

TM02/TU16

BASIC FUNCTION TEST
MD-11-DZTUB-E

EP-DZTUB-E-DL-A

NOV 1976

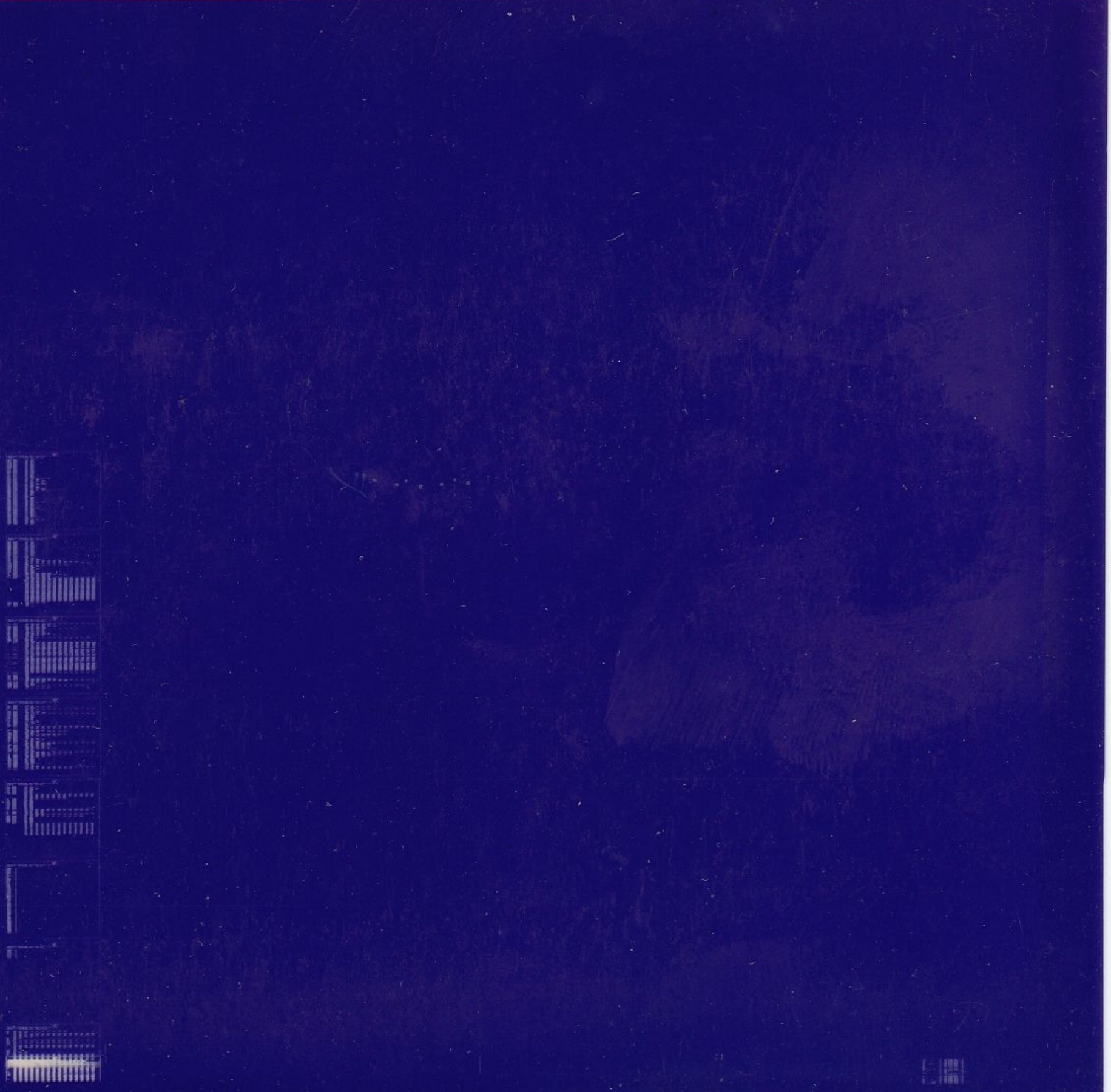
COPYRIGHT © 1976

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames, each containing test data for the MD-11-DZTUB-E. The data is organized into columns and rows, with some frames containing headers and footers. The frames are arranged in a grid that is approximately 10 columns wide and 15 rows high. The data in the frames is too small to read clearly but appears to be a series of test results or parameters.



.REM %

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTUB-E-D
PRODUCT TITLE: TM02/TU16 BASIC FUNCTION TEST
DATE CREATED: 21 APRIL 76
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: R.B. BARNES
REVISED: SUPPORTS SOFTWARE SWITCH REGISTER

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, 1976 BY DIGITAL EQUIPMEN CORPORATION

1102/TU16 BASIC FUNCTION TEST
DZTUBE.P11

TABLE OF CONTENTS

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

11-03-76 13:04

1009896
1009897
1009898
1009899
1009900
1009901
1009902
1009903
1009904
1009905
1009906
1009907
1009908
1009909
1009910
1009911
1009912
1009913
1009914
1009915
1009916
1009917
1009918
1009919
1009920
1009921
1009922

4.1 SAMPLE START AT 200(B): 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)
TMO2-TU16 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.)

123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160

5. 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.

161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187

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.

188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223

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

224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265

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: (SWO-SW4)

WHEN SWO-4 ARE SET TO ZERO (00) THE SCHEDULAR WILL EXECUTE ALL OF THE TESTS IN SEQUENCE (1-24). IF SWO-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 SWID 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.

(PAGE 5)

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 TMO2/TU16 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 TMO2 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. ****

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
307
308
309
310
311
312
313

(PAGE 6)

B. TMO2/TU16 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. 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.

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
364
365
366
367
368
369

LO1

370
371
372

9. END

(PAGE 7)

373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428

FT15: ERASE TEST:

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

1. REWIND TO BOT
2. ISSUE 200 ERASE 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 REFLECT
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

NO1

TMOZ/TU16 BASIC FUNCTION TEST MACY11 27(732) 11-OCT-76 13:04 PAGE 14
DZTUBE.P11

429
430

10. END

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
477
478
479
480
481
482
483
484
485
486

(PAGE 8)

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 TMO2 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

CO2

TMD2/TU16 BASIC FUNCTION TEST MACY11 27(732) 11-OCT-76 13:04 PAGE 16
OZTUBE.P11

487

488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518

(PAGE 9)

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

519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600

9. LISTING

%
.TITLE TM02/TU16 BASIC FUNCTION TEST
:MAINDEC-11-DZTUB-E-D
:21 SEPT 75
:R. BARNES
:REVISED APRIL 1976 BY S. CARPENTER
: 1) SUPPORTS SOFTWARE SWITCH REGISTER
: 2) SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
.ABS

