE TESTED IN BOTH PE AND NRZ.

1. REWIND TO BOT.
2. WRITE A 400 FRAME RECORD USING DATA PATTERN 3 (125125).
3. ASSURE NO ERRORS OCCURED.
4. ISSUE A REVERSE WRITE CHECK COMMAND.
5. ASSURE NO ERRORS OCCURED.
6. REPEAT STEP 5 FOR A FORWARD WRITE CHECK.
7. REPEAT STEPS 1 THRU 6 FOR PE.
8. END

FT21: ERASE HEAD TEST: THIS TEST WILL ASSURE THAT THE ERASE HEAD ITSELF IS OPERATING.

1. REWIND TO BOT.
2. WRITE 2 RECORDS OF 800(10) FRAMES EACH. EACH RECORD WILL BE 1 INCH OF TAPE. DATA IS NOT ALL ONES.
3. REWIND TO BOT.
4. NOW WRITE A 400(10) FRAME RECORD. THIS RECORD WILL BE ONE HALF INCH OF TAPE. THE ERASE HEAD SHOULD CLEAR THE REMAINDER OF THE FIRST RECORD (ONE HALF INCH).
5. REWIND TO BOT.
6. NOW READ THE SHORT FIRST RECORD. IT SHOULD BE 400(10) FRAMES.
7. NOW READ THE SECOND RECORD. IT SHOULD BE STILL 800(10) FRAMES.
8. IF THE SECOND RECORD IS TOO LONG, THE ERASE HEAD DID NOT FUNCTION OR IT IS IN THE WRONG POLARITY.
10. END

FT22: BUFFERED COMMAND: THIS TEST WILL ASSURE THAT THE TM02 WILL ACCEPT AND EXECUTE ANOTHER COMMAND WHILE ITS SELECTED SLAVE IS REWINDING.

1. REWIND TO BOT.
2. ISSUE 3 LONG WRITE COMMANDS TO ASSURE BEING OFF BOT.
3. ISSUE A REWIND COMMAND.
4. AS SOON AS DRIVE READY BECOMES SET, ISSUE ANOTHER WRITE COMMAND.
5. THE NEXT DRIVE READY SHOULD BE AFTER THE TAPE HAS REACHED BOT AND EXECUTED THE BUFFERED WRITE COMMAND.
6. ASSURE NO ERRORS OCCURED.
7. END

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

FT23: READ IN PRESET: THIS TEST WILL ASSURE THAT UNIT 0
IS REWOUND AND SET TO 800 BPI NORMAL.
(ONLY IF SLAVE 0 IS SELECTED).

1. ISSUE A WRITE COMMAND TO ASSURE
BEING OFF BOT.
2. ISSUE THE READ-IN PRESET COMMAND.
3. AWAIT MOTION STOP.
4. ASSURE THAT BOT WAS REACHED.
5. ASSURE THAT THE TAPE CONTROL REGISTER
IS SET TO 800 BPI,NORMAL,ODD.
6. END

(THIS TEST IS ONLY PERFORMED IF THE SELECTED SLAVE IS ZERO (0).

FT24: REWIND: OFF LINE THIS TEST WILL ASSURE
THAT THE UNIT WILL REWIND AND
GO OFF LINE. (NOT IF IN CONTINUOUS CYCLE)

1. ISSUE THE REWIND OFF-LINE COMMAND.
2. ASSURE THAT MOL (BIT 12 OF DRIVE STATUS)
IS RESET INDICATING THE UNIT WENT OFF LINE.
3. END

(THIS TEST IS NOT PERFORMED WHEN CONTINUOUS CYCLE OPERATION IS S

506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535

9. LISTING

E

.TITLE CZTUBHO TM02 - TU16/TE16 BSC FC
;ZZ - CZTUB-M-0
;15 AUGUST 1983
;R. BARNES
;REVISED APRIL .1976 BY S. CARPENTER
;REVISED NOV .1977 BY CLEM WALSH
; 1) SUPPORTS SOFTWARE SWITCH REGISTER
; 2) SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
;REVISED SEP .1983 BY B.T. LEBLANC
; 1) TEST FOR ATA SET IN DS AFTER NON-TRANSFER OPERATION CC0001619

;CONSOLE SWITCHES*****
;
;SW1(10000): 1=HALT ON ERROR
; 0=CONTINUE
;SW14(040000) 1=LOOP ON ERROR (SCOPE(040000) RM TESTS ONLY)
; 0=CONTINUE
;SW13(02000): 1=DO NOT PRINT ERRORS
; 0=PRINT ERRORS
;SW12(010000): 1=INHIBIT ITERATIONS
; 0=DO ITERATIONS
;SW11(004000): 1=CONTINUOUS CYCLE
; 0=HALT AT END OF PASS
;SW10(002000): 1=HALT AT END OF EACH TEST
; 0=CONTINUE
;SW0-4: SELECT TEST NUMBER :: 00=ALL TESTS


```

584          .ENABL ABS
585          ;REGISTER EQUIVS*****
586
587          000000          R0=#0
588          000001          R1=#1
589          000002          R2=#2
590          000003          R3=#3
591          000004          R4=#4
592          000005          R5=#5
593          000006          SP=#6
594          000007          PC=#7
595
596          ;TRAP CATCHERS*****
597
598          000000          . = 0
599          000200          .REPT 200
600
601          .+2
602          HALT
603          .ENDR
604          ;
605          ; *****
606          ; ACT11 HOOKS
607          ; *****
608          001000          $SVPC=.
609
610          000040          . = 40
611          000040          000          DRIVE: .BYTE 0          ;DRIVE # FOR XXDP LOAD MEDIUM
612
613
614          000041          000          MEDIUM: . = 41          ;XXDP LOAD MEDIUM
615          000041          000          .BYTE 0          ;ASSEMBLE AS A 0
616
617
618          000042          000000          . = 42          ;LOCATION INDICATOR - AUTOM/MAN MODE
619          000042          000000          .WORD 0          ;ASSEMBLE AS A 0
620
621
622          000046          000046          . = 46          ;SET TO $ENDAD IN .$EOP
623          000046          003334          .WORD $ENDAD
624
625          000052          000052          . = 52          ;CHARACTERISTICS OF PROGRAM
626          000052          000000          .WORD 0          ;SET TO 0
627
628
629          001000          .=$SVPC          ;RESTORE PC
630
631          ; *****
632
633          ;TTY INTERRUPT VECTOR*****
634
635
636          000060          000060          . = 60
637          000060          012630          TTYINT          ;TTY INTERRUPT HEADER ADDRESS
638          000062          000000          0
639

```

```

640 ;SOFTWARE SWITCH REGISTER*****
641
642 . = 176
643 000176 000176 SWREG: 0 ;SOFTWARE SWITCH REGISTER
644
645
646 ;*****
647
648 ;THIS PROGRAM SUPPORTS THE SOFTWARE SWITCH REGISTER LOC.176.
649 ;REFER TO SECTION 5 OF DOCUMENT FOR DESCRIPTION
650
651 ;*****
652 ;START ADDRESS*****
653
654 . = 200
655 000200 005000 CLR R0
656 000202 000167 001372 JMP START ;PROGRAM START
657
658 . = 210
659 000210 000240 NOP
660 000212 012700 000001 MOV #1,R0 ;SET NO HEADER FLAG
661 000216 000167 001356 JMP START
662
663 ;TM02 INTERRUPT VECTOR*****
664
665 . = 224
666 000224 012614 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
667 000226 000340 340
668
669
670 . = 600
671 000600 ;MASS BUS REGISTER EQUIVS*****
672
673 C1: 172440
674 WC: 172442
675 BA: 172444
676 FC: 172446
677 CS: 172450
678 DS: 172452
679 ER: 172454
680 AS: 172456
681 CC: 172460
682 DB: 172462
683 MR: 172464
684 DT: 172466
685 SN: 172470
686 TC: 172472
687 BAE: 172474
688
689 ;CONSTANTS*****
690
691 PSW: 177776 ;PROCESSOR STATUS
692 SWR: 177570 ;SWITCH REGISTER
693 TKS: 177560 ;TTY READER STATUS
694 TKB: 177562 ;TTY READ BUFFER
695 TPS: 177564 ;TTY PUNCH STATUS

```

696 000650 177566
 697 000652 177777
 698 000654 000011
 699 000656 000010
 700 000660 000224
 701 000662 172440
 702 000664 000004
 703 000666 000006
 704
 705
 706
 707
 708
 709
 710 000670 000000
 711 000672 000
 712 000673 000
 713 000674 000
 714 000675 000
 715
 716

TPB: 177566 ;TTY PUNCH BUFFER
 SERNUM: 177777 ;SERIAL NUMBER
 DRVTP: 011 ;DRIVE TYPE
 ITAMT: 10 ;ITERATION AMOUNT
 VECT: 224 ;INTERRUPT VECTOR(RH)
 REGS: 172440 ;STARTING REGISTER ADDRESS
 BTRP: 4 ;BUS TRAP ADDRESS
 BTRP2: 6 ;BUS TRAP PRIORITY LEVEL 7

```

; *****
;               ACT11  MODE INDICATORS
; *****
AUTOM:  .WORD  0      ;AUTOMATIC MODE INDICATOR
ACT11M: .BYTE  0      ;ACT11 AUTO MODE INDICATOR
XXDPM:  .BYTE  0      ;XXDP AUTO MODE INDICATOR
ADUMP:  .BYTE  0      ;ACT11 DUMP INDICATOR
XDUMP:  .BYTE  0      ;XXDP DUMP MODE INDICATOR
; *****

```

```

718                                     ;FLAGS AND COUNTERS*****
719
720 000676 000000 TOB: 0
721 000700 000000 TIB: 0
722 000702 000000 RH17F: 0
723 000704 000000 HDRFL: 0
724 000706 000000 EMADDR: 0
725 000710 000000 DRVN: 0
726 000712 000000 SLVN: 0
727 000714 000000 BADDR: 0
728 000716 000000 FCNT: 0
729 000720 000000 WCNT: 0
730 000722 000000 RCNT: 0
731 000724 000000 ERRP: 0
732 000726 000000 ERRP1: 0
733 000730 000000 RRD: 0
734 000732 000000 RFD: 0
735 000734 000000 RDYDX: 0
736 000736 000000 OPDYX: 0
737 000740 000000 SCNT: 0
738 000742 000000 PFLG: 0
739 000744 000000 RTRN: 0
740 000746 000000 ERADD: 0
741 000750 000000 TEMP1: 0
742 000752 000000 TEMP2: 0
743 000754 000000 TEMP3: 0
744 000756 000000 STMSK: 0
745 000760 000000 ITCNT: 0
746 000762 000000 DSAV: 0
747 000764 000000 SAV1: 0
748 000766 000000 SAV2: 0
749 000770 000000 SAV3: 0
750 000772 000000 SCOLP: 0
751 000774 000000 ITRLP: 0
752 000776 000000 EXFL: 0
753 001000 000000 PEXFL: 0
754 001002 000000 STFLG: 0
755 001004 000000 LTADD: 0
756 001006 000000 FUN: 0
757 001010 000000 SERFL: 0
758 001012 000000 CRCNT: 0
759 001014 000000 UDES: 0
760 001016 000000 PATRN: 0
761 001020 000000 RHTF: 0
762 001022 000000 NRZOF: 0
763 001024 000000 RHOF: 0
764 001026 000000 PCNTR: 0
765 001030 000000 TEMPST: 0
766 001032 000000 COUNT: 0
767 001034 000000 RDSW: 0
768 001036 000000 PAFLG: 0
769

```

771
772

;DATA PATTERN GENERATORS*****

773 001040 000000
774 001042 012364
775 001044 012406
776 001046 012414
777 001050 012424

DATBL: 0
DATA0: DAT1 ;ALL ONE BITS
DATA1: DAT2 ;ALL ZERO BITS
DATA2: DAT3 ;ALTERNATING ONE/ZERO BITS
DATA3: DAT4 ;ALL BITS 0-377

Line	Code	Value	Label
779			
780			:LOGIC TEST ENTRY TABLE*****
781			
782	001052	000000	TSTTBL: 0
783	001054	000000	0
784	001056	003412	FT1
785	001060	003412	FT1
786	001062	003512	FT2
787	001064	003512	FT2
788	001066	004036	FT3
789	001070	004036	FT3
790	001072	004254	FT4
791	001074	004254	FT4
792	001076	004402	FT5
793	001100	004402	FT5
794	001102	004604	FT6
795	001104	004604	FT6
796	001106	005072	FT7
797	001110	005072	FT7
798	001112	005166	FT10
799	001114	005166	FT10
800	001116	005322	FT11
801	001120	005322	FT11
802	001122	005440	FT12
803	001124	005440	FT12
804	001126	005552	FT13
805	001130	005552	FT13
806	001132	006104	FT14
807	001134	006104	FT14
808	001136	007000	FT15
809	001140	007000	FT15
810	001142	007200	FT16
811	001144	007200	FT16
812	001146	007442	FT17
813	001150	007442	FT17
814	001152	010056	FT20
815	001154	010056	FT20
816	001156	010316	FT21
817	001160	010316	FT21
818	001162	010646	FT22
819	001164	010646	FT22
820	001166	011052	FT23
821	001170	011052	FT23
822	001172	011272	FT24
823	001174	011272	FT24
824	001176	000000	0
825	001200	000000	0
826	001202	000000	0
827	001204	000000	0

```

829          001600          . = 1600
830          ;PROGRAM START AND HOUSEKEEPING*****
831
832 001600 000240          START: NOP
833 001602 005067 177230  CLR      PAFLG          ;INIT PASS FLAG
834 001606 005067 177214  CLR      PCNTR         ;INIT PASS COUNTER
835 001612 012777 000340 177016 RESTRT: MOV     #340,SPSW    ;SET PRIORITY
836 001620 012706 000500  MOV     #500,SP      ;SET STACK POINTER
837
838          ; *****
839          ;          DIAGNOSTIC SETUP FOR EXECUTION
840          ;          UNDER ACT11.
841          ; *****
842
843 001624 004767 012474  JSR     PC,CKMODE      ;CHECK FOR MODE OF OPERATION
844 001630 005767 177034  TST     AUTOM          ;IS IT AUTOMATIC MODE
845 001634 001001          BNE     1$            ;BRANCH - IF YES
846 001636 000412          BR      SUSWR          ;CHECK SWR IN DUMPM - IF NOT
847 001640 032737 020000 000052 1$: BIT     #20000,#52    ;SET UP FOR MANUAL INTERVENTION?
848 001646 001406          BEQ     SUSWR          ;BRANCH - IF NO
849 001650 012704 014460  MOV     #MSGC,R4      ;GET MESSAGE
850 001654 004767 011334  JSR     PC,TTOUT      ;TYPE MESSAGE
851 001660 000167 012540  JMP     ABORT          ;AND ABORT THE PROGRAM
852
853          ; *****
854
855 001664 013746 000006  SUSWR: MOV     #06,-(SP)    ;SAVE VECTORS
856 001670 013746 000004  MOV     #04,-(SP)
857 001674 012737 001714 000004  MOV     #1$,#04      ;SET UP FOR TIMEOUT
858 001702 022777 177777 176730  CMP     #-1,#SWR     ;REFERENCE HARDWARE SWITCH REGISTER
859 001710 001402          BEQ     2$
860 001712 000404          BR      3$
861 001714 022626          1$: CMP     (SP)+,(SP)+    ;ADJUST STACK
862 001716 012767 000176 176714  2$: MOV     #SWREG,SWR  ;POINT TO SOFTWARE SWITCH REG
863 001724 012637 000004          3$: MOV     (SP)+,#04    ;RESTORE VECTORS
864 001730 012637 000006  MOV     (SP)+,#06
865 001734 023727 000640 000176  CMP     #SWR,#SWREG  ;IS SOFTWARE REG USED
866 001742 001002          BNE     4$            ;BRANCH IF NO
867 001744 004767 012112  JSR     PC,CNTLU     ;ALLOW SOFTWARE SWITCH REGISTER TO BE CHANGED
868
869 001750          4$:
870
871          ; *****
872          ;          IF IN ACT11 MODE INHIBIT TYPING PROGRAM
873          ;          IDENTIFICATION AND MANUAL INTERVENTION
874          ; *****
875
876 001750 005767 176716  TST     ACT11M        ;CHECK FOR ACT11 MODE
877 001754 001104          BNE     ST            ;BRANCH - IF ACT11
878          ; *****
879
880 001756 005700          TST     R0            ;SEE IF PRINT HEADER
881 001760 001402          BEQ     STOA         ;IF SO: BR
882 001762 000167 000764  JMP     ST4           ;ELSE SKIP
883 001766 012704 014664  STOA: MOV     #MSG3,R4
884 001772 004767 011216  JSR     PC,TTOUT     ;PRINT TITLE