:CONSOLE SWITCHES*****
:SW1(100000): 1=HALT ON ERROR
: 0=CONTINUE
:SW14(040000) 1=LOOP ON ERROR (SCOPE(040000) RH 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


```

593                                     ;REGISTER EQUIVS*****
594                                     ;
595         000000                       R0=%0
596         000001                       R1=%1
597         000002                       R2=%2
598         000003                       R3=%3
599         000004                       R4=%4
600         000005                       R5=%5
601         000006                       SP=%6
602         000007                       PC=%7
603
604                                     ;TRAP CATCHERS*****
605
606         000000                       .=0
607         000000 000002                 .+2
608         000002 000000                 HALT
609         000004 000006                 .+2
610         000006 000000                 HALT
611         000010 000012                 .+2
612         000012 000000                 HALT
613         000014 000016                 .+2
614         000016 000000                 HALT
615         000020 000022                 .+2
616         000022 000000                 HALT
617         000024 000026                 .+2
618         000026 000000                 HALT
619         000030 000032                 .+2
620         000032 000000                 HALT
621         000034 000036                 .+2
622         000036 000000                 HALT
623         000040 000042                 .+2
624         000042 000000                 HALT
625         000044 000046                 .+2
626         000046 000000                 HALT
627         000050 000052                 .+2
628         000052 000000                 HALT
629         000054 000056                 .+2
630         000056 000000                 HALT
631         000060 000062                 .+2
632         000062 000000                 HALT
633         000064 000066                 .+2
634         000066 000000                 HALT
635         000070 000072                 .+2
636         000072 000000                 HALT
637         000074 000076                 .+2
638         000076 000000                 HALT
639         000100 000102                 .+2
640         000102 000000                 HALT
641         000104 000106                 .+2
642         000106 000000                 HALT
643         000110 000112                 .+2
644         000112 000000                 HALT
645         000114 000116                 .+2
646         000116 000000                 HALT
647         000120 000122                 .+2
648         000122 000000                 HALT

```

649	000124	000126	.+2
650	000126	000000	HALT
651	000130	000132	.+2
652	000132	000000	HALT
653	000134	000136	.+2
654	000136	000000	HALT
655	000140	000142	.+2
656	000142	000000	HALT
657	000144	000146	.+2
658	000146	000000	HALT
659	000150	000152	.+2
660	000152	000000	HALT
661	000154	000156	.+2
662	000156	000000	HALT
663	000160	000162	.+2
664	000162	000000	HALT
665	000164	000166	.+2
666	000166	000000	HALT
667	000170	000172	.+2
668	000172	000000	HALT
669	000174	000176	.+2
670	000176	000000	HALT
671	000200	000202	.+2
672	000202	000000	HALT
673	000204	000206	.+2
674	000206	000000	HALT
675	000210	000212	.+2
676	000212	000000	HALT
677	000214	000216	.+2
678	000216	000000	HALT
679	000220	000222	.+2
680	000222	000000	HALT
681	000224	000226	.+2
682	000226	000000	HALT
683	000230	000232	.+2
684	000232	000000	HALT
685	000234	000236	.+2
686	000236	000000	HALT
687	000240	000242	.+2
688	000242	000000	HALT
689	000244	000246	.+2
690	000246	000000	HALT
691	000250	000252	.+2
692	000252	000000	HALT
693	000254	000256	.+2
694	000256	000000	HALT
695	000260	000262	.+2
696	000262	000000	HALT
697	000264	000266	.+2
698	000266	000000	HALT
699	000270	000272	.+2
700	000272	000000	HALT
701	000274	000276	.+2
702	000276	000000	HALT
703	000300	000302	.+2
704	000302	000000	HALT

705	000304	000306	.+2
706	000306	000000	HALT
707	000310	000312	.+2
708	000312	000000	HALT
709	000314	000316	.+2
710	000316	000000	HALT
711	000320	000322	.+2
712	000322	000000	HALT
713	000324	000326	.+2
714	000326	000000	HALT
715	000330	000332	.+2
716	000332	000000	HALT
717	000334	000336	.+2
718	000336	000000	HALT
719	000340	000342	.+2
720	000342	000000	HALT
721	000344	000346	.+2
722	000346	000000	HALT
723	000350	000352	.+2
724	000352	000000	HALT
725	000354	000356	.+2
726	000356	000000	HALT
727	000360	000362	.+2
728	000362	000000	HALT
729	000364	000366	.+2
730	000366	000000	HALT
731	000370	000372	.+2
732	000372	000000	HALT
733	000374	000376	.+2
734	000376	000000	HALT
735	000400	000402	.+2
736	000402	000000	HALT
737	000404	000406	.+2
738	000406	000000	HALT
739	000410	000412	.+2
740	000412	000000	HALT
741	000414	000416	.+2
742	000416	000000	HALT
743	000420	000422	.+2
744	000422	000000	HALT
745	000424	000426	.+2
746	000426	000000	HALT
747	000430	000432	.+2
748	000432	000000	HALT
749	000434	000436	.+2
750	000436	000000	HALT
751	000440	000442	.+2
752	000442	000000	HALT
753	000444	000446	.+2
754	000446	000000	HALT
755	000450	000452	.+2
756	000452	000000	HALT
757	000454	000456	.+2
758	000456	000000	HALT
759	000460	000462	.+2
760	000462	000000	HALT

761	000464	000466	.+2
762	000466	000000	HALT
763	000470	000472	.+2
764	000472	000000	HALT
765	000474	000476	.+2
766	000476	000000	HALT
767	000500	000502	.+2
768	000502	000000	HALT
769	000504	000506	.+2
770	000506	000000	HALT
771	000510	000512	.+2
772	000512	000000	HALT
773	000514	000516	.+2
774	000516	000000	HALT
775	000520	000522	.+2
776	000522	000000	HALT
777	000524	000526	.+2
778	000526	000000	HALT
779	000530	000532	.+2
780	000532	000000	HALT
781	000534	000536	.+2
782	000536	000000	HALT
783	000540	000542	.+2
784	000542	000000	HALT
785	000544	000546	.+2
786	000546	000000	HALT
787	000550	000552	.+2
788	000552	000000	HALT
789	000554	000556	.+2
790	000556	000000	HALT
791	000560	000562	.+2
792	000562	000000	HALT
793	000564	000566	.+2
794	000566	000000	HALT
795	000570	000572	.+2
796	000572	000000	HALT
797	000574	000576	.+2
798	000576	000000	HALT
799	000600	000602	.+2
800	000602	000000	HALT
801	000604	000606	.+2
802	000606	000000	HALT
803	000610	000612	.+2
804	000612	000000	HALT
805	000614	000616	.+2
806	000616	000000	HALT
807	000620	000622	.+2
808	000622	000000	HALT
809	000624	000626	.+2
810	000626	000000	HALT
811	000630	000632	.+2
812	000632	000000	HALT
813	000634	000636	.+2
814	000636	000000	HALT
815	000640	000642	.+2
816	000642	000000	HALT


```

873 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
874
875
876 ;*****
877
878 ;THIS PROGRAM SUPPORTS THE SOFTWARE SWITCH REGISTER LOC.176.
879 ;REFER TO SECTION 5 OF DOCUMENT FOR DESCRIPTION
880
881 ;*****
882 ;START ADDRESS*****
883
884 .=200
885 000200 005000 CLR RO
886 000202 000167 001372 JMP START ;PROGRAM START
887
888 .=210
889 000210 000240 NOP
890 000212 012700 000001 MOV #1,RO ;SET NO HEADER FLAG
891 000216 000167 001356 JMP START
892
893 ;TM02 INTERRUPT VECTOR*****
894
895 .=224
896 000224 012102 MTINT
897 000226 000340 ;TAPE INTERRUPT HANDLER ADDRESS
898