```

```

885 001776 012704 014766      ST0B:  MOV    #MSG4,R4
886 002002 004767 011206      JSR    PC,TTOUT      ;REQUEST REGISTER ADDRESS
887 002006 016703 176650      MOV    REGS,R3
888 002012 004767 011342      JSR    PC,OCTP      ;PRINT CURRENT ADDRESS
889 002016 012705 000662      MOV    #REGS,R5      ;SET ADDRESS SAVE LOC
890 002022 012701 000006      MOV    #6,R1         ;SET SIZE OF RESPONSE
891 002026 012702 176400      MOV    #176400,R2    ;SET UPPER LIMIT
892 002032 012703 172300      MOV    #172300,R3    ;SET LOWER LIMIT
893 002036 004767 010716      JSR    PC,TTR        ;GO GET RESPONSE
894 002042 012704 015011      MOV    #MSG5,R4
895 002046 004767 011142      JSR    PC,TTOUT      ;REQUEST VECTOR
896 002052 016703 176602      MOV    VECT,R3
897 002056 004767 011276      JSR    PC,OCTP      ;PRINT CURRENT VECTOR
898 002062 012705 000660      MOV    #VECT,R5      ;SET ADDRESS SAVE LOC
899 002066 012701 000003      MOV    #3,R1         ;SET SIZE OF RESPONSE
900 002072 012702 000224      MOV    #224,R2       ;SET UPPER LIMIT
901 002076 012703 000150      MOV    #150,R3       ;SET LOWER LIMIT
902 002102 004767 010652      JSR    PC,TTR        ;GO GET RESPONSE
903 002106 016700 176546      MOV    VECT,R0       ;GET VECTOR
904 002112 012720 012614      MOV    #MTINT,(R0)+  ;LOAD INTERRUPT ADDRESS IN VECTOR
905 002116 012710 000340      MOV    #340,(R0)     ;LOAD PRIORITY
906 002122 016700 176534      MOV    REGS,R0       ;GET START OF REGS
907 002126 012701 000016      MOV    #16,R1        ;SET NUMBER OF REGS
908 002132 012702 000600      MOV    #C1,R2        ;GET START OF TABLE
909 002136 010022 000002      ST0:  MOV    R0,(R2)+  ;BUILD TABLE
910 002140 062700 000002      ADD    #2,R0         ;BUMP ADDRESS
911 002144 005301 000000      DEC    R1            ;SEE IF DONE
912 002146 001373 000000      BNE    ST0          ;IF NOT: BR
913 002150 012702 000676      MOV    #TOB,R2
914 002154 012700 000054      MOV    #54,R0
915 002160 005022 000000      ST1:  CLR    (R2)+     ;CLEAR FLAGS + COUNTERS
916 002162 005300 000000      DEC    R0
917 002164 001375 000000      BNE    ST1
918 002166 012767 000001 176624 ST:  MOV    #1,RHTF      ;SET ADDRESS TEST FLAG
919 002174 000167 000650      JMP    TSRH         ;GO DO INITIAL ADDRESS TEST PASS
920
921 002200      ST1A:
922
923      ; *****
924 002200 005767 176464      TST    AUTOM        ;CHECK FOR AUTOMATIC MODE
925 002204 001017 000000      BNE    1$          ;BRANCH - IF YES
926      ; *****
927
928 002206 012704 015072      MOV    #MSG10,R4
929 002212 004767 010776      JSR    PC,TTOUT      ;REQUEST DRIVE NUMBER
930 002216 012705 000710      MOV    #DRVN,R5      ;SET ADDRESS OF DRIVE NUMBER SAVE
931 002222 012701 000001      MOV    #1,R1         ;SET SIZE OF RESPONSE
932 002226 012702 000007      MOV    #7,R2         ;SET UPPER LIMIT
933 002232 012703 000000      MOV    #0,R3         ;SET LOWER LIMIT
934 002236 004767 010516      JSR    PC,TTR        ;GO GET RESPONSE
935 002242 000434 000000      BR     CONT1        ;EXIT
936
937      ; *****
938      ; AUTOMATICALLY SIZE FOR DRIVES
939      ; *****
940 002244      1$:
    
```

```

941 002244 012767 177777 176436      MOV      #-1,DRVN      ;INIT DRIVE #
942 002252 012767 177777 176432  NXTDRV: MOV      #-1,SLVN      ;INIT SLAVE #
943 002260 012777 000040 176322  1$:      MOV      #40,@CS      ;INIT CONTROLLER
944 002266 005267 176416          INC      DRVN          ;STEP DRIVE #
945 002272 022767 000010 176410      CMP      #10,DRVN      ;ALL DRIVES TESTED?
946 002300 001002          BNE      2$           ;BRANCH - IF NOT
947 002302 000167 000764          JMP      TENDO         ;EXIT
948 002306 016777 176376 176274  2$:      MOV      DRVN,@CS      ;LOAD DRIVE #
949 002314 005777 176260          TST      @C1           ;ACCESS DRIVE
950 002320 032777 010000 176262      BIT      #10000,@CS    ;NON-EXISTANT DRIVE?
951 002326 001354          BNE      1$           ;BRANCH - IF YES (NED=1)
952 002330 000167 000106          JMP      NXTSLV        ;EXIT TO SIZE FOR SLAVES
953
954      ; *****
955
956 002334 012777 000040 176246  CONT1:  MOV      #40,@CS      ;SET INIT
957 002342 056777 176342 176240      BIS      DRVN,@CS      ;SET DRIVE NUMBER
958 002350 005777 176224          TST      @C1           ;ACCESS DRIVE
959 002354 032777 010000 176226      BIT      #10000,@CS    ;SEE IF NED
960 002362 001405          BEQ      ST2           ;IF NOT: BR
961 002364 012704 016024          MOV      #MSG41,R4
962 002370 004767 010620          JSR      PC,TTOUT      ;PRINT NOT AVAIL
963 002374 000701          BR       ST1A          ;REDO DRIVE REQUEST
964
965 002376          ST2:
966
967      ; *****
968 002376 005767 176266          TST      AUTOM          ;CHECK FOR AUTOMATIC MODE
969 002402 001017          BNE      1$           ;BRANCH - IF YES
970      ; *****
971
972 002404 012704 015112          MOV      #MSG11,R4
973 002410 004767 010600          JSR      PC,TTOUT      ;REQUEST SLAVE NUMBER
974 002414 012705 000712          MOV      #SLVN,R5      ;SET ADDRESS OF SLAVE SAVE
975 002420 012701 000001          MOV      #1,R1         ;SET SIZE OF RESPONSE
976 002424 012702 000007          MOV      #7,R2         ;SET UPPER LIMIT
977 002430 012703 000000          MOV      #0,R3         ;SET LOWER LIMIT
978 002434 004767 010320          JSR      PC,TTR        ;GO GET RESPONSE
979 002440 000432          BR       CONT2        ;AND EXIT
980
981      ; *****
982      ; AUTOMATICALLY SIZE FOR SLAVES
983      ; *****
984 002442          1$:
985 002442 005267 176244          NXTSLV: INC      SLVN          ;STEP SLAVE #
986 002446 001010          BNE      1$           ;BRANCH - IF NOT SLAVE 0
987 002450 005767 176234          TST      DRVN          ;DRIVE 0?
988 002454 001005          BNE      1$           ;BRANCH - IF NOT
989 002456 105767 176211          TSTB    XXDPM          ;CHAIN MODE?
990 002462 001402          BEQ      1$           ;BRANCH - IF NOT
991 002464 005267 176222          INC      SLVN          ;STEP TO NEXT SLAVE
992 002470 022767 000010 176214  1$:      CMP      #10,SLVN      ;ALL SLAVES TESTED?
993 002476 001665          BEQ      NXTDRV        ;BRANCH - IF YES
994 002500 016777 176206 176124      MOV      SLVN,@TC      ;LOAD SLAVE UNIT #
995 002506 032777 002000 176112      BIT      #2000,@DT     ;SLAVE PRESENT
996 002514 001752          BEQ      NXTSLV        ;BRANCH - IF NOT (SPR=0)

```

```

997 002516 032777 140000 176102 BIT #140000,@DT ;IS DRIVE A TAPE UNIT
998 002524 001746 BEQ NXTSLV ;BRANCH - IF NOT
999
1000 ; *****
1001
1002 002526 012777 000040 176054 CONT2: MOV #40,@CS ;INIT
1003 002534 056777 176150 176046 BIS DRVN,@CS ;SET DRIVE NUMBER
1004 002542 016777 176144 176062 MOV SLVN,@TC ;LOAD SLAVE NUMBER
1005 002550 032777 002000 176050 BIT #2000,@DT ;SEE IF SLAVE PRESENT
1006 002556 001005 BNE ST3 ;IF SO: BR
1007 002560 012704 016045 MOV #MSG42,R4
1008 002564 004767 010424 JSR PC,TTOUT ;PRINT NON-EXIST SLAVE
1009 002570 000702 BR ST2 ;REDO SLAVE REQUEST
1010 002572 012704 016066 ST3: MOV #MSG43,R4
1011 002576 004767 010412 JSR PC,TTOUT ;PRINT SERIAL NUMBER TAG
1012 002602 017703 176022 MOV @SN,R3
1013 002606 004767 011074 JSR PC,SNPT ;PRINT SERIAL NUMBER
1014
1015 ; *****
1016 002612 005767 176052 TST AUTOM ;CHECK FOR AUTOMATIC MODE
1017 002616 001057 BNE TSCD ;BRANCH - IF YES
1018 ; *****
1019
1020 002620 012704 016650 MOV #MSG61,R4
1021 002624 004767 010364 JSR PC,TTOUT ;REQUEST RH11 OR RH70
1022 002630 012705 000702 MOV #RH17F,R5 ;GET ADDRESS OF FLAG
1023 002634 012701 000001 MOV #1,R1 ;SET SIZE OF RESPONSE
1024 002640 012702 000001 MOV #1,R2 ;SET UPPER LIMIT
1025 002644 012703 000000 MOV #0,R3 ;SET LOWER LIMIT
1026 002650 004767 010104 JSR PC,TTR ;GET RESPONSE
1027 002654 012704 016670 MOV #MSG62,R4
1028 002660 004767 010330 JSR PC,TTOUT ;REQUEST RH11 ONLY RESPONSE
1029 002664 012705 001024 MOV #RHOF,R5 ;SET FLAG ADDRESS
1030 002670 012701 000001 MOV #1,R1 ;SET SIZE OF RESPONSE
1031 002674 012702 000001 MOV #1,R2 ;SET UPPER LIMIT
1032 002700 012703 000000 MOV #0,R3 ;SET LOWER LIMIT
1033 002704 004767 010050 JSR PC,TTR ;GO GET RESPONSE
1034 002710 005767 176110 TST RHOF ;SEE IF RH11 ONLY
1035 002714 001016 BNE ST4 ;IF SO: BR
1036 002716 012704 016541 MOV #MSG55,R4
1037 002722 004767 010266 JSR PC,TTOUT ;REQUEST NRZ ONLY RESPONSE
1038 002726 012705 001022 MOV #NRZOF,R5 ;SET FLAG ADDRESS
1039 002732 012701 000001 MOV #1,R1 ;SET SIZE OF RESPONSE
1040 002736 012702 000001 MOV #1,R2 ;SET UPPER LIMIT
1041 002742 012703 000000 MOV #0,R3 ;SET LOWER LIMIT
1042 002746 004767 010006 JSR PC,TTR ;GO GET RESPONSE
1043 002752 005067 176050 ST4: CLR PCNTR ;CLEAR PASS COUNTER

```

```

1045 ;TEST SCHEDULAR*****
1046
1047 002756 000240 TSCD: NOP
1048 002760 005067 176016 CLR STFLG ;CLEAR SINGLE TEST FLAG
1049
1050 ; *****
1051
1052 002764 005067 175712 CLR RH17F ;SET RH11 INDICATOR
1053 002770 013746 000004 MOV @#4,-(SP) ;SAVE ERROR TRAP AND VECTORS
1054 002774 013746 000006 MOV @#6,-(SP) ;SAVE PRIORITY
1055 003000 012737 003024 000004 MOV @1,@#4 ;SET TIME OUT
1056 003006 005037 000006 CLR @#6 ;SET LOW PRIORITY
1057 003012 005777 175616 TST @BAE ;REFERENCE BAE REGISTER
1058 003016 012767 000001 175656 MOV @1,RH17F ;SET RH70 INDICATOR
1059 003024 012637 000006 1$: MOV (SP)+,@#6 ;RESTORE ERROR TYPE
1060 003030 012637 000004 MOV (SP)+,@#4
1061
1062 ; *****
1063
1064 003034 017700 175600 MOV @SWR,R0
1065 003040 042700 177740 BIC @177740,R0
1066 003044 005700 TST R0
1067 003046 001055 BNE STSCD ;GO SELECT SINGLE TEST
1068 003050 012767 001052 175726 TSRH: MOV @TSTTBL,LTADD
1069 003056 062767 000004 175720 TSCD0: ADD @4,LTADD
1070 003064 016767 175714 175702 MOV LTADD,ITRLP
1071 003072 062767 000002 175674 ADD @2,ITRLP ;SET ITERATION ADDRESS
1072 003100 005777 175700 TST @LTADD
1073 003104 001002 BNE TSCD1
1074 003106 000167 000144 JMP TEND ;GO TO END ROUTINE
1075 003112 000240 TSCD1: NOP
1076 003114 005067 175636 CLR STMSK
1077 003120 005067 175600 CLR ERRP
1078 003124 005067 175554 CLR HDRFL ;CLEAR PRINT HEADER FLAG
1079 003130 017700 175650 MOV @LTADD,R0 ;SET POINTER TO TEST
1080 003134 000110 JMP (R0) ;GO TO TEST
1081 003136 000240 TSCD2: NOP
1082 003140 032777 002000 175472 BIT @2000,@SWR ;SEE IF HALT ON TEST
1083 003146 001401 BEQ TSCD3 ;IF NOT: BR
1084 003150 000000 HALT
1085 003152 004767 010632 TSCD3: JSR PC,CKSWR ;CHECK FOR CNTL G
1086 003156 000240 NOP
1087 003160 005767 175616 TST STFLG ;SE IF SINGLE TEST
1088 003164 001734 BEQ TSCD0 ;IF NOT: BR
1089 003166 017700 175446 MOV @SWR,R0
1090 003172 042700 177740 BIC @177740,R0 ;MASK TEST NUMBER
1091 003176 005700 TST R0 ;SEE IF RETURN TO ALL
1092 003200 001666 BEQ TSCD ;IF SO: BR
1093 003202 000240 STSCD: NOP
1094 003204 012767 000001 175570 MOV @1,STFLG ;SET SINGLE TEST FLAG
1095 003212 022700 000025 CMP @25,R0 ;SEE IF EXCEEDED TESTS
1096 003216 003417 BLE TEND ;IF SO: BR
1097 003220 000241 CLC
1098 003222 006100 ROL R0
1099 003224 006100 ROL R0 ;SET TABLE MODIFIER
1100 003226 012767 001052 175550 MOV @TSTTBL,LTADD
  
```

```

1101 003234 060067 175544          ADD    RO,LTADD      ;SET TEST POINTER
1102 003240 016767 175540 175526  MOV    LTADD,ITRLP
1103 003246 062767 000002 175520  ADD    @2,ITRLP     ;SET ITERATION POINTER
1104 003254 000716          BR     TSCD1
1105
1106 ; .....
1107
1108 003256 000240          TEND:  NOP
1109 003260 005767 175404          TST    AUTOM        ;CHECK FOR AUTO MODE
1110 003264 001402          BEQ    TENDO        ;BRANCH - IF NOT
1111 003266 000167 177150          JMP    NXTSLV       ;GET ANOTHER SLAVE DEVICE
1112 003272
1113 TENDO:
1114 ; .....
1115
1116 003272 012704 015024          MOV    @MSG6,R4
1117 003276 004767 007712          JSR    PC,TTOUT     ;PRINT END OF PASS
1118 003302 016703 175520          MOV    PCNTR,R3
1119 003306 004767 010046          JSR    PC,OCTP      ;PRINT PASS NUMBER
1120
1121 ; .....
1122 ; AUTOMATIC MODE END OF PASS
1123 ; .....
1124
1125 003312 005767 175520          TST    PAFLG        ;PASS INDICATOR SET?
1126 003316 001002          BNE    3$           ;BRANCH - IF SET
1127 003320 005267 175512          INC    PAFLG        ;SET PASS INDICATOR
1128 003324 013704 000042          3$:  MOV    @42,R4    ;CONTENTS OF 42 TO R4
1129 003330 001405          BEQ    HERE         ;BRANCH - IF NOT AUTO MODE
1130 003332 000005          RESET              ;CLEAR THE WORLD
1131 003334 004714          $ENDAD: JSR    PC,(R4) ;RETURN TO MONITOR
1132 003336 000240          NOP
1133 003340 000240          NOP
1134 003342 000240          NOP
1135 003344
1136 003344 005767 175320          HERE:  TST    AUTOM        ;CHECK FOR AUTOMATIC MODE
1137 003350 001005          BNE    TENDX        ;BRANCH - IF YES
1138 003352 032777 004000 175260  BIT    @4000,BSWR   ;SEE IF HALT ON PASS
1139 003360 001001          BNE    TENDX        ;IF NOT: BR
1140 003362 000000          HALT
1141 TENDX:
1142 003364 005767 175300          TST    AUTOM        ;CHECK FOR AUTO MODE
1143 003370 001402          BEQ    1$           ;BRANCH - IF NOT
1144 003372 000167 000004          JMP    EXIT         ;RESTART
1145 003376 004767 010406          1$:  JSR    PC,CKSWR   ;CHECK FOR CNTL G
1146 003402 005267 175420          EXIT:  INC    PCNTR   ;BUMP PASS COUNTER
1147 003406 000167 176200          JMP    RESTRT      ;RESTART
1148
1149 ; .....