```

```

899
900          000600          . =600
901                                     ; MASS BUS REGISTER EQUIVS*****
902
903 000600 172440          CI:      172440
904 000602 172442          WC:      172442
905 000604 172444          BA:      172444
906 000606 172446          FC:      172446
907 000610 172450          CS:      172450
908 000612 172452          DS:      172452
909 000614 172454          ER:      172454
910 000616 172456          AS:      172456
911 000620 172460          CC:      172460
912 000622 172462          DB:      172462
913 000624 172464          MR:      172464
914 000626 172466          DT:      172466
915 000630 172470          SN:      172470
916 000632 172472          TC:      172472
917
918                                     ; CONSTANTS*****
919
920 000634 177776          PSW:      177776          ; PROCESSOR STATUS
921 000636 177570          SWR:      177570          ; SWITCH REGISTER
922 000640 177560          TKS:      177560          ; TTY READER STATUS
923 000642 177562          TKB:      177562          ; TTY READ BUFFER
924 000644 177564          TPS:      177564          ; TTY PUNCH STATUS
925 000646 177566          TPB:      177566          ; TTY PUNCH BUFFER
926 000650 177777          SERNUM: 177777          ; SERIAL NUMBER
927 000652 000011          DRVTP: 011             ; DRIVE TYPE
928 000654 000010          ITAMT: 10             ; ITERATION AMOUNT
929 000656 000224          VECT: 224             ; INTERRUPT VECTOR(RH)
930 000660 172440          REGS: 172440          ; STARTING REGISTER ADDRESS
931 000662 000004          BTRP: 4              ; BUS TRAP ADDRESS
932 000664 000006          BTRP2: 6             ; BUS TRAP PRIORITY LEVEL 7

```

```

933                                     ;FLAGS AND COUNTERS*****
934
935 000666 000000 TOB: 0
936 000670 000000 TIB: 0
937 000672 000000 RH17F: 0
938 000674 000000 HDRFL: 0
939 000676 000000 EMADDR: 0
940 000700 000000 DRVN: 0
941 000702 000000 SLVN: 0
942 000704 000000 BADDR: 0
943 000706 000000 FCNT: 0
944 000710 000000 WCNT: 0
945 000712 000000 RCNT: 0
946 000714 000000 ERRP: 0
947 000716 000000 ERRP1: 0
948 000720 000000 RRD: 0
949 000722 000000 RFD: 0
950 000724 000000 RDYDX: 0
951 000726 000000 OPDYX: 0
952 000730 000000 SCNT: 0
953 000732 000000 PFLG: 0
954 000734 000000 RTRN: 0
955 000736 000000 ERADD: 0
956 000740 000000 TEMP1: 0
957 000742 000000 TEMP2: 0
958 000744 000000 TEMP3: 0
959 000746 000000 STMSK: 0
960 000750 000000 ITCNT: 0
961 000752 000000 DSAV: 0
962 000754 000000 SAV1: 0
963 000756 000000 SAV2: 0
964 000760 000000 SAV3: 0
965 000762 000000 SCOLP: 0
966 000764 000000 ITRLP: 0
967 000766 000000 EXFL: 0
968 000770 000000 PEXFL: 0
969 000772 000000 STFLG: 0
970 000774 000000 LTADD: 0
971 000776 000000 FUN: 0
972 001000 000000 SERFL: 0
973 001002 000000 CRCNT: 0
974 001004 000000 UDES: 0
975 001006 000000 PATRN: 0
976 001010 000000 RHTF: 0
977 001012 000000 NRZOF: 0
978 001014 000000 RHOF: 0
979 001016 000000 PCNTR: 0
980 001020 000000 TEMPST: 0
981 001022 000000 COUNT: 0
982 001024 000000 RDSW: 0
983
984                                     ;DATA PATTERN GENERATORS*****
985
986 001026 000000 DATBL: 0
987 001030 011660 DATA0: DAT1 ;ALL ONE BITS
988 001032 011702 DATA1: DAT2 ;ALL ZERO BITS

```

989 001034 011710
990 001036 011720

DATA2: DWT3 :ALTERNATING ONE/ZERO BITS
DATA3: DAT4 :ALL BITS 0-377

;LOGIC TEST ENTRY TABLE*****

Line No.	Address	Value	Label
991			
992			
993			
994	001040	000000	TSTTBL: 0
995	001042	000000	0
996	001044	002774	FT1
997	001046	002774	FT1
998	001050	003074	FT2
999	001052	003074	FT2
1000	001054	003420	FT3
1001	001056	003420	FT3
1002	001060	003636	FT4
1003	001062	003636	FT4
1004	001064	003764	FT5
1005	001066	003764	FT5
1006	001070	004166	FT6
1007	001072	004166	FT6
1008	001074	004454	FT7
1009	001076	004454	FT7
1010	001100	004550	FT10
1011	001102	004550	FT10
1012	001104	004704	FT11
1013	001106	004704	FT11
1014	001110	005022	FT12
1015	001112	005022	FT12
1016	001114	005134	FT13
1017	001116	005134	FT13
1018	001120	005466	FT14
1019	001122	005466	FT14
1020	001124	006362	FT15
1021	001126	006362	FT15
1022	001130	006562	FT16
1023	001132	006562	FT16
1024	001134	007024	FT17
1025	001136	007024	FT17
1026	001140	007440	FT20
1027	001142	007440	FT20
1028	001144	007700	FT21
1029	001146	007700	FT21
1030	001150	010202	FT22
1031	001152	010202	FT22
1032	001154	010406	FT23
1033	001156	010406	FT23
1034	001150	010626	FT24
1035	001162	010626	FT24
1036	001164	000000	0
1037	001166	000000	0
1038	001170	000000	0
1039	001172	000000	0

```

001600      . = 1600
;PROGRAM START AND HOUSEKEEPING*****
1070      001600 000240      START:  NOP
1071      001602 012777 000340 177024  MOV    #340, @PSW      ;SET PRIORITY
1072      001610 012706 000500      MOV    #500, SP       ;SET STACK POINTER
1073      001614 013746 000006      SUSWR:  MOV    @#6, -(SP)      ;SAVE VECTORS
1074      001620 013746 000004      MOV    @#4, -(SP)
1075      001624 012737 001644 000004  MOV    #15, @#4       ;SET UP FOR TIMEOUT
1076      001632 022777 177777 176776  CMP    #-1, @SWR     ;REFERENCE HARDWARE SWITCH REGISTER
1077      001640 001402      BEQ    2$
1078      001642 000404      BR     3$
1079      001644 022626      1$:  CMP    (SP)+, (SP)+      ;ADJUST STACK
1080      001646 012767 000176 176762  2$:  MOV    #SWREG, SWR   ;POINT TO SOFTWARE SWITCH REG
1081      001654 012637 000004      3$:  MOV    (SP)+, @#4     ;RESTORE VECTORS
1082      001660 012637 000006      MOV    (SP)+, @#6
1083      001664 023727 000636 000176  CMP    @#SWR, #SWREG  ;IS SOFTWARE REG USED
1084      001672 001002      BNE    4$
1085      001674 004767 011452      JSR    PC, CNTLU     ;BRANCH IF NO
1086      001700 005700      4$:  TST    RO           ;ALLOW SOFTWARE SWITCH REGISTER TO BE CHANGED
1087      001702 001402      BEQ    5$           ;SEE IF PRINT HEADER
1088      001704 000167 000562      JMP    ST4          ;IF SO: BR
1089      001710 012704 013720      STOA:  MOV    #MSG3, R4 ;ELSE SKIP
1090      001714 004767 010564      JSR    PC, TTOUT    ;PRINT TITLE
1091      001720 012704 014025      STOB:  MOV    #MSG4, R4
1092      001724 004767 010554      JSR    PC, TTOUT    ;REQUEST REGISTER ADDRESS
1093      001730 016703 176724      MOV    REGS, R3
1094      001734 004767 010710      JSR    PC, OCTP     ;PRINT CURRENT ADDRESS
1095      001740 012705 000660      MOV    #REGS, R5   ;SET ADDRESS SAVE LOC
1096      001744 012701 000006      MOV    #6, R1      ;SET SIZE OF RESPONSE
1097      001750 012702 176400      MOV    #176400, R2 ;SET UPPER LIMIT
1098      001754 012703 172300      MOV    #172300, R3 ;SET LOWER LIMIT
1099      001760 004767 010262      JSR    PC, TTR      ;GO GET RESPONSE
1100      001764 012704 014050      MOV    #MSG5, R4
1101      001770 004767 010510      JSR    PC, TTOUT    ;REQUEST VECTOR
1102      001774 016703 176656      MOV    VECT, R3
1103      002000 004767 010644      JSR    PC, OCTP     ;PRINT CURRENT VECTOR
1104      002004 012705 000656      MOV    #VECT, R5   ;SET ADDRESS SAVE LOC
1105      002010 012701 000003      MOV    #3, R1      ;SET SIZE OF RESPONSE
1106      002014 012702 000224      MOV    #224, R2    ;SET UPPER LIMIT
1107      002020 012703 000150      MOV    #150, R3    ;SET LOWER LIMIT
1108      002024 004767 010216      JSR    PC, TTR      ;GO GET RESPONSE
1109      002030 016700 176622      MOV    VECT, RO    ;GET VECTOR
1110      002034 012720 012102      MOV    #MTINT, (RO)+ ;LOAD INTERRUPT ADDRESS IN VECTOR
1111      002040 012710 000340      MOV    #340, (RO)  ;LOAD PRIORITY
1112      002044 016700 176610      MOV    REGS, RO    ;GET START OF REGS
1113      002050 012701 000016      MOV    #16, R1     ;SET NUMBER OF REGS
1114      002054 012702 000600      MOV    #C1, R2     ;GET START OF TABLE
1115      002060 010022      STO:  MOV    RO, (R2)+ ;BUILD TABLE
1116      002062 062700 000002      ADD    #2, RO      ;BUMP ADDRESS
1117      002066 005301      DEC    R1          ;SEE IF DONE
1118      002070 001373      BNE    STO         ;IF NOT: BR
1119      002072 012702 000566      MOV    #TOB, R2
1120      002076 012700 000054      MOV    #54, RO
1121      002102 005022      ST1:  CLR    (R2)+    ;CLEAR FLAGS + COUNTERS
    
```

1096	002104	005300			DEC	R0	
1097	002106	001375			BNE	ST1	
1098	002110	012767	000001	176672	MOV	#1,RHTF	;SET ADDRESS TEST FLAG
1099	002116	000167	000376		JMP	TSRH	;GO DO INITIAL ADDRESS TEST PASS
1100	002122	012704	014131		MOV	#MSG10,R4	
1101	002126	004767	010352		JSR	PC,TTOUT	;REQUEST DRIVE NUMBER
1102	002132	012705	000700		MOV	#DRVN,R5	;SET ADDRESS OF DRIVE NUMBER SAVE
1103	002136	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
1104	002142	012702	000007		MOV	#7,R2	;SET UPPER LIMIT
1105	002146	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
1106	002152	004767	010070		JSR	PC,TTR	;GO GET RESPONSE
1107	002156	012777	000040	176424	MOV	#40,ACS	;SET INIT
1108	002164	056777	176510	176416	BIS	DRVN,ACS	;SET DRIVE NUMBER
1109	002172	005777	176402		TST	ACS	;ACCESS DRIVE
1110	002176	032777	010000	176404	BIT	#10000,ACS	;SEE IF NED
1111	002204	001405			BEQ	ST2	;IF NOT: BR
1112	002206	012704	015063		MOV	#MSG41,R4	
1113	002212	004767	010266		JSR	PC,TTOUT	;PRINT NOT AVAIL
1114	002216	000741			BR	ST1A	;REDO DRIVE REQUEST
1115	002220	012704	014151		MOV	#MSG11,R4	
1116	002224	004767	010254		JSR	PC,TTOUT	;REQUEST SLAVE NUMBER
1117	002230	012705	000702		MOV	#SLVN,R5	;SET ADDRESS OF SLAVE SAVE
1118	002234	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
1119	002240	012702	000007		MOV	#7,R2	;SET UPPER LIMIT
1120	002244	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
1121	002250	004767	007772		JSR	PC,TTR	;GO GET RESPONSE
1122	002254	012777	000040	176326	MOV	#40,ACS	;INIT
1123	002262	056777	176412	176320	BIS	DRVN,ACS	;SET DRIVE NUMBER
1124	002270	016777	176406	176334	MOV	SLVN,ACS	;LOAD SLAVE NUMBER
1125	002276	032777	002000	176322	BIT	#2000,ACS	;SEE IF SLAVE PRESENT
1126	002304	001005			BNE	ST3	;IF SO: BR
1127	002306	012704	015104		MOV	#MSG42,R4	
1128	002312	004767	010166		JSR	PC,TTOUT	;PRINT NON-EXIST SLAVE
1129	002316	000740			BR	ST2	;REDO SLAVE REQUEST
1130	002320	012704	015125		MOV	#MSG43,R4	
1131	002324	004767	010154		JSR	PC,TTOUT	;PRINT SERIAL NUMBER TAG
1132	002330	017703	176274		MOV	ACS,R3	
1133	002334	004767	010636		JSR	PC,SNPT	;PRINT SERIAL NUMBER
1134	002340	012704	015707		MOV	#MSG61,R4	
1135	002344	004767	010134		JSR	PC,TTOUT	;REQUEST RH11 OR RH70
1136	002350	012705	000672		MOV	#RH17F,R5	;GET ADDRESS OF FLAG
1137	002354	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
1138	002360	012702	000001		MOV	#1,R2	;SET UPPER LIMIT
1139	002364	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
1140	002370	004767	007652		JSR	PC,TTR	;GET RESPONSE
1141	002374	012704	015727		MOV	#MSG62,R4	
1142	002400	004767	010100		JSR	PC,TTOUT	;REQUEST RH11 ONLY RESPONSE
1143	002404	012705	001014		MOV	#RH0F,R5	;SET FLAG ADDRESS
1144	002410	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
1145	002414	012702	000001		MOV	#1,R2	;SET UPPER LIMIT
1146	002420	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
1147	002424	004767	007616		JSR	PC,TTR	;GO GET RESPONSE
1148	002430	005767	176360		TST	RH0F	;SEE IF RH11 ONLY
1149	002434	001016			BNE	ST4	;IF SO: BR
1150	002436	012704	015600		MOV	#MSG55,R4	
1151	002442	004767	010036		JSR	PC,TTOUT	;REQUEST NRZ ONLY RESPONSE