```

```

1151
1152
1153
1154 003412 012767 016755 175266 FT1:  MOV    #MSFT1,EMADDR    ;SET HEADER
1155 003420 012777 012640 175236      MOV    @TRAP,@BTRP    ;SET TRAP HANDLER ADDRESS
1156 003426 012777 000340 175232      MOV    #340,@BTRP2
1157 003434 012700 000016      MOV    #16,R0        ;SET NUMBER OF REGISTER
1158 003440 016701 175134      MOV    C1,R1        ;GET FIRST ADDRESS (CS1)
1159 003444 005711      FT1A:  TST    (R1)      ;REFERENCE REGISTER
1160 003446 000240      NOP
1161 003450 005300      FT1B:  DEC    R0        ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
1162 003452 001403      BEQ    FT1X          ;SEE IF DONE ALL
1163 003454 062701 000002      ADD    #2,R1        ;IF SO: BR
1164 003460 000771      BR     FT1A          ;BUMP ADDRESS POINTER
1165 003462 012777 000006 175174 FT1X:  MOV    #6,@BTRP    ;CONTINUE
1166 003470 005767 175324      TST    RHTF         ;RESET TRAP CATCHER
1167 003474 001404      BEQ    FT1XX        ;SEE IF INITIAL ADDRESS TEST PASS
1168 003476 005067 175316      CLR    RHTF         ;IF NOT: BR
1169 003502 000167 176472      JMP    ST1A         ;CLEAR FLAG
1170 003506 000167 177424      FT1XX: JMP    TSCD2    ;RETURN
                                ;RETURN TO SCHEDULAR

```

```

1172
1173 ;RM REGISTER BITS READ/WRITE*****
1174
1175 003512 012767 017002 175166 FT2:  MOV    #MSFT2,EMADDR ;SET TEST HEADER
1176 003520 012701 177777          MOV    @-1,R1      ;SET ALL ONES PATTERN
1177 003524 004767 007036          FT2A: JSR    PC,INIT1   ;GO INIT
1178 003530 016700 175046          MOV    WC,R0      ;GET ADDRESS OF WORD COUNT
1179 003534 010102          MOV    R1,R2     ;SET EXPT REGISTER BIT PATTERN
1180 003536 010110          MOV    R1,(R0)   ;LOAD PATTERN
1181 003540 021002          CMP    (R0),R2   ;SEE IF EXPT=RCVD
1182 003542 001410          BEQ    FT2B      ;IF SO: BR
1183 003544 012767 015352 175174  MOV    #MSG25,ERADD ;SET CODE
1184 003552 012767 003524 175212  MOV    #FT2A,SCOLP ;SET SCOPE
1185 003560 004767 000116          JSR    PC,FT2ER  ;GO DO ERROR
1186 003564 016700 175014          FT2B: MOV    BA,R0     ;GET ADDRESS OF BUS ADDRESS
1187 003570 010102          MOV    R1,R2
1188 003572 042702 000001          BIC    @1,R2     ;SET EXPT PATTERN
1189 003576 010110          MOV    R1,(R0)  ;LOAD PATTERN
1190 003600 020210          CMP    R2,(R0)  ;SEE IF EXPT=RCVD
1191 003602 001410          BEQ    FT2C      ;IF SO:BR
1192 003604 012767 015360 175134  MOV    #MSG26,ERADD ;SET ERROR CODE
1193 003612 012767 003564 175152  MOV    #FT2B,SCOLP ;SET SCOPE ADDRESS
1194 003620 004767 000056          JSR    PC,FT2ER  ;GO DO ERROR
1195 003624 016700 174772          FT2C: MOV    DB,R0     ;GET ADDRESS OF DATA BUFFER
1196 003630 010102          MOV    R1,R2
1197 003632 010110          MOV    R1,(R0)  ;LOAD PATTERN
1198 003634 012703 004000          MOV    #4000,R3
1199 003640 005303          FT2D: DEC    R3      ;DELAY
1200 003642 001376          BNE    FT2D
1201 003644 020210          CMP    R2,(R0)  ;SEE IF EXPT=RCVD
1202 003646 001410          BEQ    FT2E      ;IF SO: BR
1203 003650 012767 015366 175070  MOV    #MSG27,ERADD ;SET ERROR CODE
1204 003656 012767 003624 175106  MOV    #FT2C,SCOLP ;SET SCOPE ADDRESS
1205 003664 004767 000012          JSR    PC,FT2ER  ;GO DO ERROR
1206 003670 005701          FT2E: TST    R1      ;SEE IF DONE RESET
1207 003672 001454          BEQ    FT2X      ;IF SO: BR
1208 003674 005001          CLR    R1      ;SET ZERO PATTERN
1209 003676 000167 177622          JMP    FT2A     ;DO ZERO BITS
1210 003702 000240          FT2ER: NOP
1211 003704 032777 020000 174726  BIT    @20000,@SWR ;SEE IF PRINT ERROR
1212 003712 001034          BNE    FT2ERB   ;IF NOT: BR
1213 003714 005767 174764          TST    HDRFL    ;SEE ID DONE HEADER
1214 003720 001004          BNE    FT2ERA   ;IF SO: BR
1215 003722 016704 174760          MOV    EMADDR,R4
1216 003726 004767 007262          JSR    PC,TTOUT ;DO HEADER
1217 003732 012767 000001 174744  FT2ERA: MOV    @1,HDRFL  ;SET FLAG
1218 003740 016704 175002          MOV    ERADD,R4
1219 003744 004767 007244          JSR    PC,TTOUT ;PRINT ERROR CODE
1220 003750 012704 015316          MOV    #MSG22,R4
1221 003754 004767 007234          JSR    PC,TTOUT ;PRINT EXPT TAG
1222 003760 010103          MOV    R1,R3
1223 003762 004767 007360          JSR    PC,OCTPE ;PRINT EXPT
1224 003766 012704 015326          MOV    #MSG23,R4
1225 003772 004767 007216          JSR    PC,TTOUT ;PRINT RCVD TAG
1226 003776 011003          MOV    (R0),R3
1227 004000 004767 007342          JSR    PC,OCTPE ;PRINT RCVD

```

1228	004004	005777	174630	FT2ERB:	TST	@SWR		;SEE IF HALT ON ERROR
1229	004010	100001			BPL	FT2ERC		;IF NOT: BR
1230	004012	000000			HALT			
1231	004014	004767	006426	FT2ERC:	JSR	PC,SCOPE		;GO SEE IF SCOPE ON ERROR
1232	004020	000240			NOP			
1233	004022	000207			RTS	PC		;IF NO SCOPE: CONTINUE TEST
1234	004024	000240		FT2X:	NOP			
1235	004026	004767	006456		JSR	PC,ITER		;GO SEE IF ITERATIONS
1236	004032	000167	177100		JMP	TSCD2		;RETURN TO SCHEDULAR

```

1238
1239
1240 ;RM INITIALIZE TEST*****
1241 004036 012767 017037 174642 FT3: MOV #MSFT3,EMADDR ;SET TEST HEADER
1242 004044 012767 004036 174720 MOV #FT3,SCOLP
1243 004052 004767 006510 JSR PC,INIT1 ;GO INIT
1244 004056 052777 020000 174524 BIS #20000,BCS ;FORCE UPE =1
1245 004064 000240 NOP
1246 004066 004767 006474 JSR PC,INIT1 ;GO INIT
1247 004072 005777 174502 TST BC1 ;SEE IF SC IS RESET
1248 004076 100005 BPL FT3A ;IF SO: BR
1249 004100 012767 015424 174640 MOV #MSG29,ERADD ;SET ERROR CODE
1250 004106 004767 000060 JSR PC,FT3ER ;GO DO ERROR
1251 004112 032777 040000 174460 FT3A: BIT #40000,BC1 ;SEE IF TRE IS RESET
1252 004120 001405 BEQ FT3B ;IF SO: BR
1253 004122 012767 015453 174616 MOV #MSG30,ERADD ;SET ERROR CODE.
1254 004130 004767 000036 JSR PC,FT3ER ;GO DO ERROR
1255 004134 017701 174450 FT3B: MOV #CS,R1 ;GET CS2
1256 004140 042701 000307 BIC #307,R1 ;MARK IR/OR
1257 004144 005701 TST R1 ;SEE IF RESET
1258 004146 001405 BEQ FT3X ;IF SO: BR
1259 004150 012767 015503 174570 MOV #MSG31,ERADD ;SET ERROR CODE
1260 004156 004767 000010 JSR PC,FT3ER ;GO DO ERROR
1261 004162 004767 006322 FT3X: JSR PC,ITER ;GO SEE IF ITERATION
1262 004166 000167 176744 JMP TSCD2 ;RETURN TO SCHEDULAR
1263 004172 032777 020000 174440 FT3ER: BIT #20000,BSWR ;SEE IF PRINT ERROR
1264 004200 001015 RNE FT3ERB ;IF NOT: BR
1265 004202 005767 174476 TST HDRFL ;SEE IF DONE HEADER
1266 004206 001006 BNE FT3ERA ;IF SO: BR
1267 004210 016704 174472 MOV EMADDR,R4
1268 004214 004767 006774 JSR PC,TTOUT ;PRINT HEADER
1269 004220 005267 174460 INC HDRFL
1270 004224 016704 174516 FT3ERA: MOV ERADD,R4
1271 004230 004767 006760 JSR PC,TTOUT ;PRINT ERROR CODE
1272 004234 005777 174400 FT3ERB: TST #SWR ;SEE IF HALT ON ERROR
1273 004240 100001 BPL FT3ERC ;IF NOT: BR
1274 004242 000000 HALT
1275 004244 000240 FT3ERC: NOP
1276 004246 004767 006174 JSR PC,SCOPE ;GO SEE IF SCOPE
1277 004252 000207 RTS PC ;IF NOT: BR

```

```

1279
1280
1281
1282 004254 005767 174422 FT4: TST RH17F
1283 004260 001141 BNE FT5X ;IF RH70: BR
1284 004262 012767 017071 174416 MOV #MSFT4,EMADDR ;SET TEST TEST HEADER
1285 004270 012777 000040 174312 MOV #40,@CS ;INIT
1286 004276 017700 174320 MOV @DB,R0 ;READ DB
1287 004302 005777 174302 TST @CS ;SEE IF DLT IS SET
1288 004306 100013 BPL FT4ER ;IF NOT: BR
1289 004310 005777 174264 TST @C1 ;SEE IF SC IS SET
1290 004314 100014 BPL FT4ERA ;IF NOT: BR
1291 004316 032777 040000 174254 BIT #40000,@C1 ;SEE IF TRE IS SET
1292 004324 001414 BEQ FT4ERB ;IF NOT: BR
1293 004326 004767 006156 FT4X: JSR PC,ITER ;GO SEE IF ITERATION
1294 004332 000167 176600 JMP TSCD2 ;RETURN TO SCHEDULAR
1295 004336 012767 015533 174402 FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE
1296 004344 000407 BR FT4ERC
1297 004346 012767 015551 174372 FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE
1298 004354 000403 BR FT4ERC
1299 004356 012767 015566 174362 FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.
1300 004364 000240 FT4ERC: NOP
1301 004366 012767 004254 174376 MOV #FT4,SCOLP ;SET SCOPE ADDRESS
1302 004374 004767 177572 JSR PC,FT3ER ;GO PRINT ERROR
1303 004400 000752 BR FT4X

```

```

1305
1306
1307
1308 004402 005767 174274          FT5:  TST      RH17F          ;SEE IF RH70
1309 004406 001066                   BNE      FT5X          ;IF SO: BR
1310 004410 012767 017121 174270    MOV      @MSFT5,EMADDR ;SET TEST HEADER
1311 004416 012767 004424 174346    MOV      @FT5A,SCOLP   ;SET SCOPE ADDRESS
1312 004424 004767 006136          FT5A:  JSR      PC,INIT1  ;GO INIT
1313 004430 032777 000100 174152    BIT      @100,@CS      ;SEE IF IR IS SET
1314 004436 001005                   BNE      FT5B          ;IF SO: BR
1315 004440 012767 015604 174300    MOV      @MSG35,ERADD  ;SET ERROR CODE
1316 004446 004767 000122          JSR      PC,FT5ER      ;GO DO ERROR
1317 004452 032777 000200 174130    FT5B:  BIT      @200,@CS ;SEE IF OR IS RESET
1318 004460 001405                   BEQ      FT5C          ;IF SO: BR
1319 004462 012767 015631 174256    MOV      @MSG36,ERADD  ;SET ERROR CODE
1320 004470 004767 000100          JSR      PC,FT5ER      ;GO DO ERROR
1321 004474 012777 000000 174120    FT5C:  MOV      @0,@DB     ;LOAD ZERO INTO SILO
1322 004502 032777 000200 174100    BIT      @200,@CS      ;SEE THAT OR RESET
1323 004510 001405                   BEQ      FT5D          ;IF IT DOES: BR
1324 004512 012767 015660 174226    MOV      @MSG37,ERADD  ;SET ERROR CODE
1325 004520 004767 000050          JSR      PC,FT5ER      ;GO DO ERROR
1326 004524 012777 177777 174070    FT5D:  MOV      @-1,@DB     ;LOAD SILO WITH -1
1327 004532 012700 004000          MOV      @4000,R0
1328 004536 032777 000200 174044    FT5E:  BIT      @200,@CS ;SEE IF OR IS SET
1329 004544 001007                   BNE      FT5X          ;IF SO: BR
1330 004546 005300                   DEC      R0
1331 004550 001372                   BNE      FT5E          ;AWAIT OR
1332 004552 012767 015660 174166    MOV      @MSG37,ERADD  ;SET ERROR CODE
1333 004560 004767 000010          JSR      PC,FT5ER      ;GO DO ERROR
1334 004564 004767 005720          FT5X:  JSR      PC,ITER   ;GO SEE IF ITERATION
1335 004570 000167 176342          JMP      TSCD2         ;RETURN TO SCHEDULAR
1336 004574 004767 177372          FT5ER: JSR      PC,FT3ER   ;GO PRINT ERROR
1337 004600 000240                   NOP
1338 004602 000207                   RTS      PC            ;CONTINUE TEST

```

```

1340
1341
1342
1343 004604 005767 174072          FT6:  TST      RH17F
1344 004610 001052                BNE      FT6X          ;IF RH70: BR
1345 004612 012767 017151 174066  MOV      #MSFT6,EMADDR ;SET TEST HEADER
1346 004620 012767 004626 174144  MOV      #FT6A,SCOLP   ;SET SCOPE ADDRESS
1347 004626 004767 005734          FT6A:  JSR      PC,INIT1 ;GO INIT
1348 004632 005000                CLR      R0           ;PRESET DATA
1349 004634 010077 173762          FT6B:  MOV      R0,&DB   ;LOAD SILO
1350 004640 005200                INC      R0           ;BUMP DATA
1351 004642 022700 000102          CMP      #102,R0      ;SEE IF FILLED ALL
1352 004646 001372                BNE      FT6B        ;IF NOT: BR
1353 004650 032777 000100 173732  BIT      #100,&CS     ;SEE IF IR IS RESET.
1354 004656 001405                BEQ      FT6C        ;IF SO: BR
1355 004660 012767 015771 174060  MOV      #MSG40,ERADD ;SET ERROR CODE
1356 004666 004767 000054          JSR      PC,FT6ER    ;GO DO ERROR
1357 004672 032777 000200 173710  FT6C:  BIT      #200,&CS ;SEE IF OR IS SET
1358 004700 001005                BNE      FT6D        ;IF SO: BR
1359 004702 012767 015717 174036  MOV      #MSG38,ERADD ;SET ERROR CODE
1360 004710 004767 000032          JSR      PC,FT6ER    ;GO DO ERROR
1361 004714 005000                FT6D:  CLR      R0           ;PRESET DATA
1362 004716 017701 173700          FT6E:  MOV      &DB,R1   ;READ SILO
1363 004722 020001                CMP      R0,R1       ;SEE IF EXPT-RCVD
1364 004724 001014                BNE      FT6DE       ;IF NOT: BR
1365 004726 005200                INC      R0           ;BUMP DATA
1366 004730 022700 000102          CMP      #102,R0     ;SEE IF DONE ALL
1367 004734 001370                BNE      FT6E        ;IF NOT: BR
1368 004736 004767 005546          FT6X:  JSR      PC,ITER ;GO SEE IF ITERATION
1369 004742 000167 176170          JMP      TSCD2       ;RETURN TO SCHEDULAR
1370 004746 000240                FT6ER:  NOP
1371 004750 004767 177216          JSR      PC,FT3ER    ;GO PRINT ERROR
1372 004754 000207                RTS      PC           ;RETURN
1373 004756 000240                FT6DE:  NOP
1374 004760 032777 020000 173652  BIT      #20000,&SWR  ;SEE IF PRINT ERROR
1375 004766 001032                BNE      FT6DEB      ;IF NOT: BR
1376 004770 005767 173710          TST      HDRFL       ;SEE IF DONE HEADER
1377 004774 016701 173706          MOV      EMADDR,R1
1378 005000 004767 006210          JSR      PC,TTOUT    ;PRINT HEADER
1379 005004 005267 173674          INC      HDRFL       ;SET FLAG
1380 005010 012704 015751          FT6DEA: MOV      #MSG39,R4
1381 005014 004767 006174          JSR      PC,TTOUT    ;PRINT SILO READ ERROR
1382 005020 012704 015316          MOV      #MSG22,R4
1383 005024 004767 006164          JSR      PC,TTOUT    ;PRINT EXPT TAG
1384 005030 010003                MOV      R0,R3
1385 005032 004767 006322          JSR      PC,OCTP     ;PRINT EXPT
1386 005036 012704 015326          MOV      #MSG23,R4
1387 005042 004767 006146          JSR      PC,TTOUT    ;PRINT RCVD TAG
1388 005046 010103                MOV      R1,&3
1389 005050 004767 006304          JSR      PC,UCTP     ;PRINT RCVD
1390 005054 005777 173560          FT6DEB: TST      &SWR ;SEE IF HALT ON ERROR
1391 005060 100001                BPL      FT6DEX      ;IF NOT: BR
1392 005062 000000                HALT
1393 005064 004767 006720          FT6DEX: JSR      PC,CKSWR ;CHECK FOR CNTL G
1394 005070 000207                RTS      PC           ;RETURN TO TEST

```

```

1396
1397
1398
1399 005072 005767 173604 FT7: TST RH17F
1400 005076 001021 BNE FT7X ;IF RH70: BR
1401 005100 012767 017201 173600 MOV #MSFT7,EMADDR ;SET TEST HEADER
1402 005106 012767 005072 173656 MOV #FT7,SCOLP ;SET SCOPE ADDRESS
1403 005114 004767 005446 JSR PC,INIT1 ;GO INIT
1404 005120 012700 000103 MOV #103,R0 ;SET SIZE OF SILO +1
1405 005124 010077 173472 FT7A: MOV R0,&DB ;LOAD SILO
1406 005130 005300 DEC R0 ;SEE IF DONE
1407 005132 001374 BNE FT7A ;IF NOT: BR
1408 005134 005777 173450 TST @CS ;SEE IF DLT IS SET
1409 005140 100004 BPL FT7ER ;IF NOT: BR
1410 005142 004767 005342 FT7X: JSR PC,ITER ;GO SEE IF ITERATION
1411 005146 000167 175764 JMP TSCD2 ;RETURN TO SCHEDULAR
1412 005152 012767 015533 173566 FT7ER: MOV #MSG32,ERADD ;SET ERROR CODE
1413 005160 004767 177006 JSR PC,FT3ER ;GO DO ERROR
1414 005164 000766 BR FT7X