1152	002446	012705	001012	MOV	#NRZOF,R5	:SET FLAG ADDRESS
1153	002452	012701	000001	MOV	#1,R1	:SET SIZE OF RESPONSE
1154	002456	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
1155	002462	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
1156	002466	004767	007554	JSR	PC,TTR	:GO GET RESPONSE
1157	002472	005067	176320	ST4: CLR	PCNTR	:CLEAR PASS COUNTER


```

1158                                     ; TEST SCHEDULAR*****
1159
1160 002476 000240          TSCD:  NOP
1161 002500 005067 176266      CLR      STFLG          ;CLEAR SINGLE TEST FLAG
1162 002504 017700 176126      MOV      @SWR,RO
1163 002510 042700 177740      BIC      #177740,RO
1164 002514 005700          TST      RO
1165 002516 001055          BNE      STSCD          ;GO SELECT SINGLE TEST
1166 002520 012767 001040 176246  TSRH:  MOV      #TSTTBL,LTADD
1167 002526 062767 000004 176240  TSCD0:  ADD      #4,LTADD
1168 002534 016767 176234 176222      MOV      LTADD,ITRLP
1169 002542 062767 000002 176214      ADD      #2,ITRLP      ;SET ITERATION ADDRESS
1170 002550 005777 176220          TST      @LTADD
1171 002554 001002          BNE      TSCD1
1172 002556 000167 000144          JMP      TEND          ;GO TO END ROUTINE
1173 002562 000240          TSCD1:  NOP
1174 002564 005067 176156      CLR      STMSK
1175 002570 005067 176120      CLR      ERFP
1176 002574 005067 176074      CLR      HDRFL        ;CLEAR PRINT HEADER FLAG
1177 002600 017700 176170      MOV      @LTADD,RO    ;SET POINTER TO TEST
1178 002604 000110          JMP      (RO)         ;GO TO TEST
1179 002606 000240          TSCD2:  NOP
1180 002610 032777 002000 176020      BIT      #2000,@SWR   ;SEE IF HALT ON TEST
1181 002616 001401          BEQ      TSCD3        ;IF NOT: BR
1182 002620 000000          HALT
1183 002622 004767 010452          TSCD3:  JSR      PC,CKSWR          ;CHECK FOR CNTL G
1184 002626 000240          NOP
1185 002630 005767 176136      TST      STFLG        ;SE IF SINGLE TEST
1186 002634 001734          BEQ      TSCD0        ;IF NOT: BR
1187 002636 017700 175774      MOV      @SWR,RO
1188 002642 042700 177740      BIC      #177740,RO   ;MASK TEST NUMBER
1189 002646 005700          TST      RO           ;SEE IF RETURN TO ALL
1190 002650 001712          BEQ      TSCD         ;IF SO: BR
1191 002652 000240          STSCD:  NOP
1192 002654 012767 000001 176110      MOV      #1,STFLG    ;SET SINGLE TEST FLAG
1193 002662 022700 000025          CMP      #25,RO      ;SEE IF EXCEEDED TESTS
1194 002666 003417          BLE      TEND        ;IF SO: BR
1195 002670 000241          CLC
1196 002672 006100          ROL      RO
1197 002674 006100          ROL      RO           ;SET TABLE MODIFIER
1198 002676 012767 001040 176070      MOV      #TSTTBL,LTADD
1199 002704 060067 176064          ADD      RO,LTADD    ;SET TEST POINTER
1200 002710 016767 176060 176046      MOV      LTADD,ITRLP
1201 002716 062767 000002 176040      ADD      #2,ITRLP    ;SET ITERATION POINTER
1202 002724 000716          BR      TSCD1
1203 002726 012704 014063          TEND:  MOV      #MSG6,R4
1204 002732 004767 007546          JSR      PC,TOUT     ;PRINT END OF PASS
1205 002736 016703 176054          MOV      PCNTR,R3
1206 002742 004767 007702          JSR      PC,OCIP     ;PRINT PASS NUMBER
1207 002746 032777 004000 175662      BIT      #4000,@SWR  ;SEE IF HALT ON PASS
1208 002754 001001          BNE      TENDX       ;IF NOT: BR
1209 002756 000000          HALT
1210 002760 004767 010314          TENDX:  JSR      PC,CKSWR          ;CHECK FOR CNTL G
1211 002764 005267 176026          INC      PCNTR       ;BUMP PASS COUNTER
1212 002770 000167 177502          JMP      TSCD        ;RESTART
    
```

```

1213
1214
1215
1216 002774 012767 015742 175674 FT1: MOV #MSFT1,EMADDR ;SET HEADER
1217 003002 012777 012126 175652 MOV #TRAP,JBTRP ;SET TRAP HANDLER ADDRESS
1218 003010 012777 000340 175646 MOV #340,JBTRP2
1219 003016 012700 000016 MOV #16,R0 ;SET NUMBER OF REGISTER
1220 003022 016701 175552 MOV C1,R1 ;GET FIRST ADDRESS (C51)
1221 003026 005711 FT1A: TST (R1) ;REFERENCE REGISTER
1222 003030 000240 NOP ;IF ADDRESS IS BAD, BLS TRAP WILL OCCUR
1223 003032 005300 FT1B: DEC R0 ;SEE IF DONE ALL
1224 003034 001403 BEQ FT1X ;IF SO: BR
1225 003036 062701 000002 ADD #2,R1 ;BUMP ADDRESS POINTER
1226 003042 000771 BR FT1A ;CONTINUE
1227 003044 002777 000006 175610 FT1X: MOV #6,JBTRP ;RESET TRAP CATCHER
1228 003052 005767 175732 TST RHTF ;SEE IF INITIAL ADDRESS TEST PASS
1229 003056 001404 BEQ FT1XX ;IF NOT: BR
1230 003060 005067 175724 CLR RHTF ;CLEAR FLAG
1231 003064 000167 177032 JMP ST1A ;RETURN
1232 003070 000167 177512 FT1XX: JMP TSCD2 ;RETURN TO SCHEDULAR
    
```

```

1233
1234
1235
1236 003074 012767 015767 175574 FT2:  MOV      #MSFT2,EMADDR ;SET TEST HEADER
1237 003102 012701 177777          MOV      #-1,R1 ;SET ALL ONES PATTERN
1238 003106 004767 006742          FT2A:  JSR      PC,INIT1 ;GO INIT
1239 003112 016700 175464          MOV      WC,R0 ;GET ADDRESS OF WORD COUNT
1240 003116 010102          MOV      R1,R2 ;SET EXPT REGISTER BIT PATTERN
1241 003120 010110          MOV      R1,(R0) ;LOAD PATTERN
1242 003122 021002          CMP      (R0),R2 ;SEE IF EXPT=RCVD
1243 003124 001410          BEQ      FT2B ;IF SO: BR
1244 003126 012767 014411 175602          MOV      #MSG25,ERADD ;SET CODE
1245 003134 012767 003106 175620          MOV      #FT2A,SCOLP ;SET SCOPE
1246 003142 004767 000116          JSR      PC,FT2ER ;GO DO ERROR
1247 003146 016700 175432          FT2B:  MOV      BA,R0 ;GET ADDRESS OF BUS ADDRESS
1248 003152 010102          MOV      R1,R2
1249 003154 042702 000001          BIC      #1,R2 ;SET EXPT PATTERN
1250 003160 010110          MOV      R1,(R0) ;LOAD PATTERN
1251 003162 020210          CMP      R2,(R0) ;SEE IF EXPT=RCVD
1252 003164 001410          BEQ      FT2C ;IF SO: BR
1253 003166 012767 014417 175542          MOV      #MSG26,ERADD ;SET ERROR CODE
1254 003174 012767 003146 175560          MOV      #FT2B,SCOLP ;SET SCOPE ADDRESS
1255 003202 004767 000056          JSR      PC,FT2ER ;GO DO ERROR
1256 003206 016700 175410          FT2C:  MOV      DB,R0 ;GET ADDRESS OF DATA BUFFER
1257 003212 010102          MOV      R1,R2
1258 003214 010110          MOV      R1,(R0) ;LOAD PATTERN
1259 003216 012703 004000          MOV      #4000,R3
1260 003222 005303          FT2D:  DEC      R3 ;DELAY
1261 003224 001376          BNE      FT2D
1262 003226 020210          CMP      R2,(R0) ;SEE IF EXPT=RCVD
1263 003230 001410          BEQ      FT2E ;IF SO: BR
1264 003232 012767 014425 175476          MOV      #MSG27,ERADD ;SET ERROR CODE
1265 003240 012767 003206 175514          MOV      #FT2C,SCOLP ;SET SCOPE ADDRESS
1266 003246 004767 000012          JSR      PC,FT2ER ;GO DO ERROR
1267 003252 005701          FT2E:  TST      R1 ;SEE IF DONE RESET
1268 003254 001454          BEQ      FT2X ;IF SO: BR
1269 003256 005001          CLR      R1 ;SET ZERO PATTERN
1270 003260 000167 177622          JMP      FT2A ;DO ZERO BITS
1271 003264 000240          FT2ER: NOP
1272 003266 032777 020000 175342          BIT      #20000,JSWR ;SEE IF PRINT ERROR
1273 003274 001034          BNE      FT2ERB ;IF NOT: BR
1274 003276 005767 175372          TST      HDRFL ;SEE IF DONE HEADER
1275 003302 001004          BNE      FT2ERA ;IF SO: BR
1276 003304 016704 175366          MOV      EMADDR,R4
1277 003310 004767 007170          JSR      PC,TTOUT ;DO HEADER
1278 003314 012767 000001 175352          FT2ERA: MOV      #1,HDRFL ;SET FLAG
1279 003322 016704 175410          MOV      ERADD,R4
1280 003326 004767 007152          JSR      PC,TTOUT ;PRINT ERROR CODE
1281 003332 012704 014355          MOV      #MSG22,R4
1282 003336 004767 007142          JSR      PC,TTOUT ;PRINT EXPT TAG
1283 003342 010103          MOV      R1,R3
1284 003344 004767 007266          JSR      PC,OCTPE ;PRINT EXPT
1285 003350 012704 014365          MOV      #MSG23,R4
1286 003354 004767 007124          JSR      PC,TTOUT ;PRINT RCVD TAG
1287 003360 011003          MOV      (R0),R3
1288 003362 004767 007250          JSR      PC,OCTPE ;PRINT RCVD
    
```