```

```

1416
1417
1418
1419 005166 005767 173510 FT10: TST RH17F
1420 005172 001034 BNE FT10X ;IF RH70: BR
1421 005174 012767 017231 173504 MOV #MSFT10,EMADDR ;SET TEST HEADER
1422 005202 012767 005166 173562 MOV #FT10,SCOLP ;SET SCOPE ADDRESS
1423 005210 012777 000040 173372 MOV #40,@CS ;INITIALIZE
1424 005216 012700 000004 MOV #4,R0 ;SET NUMBER OF SILO WRITER
1425 005222 010077 173374 FT10A: MOV R0,@DB ;WRITE SILO
1426 005226 005300 DEC R0 ;SEE IF DONE
1427 005230 001374 BNE FT10A ;IF NOT: BR
1428 005232 052777 000040 173350 BIS #40,@CS ;INITIALIZE
1429 005240 012777 177777 173354 MOV #-1,@DB ;WRITE SILO
1430 005246 017701 173350 MOV @DB,R1 ;READ SILO 1
1431 005252 017701 173344 MOV @DB,R1 ;READ SILO 2
1432 005256 005777 173326 TST @CS ;SEE IF DLT IS SET
1433 005262 100011 BPL FT10ER ;IF NOT: BR
1434 005264 004767 005220 FT10X: JSR PC,ITER ;GO SEE IF ITERATION
1435 005270 005767 173530 TST RHOF ;SEE IF RH11 ONLY
1436 005274 001402 BEQ FT10XX ;IF NOT: BR
1437 005276 000167 175754 JMP TEND ;ELSE GO TO END
1438 005302 000167 175630 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR
1439 005306 012767 015533 173432 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE
1440 005314 004767 176652 JSR PC,FT3ER ;GO DO ERROR
1441 005320 000761 BR FT10X

```



```

1531                                     ;SPACE TEST*****
1532 006104 000240                       FT14:  NOP
1533 006106 012767 017354 172572        MOV     #MSFT14,EMADDR ;SET TEST HEADER
1534 006114 012767 001700 172672        MOV     #1700,UDES     ;SET NRZ,NORMAL
1535 006122 004767 003400                 FT14A1: JSR     PC,RWIND  ;GO INITIALIZE
1536 006126 012767 000100 172566        MOV     #100,RCNT     ;SET NUMBER OF RECORDER
1537 006134 012767 177777 011600        MOV     #-1,WDATA    ;SET DATA PATTERN
1538 006142 012767 177700 172546        MOV     #-100,FCNT   ;PRESET FRAME CNT
1539 006150 012767 177740 172542        MOV     #-40,WCNT   ;PRESET WORD CNT
1540 006156 004767 004404                 FT14A:  JSR     PC,INIT1 ;GO REWIND
1541 006162 012767 001000 172546        MOV     #1000,OPDYX
1542 006170 012767 040000 172536        MOV     #40000,RDYDX
1543 006176 012767 000061 172602        MOV     #61,FUN      ;SET WRITE OP-CODE
1544 006204 012767 102300 172544        MOV     #102300,STMSK ;MASK DATA RELATED ERRORS
1545 006212 052777 000010 172370        BIS     #10,BCS      ;INHIBIT BUS ADDRESS INCREMENT
1546 006220 004767 003150                 JSR     PC,EXEC      ;GO EXECUTE COMMAND
1547 006224 000240                       NOP
1548 006226 012767 016205 172470        MOV     #MSG46,ERRP  ;SET ERROR CODE
1549 006234 004767 003364                 JSR     PC,ERCHK    ;GO CHECK ERRORS
1550 006240 005767 172544                 TST     SERFL       ;SEE IF ERROR
1551 006244 001402                       BEQ     FT14A2      ;IF NOT: BR
1552 006246 000167 000466                 JMP     FT14X       ;ELSE EXIT
1553 006252 162767 000001 172436        FT14A2: SUB     #1,FCNT   ;BUMP FC
1554 006260 032767 000001 172430        BIT     #1,FCNT     ;SEE IF SHOULD BUMP WC
1555 006266 001403                       BEQ     FT14A3      ;IF NOT: BR
1556 006270 162767 000001 172422        SUB     #1,WCNT     ;BUMP WC
1557 006276 005367 172420                 FT14A3: DEC     RCNT  ;SEE IF DONE ALL
1558 006302 001325                       BNE     FT14A      ;WRITE ALL RECORDS
1559 006304 000240                       NOP
1560 006306 012767 000100 172414        MOV     #100,RRD    ;PRESET RECORD POSITION
1561 006314 012767 000176 172410        MOV     #176,RFD
1562 006322 000240                       NOP
1563 006324 012767 177701 172406        MOV     #-77,SCNT   ;SET SPACE AMOUNT
1564 006332 012767 000033 172446        FT14B:  MOV     #33,FUN  ;SET OP-CODE SPACE REVERSE
1565 006340 004767 003030                 JSR     PC,EXEC     ;GO EXECUTE COMMAND
1566 006344 012767 016256 172352        MOV     #MSG48,ERRP ;SET ERROR CODE
1567 006352 004767 003246                 JSR     PC,ERCHK    ;GO CHECK ERRORS
1568 006356 005767 172426                 TST     SERFL       ;SEE IF ERROR
1569 006362 001166                       BNE     FT14X       ;IF SO: BR
1570 006364 004767 000070                 JSR     PC,FT14RR   ;GO READ REVERSE + CHECK DATA
1571 006370 000240                       NOP
1572 006372 012767 000031 172406        MOV     #31,FUN     ;SET SPACE FORWARD OP-CODE
1573 006400 005267 172334                 INC     SCNT        ;SET SPACE AMOUNT
1574 006404 001555                       BEQ     FT14X       ;IF DONE: BR
1575 006406 004767 002762                 JSR     PC,EXEC     ;GO EXECUTE COMMAND
1576 006412 012767 016231 172304        MOV     #MSG47,ERRP ;SET ERROR CODE
1577 006420 004767 003200                 JSR     PC,ERCHK    ;GO CHECK ERROR
1578 006424 005767 172360                 TST     SERFL       ;SEE IF ERROR FLAG
1579 006430 001143                       BNE     FT14X       ;IF NO: BR
1580 006432 004767 000064                 JSR     PC,FT14RF   ;GO READ FORWARD FOR POSITION CHECK
1581 006436 000240                       NOP
1582 006440 005267 172274                 INC     SCNT        ;DECREMENT SPACE AMOUNT
1583 006444 001535                       BEQ     FT14X       ;IF DONE: BR
1584 006446 005267 172256                 INC     RRD         ;BUMP DATA EXPT
1585 006452 005367 172254                 DEC     RFD         ;BUMP DATA EXPT
1586 006456 000725                       BR      FT14B

```

1587	006460	000240			FT14RR:	NOP			
1588	006462	012767	021454	172224		MOV	#RDATA,BADDR		;SET BA
1589	006470	012767	000077	172310		MOV	#77,FUN		;SET READ REVERSE OP-CODE
1590	006476	004767	002672			JSR	PC,EXEC		;GO EXECUTE COMMAND
1591	006502	000240				NOP			
1592	006504	016705	172220			MOV	RRD,R5		
1593	006510	020577	172072			CMP	R5,#FC		;SEE IF CORPECT RECORD
1594	006514	001020				BNE	FT14RER		;IF NOT: BR
1595	006516	000167	000026			JMP	FT14EC		;GO CLEAR RM11 ERROR BIT
1596	006522	000240			FT14RF:	NOP			
1597	006524	012767	000071	172254		MOV	#71,FUN		;SET READ FORWARD OP-CODE
1598	006532	004767	002636			JSR	PC,EXEC		;GO EXECUTE COMMAND
1599	006536	016705	172170			MOV	RFD,R5		
1600	006542	020577	172040			CMP	R5,#FC		;SEE IF CORRECT RECORD
1601	006546	001003				BNE	FT14RER		;IF NOT: BR
1602	006550	004767	004012		FT14EC:	JSR	PC,INIT1		;CLEAR RM
1603	006554	000207				RTS	PC		;RETURN
1604	006556	000240			FT14RER:	NOP			
1605	006560	032777	020000	172052		BIT	#20000,#SWR		;SEE IF PRINT INHIBITED
1606	006566	001060				BNE	FT14R3		;IF SO: BR
1607	006570	012704	017354			MOV	#MSFT14,R4		
1608	006574	004767	004414			JSR	PC,TTOUT		;PRINT HEADER
1609	006600	012704	015050			MOV	#MSG9,R4		
1610	006604	004767	004404			JSR	PC,TTOUT		;PRINT ERROR TYPE
1611	006610	012704	015303			MOV	#MSG20,R4		;SET NRZ TAG POINTER
1612	006614	032767	002000	172172		BIT	#2000,UDES		;SEE IF PE
1613	006622	001402				BEQ	FT14R0		;IF NOT: BR
1614	006624	012704	015311			MOV	#MSG21,R4		;ELSE SET PE TAG POINTER
1615	006630	004767	004360		FT14R0:	JSR	PC,TTOUT		;PRINT TAG
1616	006634	032767	000002	172144		BIT	#2,FUN		;SEE IF READ REVERSE
1617	006642	001003				BNE	FT14R1		;IF SO: BR
1618	006644	012704	015263			MOV	#MSG17,R4		
1619	006650	000402				BR	FT14R2		;GO PRINT
1620	006652	012704	015243		FT14R1:	MOV	#MSG16,R4		
1621	006656	004767	004332		FT14R2:	JSR	PC,TTOUT		;PRINT FRWD/REV
1622	006662	012704	015316			MOV	#MSG22,R4		
1623	006666	004767	004322			JSR	PC,TTOUT		;PRINT EXPT TAG
1624	006672	010503				MOV	R5,R3		
1625	006674	042703	177700			BIC	#177700,R3		;MASK RECORD NUMBER
1626	006700	004767	004454			JSR	PC,OCTP		;PRINT EXPT RECORD NUMBER
1627	006704	012704	015326			MOV	#MSG23,R4		
1628	006710	004767	004300			JSR	PC,TTOUT		;PRINT RCVD TAG
1629	006714	017703	171666			MOV	#FC,R3		
1630	006720	042703	177700			BIC	#177700,R3		;MASK RECORD NUMBER
1631	006724	004767	004430			JSR	PC,OCTP		;PRINT ACTUAL RECORD NUMBER
1632	006730	005777	171704		FT14R3:	TST	#SWR		;SEE IF HALT ON ERROR
1633	006734	100001				BPL	FT14X		;IF NOT: BR
1634	006736	000000				HALT			
1635	006740	004767	005044		FT14X:	JSR	PC,CKSWR		;CHECK FOR CNTL G
1636	006744	005767	172052			TST	NRZOF		;SEE IF NRZ ONLY
1637	006750	001011				BNE	FT14XX		;IF SO: BR
1638	006752	032767	002000	172034		BIT	#2000,UDES		;SEE IF DONE PE
1639	006760	001005				BNE	FT14XX		;IF SO: BR
1640	006762	012767	002300	172024		MOV	#2300,UDES		;SET TO PE
1641	006770	000167	177126			JMP	FT14A1		;DO IN PE
1642	006774	000167	174136		FT14XX:	JMP	TSCD2		;RETURN TO SCHEDULAR

```

1644                                     ;ERASE TEST*****
1645
1646 007000 000240                      FT15:  NOP
1647 007002 005067 171750                CLR      STMSK
1648 007006 012767 000100 171720         MOV      @100,RDYDX
1649 007014 012767 000010 171714         MOV      @10,OPDYX
1650 007022 012767 017376 171656         MOV      @MSFT15,EMADDR ;SET TEST HFADER
1651 007030 004767 002472                JSR      PC,RWND ;REWIND
1652 007034 012767 021454 171652         MOV      @RDATA,BADDR ;SET BA
1653 007042 012767 001700 171744         MOV      @1700,UDES ;SET NRZ, NORMAL
1654 007050 012767 000025 171730 FT15A: MOV      @25,FUN ;SET ERASE OP-CODE
1655 007056 012767 000200 171636 FT15A: MOV      @200,RCNT ;SET TO ERASE 128 TIMES
1656 007064 004767 002304                FT15B: JSR      PC,EXEC ;GO EXECUTE COMMAND
1657 007070 012767 016205 171626         MOV      @MSG46,ERRP ;SET ERROR CODE
1658 007076 004767 002522                JSR      PC,ERCHK ;GO CHECK ERRORS
1659 007102 005767 171702                TST      SERFL ;SEE IF ANY ERRORS
1660 007106 001032                        BNE      FT15X ;IF SO EXIT
1661 007110 005367 171606                DEC      RCNT ;SEE IF DONE ERASING
1662 007114 001363                        BNE      FT15B ;IF NOT: BR
1663 007116 000240                        NOP
1664 007120 004767 002402                JSR      PC,RWND ;REWIND
1665 007124 012767 177600 171566         MOV      @-200,WCNT ;SET WC
1666 007132 012767 000071 171646         MOV      @71,FUN ;SET READ FORWARD OP-CODE
1667 007140 012767 000040 171566         MOV      @40,RDYDX ;SET DELAY
1668 007146 004767 002222                JSR      PC,EXEC ;GO EXECUTE COMMAND
1669 007152 000240                        NOP
1670 007154 012767 016601 171542         MOV      @MSG60,ERRP ;SET ERROR CODE
1671 007162 012767 020000 171566         MOV      @20000,STMSK
1672 007170 004767 002430                JSR      PC,ERCHK ;GO CHECK ERRORS
1673 007174 000167 173736                FT15X: JMP      TSCD2 ;RETURN TO SCHEDULAR

```

```

1675                                     ;TAPE MARK WRITE/READ TEST*****
1676
1677 007200 000240                      FT16:  NOP
1678 007202 012767 000001 171524      MOV     @1,RDYDX
1679 007210 012767 001000 171520      MOV     @1000,OPDYX
1680 007216 012767 017420 171462      MOV     @MSFT16,EMADDR ;SET HEADER
1681 007224 012767 001700 171562      MOV     @1700,UDES ;SET TO NRZ NORMAL,ODD
1682 007232 004767 002270              FT16A: JSR     PC,RWIND ;REWIND
1683 007236 012767 177760 171452      FT16B: MOV     @-20,FCNT ;FC-20
1684 007244 012767 177770 171446      MOV     @-10,WCNT ;WC-10
1685 007252 012767 000027 171526      MOV     @27,FUN ;SET WRITE TAPE MARK OP-CODE
1686 007260 004767 002110              JSR     PC,EXEC ;GO EXECUTE COMMAND
1687 007264 012767 001000 171464      MOV     @1000,STMSK ;SET FOR FCE MASK
1688 007272 012767 015132 171424      MOV     @MSG12,ERRP ;SET ERROR CODE
1689 007300 004767 002320              JSR     PC,ERCHK ;GO CHECK ERROR
1690 007304 004767 002722              JSR     PC,TMCHK ;GO SEE IF TM SET
1691 007310 012767 000077 171470      MOV     @77,FUN ;SET USED REVERSE OP-CODE
1692 007316 004767 002052              JSR     PC,EXEC ;GO EXECUTE COMMAND
1693 007322 012767 001000 171426      MOV     @1000,STMSK ;SET FCE ERROR MASK
1694 007330 012767 015150 171366      MOV     @MSG13,ERRP ;SET ERROR CODE
1695 007336 004767 002262              JSR     PC,ERCHK ;GO CHECK ERRORS
1696 007342 004767 002664              JSR     PC,TMCHK ;GO SEE IF TM SET
1697 007346 012767 000071 171432      MOV     @71,FUN ;SET READ FORWARD OP-CODE
1698 007354 004767 002014              JSR     PC,EXEC ;GO EXECUTE COMMAND
1699 007360 012767 015175 171336      MOV     @MSG14,ERRP ;SET ERROR CODE
1700 007366 004767 002232              JSR     PC,ERCHK ;TO CHECK ERRORS
1701 007372 004767 002634              JSR     PC,TMCHK ;GO SEE IF TM SET
1702 007376 032767 002000 171410      BIT     @2000,UDES ;SEE IF DONE PE
1703 007404 001012                      BNE     FT16X ;IF SO: BR
1704 007406 005767 171410              TST     NRZOF ;SEE IF NRZ ONLY
1705 007412 001007                      BNE     FT16X ;IF SO: BR
1706 007414 012767 002300 171372      MOV     @2300,UDES ;SET PE, NORMAL
1707 007422 004767 003140              JSR     PC,INIT1 ;INITIALIZE
1708 007426 000167 177604              JMP     FT16B ;DO IN PE
1709 007432 004767 003052              FT16X: JSR     PC,ITER ;DO ITERATIONS
1710 007436 000167 173474              JMP     TSCD2 ;RETURN TO SCHEDULAR
1711

```

```

1713
1714
1715
1716 007442 005067 171254 FT17: CLR RCNT
1717 007446 012767 017461 171232 MOV #MSFT17,EMADDR ;SET HEADER
1718 007454 012767 001700 171332 MOV #1700,UDES ;SET TO NRZ
1719 007462 004767 002040 FT17A: JSR PC,RWIND ;REWIND TAPF
1720 007466 012767 000027 171312 FT17B: MOV #27,FUN
1721 007474 012767 040000 171232 MOV #40000,RDYDX ;SET DRY DELAY
1722 007502 012767 040000 171226 MOV #40000,OPDYX ;SET OP DELAY
1723 007510 004767 001660 JSR PC,EXEC ;GO WRITE TM
1724 007514 012767 102300 171234 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1725 007522 012767 015222 171174 MOV #MSG15,ERRP ;SET ERROR TYPE
1726 007530 004767 002070 JSR PC,ERCHK ;GO CHECK ERROR
1727 007534 005767 171250 TST SERFL ;SEE IF ERROR
1728 007540 001144 BNE FT17X ;IF SO: BR
1729 007542 004767 002464 JSR PC,TMCHK ;GO SEE IF TM SET
1730 007546 000240 NOP
1731 007550 000240 NOP
1732 007552 032767 000100 171142 BIT #100,RCNT ;SEE IF DONE PATTERN
1733 007560 001046 BNE FT17D ;IF SO: BR
1734 007562 062767 000020 171132 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1735 007570 016767 171126 171152 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1736 007576 012767 177600 171114 MOV #-200,WCNT ;WC=128
1737 007604 012767 177400 171104 MOV #-400,FCNT ;FC=256
1738 007612 012767 017742 171074 MOV #MDATA,BADDR ;BA=WRITE BUFFER
1739 007620 012767 000061 171160 MOV #61,FUN ;SET WRITE OP CODE
1740 007626 000240 FT17C: NOP
1741 007630 000240 NOP
1742 007632 004767 001536 JSR PC,EXEC ;GO WRITE
1743 007636 012767 015132 171060 MOV #MSG12,ERRP ;SET ERROR CODE
1744 007644 012767 102300 171104 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1745 007652 004767 001746 JSR PC,ERCHK ;GO CHECK ERROR
1746 007656 005767 171126 TST SERFL ;SEE IF ERROR
1747 007662 001073 BNE FT17X ;IF SO: BR
1748 007664 005367 171060 DEC TEMP1 ;SEE IF DONE ALL
1749 007670 001356 BNE FT17C ;IF NOT: BR
1750 007672 000167 177570 JMP FT17B ;ELSE GO DO TM
1751 007676 000240 FT17D: NOP
1752 007700 012767 000033 171100 MOV #33,FUN ;SET SPACE REVERSE
1753 007706 012767 015243 171010 MOV #MSG16,ERRP ;SET ERROR CODE
1754 007714 012767 177600 171016 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1755 007722 012767 000005 170772 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1756 007730 004767 002632 FT17E: JSR PC,INIT1 ;GO INIT
1757 007734 004767 001434 JSR PC,EXEC ;GO SPACE
1758 007740 012767 001000 171010 MOV #1000,STMSK ;SET ERROR MASK
1759 007746 004767 001652 JSR PC,ERCHK ;GO CHECK ERROR
1760 007752 005767 171032 TST SERFL ;SEE IF ERROR
1761 007756 001035 BNE FT17X ;IF SO: BR
1762 007760 004767 002246 JSR PC,TMCHK ;GO SEE IF TM SET
1763 007764 005367 170732 DEC RCNT ;SEE IF DONE SPACES
1764 007770 001357 BNE FT17E ;IF NOT: BR
1765 007772 022767 000031 171006 CMP #31,FUN ;SEE IF DONE FORWARD
1766 010000 001410 BEQ FT17F ;IF SO: BR
1767 010002 012767 015263 170714 MOV #MSG17,ERRP ;SET ERROR CODE
1768 010010 012767 000031 170770 MOV #31,FUN ;SET TO SPACE FORWARD

```

1769	010016	000167	177672			JMP	FT17D1		:DO FORWARD
1770	010022	032767	002000	170764	FT17F:	BIT	#2000, UDES		:SEE IF DONE PE
1771	010030	001010				BNE	FT17X		:IF SO: BR
1772	010032	005767	170764			TST	NRZOF		:SEE IF NRZ ONLT
1773	010036	001005				BNE	FT17X		:IF SO: BR
1774	010040	012767	002300	170746		MOV	#2300, UDES		:SET TO PE
1775	010046	000167	177410			JMP	FT17A		:GO PE
1776	010052	000167	173060		FT17X:	JMP	TSCD2		:RETURN TO SCHEDULAR

```

1778
1779
1780
1781 010056 000240
1782 010060 012767 017507 170620
1783 010066 004767 001434
1784 010072 012767 000003 170716
1785 010100 004767 002166
1786 010104 012767 017742 170602
1787 010112 012767 177400 170576
1788 010120 012767 177600 170572
1789 010126 012767 001700 170660
1790 010134 012767 000061 170644
1791 010142 004767 001226
1792 010146 012767 016205 170550
1793 010154 004767 001444
1794 010160 005767 170624
1795 010164 001050
1796 010166 012767 015243 170530
1797 010174 012767 000057 170604
1798 010202 062767 000376 170504
1799 010210 004767 001160
1800 010214 004767 001404
1801 010220 012767 015263 170476
1802 010226 012767 000051 170552
1803 010234 162767 000376 170452
1804 010242 004767 001126
1805 010246 004767 001352
1806 010252 032767 002000 170534
1807 010260 001012
1808 010262 005767 170534
1809 010266 001007
1810 010270 012767 002300 170516
1811 010276 004767 002264
1812 010302 000167 177626
1813 010306 004767 002176
1814 010312 000167 172620

```

;WRITE CHECK TEST*****

```

FT20:  NOP
      MOV #MSFT20,EMADDR ;SET HEADER
      JSR PC,RWIND ;REWIND
      MOV #3,PATRN
      JSR PC,DSUP ;GO SET PATTERN 3
      MOV #WDATA,BADDR ;SET BA
      MOV #-400,FCNT ;SET FC
      MOV #-200,WCNT ;SET WC
      MOV #1700,UDES ;SET NRZ NORMAL
FT20A: MOV #61,FUN ;SET WRITE OP CODE
      JSR PC,EXEC ;GO WRITE RECORD
      MOV #MSG46,ERRP ;SET ERROR CODE
      JSR PC,ERCHK ;GO CHECK ERROR
      TST SERFL ;SEE IF ERORR
      BNE FT20X ;IF SO: BR
      MOV #MSG16,ERRP ;SET REVERSE ERROR TAG
      MOV #57,FUN ;SET REVERSE WRITE CHECK OP-CODE
      ADD #376,BADDR ;SET BA FOR REVERSE CHECK
      JSR PC,EXEC ;GO DO REVERSE CHECK
      JSR PC,ERCHK ;GO CHECK ERROR
FT20B: MOV #MSG17,ERRP ;SET FORWARD TAG
      MOV #51,FUN ;SET FORWARD CHECK OP CODE
      SUB #376,BADDR ;SET BA FOR FORWARD CHECK
      JSR PC,EXEC ;GO DO FORWARD CHECK
      JSR PC,ERCHK ;GO CHECK ERROR
FT20C: BIT #2000,UDES ;SEE IF DONE PE
      BNE FT20X ;IF SO: BR
      TST NRZOF ;SEE IF NRZ ONLY
      BNE FT20X ;IF SO: BR
      MOV #2300,UDES ;ELSE SET PE
      JSR PC,INIT1 ;GO INIT
      JMP FT20A ;DO IN PE
FT20X: JSR PC,ITER ;DO ITERATIONS
      JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

1816
1817
1818
1819 010316 012767 017540 170362 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1820 010324 004767 001176 JSR PC,RWIND ;GO REWIND
1821 010330 012767 000003 170460 MOV #3,PATRN
1822 010336 004767 001730 JSR PC,DSUP ;GO SET PATTERN 3
1823 010342 012767 017742 170344 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1824 010350 012767 176340 170340 MOV #-1440,FCNT ;SET FC=800
1825 010356 012767 177160 170334 MOV #-620,WCNT ;SET WC=400
1826 010364 012767 001700 170422 MOV #1700,UDES ;SET NRZ, NORMAL
1827 010372 012767 000061 170406 MOV #61,FUN ;SET WRITE OP-CODE
1828 010400 004767 000770 JSR PC,EXEC ;GO DO WRITE 1
1829 010404 012767 015132 170312 MOV #MSG12,ERRP ;SET ERROR CODE
1830 010412 004767 001206 JSR PC,ERCHK ;GO CHECK FOR ERROR
1831 010416 004767 000752 JSR PC,EXEC ;YES DO WRITE 2
1832 010422 004767 001176 JSR PC,ERCHK ;YES CHECK FOR ERROR
1833 010426 000240 NOP
1834 010430 004767 001072 JSR PC,RWIND ;GO REWIND
1835 010434 012767 177160 170254 MOV #-620,FCNT ;SET FC=400
1836 010442 012767 177470 170250 MOV #-310,WCNT ;SET WC=200
1837 010450 004767 000720 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1838 010454 000240
1839 010456 004767 001044 FT21A: NOP
1840 010462 012767 021454 170224 JSR PC,RWIND ;REWIND
1841 010470 012767 177160 170220 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1842 010476 012767 177470 170214 MOV #-620,FCNT ;SET FC=400
1843 010504 012767 000071 170274 MOV #-310,WCNT ;SET WC=200
1844 010512 004767 000656 JSR PC,EXEC ;SET READ OP-CODE
1845 010516 012767 015175 170200 MOV #MSG14,ERRP ;GO READ RECORD 1
1846 010524 004767 001074 JSR PC,ERCHK ;SET ERROR CODE
1847 010530 000240 NOP ;GO CHECK FOR ERROR
1848 010532 052777 000010 170050 BIS #10,BCS ;INHIBIT BA INCREMENT
1849 010540 012767 176340 170150 MOV #-1440,FCNT ;SET FC=800
1850 010546 012767 177160 170144 MOV #-620,WCNT ;SET WC=400
1851 010554 004767 000614 JSR PC,EXEC ;GO READ RECORD 2
1852 010560 022777 001440 170020 CMP #1440,SFC ;SEE IF READ RECORD 2
1853 010566 001423 BEQ FT21X ;IF SO: BR
1854 010570 022777 001441 170010 CMP #1441,SFC ;++F CHECK FOR 801 FRAMES
1855 010576 001417 BEQ FT21X ;++F IF SO: BR
1856 010600 012767 016152 170140 MOV #MSG45,ERADD ;++F SET ERROR CODE
1857 010606 022777 001440 167772 CMP #1440,SFC ;++F MORE THAN 801 FRAMES ?
1858 010614 101403 BLOS 1 ;++F IF SO: BR
1859 010616 012767 016103 170122 MOV #MSG44,ERADD ;++F SET ERROR CODE
1860 010624 012767 010454 170140 1$: MOV #FT21A,SCOLP ;SET SCOPE ADDRESS
1861 010632 004767 173334 JSR PC,FT3ER ;GO PRINT ERROR
1862 010636 004767 001646 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1863 010642 000167 172270 JMP TSCD2 ;RETURN TO SCHEDULAR
1864
1865

```

```

1867                                     ;BUFFERED COMMAND TEST*****
1868
1869 010646 012767 017567 170032 FT22:  MOV    #MSFT22,EMADDR ;SET TEST HEADER
1870 010654 004767 000646          JSR    PC,RWND      ;GO REWIND
1871 010660 012700 000003          MOV    #3,R0        ;SET NUMBER OF WRITES
1872 010664 012767 001700 170122  MOV    #1700,UDES   ;SET TO NRZ NORMAL
1873 010672 012767 017742 170014  MOV    #WDATA,BADDR ;SET BA=WRITE BUFFER
1874 010700 012767 177000 170010  MOV    #-1000,FCNT  ;SET FC=1000
1875 010706 012767 177400 170004  MOV    #-400,WCNT   ;SET WC=400
1876 010714 012767 000061 170064  MOV    #61,FUN      ;SET WRITE OP-CODE
1877 010722 004767 000446          FT22A: JSR    PC,EXEC     ;GO DO WRITE
1878 010726 005300          DEC    R0           ;SEE IF DONE ALL
1879 010730 001374          BNE    FT22A        ;IF NOT: BR
1880 010732 000240          NOP
1881 010734 012777 000007 167636  MOV    #7,SC1       ;START REWIND
1882 010742 032777 000200 167642  FT22B: BIT    #200,BDS
1883 010750 001774          BEQ    FT22B
1884 010752 004767 001610          JSR    PC,INIT1    ;INITIALIZE
1885 010756 012767 000010 167750  MOV    #10,RDYDX    ;SET LONG READY DELAY
1886 010764 004767 000404          JSR    PC,EXEC     ;ISSUE BUFFERED WRITE
1887 010770 000240          NOP
1888 010772 012767 016303 167724  MOV    #MSG49,ERRP  ;SET ERROR CODE
1889 011000 012767 102300 167750  MOV    #102300,STMSK ;MARK DATA ERROR
1890 011006 004767 000612          JSR    PC,ERCHK    ;GO CHECK ERROR
1891 011012 032777 000002 167572  BIT    #2,BDS       ;SEE IF BOT IS SET
1892 011020 001410          BEQ    FT22X        ;IF NOT: BR
1893 011022 012767 016331 167716  MOV    #MSG50,ERADD ;SET ERROR CODE
1894 011030 012767 010646 167734  MOV    #FT22,SCOLP
1895 011036 004767 173130          JSR    PC,FT3ER    ;GO DO ERROR
1896 011042 004767 001442          FT22X: JSR    PC,ITER  ;GO SEE IF ITERATION
1897 011046 000167 172064          JMP    TSCD2       ;RETURN TO SCHEDULAR
1898
1899

```

```

1901                                     ;READ-IN PRESET TEST*****
1902
1903 011052 005767 167634          FT23: TST      SLVN          ;SEE IF SLAVE SELECT=0
1904 011056 001103                    BNE      FT23X        ;IF NOT:BR
1905 011060 012767 017624 167620  MOV      @MSFT23,EMADDR ;SET TEST HEADER
1906 011066 004767 001474                    JSR      PC,INIT1     ;GO INIT
1907 011072 012767 001700 167714  MOV      @1700,UDES    ;SET TO NRZ NORMAL
1908 011100 012767 017742 167606  MOV      @WDATA,BADDR ;SET BA=WRITE BUFFER
1909 011106 012767 177400 167602  MOV      @-400,FCNT    ;SET FC=400
1910 011114 012767 177600 167576  MOV      @-200,WCNT    ;SET WC=200
1911 011122 012767 000061 167656  MOV      @61,FUN       ;SET WRITE OP-CODE
1912 011130 004767 000240                    JSR      PC,EXEC      ;GO DO WRITE
1913 011134 000240                    NOP
1914 011136 004767 001424                    JSR      PC,INIT1     ;INITIALIZE
1915 011142 012767 000021 167636  MOV      @21,FUN       ;SET READ-IN PRESET OP CODE
1916 011150 004767 000220                    JSR      PC,EXEC      ;GO DO COMMAND
1917 011154 005000                    CLR      R0
1918 011156 012703 000004                    MOV      @4,R3        ;SET MULT
1919 011162 032777 020000 167422  FT23A: BIT      @20000,@DS ;SEE IF PIP RESET
1920 011170 001404                    BEQ      FT23B        ;IF SO: BR
1921 011172 005300                    DEC      R0
1922 011174 001372                    BNE      FT23A        ;AWAIT PIP RESET
1923 011176 005303                    DEC      R3
1924 011200 001370                    BNE      FT23A        ;DELAY
1925 011202 032777 000002 167402  FT23B: BIT      @2,@DS  ;SEE IF BOT
1926 011210 001010                    BNE      FT23C        ;IF SO: BR
1927 011212 012767 016367 167526  MOV      @MSG51,ERADD ;SET ERROR CODE
1928 011220 012767 011052 167544  MOV      @FT23,SCOLP
1929 011226 004767 172740                    JSR      PC,FT3ER     ;GO DO ERROR
1930 011232 012701 141000          FT23C: MOV      @141000,R1 ;SET EXPT TC
1931 011236 016700 167370          MOV      TC,R0        ;SET TC ADDRESS
1932 011242 020110                    CMP      R1,(R0)      ;SEE IF EXPT=RCVD
1933 011244 001410                    BEQ      FT23X        ;IF SO: BR
1934 011246 012767 016423 167472  MOV      @MSG52,ERADD ;SET ERROR CODE
1935 011254 012767 011052 167510  MOV      @FT23,SCOLP ;CLEAR SCOPE ADDRESS
1936 011262 004767 172414                    JSR      PC,FT2ER     ;GO DO ERROR
1937 011266 000167 171644          FT23X: JMP      TSCD2   ;RETURN TO SCHEDULAR
1938
1939

```

```

1941                                     ;REWIND: OFF LINE TEST*****
1942
1943 011272 032777 000000 167340 FT24: BIT    #0,@SWR          ;SEE IF IN CONTINUOUS MODE
1944 011300 001433                BEQ    FT24XX          ;IF SO: BR
1945 011302 012767 017657 167376      MOV    #MSFT24,EMADDR ;SET TEST HEADER
1946 011310 004767 001252                JSR    PC,INIT1       ;GO INITIAIZE
1947 011314 012777 000003 167256      MOV    #3,@C1         ;ISSUE REWIND: OFF LINE COMMAND
1948 011322 012700 004000                MOV    #4000,R0
1949 011326 005300                FT24A: DEC    R0          ;DELAY
1950 011330 001376                BNE    FT24A
1951 011332 032777 010000 167252      BIT    #10000,@DS     ;SEE IF MOL IS RESET
1952 011340 001407                BEQ    FT24X          ;IF SO: BR
1953 011342 005067 167424                CLR    SCOLP          ;ASSURE NO SCOPE
1954 011346 012767 016442 167372      MOV    #MSG53,ERADD   ;SET ERROR CODE
1955 011354 004767 172612                JSR    PC,FT3ER       ;GO DC' ERROR
1956 011360 012704 016466                FT24X: MOV    #MSG54,R4
1957 011364 004767 001624                JSR    PC,TTOUT       ;PRINT ON LINE REQUEST
1958 011370 000167 171542                FT24XX: JMP    TSCD2   ;RETURN TO SCHEDULAR
1959
1960

```

```

1962                                     ;COMMAND EXECUTE SUBROUTINE*****
1963
1964 011374 000240                               EXEC:  NOP
1965 011376 056777 167412 167226                BIS      UDES,@TC          ;LOAD TAPE CONT
1966 011404 016777 167310 167170                MOV      WCNT,@WC         ;LOAD WC
1967 011412 016777 167300 167166                MOV      FCNT,@FC        ;LOAD FC
1968 011420 016777 167270 167156                MOV      BADDR,@BA      ;LOAD BA
1969 011426 022767 000031 167352                CMP      #31,FUN         ;SEE IF SPACE FORWARD
1970 011434 001404                               BEQ      EXECA           ;IF SO: BR
1971 011436 022767 000033 167342                CMP      #33,FUN         ;SEE IF SPACE REVERSE
1972 011444 001003                               BNE      EXECB           ;IF NOT: BR
1973 011446 016777 167266 167132                EXECA:  MOV      SCNT,@FC ;SET SPACE COUNT
1974 011454 000240                               EXECB:  NOP
1975 011456 016777 167324 167114                MOV      FUN,@C1         ;LOAD OP-CODE + GO
1976 011464 000240                               NOP
1977 011466 016703 167242                MOV      RDYDX,R3        ;SET DELAY
1978 011472 005004                               CLR      R4
1979 011474 032777 000200 167110                EXECB:  BIT      #200,@DS ;SEE IF DRY
1980 011502 001004                               BNE      EXECX           ;IF SO: BR
1981 011504 005304                               DEC      R4
1982 011506 001372                               BNE      EXECB
1983 011510 005303                               DEC      R3              ;DELAY FOR DRY
1984 011512 001370                               BNE      EXECB
1985 011514 016703 167216                EXECX:  MOV      OPDYX,R3
1986 011520 005303                EXECXA: DEC      R3              ;DELAY
1987 011522 001376                BNE      EXECXA
1988 011524 000207                EXECXX: RTS      PC          ;RETURN TO CALLER
1989