1289	003366	005777	175244	FT2ERB:	TST	JSWR	;SEE IF HALT ON ERROR
1290	003372	100001			BPL	FT2ERC	;IF NOT: BR
1291	003374	000000			HALT		
1292	003376	004767	006340	FT2ERC:	JSR	PC,SCOPE	;GO SEE IF SCOPE ON ERROR
1293	003402	000240			NOP		
1294	003404	000207			RTS	PC	;IF NO SCOPE: CONTINUE TEST
1295	003406	000240		FT2X:	NOP		
1296	003410	004767	006370		JSR	PC,ITER	;GO SEE IF ITERATIONS
1297	003414	000167	177166		JMP	TSC02	;RETURN TO SCHEDULAR

```

1298
1299
1300 ;RM INITIALIZE TEST*****
1301 003420 012767 016024 175250 FT3: MOV #MSFT3,EMADDR ;SET TEST HEADER
1302 003426 012767 003420 175326 MOV #FT3,SCOLP
1303 003434 004767 006414 JSR PC,INIT1 ;GO INIT
1304 003440 052777 020000 175142 BIS #20000,@CS ;FORCE UPE =1
1305 003446 000240 NOP
1306 003450 004767 006400 JSR PC,INIT1 ;GO INIT
1307 003454 005777 175120 TST @C1 ;SEE IF SC IS RESET
1308 003460 100005 BPL FT3A ;IF SO: BR
1309 003462 012767 014463 175246 MOV #MSG29,ERADD ;SET ERROR CODE
1310 003470 004767 000060 JSR PC,FT3ER ;GO DO ERROR
1311 003474 032777 040000 175076 FT3A: BIT #40000,@C1 ;SEE IF TRE IS RESET
1312 003502 001405 BEQ FT3B ;IF SO: BR
1313 003504 012767 014512 175224 MOV #MSG30,ERADD ;SET ERROR CODE.
1314 003512 004767 000036 JSR PC,FT3ER ;GO DO ERROR
1315 003516 017701 175066 FT3B: MOV @CS,R1 ;GET CS2
1316 003522 042701 000307 BIC #307,R1 ;MARK IR/OR
1317 003526 005701 TST R1 ;SEE IF RESET
1318 003530 001405 BEQ FT3X ;IF SO: BR
1319 003532 012767 014542 175176 MOV #MSG31,ERADD ;SET ERROR CODE
1320 003540 004767 000010 JSR PC,FT3ER ;GO DO ERROR
1321 003544 004767 006234 FT3X: JSR PC,ITER ;GO SEE IF ITERATION
1322 003550 000167 177032 JMP TSCD2 ;RETURN TO SCHEDULAR
1323 003554 032777 020000 175054 FT3ER: BIT #20000,@SWR ;SEE IF PRINT ERROR
1324 003562 001015 BNE FT3ERB ;IF NOT: BR
1325 003564 005767 175104 TST HDRFL ;SEE IF DONE HEADER
1326 003570 001006 BNE FT3ERA ;IF SO: BR
1327 003572 016704 175100 MOV EMADDR,R4
1328 003576 004767 006702 JSR PC,TTOUT ;PRINT HEADER
1329 003602 005267 175066 INC HDRFL
1330 003606 016704 175124 FT3ERA: MOV ERADD,R4
1331 003612 004767 006666 JSR PC,TTOUT ;PRINT ERROR CODE
1332 003616 005777 175014 FT3ERB: TST @SWR ;SEE IF HALT ON ERROR
1333 003622 100001 BPL FT3ERC ;IF NOT: BR
1334 003624 000000 HALT
1335 003626 000240 FT3ERC: NOP
1336 003630 004767 006106 JSR PC,SCOPE ;GO SEE IF SCOPE
1337 003634 000207 RTS PC ;IF NOT: BR