```

```

1991                                     ;REWIND SUBROUTINE*****
1992
1993 011526 000240                       RWND:  NOP
1994 011530 004767 001032                 JSR    PC,INIT1           ;INIT
1995 011534 012777 000007 167036         MOV    #7,@C1           ;START REWIND
1996 011542 012700 040000                 MOV    #40000,R0
1997 011546 005300                       RWNDA:  DEC    R0
1998 011550 001376                       BNE    RWNDA           ;DELAY
1999 011552 032777 020000 167032         RWNDB:  BIT    #20000,@DS
2000 011560 001374                       BNE    RWNDB           ;AWAIT PIP
2001 011562 032777 000002 167022         BIT    #2,@DS          ;SEE IF BOT
2002 011570 001012                       BNE    RWNDX           ;IF SO: BR
2003 011572 016704 167110                 MOV    EMADDR,R4
2004 011576 004767 001412                 JSR    PC,TTOUT         ;PRINT HEADER
2005 011602 012704 014646                 MOV    #MSG2,R4
2006 011606 004767 001402                 JSR    PC,TTOUT         ;PRINT REWIND ERROR
2007 011612 000167 171320                 JMP    TSCD2           ;RETURN TO SECHEDULAR
2008 011616 004767 000744                 RWNDX:  JSR    PC,INIT1 ;INIT
2009 011622 000207                       RTS    PC              ;RETURN TO CALLER
2010

```

```

2012                                     ;ERROR CHECK SUBROUTINE*****
2013
2014 011624 005067 167160          ERCHK: CLR      SERFL      ;CLEAR FLAG
2015 011630 017767 166756 167124  MOV     @DS,DSAV   ;SAVE DRIVE STATUS REGISTER
2016 011636 032777 040000 166746  BIT     @40000,@DS ;SEE IF ERROR
2017 011644 001021                    BNE     ERPT       ;IF SO: BR
2018 011646 022777 007000 167130  CMP     @FT15,@LTADD ;CHECK FOR TEST 15
2019 011654 001404                    BEQ     1@         ;BRANCH IF TEST 15
2020 011656 022777 007200 167120  CMP     @FT16,@LTADD ;CHECK FOR TEST 16
2021 011664 001010                    BNE     2@         ;BRANCH IF NOT TEST 16
2022 011666 032777 100000 166716  1@:    BIT     @100000,@DS ;TEST FOR ATA
2023 011674 001004                    BNE     2@         ;EXIT IF ATA IS SET
2024 011676 012767 016703 167020  MOV     @MSG63,ERRP ;SET UP ERROR CODE
2025 011704 000415                    BR      ERPTG      ;GO REPORT ERROR
2026 011706 000207                    2@:    RTS     PC      ;RETURN
2027 011710 017704 166700          ERPT:  MOV     @ER,R4  ;GET ERROR REGISTER
2028 011714 032767 002000 167072  BIT     @2000,UDES  ;SEE IF PE
2029 011722 001403                    BEQ     ERPTA1     ;IF SO: BR
2030 011724 042767 000200 167024  BIC     @20@,STMSK ;RESET PEF MASK
2031 011732 046704 167020          ERPTA1: BIC     STMSK,R4  ;MASK DONT CARE BITS
2032 011736 001530                    BEQ     ERPTX      ;IF NO UNEXPECTED ERRORS: BR
2033 011740 012767 000001 167042  ERPTG: MOV     @1,SERFL ;SET FLAG
2034 011746 032777 020000 166664  BIT     @20000,@SWR ;SEE IF SHOULD PRINT ERRORS
2035 011754 001115                    BNE     ERPTD      ;IF NOT: BR
2036 011756 005767 166722          TST     HDRFL      ;SEE IF DONE HEADER
2037 011762 001006                    BNE     ERPTA      ;IF SO: BR
2038 011764 005267 166714          INC     HDRFL      ;SET HEADER FLAG
2039 011770 016704 166712          MOV     EMADDR,R4
2040 011774 004767 001214          JSR     PC,TTOUT   ;PRINT HEADER
2041 012000 016704 166720          ERPTA: MOV     ERRP,R4 ;GET ERROR CODE
2042 012004 001414                    BEQ     ERPTB      ;IF NONE: BR
2043 012006 004767 001202          JSR     PC,TTOUT   ;PRINT ERROR CODE
2044 012012 012704 015303          MOV     @MSG20,R4  ;SET NRZ TAG
2045 012016 032777 002000 166606  BIT     @2000,@TC  ;SEE IF PE
2046 012024 001402                    BEQ     ERPT1A     ;IF NOT: BR
2047 012026 012704 015311          MOV     @MSG21,R4  ;ELSE SET PE TAG
2048 012032 004767 001156          ERPT1A: JSR     PC,TTOUT ;PRINT TAG
2049 012036 016704 166664          ERPTB: MOV     ERRP1,R4 ;SEE IF CODE 2
2050 012042 001402                    BEQ     ERPTB1     ;IF NOT: BR
2051 012044 004767 001144          JSR     PC,TTOUT   ;PRINT CODE 2
2052 012050 032777 010000 166562  ERPTB1: BIT     @10000,@SWR ;SEE IF ITERATION
2053 012056 001010                    BNE     ERPTC      ;IF NOT: BR
2054 012060 012704 016555          MOV     @MSG56,R4
2055 012064 004767 001124          JSR     PC,TTOUT   ;PRINT ITER TAG
2056 012070 016703 166664          MOV     ITCNT,R3
2057 012074 004767 001260          JSR     PC,OCTP    ;PRINT ITERATION
2058 012100 012704 014560          ERPTC: MOV     @MSG1,R4
2059 012104 004767 001104          JSR     PC,TTOUT   ;PRINT REGISTER TAG
2060 012110 017703 166464          MOV     @C1,R3
2061 012114 004767 001226          JSR     PC,OCTPE   ;PRINT CS1
2062 012120 017703 166456          MOV     @MC,R3
2063 012124 004767 001216          JSR     PC,OCTPE   ;PRINT WC
2064 012130 017703 166450          MOV     @BA,R3
2065 012134 004767 001206          JSR     PC,OCTPE   ;PRINT BA
2066 012140 017703 166442          MOV     @FC,R3
2067 012144 004767 001176          JSR     PC,OCTPE   ;PRINT FC

```

```

2068 012150 017703 166434      MOV      @CS,R3
2069 012154 004767 001166      JSR      PC,CTPE      ;PRINT CS2
2070 012160 017703 166426      MOV      @DS,R3
2071 012164 004767 001156      JSR      PC,CTPE      ;PRINT DS
2072 012170 017703 166420      MOV      @ER,R3
2073 012174 004767 001146      JSR      PC,CTPE      ;PRINT ER
2074 012200 017703 166426      MOV      @TC,R3
2075 012204 004767 001136      JSR      PC,CTPE      ;PRINT TC
2076 012210 005777 166424      ERPTD:  TST      @SWR
2077 012214 100001                BPL      ERPTX        ;SEE IF HALT ON ERROR
2078 012216 000000                HALT
2079 012220 004767 001564      ERPTX:  JSR      PC,CKSWR ;CHECK FOR CNTL G
2080 012224 004767 000336      JSR      PC,INIT1
2081 012230 000207      ERPTXX: RTS      PC      ;INIT
2082
2083

```

```

2085                                     ;TAPE MARK STATUS CHECK*****
2086
2087 012232 032767 000004 166522 TMCHK: BIT    #4,DSAV      ;SEE IF TM SET
2088 012240 001401                BEQ    TMCHK1      ;IF NOT: BR
2089 012242 000207                TMCHK0: RTS    PC      ;ELSE RETURN
2090 012244 005767 166540                TMCHK1: TST    SERFL   ;SEE IF HAD ERROR
2091 012250 001374                BNE    TMCHK0      ;IF SO: BR
2092 012252 012767 016565 166446        MOV    #MSG57,ERRP1 ;SET ERROR CODE 2
2093 012260 004767 177454                JSR    PC,ERPTG    ;GO PRINT TM ERROR
2094 012264 005067 166436                CLR    ERRP1      ;CLEAR CODE 2 FLAG
2095 012270 000207                RTS    PC          ;RETURN
2096
2097                                     ;DATA SETUP ROUTINE*****
2098
2099 012272 000240                DSUP:  NOP
2100 012274 012703 017742                DS0:  MOV    #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
2101 012300 016701 166512                MOV    PATRN,R1     ;R1 = PATTERN SELECTOR
2102 012304 000241                CLC
2103 012306 006101                ROL    R1           ;MAKE PATTERN SELECTOR EVEN
2104 012310 000171 001040                JMP    @DATBL(R1)   ;GO GENERATE PATTERN
2105 012314 032777 010000 166304        DS1:  BIT    #10000,BDT ;SEE IF SEVEN TRACK
2106 012322 001410                BEQ    DS3          ;IF NOT: BR
2107 012324 012702 000640                MOV    #640,R2     ;SET BUFFER SIZE
2108 012330 012701 017742                MOV    #WDATA,R1   ;SET START OF BUFFER
2109 012334 042721 140300                DS2:  BIC    #140300,(R1) ;MASK FOR 7 CH
2110 012340 005302                DEC    R2          ;SEE IF DONE
2111 012342 001374                BNE    DS2         ;IF NOT: BR
2112 012344 012702 000640                DS3:  MOV    #640,R2 ;R2=BUFFER SIZE *2
2113 012350 012701 021454                MOV    #RDATA,R1   ;R1=READ DATA START
2114 012354 005021                DS4:  CLR    (R1)   ;CLEAR BUFFER
2115 012356 005302                DEC    R2          ;SEE IF DONE ALL
2116 012360 001375                BNE    DS4         ;IF NOT: BR
2117 012362 000207                RTS    PC          ;EXIT
2118
2119                                     ;ALL ONES*****
2120
2121 012364 012701 177777                DAT1:  MOV    #-1,R1 ;R1=DATA
2122 012370 012702 000640                DAT1A: MOV    #640,R2 ;R2=WORD COUNT *2
2123 012374 010123                DAT1B: MOV    R1,(R3) ;LOAD BUFFER
2124 012376 005302                DEC    R2          ;SEE IF DONE
2125 012400 001375                BNE    DAT1B       ;IF NOT: BR
2126 012402 000167 177706                JMP    DS1         ;RETURN
2127
2128                                     ;ALL ZEROS*****
2129
2130 012406 005001                DAT2:  CLR    R1     ;R1=DATA
2131 012410 000167 177754                JMP    DAT1A       ;LOAD BUFFER
2132

```



```
2190  
2191 012566 000240  
2192 012570 012777 000040 166012  
2193 012576 016777 166106 166004  
2194 012604 016777 166102 166020  
2195 012612 000207  
2196  
  
INIT1:  NOP  
        MOV  @40,@CS  
INIT2:  MOV  DRVN,@CS  
        MOV  SLVN,@TC  
        RTS  PC  
  
;INIT  
;SELECT DRIVE  
;SELECT SLAVE  
;RETURN
```

```

2198                                     ;MAG TAPE INTERRUPT HANDLER*****
2199
2200 012614 000240          MTINT:  NOP
2201 012616 022626          CMP      (SP), (SP)      ;RESET STACK POINTER
2202 012620 000240          NOP
2203 012622 000240          NOP
2204 012624 000177 166114  JMP      @RTN          ;RETURN TO CALLER
2205
2206                                     ;TTY INTERRUPT HANDLER*****
2207
2208 012630 000240          TTINT:  NOP
2209 012632 000240          NOP
2210 012634 000240          NOP
2211 012636 000002          RTI
2212
2213                                     ;BUS ADDRESS TRAP HANDLER*****
2214
2215 012640 000240          TRAP:   NOP
2216 012642 032777 020000 165770  BIT      @20000, @SWR    ;SEE IF SHOULD PRINT ERRORS
2217 012650 001020          BNE     TRAP2          ;IF NOT: BR
2218 012652 005767 166026  TST     HDRFL          ;SEE IF DONE HEADER
2219 012656 001006          BNE     TRAP1          ;IF SO: BR
2220 012660 005267 166020  INC     HDRFL          ;ELSE SET HEADER FLAG
2221 012664 016704 166016  MOV     EMADDR, R4
2222 012670 004767 000320  JSR     PC, TTOUT      ;PRINT HEADER
2223 012674 012704 015336  TRAP1:  MOV     @MSG24, R4
2224 012700 004767 000310  JSR     PC, TTOUT      ;PRINT ERROR
2225 012704 010103          MOV     R1, R3
2226 012706 004767 000446  JSR     PC, OCTP       ;PRINT ADDRESS OF TRAP
2227 012712 005777 165722  TRAP2:  TST     @SWR      ;SEE IF HALT ON ERROR
2228 012716 100001          BPL     TRAPX          ;IF NOT: BR
2229 012720 000000          HALT
2230 012722 004767 001062  TRAPX:  JSR     PC, CKSWR    ;CHECK FOR CNTL G
2231 012726 022626          CMP     (SP), (SP)    ;RESET STACK
2232 012730 012767 003444 166034  MOV     @FT1A, SCOLP   ;SET SCOPE ADDRESS
2233 012736 004767 177504  JSR     PC, SCOPE      ;GO SEE IF SCOPE LOOP
2234 012742 005767 166052  TST     RHTF           ;SEE IF INITIAL ADDRESS TEST
2235 012746 001402          BEQ     TRAPXX        ;IF NOT: BR
2236 012750 000167 167022  JMP     STOB           ;ELSE REDO ADDRESS REQUEST
2237 012754 000167 170470  TRAPXX: JMP     FT1B      ;RETURN TO TEST 1
2238

```

```

2240 ;*****
2241 ;TTY ENTRY SUBROUTINE:
2242 ;
2243 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
2244 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
2245 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
2246 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
2247 ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
2248 ;THE CALLING ROUTINE.
2249 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
2250 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
2251 ;MAY BE REENTERED.
2252 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
2253 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
2254 ;CARRIAGE RETURN
2255 ;*****
2256
2257 012760 005067 165764 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
2258 012764 005000 CLR RO
2259 012766 004767 000152 TTR0: JSR PC,TIN ;GO READ CHARACTER
2260 012772 122767 000015 165700 CMPB #15,TIB ;SEE IF CR
2261 013000 001005 BNE TTR1 ;IF NOT: BR
2262 013002 005767 165742 TST TEMP1 ;SEE IF FIRST CHARACTER
2263 013006 001446 BEQ TTR5 ;IF SO: BR
2264 013010 000167 000066 JMP TTR2 ;ELSE GO LOAD VALUE
2265 013014 122767 000060 165656 TTR1: CMPB #60,TIB ;SEE IF CHAR IS LESS THAN 0
2266 013022 101402 BLOS TTR1A ;IF NOT: BR
2267 013024 000167 000076 JMP TINER ;ELSE GO TO ERROR
2268 013030 122767 000070 165642 TTR1A: CMPB #70,TIB ;SEE IF CHAR IS GREATER THAN 7
2269 013036 101002 BHI TTR1B ;IF NOT: BR
2270 013040 000167 000062 JMP TINER ;ELSE GO TO ERROR
2271 013044 005267 165700 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
2272 013050 000241 CLC
2273 013052 006100 ROL RO
2274 013054 000241 CLC
2275 013056 006100 ROL RO ;SHIFT 3 LEFT
2276 013060 000241 CLC
2277 013062 006100 ROL RO
2278 013064 042767 177770 165606 BIC #177770,TIB ;STRIP ASCII
2279 013072 056700 165602 BIS TIB,RO ;LOAD CHARACTER
2280 013076 005301 DEC R1 ;SEE IF DONE
2281 013100 001332 BNE TTR0 ;IF NOT: BR
2282 013102 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
2283 013104 101402 BLOS TTR3 ;IF NOT: BR
2284 013106 000167 000014 JMP TINER ;ELSE GO TO ERROR
2285 013112 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
2286 013114 101402 BLOS TTR4 ;IF NOT: BR
2287 013116 000167 000004 JMP TINER ;ELSE GO TO ERROR
2288 013122 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
2289 013124 000207 TTR5: RTS PC ;EXIT
2290

```

```

2292                                     ;TTY ENTRY ERROR SUBROUTINE*****
2293
2294 013126 012704 015042          TINER:  MOV    @MSG7,R4
2295 013132 004767 000056          JSR    PC,TTOUT          ;PRINT?
2296 013136 162716 000020          SUB    @20,(SP)         ;RESET SP TO START OF VALUE ROUTINE
2297 013142 000207                  RTS    PC                ;REDO VALUE ENTRY
2298
2299                                     ;TTY READ SUBROUTINE*****
2300
2301 013144 005077 165472          TTIN:  CLR    @TKS
2302 013150 005077 165470          CLR    @TKB
2303 013154 105777 165462          TSTB   @TKS
2304 013160 100375                  BPL    .-4
2305 013162 017767 165456 165510  MOV    @TKB,TIB
2306 013170 042767 177600 165502  BIC    @177600,TIB
2307 013176 105777 165444          TTIN2: TSTB   @TPS
2308 013202 100375                  BPL    TTIN2
2309 013204 116777 165470 165436  MOVB   TIB,@TPB
2310 013212 000207                  RTS    PC
2311
2312                                     ;TTY OUTPUT SUBROUTINE*****
2313
2314 013214 112467 165456          TTOUT: MOVB   (R4)+,TOB
2315 013220 122767 000043 165450  CMPB   @43,TOB
2316 013226 001446                  BEQ    TEX
2317 013230 122767 000045 165440  CMPB   @45,TOB
2318 013236 001403                  BEQ    TCRLF
2319 013240 004767 000064          JSR    PC,TOG
2320 013244 000763                  BR     TTOUT
2321 013246 112767 000015 165422  TCRLF: MOVB   @15,TOB
2322 013254 004767 000050          JSR    PC,TOG
2323 013260 012703 000004          MOV    @4,R3
2324 013264 005067 165406          TCRLFA: CLR    TOB
2325 013270 004767 000034          JSR    PC,TOG
2326 013274 005303                  DEC    R3
2327 013276 001372                  BNE    TCRLFA          ;DO FILLERS
2328 013300 112767 000012 165370  MOVB   @12,TOB
2329 013306 004767 000016          JSR    PC,TOG
2330 013312 105767 165516          TSTB   RDSW
2331 013316 100401                  BMI    1#
2332 013320 000735                  BR     TTOUT
2333 013322 005067 165506          1#:   CLR    RDSW
2334 013326 000406                  BR     TEX
2335 013330 105777 165312          TOG:   TSTB   @TPS
2336 013334 100375                  BPL    TOG
2337 013336 116777 165334 165304  MOVB   TOB,@TPB
2338 013344 000207                  RTS    PC
2339
2340

```

```

2342                                     ;OCTAL OUTPUT SUBROUTINE*****
2343
2344 013346 012767 000001 000222 OCTPC: MOV #1,OFL
2345 013354 010304          MOV R3,R4
2346 013356 000410          BR OCTP0
2347 013360 005067 000212          OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
2348 013364 010304          OCTPe1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
2349 013366 001004          BNE OCTP0 ;IF NOT ZERO: BR
2350 013370 004767 000162          JSR PC,OCTPG1 ;ELSE PRINT ZERO
2351 013374 000167 000120          JMP OCTP3 ;SPACE AND EXIT
2352 013400 032704 100000          OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
2353 013404 001406          BEQ OCTP1 ;IF NOT: BR
2354 013406 012704 000001          MOV #1,R4
2355 013412 004767 000116          JSR PC,OCTPG ;PRINT 1
2356 013416 000167 000006          JMP OCTP2
2357 013422 005004          OCTP1: CLR R4
2358 013424 004767 000104          JSR PC,OCTPG ;PRINT 0
2359 013430 010304          OCTP2: MOV R3,R4
2360 013432 006004          ROR R4
2361 013434 006004          ROR R4
2362 013436 006004          ROR R4 ;POSITION DIGIT
2363 013440 006004          ROR R4
2364 013442 000304          SWAB R4
2365 013444 004767 000064          JSR PC,OCTPG ;PRINT DIGIT 2
2366 013450 010304          MOV R3,R4
2367 013452 006004          ROR R4
2368 013454 000304          SWAB R4
2369 013456 004767 000052          JSR PC,OCTPG ;PRINT DIGIT 3
2370 013462 010304          MOV R3,R4
2371 013464 006104          ROL R4
2372 013466 006104          ROL R4
2373 013470 000304          SWAB R4
2374 013472 004767 000036          JSR PC,OCTPG ;PRINT DIGIT 4
2375 013476 010304          MOV R3,R4
2376 013500 006004          ROR R4
2377 013502 006004          ROR R4
2378 013504 006004          ROR R4
2379 013506 004767 000022          JSR PC,OCTPG
2380 013512 010304          MOV R3,R4
2381 013514 004767 000014          JSR PC,OCTPG ;PRINT DIGIT 5
2382 013520 012767 000240 165150 OCTP3: MOV #240,TOB
2383 013526 004767 177576          JSR PC,TOB ;PRINT SPACE
2384 013532 000207          RTS PC ;EXIT
2385 013534 042704 177770          OCTPG: BIC #177770,R4
2386 013540 001004          BNE OCTPG0
2387 013542 005767 000030          TST OFL
2388 013546 001001          BNE OCTPG0
2389 013550 000207          RTS PC
2390 013552 005267 000020          OCTPG0: INC OFL
2391 013556 052704 000260          OCTPG1: BIS #260,R4
2392 013562 010467 165110          MOV R4,TOB
2393 013566 004767 177536          JSR PC,TOB
2394 013572 010304          MOV R3,R4
2395 013574 000207          RTS PC
2396 013576 000000          OFL: 0 ;FIRST CHAR FLAG
2397

```

```

2398                                     ;DATA CHARACTER OUTPUT SUBROUTINE*****
2399
2400 013600 005067 165072          DOUT: CLR      TOB
2401 013604 012704 000010          MOV      #10,R4          ;SET NUMBER TO PRINT
2402 013610 110367 165062          MOVB    R3,TOB
2403 013614 105777 165026          DOUT1: TSTB    @TPS
2404 013620 100375                   BPL     DOUT1
2405 013622 132767 000200 165046   BITB    #200,TOB
2406 013630 001404                   BEQ     DOUT2
2407 013632 012777 000061 165010   MOV     #061,@TPB
2408 013640 000403                   BR      DOUT3
2409 013642 012777 000060 165000   DOUT2: MOV     #060,@TPB
2410 013650 006167 165022          DOUT3: ROL     TOB
2411 013654 005304                   DEC     R4
2412 013656 001356                   BNE    DOUT1
2413 013660 000207                   RTS     PC
2414 013662 016703 165066          DOUTD: MOV     TEMP3,R3
2415 013666 000303                   SWAB   R3
2416 013670 004767 177704          JSR    PC,DOUT
2417 013674 016703 165054          MOV     TEMP3,R3
2418 013700 004767 177674          JSR    PC,DOUT
2419 013704 000207                   RTS     PC

```

```

2420                                     ;TU16/TE16 SERIAL NUMBER PRINT SUBROUTINE*****
2421
2422
2423 013706 010304          SNPT: MOV      R3,R4
2424 013710 000304          SWAB   R4
2425 013712 006004          ROR    R4
2426 013714 006004          ROR    R4
2427 013716 006004          ROR    R4
2428 013720 006004          ROR    R4
2429 013722 004767 000036   JSR    PC,SNPG          ;GET FIRST DIGIT
2430 013726 010304          MOV     R3,R4          ;GO PRINT
2431 013730 000304          SWAB   R4
2432 013732 004767 000026   JSR    PC,SNPG          ;GET SECOND DIGIT
2433 013736 010304          MOV     R3,R4          ;GO PRINT
2434 013740 006004          ROR    R4
2435 013742 006004          ROR    R4
2436 013744 006004          ROR    R4
2437 013746 006004          ROR    R4
2438 013750 004767 000010   JSR    PC,SNPG          ;GET THIRD DIGIT
2439 013754 010304          MOV     R3,R4          ;GO PRINT
2440 013756 004767 000002   JSR    PC,SNPG          ;GET FOURTH DIGIT
2441 013762 000207          RTS     PC          ;GO PRINT
2442 013764 012767 000260 164704 SNPG: MOV     #260,TOB          ;EXIT
2443 013772 042704 177760          BIC    #177760,R4      ;SET BASE = 0
2444 013776 050467 164674          BIS    R4,TOB          ;MASK DIGIT
2445 014002 004767 177322          JSR    PC,TOG          ;SET ASCII
2446 014006 000207          RTS     PC          ;TYPE DIGIT
2447                                     ;RETURN

```

```

2448                                     ;CKSWR ROUTINE THAT ALLOWS THE LOADING OF LOC.176, SWREG*****
2449                                     ;FROM THE TTY AT SELECTED POINTS WITHIN THE PROGRAM*****
2450
2451 014010 022767 000176 164622 CKSWR: CMP     #SWREG,SWR          ;SOFTWARE SWITCH REG PRESENT
2452 014016 001041          BNE    OUT              ;NO, GET OUT
2453 014020 105777 164616          TSTB   @TKS              ;YES, WAIT FOR

```

```

2454 014024 100036          BPL      OUT          ;READY, GET CHARACTER
2455 014026 017767 164612 164644  MOV     @TKB,TIB     ;AND STRIP OFF
2456 014034 042767 177600 164636  BIC     @177600,TIB  ;THE GARBAGE
2457 014042 022767 000007 164630  CMP     @7,TIB      ;IS IT A <+G>
2458 014050 001024          BNE     OUT
2459 014052 012704 017713  MOV     @CNTG,R4
2460 014056 004767 177132  JSR     PC,TTOUT
2461 014062 012704 017717  CNTLU: MOV     @MSWR,R4
2462 014066 004767 177122  JSR     PC,TTOUT
2463 014072 017703 164542  MOV     @SWR,R3
2464 014076 004767 177244  JSR     PC,OCTPE
2465 014102 012704 017726  MOV     @MNEW,R4
2466 014106 004767 177102  JSR     PC,TTOUT
2467 014112 005037 001030  CLR     @TEMPST
2468 014116 004767 000002  JSR     PC,$READ
2469 014122 000207  OUT:    RTS      PC          ;GO READ A LINE
2470                                ;RETURN TO MAIN BODY OF PROGRAM
2471 014124 005067 164700  $READ: CLR     TEMPST
2472 014130 012767 000007 164674  MOV     @7,COUNT
2473 014136 004767 177002  1$:    JSR     PC,TTIN     ;GO READ A CHARACTER
2474 014142 042767 177600 164530  BIC     @177600,TIB  ;STRIP OFF GARBAGE
2475 014150 122767 000025 164522  CMPB   @25,TIB      ;IS IT A +U?
2476 014156 001002          BNE     2$           ;BRANCH IF NOT
2477 014160 005726          3$:    TST     (SP)+    ;POP THE STACK
2478 014162 000737          BR     CNTLU        ;START OVER
2479 014164 122767 000015 164506  2$:    CMPB   @15,TIB    ;IS IT A <CR>?
2480 014172 001013          BNE     4$           ;BRANCH IF NOT
2481 014174 012767 000200 164632  MOV     @200,RDSW
2482 014202 004767 177040  JSR     PC,TCRLF
2483 014206 022767 000007 164616  CMP     @7,COUNT
2484 014214 001037          BNE     7$           ;ECHO IT WITH <LF>
2485 014216 005726          8$:    TST     (SP)+    ;WAS IT FIRST CHARACTER
2486 014220 000740          BR     OUT          ;CHANGE SWR IF NOT FIRST ONE
2487 014222 122767 000060 164450  4$:    CMPB   @60,TIB    ;POP THE STACK
2488 014230 003004          BGT     5$           ;GET OUT
2489 014232 122767 000067 164440  CMPB   @67,TIB
2490 014240 003005          BGT     6$
2491 014242 012704 017736  5$:    MOV     @REQUEST,R4
2492 014246 004767 176742  JSR     PC,TTOUT
2493 014252 000742          BR     3$           ;START OVER IF NOT LEGAL CHARACTER
2494 014254 006367 164550  6$:    ASL     TEMPST
2495 014260 006367 164544  ASL     TEMPST
2496 014264 006367 164540  ASL     TEMPST
2497 014270 142767 000060 164402  BICB   @60,TIB      ;GET NITTY-GRITTY
2498 014276 156767 164376 164524  BISB   TIB,TEMPST
2499 014304 005367 164522  DEC     COUNT
2500 014310 001754          BEQ     5$           ;ONLY WANT 6 DIGITS
2501 014312 000711          BR     1$
2502 014314 016777 164510 164316  7$:    MOV     TEMPST,@SWR ;CHANGE SWITCH REGISTER CONTENTS
2503 014322 000735          BR     8$
2504

```

```

2506
2507 ; *****
2508 ; CHECK FOR DUMP MODE OR AUTO MODE
2509 ; *****
2510
2511 014324 005067 164340 CKMODE: CLR AUTOM ;INIT AUTO MODE
2512 014330 005737 000042 TST @#42 ;AUTO MODE?
2513 014334 001417 BEQ 2# ;BRANCH - IF NOT
2514 014336 005267 164326 INC AUTOM ;SET AUTO MODE INDICATORE
2515 014342 023737 000042 000046 CMP @#42,@#46 ;ACT11 MODE?
2516 014350 001403 BEQ 1# ;BRANCH - IF YES
2517 014352 105267 164315 INCB XXDPM ;INDICATE XXDP AUTO MODE
2518 014356 000421 BR 5# ;AND EXIT
2519 014360 105267 164306 1#: INCB ACT11M ;INDICATE ACT11 AUTO MODE
2520 014364 052777 104000 164246 BIS @104000,@SWR ;SET FOR CON:CYCLE & HALT ON ERROR
2521 014372 000413 BR 5# ;AND EXIT
2522 014374 105737 000041 2#: TSTB @#41 ;MAN: MODE VIA ACT11/PAPER TAPE?
2523 014400 001003 BNE 3# ;BRANCH - IF NOT
2524 014402 105267 164266 INCB ADUMPM ;INDICATE MAN: MODE VIA ACT11/PAPER TAPE
2525 014406 000402 BR 4# ;AND EXIT THRU M.I
2526 014410 105267 164261 3#: INCB XDUMPM ;INDICATE MANUAL MODE VIA XXDP
2527 014414 052737 020000 000052 4#: BIS @20000,@#52 ;ALLOW MANUAL INTERVENTION
2528 014422 000207 5#: RTS PC ;RETURN
2529
2530 ; *****
2531

```

```

2533
2534 ; *****
2535 ; DISCONTINUE TESTING FOR
2536 ; ILLEGAL CONDITIONS
2537 ; *****
2538
2539 ABORT:
2540 014424 000005 RESET ;INITIALIZE THE WORLD
2541 014426 012704 014536 MOV #MSGD,R4 ;GET ABORT MESSAGE
2542 014432 004767 176556 JSR PC,TTOUT ;TYPE ABORT MESSAGE
2543 014436 105767 164231 TSTB XXDPM ;XXDP AUTO MODE?
2544 014442 001405 BEQ 1$ ;BRANCH - IF NOT
2545 014444 013700 000042 MOV @#42,R0 ;GET MONITOR EXIT ADDRESS
2546 014450 005037 000042 CLR @#42 ;USE AS ABORT FLAG
2547 014454 004710 JSR PC,(R0) ;EXIT TO XXDP MONITOR
2548 014456 000777 1$: BR . ;AND HANG
2549 ;
2550 ; *****

```

```

2552                                     ;MESSAGE TABLE*****
2553                                     ;*****
2554 014460 050045 047522 051107 MSGC: .ASCII /PROGRAM DISABLED ? NO MANUAL INTERVENTION.#/
      014466 046501 042040 051511
      014474 041101 042514 020104
      014502 037440 047040 020117
      014510 040515 052516 046101
      014516 044440 052116 051105
      014524 042526 052116 047511
      014532 027116 021445
2555 014536 050045 047522 051107 MSGD: .ASCII /PROGRAM ABORTED#/
      014544 046501 040440 047502
      014552 052122 042105 021445
2556
2557                                     ;*****
2558 014560 041445 030523 020040 MSG1: .ASCII /CS1 WC BA FC CS2 /
      014566 020040 041527 020040
      014574 020040 041040 020101
      014602 020040 020040 041506
      014610 020040 020040 041440
      014616 031123 020040 020040
2559 014624 051504 020040 020040 .ASCII /DS ER TC#/
      014632 042440 020122 020040
      014640 020040 041524 021445
2560 014646 051045 053505 047111 MSG2: .ASCII /REWIND ERROR#/
      014654 020104 051105 047522
      014662 021522
2561 014664 022445 046524 031060 MSG3: .ASCII /TM02 - TU16 - TE16 BSC FC (CZTUBHO) #/
      014672 026440 052040 030525
      014700 020066 020055 042524
      014706 033061 041040 041523
      014714 043040 020103 041450
      014722 052132 041125 030110
      014730 020051 045
2562 014733 105 052116 051105 .ASCII /ENTER CONDITIONS IN OCTAL#/
      014740 041440 047117 044504
      014746 044524 047117 020123
      014754 047111 047440 052103
      014762 046101 021445
2563 014766 051045 043505 051511 MSG4: .ASCII /REGISTER START = #/
      014774 042524 020122 052123
      015002 051101 020124 020075
      015010 043
2564 015011 045 042526 052103 MSG5: .ASCII /VECTOR = #/
      015016 051117 036440 021440
2565 015024 042445 042116 047440 MSG6: .ASCII /END OF PASS #/
      015032 020106 040520 051523
      015040 021440
2566 015042 037440 021440 MSG7: .ASCII / ? #/
2567 015046 021445 MSG8: .ASCII /# #/
2568 015050 050045 051517 052111 MSG9: .ASCII /POSITION ERROR: #/
      015056 047511 020116 051105
      015064 047522 035122 021440
2569 015072 042045 044522 042526 MSG10: .ASCII /DRIVE NUMBER: #/
      015100 047040 046525 042502
      015106 035122 021440

```

2570	015112	051445	040514	042526	MSG11:	.ASCII	/#SLAVE NUMBER: 0/
	015120	047040	046525	042502			
	015126	035122	021440				
2571	015132	053445	044522	042524	MSG12:	.ASCII	/#WRITE ERROR 0/
	015140	042440	051122	051117			
	015146	021440					
2572	015150	051045	040505	020104	MSG13:	.ASCII	/#READ REVERSE ERROR 0/
	015156	042522	042526	051522			
	015164	020105	051105	047522			
	015172	020122	043				
2573	015175	045	042522	042101	MSG14:	.ASCII	/#READ FORWARD ERROR 0/
	015202	043040	051117	040527			
	015210	042122	042440	051122			
	015216	051117	021440				
2574	015222	053445	044522	042524	MSG15:	.ASCII	/#WRITE TM ERROR 0/
	015230	052040	020115	051105			
	015236	047522	020122	043			
2575	015243	045	042522	042526	MSG16:	.ASCII	/#REVERSE ERROR 0/
	015250	051522	020105	051105			
	015256	047522	020122	043			
2576	015263	045	047506	053522	MSG17:	.ASCII	/#FORWARD ERROR 0/
	015270	051101	020104	051105			
	015276	047522	020122	043			
2577	015303	040	051116	020132	MSG20:	.ASCII	/ NRZ 0/
	015310	043					
2578	015311	040	042520	021440	MSG21:	.ASCII	/ PE 0/
2579	015316	042440	050130	035124	MSG22:	.ASCII	/ EXPT: 0/
	015324	021440					
2580	015326	051040	05103	035104	MSG23:	.ASCII	/ RCVD: 0/
	015334	021440					
2581	015336	041045	051525	052040	MSG24:	.ASCII	/#BUS TRAP: 0/
	015344	040522	035120	021440			
2582	015352	053445	035103	021440	MSG25:	.ASCII	/#MC: 0/
2583	015360	041045	035101	021440	MSG26:	.ASCII	/#BA: 0/
2584	015366	042045	035102	021440	MSG27:	.ASCII	/#DB: 0/
2585	015374	044445	044516	020124	MSG28:	.ASCII	/#INIT DID NOT CLEAR RM 0/
	015402	044504	020104	047516			
	015410	020124	046103	040505			
	015416	020122	044122	021440			
2586	015424	051445	020103	047516	MSG29:	.ASCII	/#SC NOT RESET BY INIT 0/
	015432	020124	042522	042523			
	015440	020124	054502	044440			
	015446	044516	020124	043			
2587	015453	045	051124	020105	MSG30:	.ASCII	/#TRE NOT RESET BY INIT 0/
	015460	047516	020124	042522			
	015466	042523	020124	054502			
	015474	044440	044516	020124			
	015502	043					
2588	015503	045	051503	020062	MSG31:	.ASCII	/#CS2 NOT RESET BY INIT 0/
	015510	047516	020124	042522			
	015516	042523	020124	054502			
	015524	044440	044516	020124			
	015532	043					
2589	015533	045	046104	020124	MSG32:	.ASCII	/#DLT NOT SET 0/
	015540	047516	020124	042523			
	015546	020124	043				