```

```

1338
1339
1340 ;RH11 SILO TEST 1: EPMTY SILO READ*****
1341 003636 005767 175030 FT4: TST RH17F
1342 003642 001141 BNE FTSX ;IF RH70: BR
1343 003644 012767 016056 175024 MOV #MSFT4,EMADDR ;SET TEST TEST HEADER
1344 003652 012777 000040 174730 MOV #40,ACS ;INIT
1345 003660 017700 174736 MOV QDB,RO ;READ DB
1346 003664 005777 174720 TST ACS ;SEE IF DLT IS SET
1347 003670 100013 BPL FT4ER ;IF NOT: BR
1348 003672 005777 174702 TST QCI ;SEE IF SC IS SET
1349 003676 100014 BPL FT4ERA ;IF NOT: BR
1350 003700 032777 040000 174672 BIT #40000,QCI ;SEE IF TRE IS SET
1351 003706 001414 BEQ FT4ERB ;IF NOT: BR
1352 003710 004767 006070 FT4X: JSR PC,ITER ;GO SEE IF ITERATION
1353 003714 000157 176666 JMP TSCD2 ;RETURN TO SCHEDULAR
1354 003720 012767 014572 175010 FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE
1355 003726 000407 BR FT4ERC
1356 003730 012767 014610 175000 FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE
1357 003736 000403 BR FT4ERC
1358 003740 012767 014625 174770 FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.
1359 003746 000240 FT4ERC: NOP
1360 003750 012767 003636 175004 MOV #FT4,SCOLP ;SET SCOPE ADDRESS
1361 003756 004767 177572 JSR PC,FT3ER ;GO PRINT ERROR
1362 003762 000752 BR FT4X
  
```

```

1363
1364
1365
1366 003764 005767 174702          FT5:  TST      RH17F          ;SEE IF RH70
1367 003770 001066                    BNE      FT5X          ;IF SO: BR
1368 003772 012767 016106 174676    MOV      #MSG35,EMADDR ;SET TEST HEADER
1369 004000 012767 004006 174754    MOV      #FT5A,$COLP   ;SET SCOPE ADDRESS
1370 004006 004767 006042          FT5A:  JSR      PC,INIT1  ;GO INIT
1371 004012 032777 000100 174570    BIT      #100,$CS      ;SEE IF IR IS SET
1372 004020 001005                    BNE      FT5B          ;IF SO: BR
1373 004022 012767 014643 174706    MOV      #MSG35,ERADD  ;SET ERROR CODE
1374 004030 004767 000122          JSR      PC,FT5ER      ;GO DO ERROR
1375 004034 032777 000200 174546    FT5B:  BIT      #200,$CS ;SEE IF OR IS RESET
1376 004042 001405                    BEQ      FT5C          ;IF SO: BR
1377 004044 012767 014670 174664    MOV      #MSG36,ERADD  ;SET ERROR CODE
1378 004052 004767 000100          JSR      PC,FT5ER      ;GO DO ERROR
1379 004056 012777 000000 174536    FT5C:  MOV      #0,$DB    ;LOAD ZERO INTO SILO
1380 004064 032777 000200 174516    BIT      #200,$CS      ;SEE THAT OR RESET
1381 004072 001405                    BEQ      FT5D          ;IF IT DOES: BR
1382 004074 012767 014717 174634    MOV      #MSG37,ERADD  ;SET ERROR CODE
1383 004102 004767 000050          JSR      PC,FT5ER      ;GO DO ERROR
1384 004106 012777 177777 174506    FT5D:  MOV      #-1,$DB   ;LOAD SILO WITH -1
1385 004114 012700 004000          MOV      #4000,R0
1386 004120 032777 000200 174462    FT5E:  BIT      #200,$CS ;SEE IF OR IS SET
1387 004126 001007                    BNE      FT5X          ;IF SO: BR
1388 004130 005300                    DEC      R0
1389 004132 001372                    BNE      FT5E          ;AWAIT OR
1390 004134 012767 014717 174574    MOV      #MSG37,ERADD  ;SET ERROR CODE
1391 004142 004767 000010          JSR      PC,FT5ER      ;GO DO ERROR
1392 004146 004767 005632          FT5X:  JSR      PC,ITER  ;GO SEE IF ITERATION
1393 004152 000167 176430          JMP      TSCD2         ;RETURN TO SCHEDULAR
1394 004156 004767 177372          FT5ER: JSR      PC,FT3ER  ;GO PRINT ERROR
1395 004162 000240                    NOP
1396 004164 000207                    RTS      PC            ;CONTINUE TEST
    
```

```

1397
1398
1399
1400 004166 005767 174500 FT6: TST RH17F
1401 004172 001052 BNE FT6X ;IF RH70: BR
1402 004174 012767 016136 174474 MOV #MSGT6,EMADDR ;SET TEST HEADER
1403 004202 012767 004210 174552 MOV #FT6A,SCOLP ;SET SCOPE ADDRESS
1404 004210 004767 005640 FT6A: JSR PC,INIT1 ;GO INIT
1405 004214 005000 CLR RO ;PRESET DATA
1406 004216 010077 174400 FT6B: MOV RO,ADB ;LOAD SILO
1407 004222 005200 INC RO ;BUMP DATA
1408 004224 022700 000102 CMP #102,RO ;SEE IF FILLED ALL
1409 004230 001372 BNE FT6B ;IF NOT: BR
1410 004232 032777 000100 174350 BIT #100,ACS ;SEE IF IR IS RESET.
1411 004240 001405 BEQ FT6C ;IF SO: BR
1412 004242 012767 015030 174466 MOV #MSG40,ERADD ;SET ERROR CODE
1413 004250 004767 000054 JSR PC,FT6ER ;GO DO ERROR
1414 004254 032777 000200 174326 FT6C: BIT #200,ACS ;SEE IF OR IS SET
1415 004262 001005 BNE FT6D ;IF SO: BR
1416 004264 012767 014756 174444 MOV #MSG38,ERADD ;SET ERROR CODE
1417 004272 004767 000032 JSR PC,FT6ER ;GO DO ERROR
1418 004276 005000 FT6D: CLR RO ;PRESET DATA
1419 004300 017701 174316 FT6E: MOV ADB,R1 ;READ SILO
1420 004304 020001 CMP RO,R1 ;SEE IF EXPT=RCVD
1421 004306 001014 BNE FT6DE ;IF NOT: BR
1422 004310 005200 INC RO ;BUMP DATA
1423 004312 022700 000102 CMP #102,RO ;SEE IF DONE ALL
1424 004316 001370 BNE FT6E ;IF NOT: BR
1425 004320 004767 005460 FT6X: JSR PC,ITER ;GO SEE IF ITERATION
1426 004324 000167 176256 JMP TSCD2 ;RETURN TO SCHEDULAR
1427 004330 000240 FT6ER: NOP
1428 004332 004767 177216 JSR PC,FT3ER ;GO PRINT ERROR
1429 004336 000207 RTS PC ;RETURN
1430 004340 000240 FT6DE: NOP
1431 004342 032777 020000 174266 BIT #20000,ASWR ;SEE IF PRINT ERROR
1432 004350 001032 BNE FT6DEB ;IF NOT: BR
1433 004352 005767 174316 TST HDRFL ;SEE IF DONE HEADER
1434 004356 016701 174314 MOV EMADDR,R1
1435 004362 004767 006116 JSR PC,TTOUT ;PRINT HEADER
1436 004366 005267 174302 INC HDRFL ;SET FLAG
1437 004372 012704 015010 FT6DEA: MOV #MSG39,R4
1438 004376 004767 006102 JSR PC,TTOUT ;PRINT SILO READ ERROR
1439 004402 012704 014355 MOV #MSG22,R4
1440 004406 004767 006072 JSR PC,TTOUT ;PRINT EXPT TAG
1441 004412 010003 MOV RO,R3
1442 004414 004767 006230 JSR PC,OCTP ;PRINT EXPT
1443 004420 012704 014365 MOV #MSG23,R4
1444 004424 004767 006054 JSR PC,TTOUT ;PRINT RCVD TAG
1445 004430 010103 MOV R1,R3
1446 004432 004767 006212 JSR PC,OCTP ;PRINT RCVD
1447 004436 005777 174174 FT6DEB: TST ASWR ;SEE IF HALT ON ERROR
1448 004442 100001 BPL FT6DEX ;IF NOT: BR
1449 004444 000000 HALT
1450 004446 004767 006626 FT6DEX: JSR PC,CKSWR ;CHECK FOR CNTL G
1451 004452 000207 RTS PC ;RETURN TO TEST