2590	015551	045	041523	047040	MSG33:	.ASCII	/#SC NOT SET #/
	015556	052117	051440	052105			
	015564	021440					
2591	015566	052045	042522	047040	MSG34:	.ASCII	/#TRE NOT SET #/
	015574	052117	051440	052105			
	015602	021440					
2592	015604	044445	020122	047516	MSG35:	.ASCII	/#IR NOT SET BY INIT #/
	015612	020124	042523	020124			
	015620	054502	044440	044516			
	015626	020124	043				
2593	015631	045	051117	047040	MSG36:	.ASCII	/#OR NOT RESET BY INIT #/
	015636	052117	051040	051505			
	015644	052105	041040	020131			
	015652	047111	052111	021440			
2594	015660	047445	020122	047516	MSG37:	.ASCII	/#OR NOT RESET BY 1 SILO ENTRY #/
	015666	020124	042522	042523			
	015674	020124	054502	030440			
	015702	051440	046111	020117			
	015710	047105	051124	020131			
	015716	043					
2595	015717	045	051117	047040	MSG38:	.ASCII	/#OR NOT SET BY SILO FULL #/
	015724	052117	051440	052105			
	015732	041040	020131	044523			
	015740	047514	043040	046125			
	015746	020114	043				
2596	015751	045	040502	020104	MSG39:	.ASCII	/#BAD SILO READ #/
	015756	044523	047514	051040			
	015764	040505	020104	043			
2597	015771	045	051111	047040	MSG40:	.ASCII	/#IR NOT RESET BY SILO FULL#/
	015776	052117	051040	051505			
	016004	052105	041040	020131			
	016012	044523	047514	043040			
	016020	046125	021514				
2598	016024	047040	047117	042455	MSG41:	.ASCII	/ NON-EXIST DRIVE#/
	016032	044530	052123	042040			
	016040	044522	042526	043			
2599	016045	040	047516	026516	MSG42:	.ASCII	/ NON-EXIST SLAVE#/
	016052	054105	051511	020124			
	016060	046123	053101	021505			
2600	016066	051440	051105	040511	MSG43:	.ASCII	/ SERIAL NO: #/
	016074	020114	047516	020072			
	016102	043					
2601	016103	045	051105	051501	MSG44:	.ASCII	/#ERASE HEAD INOPERATIVE/
	016110	020105	042510	042101			
	016116	044440	047516	042520			
	016124	040522	044524	042526			
2602	016132	041445	042510	045503		.ASCII	/#CHECK POLARITY#/
	016140	050040	046117	051101			
	016146	052111	021531				
2603	016152	042445	040522	042523	MSG45:	.ASCII	/#ERASE HEAD POLARITY WRONG#/
	016160	044040	040505	020104			
	016166	047520	040514	044522			
	016174	054524	053440	047522			
	016202	043516	043				
2604	016205	045	042523	026524	MSG46:	.ASCII	/#SET-UP WRITE ERROR#/
	016212	050125	053440	044522			

	016220	042524	042440	051122			
	016226	051117	043				
2605	016231	045	050123	041501	MSG47:	.ASCII	/SPACE FORWARD ERROR/
	016236	020105	047506	053522			
	016244	051101	020104	051105			
	016252	047522	021522				
2606	016256	051445	040520	042503	MSG48:	.ASCII	/SPACE REVERSE ERROR/
	016264	051040	053105	051105			
	016272	042523	042440	051122			
	016300	051117	043				
2607	016303	045	052502	043106	MSG49:	.ASCII	/BUFFERED WRITE ERROR/
	016310	051105	042105	053440			
	016316	044522	042524	042440			
	016324	051122	051117	043			
2608	016331	045	047502	020124	MSG50:	.ASCII	/BOT SET AFTER BUFFERED WRITE/
	016336	042523	020124	043101			
	016344	042524	020122	052502			
	016352	043106	051105	042105			
	016360	053440	044522	042524			
	016366	043					
2609	016367	045	047516	041040	MSG51:	.ASCII	/NO BOT FROM READ IN PRESET/
	016374	052117	043040	047522			
	016402	020115	042522	042101			
	016410	044440	020116	051120			
	016416	051505	052105	043			
2610	016423	045	041524	044440	MSG52:	.ASCII	/TC INCORRECT /
	016430	041516	051117	042522			
	016436	052103	021440				
2611	016442	051445	040514	042526	MSG53:	.ASCII	/SLAVE NOT OFF LINE/
	016450	047040	052117	047440			
	016456	043106	046040	047111			
	016464	021505					
2612	016466	022445	042522	042523	MSG54:	.ASCII	/RESET SLAVE TO ON LINE BEFORE CONTINUING/
	016474	020124	046123	053101			
	016502	020105	047524	047440			
	016510	020116	044514	042516			
	016516	041040	043105	051117			
	016524	020105	047503	052116			
	016532	047111	044525	043516			
	016540	043					
2613	016541	045	051116	020132	MSG55:	.ASCII	/NRZ ONLY: /
	016546	047117	054514	020072			
	016554	043					
2614	016555	040	052111	051105	MSG56:	.ASCII	/ ITER: /
	016562	020072	043				
2615	016565	045	046524	047040	MSG57:	.ASCII	/TM NOT SET/
	016572	052117	051440	052105			
	016600	043					
2616	016601	045	044505	044124	MSG60:	.ASCII	/EITHER TAPE NOT ERASED OR OPI PROBLEM/
	016606	051105	052040	050101			
	016614	020105	047516	020124			
	016622	051105	051501	042105			
	016630	047440	020122	050117			
	016636	020111	051120	041117			
	016644	042514	021515				
2617	016650	051045	030510	020061	MSG61:	.ASCII	/RH11 OR RH70: /

	016656	051117	051040	033510	
	016664	035060	021440		
2618	016670	051045	020110	047117	MSG62: .ASCII /MRM ONLY: #/
	016676	054514	020072	043	
2619	016703	045	052101	020101	MSG63: .ASCII /DATA NOT SET AFTER NON-TRANSFER OPERATION#/
	016710	047516	020124	042523	
	016716	020124	043101	042524	
	016724	020122	047516	026516	
	016732	051124	047101	043123	
	016740	051105	047440	042520	
	016746	040522	044524	047117	
	016754	043			
2620					

```

2622                                     ;TEST HEADERS*****
2623
2624 016755      045  043045  030524  MSFT1:  .ASCII  /MSFT1:RH ADDRESSING #/
      016762  051072  020110  042101
      016770  051104  051505  044523
      016776  043516  021440
2625 017002  022445  052106  035062  MSFT2:  .ASCII  /MSFT2:RH REGISTER BITS TEST #/
      017010  044122  051040  043505
      017016  051511  042524  020122
      017024  044502  051524  052040
      017032  051505  020124    043
2626 017037      045  043045  031524  MSFT3:  .ASCII  /MSFT3:RH INITIALIZE TEST #/
      017044  051072  020110  047111
      017052  052111  040511  044514
      017060  042532  052040  051505
      017066  020124    043
2627 017071      045  043045  032124  MSFT4:  .ASCII  /MSFT4:RH1 SILO TEST 1 #/
      017076  051072  030510  020061
      017104  044523  047514  052040
      017112  051505  020124  020061
      017120    043
2628 017121      045  043045  032524  MSFT5:  .ASCII  /MSFT5:RH1 SILO TEST 2 #/
      017126  051072  030510  020061
      017134  044523  047514  052040
      017142  051505  020124  020062
      017150    043
2629 017151      045  043045  033124  MSFT6:  .ASCII  /MSFT6:RH1 SILO TEST 3 #/
      017156  051072  030510  020061
      017164  044523  047514  052040
      017172  051505  020124  020063
      017200    043
2630 017201      045  043045  033524  MSFT7:  .ASCII  /MSFT7:RH1 SILO TEST 4 #/
      017206  051072  030510  020061
      017214  044523  047514  052040
      017222  051505  020124  020064
      017230    043
2631 017231      045  043045  030524  MSFT10: .ASCII  /MSFT10:RH1 SILO TEST 5 #/
      017236  035060  044122  030461
      017244  051440  046111  020117
      017252  042524  052123  032440
      017260  021440
2632 017262  022445  052106  030461  MSFT11: .ASCII  /MSFT11:NOP TEST#/
      017270  047072  050117  052040
      017276  051505  021524
2633 017302  022445  052106  031061  MSFT12: .ASCII  /MSFT12:REWIND TEST#/
      017310  051072  053505  047111
      017316  020104  042524  052123
      017324    043
2634 017325      045  043045  030524  MSFT13: .ASCII  /MSFT13:WRITE-READ TEST#/
      017332  035063  051127  052111
      017340  026505  042522  042101
      017346  052040  051505  021524
2635 017354  022445  052106  032061  MSFT14: .ASCII  /MSFT14:SPACE TEST#/
      017362  051472  040520  042503
      017370  052040  051505  021524
2636 017376  022445  052106  032461  MSFT15: .ASCII  /MSFT15:ERASE TEST#/

```

2637	017404	042472	040522	042523	
	017412	052040	051505	021524	
	017420	022445	052106	033061	MSFT16: .ASCII /MSFT16:TAPE MARK WRITE-READ TEST#/
	017426	052072	050101	020105	
	017434	040515	045522	053440	
	017442	044522	042524	051055	
	017450	040505	020104	042524	
	017456	052123	043		
2638	017461	045	043045	030524	MSFT17: .ASCII /MSFT17:TM SPACE TEST #/
	017466	035067	046524	051440	
	017474	040520	042503	052040	
	017502	051505	020124	043	
2639	017507	045	043045	031124	MSFT20: .ASCII /MSFT20:WRITE CHECK TEST #/
	017514	035060	051127	052111	
	017522	020105	044103	041505	
	017530	020113	042524	052123	
	017536	021440			
2640	017540	022445	052106	030462	MSFT21: .ASCII /MSFT21:ERASE HEAD TEST#/
	017546	042472	040522	042523	
	017554	044040	040505	020104	
	017562	042524	052123	043	
2641	017567	045	043045	031124	MSFT22: .ASCII /MSFT22:BUFFERED COMMAND TEST#/
	017574	035062	052502	043106	
	017602	051105	042105	041440	
	017610	046517	040515	042116	
	017616	052040	051505	021524	
2642	017624	022445	052106	031462	MSFT23: .ASCII /MSFT23:READ IN PRESET TEST#/
	017632	051072	040505	020104	
	017640	047111	050040	042522	
	017646	042523	020124	042524	
	017654	052123	043		
2643	017657	045	043045	031124	MSFT24: .ASCII /MSFT24:REWIND-OFF LINE TEST#/
	017664	035064	042522	044527	
	017672	042116	047455	043106	
	017700	046040	047111	020105	
	017706	042524	052123	043	
2644	017713	045	043536	043	\$CNTG: .ASCII /\$+G#/
2645	017717	045	053523	036522	\$MSWR: .ASCII /\$SWR= #/
	017724	021440			
2646	017726	020040	042516	036527	\$MNEW: .ASCII / NEW= #/
	017734	021440			
2647	017736	022477	043		\$QUEST: .ASCII /?##/
2648					
2649					
2650		017742			.EVEN
2651	017742	000000			WDATA: 0
2652		021454			. = ..+1510
2653	021454	000000			RDATA: 0
2654					
2655		000001			.END

ABORT	014424	851	2539*												
ACT11M	000672	711*	876	2519*											
ADUMPM	000674	713*	2524*												
AS	000616	680*													
AUTOM	000670	710*	844	924	968	1016	1109	1136	1142	2511*	2514*				
BA	000604	675*	1186	1968*	2064										
BADDR	000714	727*	1451*	1470*	1495*	1504*	1505*	1512*	1588*	1652*	1738*	1786*	1798*	1803*	
		1823*	1840*	1873*	1908*	1968									
BAE	000634	687*	1057												
BTRP	000664	702*	1155*	1165*											
BTRP2	000666	703*	1156*												
CC	000620	681*													
CKMODE	014324	843	2511*												
CKSMR	014010	1085	1145	1393	1635	2079	2152	2167	2230	2451*					
CNTLU	014062	867	2461*	2478											
CONT1	002334	935	956*												
CONT2	002526	979	1002*												
COUNT	001032	766*	2472*	2483	2499*										
CRCNT	001012	758*													
CS	000610	677*	943*	948*	950	956*	957*	959	1002*	1003*	1244*	1255	1285*	1287	
		1313	1317	1322	1328	1353	1357	1408	1423*	1428*	1432	1545*	1848*	2068	
		2192*	2193*												
C1	000600	673*	908	949	958	1158	1247	1251	1289	1291	1881*	1947*	1975*	1995*	
		2060													
DATA0	001042	774*													
DATA1	001044	775*													
DATA2	001046	776*													
DATA3	001050	777*													
DATBL	001040	773*	2104												
DAT1	012364	774	2121*												
DAT1A	012370	2122*	2131	2137											
DAT1B	012374	2123*	2125												
DAT2	012406	775	2130*												
DAT3	012414	776	2136*												
DAT4	012424	777	2141*												
DAT4A	012432	2143*	2146												
DB	000622	682*	1195	1286	1321*	1326*	1349*	1362	1405*	1425*	1429*	1430	1431		
DOUT	013600	2400*	2416	2418											
DOUTD	013662	2414*													
DOUT1	013614	2403*	2404	2412											
DOUT2	013642	2406	2409*												
DOUT3	013650	2408	2410*												
DRIVE	000040	611*													
DRVN	000710	725*	930	941*	944*	945	948	957	987	1003	2193				
DRVTP	000654	698*													
DS	000612	678*	1474	1882	1891	1919	1925	1951	1979	1999	2001	2015	2016	2022	
		2070													
DSAV	000762	746*	2015*	2087											
DSUP	012272	1490	1785	1822	2099*										
DS0	012274	2100*													
DS1	012314	2105*	2126	2147											
DS2	012334	2109*	2111												
DS3	012344	2106	2112*												
DS4	012354	2114*	2116												
DT	000626	684*	995	997	1005	2105									
EMADDR	000706	724*	1154*	1175*	1215	1241*	1267	1284*	1310*	1345*	1377	1401*	1421*	1457*	

FT14A	006156	1540#	1558					
FT14A1	006122	1535#	1641					
FT14A2	006252	1551	1553#					
FT14A3	006276	1555	1557#					
FT14B	006332	1564#	1586					
FT14EC	006550	1595	1602#					
FT14RE	006556	1594	1601	1604#				
FT14RF	006522	1580	1596#					
FT14RR	006460	1570	1587#					
FT14R0	006630	1613	1615#					
FT14R1	006652	1617	1620#					
FT14R2	006656	1619	1621#					
FT14R3	006730	1606	1632#					
FT14X	006740	1552	1569	1574	1579	1583	1633	1635#
FT14XX	006774	1637	1639	1642#				
FT15	007000	808	809	1646#	2018			
FT15A	007050	1654#						
FT15B	007064	1656#	1662					
FT15X	007174	1660	1673#					
FT16	007200	810	811	1677#	2020			
FT16A	007232	1682#						
FT16B	007236	1683#	1708					
FT16X	007432	1703	1705	1709#				
FT17	007442	812	813	1716#				
FT17A	007462	1719#	1775					
FT17B	007466	1720#	1750					
FT17C	007626	1740#	1749					
FT17D	007676	1733	1751#					
FT17D1	007714	1754#	1769					
FT17E	007730	1756#	1764					
FT17F	010022	1766	1770#					
FT17X	010052	1728	1747	1761	1771	1773	1776#	
FT2	003512	786	787	1175#				
FT2A	003524	1177#	1184	1209				
FT2B	003564	1172	1186#	1193				
FT2C	003624	1191	1195#	1204				
FT2D	003640	1199#	1200					
FT2E	003670	1202	1206#					
FT2ER	003702	1185	1194	1205	1210#	1936		
FT2ERA	003732	1214	1217#					
FT2ERB	004004	1212	1228#					
FT2ERC	004014	1229	1231#					
FT2X	004024	1207	1234#					
FT20	010056	814	815	1781#				
FT20A	010134	1790#	1812					
FT20B	010220	1801#						
FT20C	010252	1806#						
FT20X	010306	1795	1807	1809	1813#			
FT21	010316	816	817	1819#				
FT21A	010454	1838#	1860					
FT21X	010636	1853	1855	1862#				
FT22	010646	818	819	1869#	1894			
FT22A	010722	1877#	1879					
FT22B	010742	1882#	1883					
FT22X	011042	1892	1896#					
FT23	011052	820	821	1903#	1928	1935		

\$CNTG	017713	2459	2644#																
\$ENDAD	003334	623	1131#																
\$MNEW	017726	2465	2646#																
\$MSWR	017717	2461	2645#																
\$QUEST	017736	2491	2647#																
\$READ	014124	2468	2471#																
\$SVPC =	001000	608#	629	608	610#	614#	618#	622#	625#	629#	636#	642#	654#	658#					
.	= 021456	598#	602	829#	2304	2548	2650#	2652#											
		665#	670#																

. ABS. 021456 000

ERRORS DETECTED: 0

.CZTUBH/CRF/NL:TOC=CZTUBH.P11
RUN-TIME: 2 5 1 SECONDS
RUN-TIME RATIO: 29/9=3.0
CORE USED: 12K (23 PAGES)