```



```

1450
1451
1452
1453
1454
1455 004454 005767 174212 FT7: TST RH17F
1456 004460 001021 BNE FT7X ;IF RH70: BR
1457 004462 012767 016166 174206 MOV #MSFT7,EMADDR ;SET TEST HEADER
1458 004470 012767 004454 174264 MOV #FT7,SCOLP ;SET SCOPE ADDRESS
1459 004476 004767 005352 JSR PC,INITI ;GO INIT
1460 004502 012700 000103 MOV #103,RO ;SET SIZE OF SILO +1
1461 004506 010077 174110 FT7A: MOV RO,ROB ;LOAD SILO
1462 004512 005300 DEC RO ;SEE IF DONE
1463 004514 001374 BNE FT7A ;IF NOT: BR
1464 004516 005777 174066 TST @CS ;SEE IF DLT IS SET
1465 004522 100004 BPL FT7ER ;IF NOT: BR
1466 004524 004767 005254 FT7X: JSR PC,ITER ;GO SEE IF ITERATION
1467 004530 000167 176052 JMP TSCD2 ;RETURN TO SCHEDULAR
1468 004534 012767 014572 174174 FT7ER: MOV #MSG32,ERADD ;SET ERROR CODE
1469 004542 004767 177006 JSR PC,FT3ER ;GO DO ERROR
1470 004546 000766 BR FT7X

```

```

1471
1472 ;RH11 SILO TEST 5: SILO RESET*****
1473
1474 004550 005767 174116 FT10: TST RH17F
1475 004554 00.034 BNE FT10X ;IF RH70: BR
1476 004556 012767 016216 174112 MOV #MSFT10,EMADDR ;SET TEST HEADER
1477 004564 012767 004550 174170 MOV #FT10,SCOLP ;SET SCOPE ADDRESS
1478 004572 012777 000040 174010 MOV #40,ACS ;INITIALIZE
1479 004600 012700 000004 MOV #4,R0 ;SET NUMBER OF SILO WRITER
1480 004604 010077 174012 FT10A: MOV R0,ADB ;WRITE SILO
1481 004610 005300 DEC R0 ;SEE IF DONE
1482 004612 001374 BNE FT10A ;IF NOT: BR
1483 004614 052777 000040 173766 BIS #40,ACS ;INITIALIZE
1484 004622 012777 177777 173772 MOV #-1,ADB ;WRITE SILO
1485 004630 017701 173766 MOV ADB,R1 ;READ SILO 1
1486 004634 017701 173762 MOV ADB,R1 ;READ SILO 2
1487 004640 005777 173744 TST ACS ;SEE IF DLT IS SET
1488 004644 100011 EPL FT10ER ;IF NOT: BR
1489 004646 004767 005132 FT10X: JSR PC,ITER ;GO SEE IF ITERATION
1490 004652 005767 174136 TST RH0F ;SEE IF RH11 ONLY
1491 004656 001402 BEQ FT10XX ;IF NOT: BR
1492 004660 000167 176042 JMP TEND ;ELSE GO TO END
1493 004664 000167 175716 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR
1494 004670 012767 014572 174040 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE
1495 004676 004767 176652 JSR PC,FT3ER ;GO DO ERROR
1496 004702 000761 BR FT10X

```

```

1497                                     ;NOP TEST*****
1498
1499 004704 000240 FT11: NOP
1500 004706 012767 004704 174046 MOV #FT11,SCOLP ;SET SCOPE ADDRESS
1501 004714 004767 005134 JSR PC,INIT1
1502 004720 012767 000300 174056 MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
1503 004726 012767 177777 173752 MOV #-1,FCNT ;SET FC= ALL OVER
1504 004734 012767 177777 173746 MOV #-1,WCNT ;SET WC= ALL OVER
1505 004742 012767 177777 173734 MOV #-1,BADDR ;SET BA= ALL OVER
1506 004750 012767 000001 173746 MOV #1,RDYDX ;SET DELAY
1507 004756 012767 000001 173742 MOV #1,OPDYX ;SET OP DELAY
1508 004764 012767 000001 174004 MOV #1,FUN ;SET NOP FUNCTIONS CODE
1509 004772 004767 003732 JSR PC,EXEC ;GO EXECUTE COMMAND
1510 004776 000240 NOP
1511 005000 012767 016247 173670 MOV #MSFT11,EMADDR
1512 005006 004767 004146 JSR PC,ERCHK ;GO CHECK REGISTER
1513 005012 004767 004766 JSR PC,ITER ;GO SEE IF ITERATIONS
1514 005016 000167 175554 JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

1515                                     ;REWIND TEST*****
1516
1517 005022 000240                                     FT12: NOP
1518 005024 012767 005022 173730                   MOV      #FT12,SCOLP
1519 005032 004767 005016                                     JSR      PC,INIT1      ;GO INITIALIZE
1520 005036 052777 001700 173566                   BIS      #1700,ATC     ;SET TO NRZ,NORMAL
1521 005044 012767 177760 173634                   MOV      #-20,FCNT    ;SET FC=20
1522 005052 012767 177770 173630                   MOV      #-10,WCNT    ;SET WC=10
1523 005060 012767 016726 173616                   MOV      #WDATA,BADDR ;SET BA=WRITE BUFFER
1524 005066 012767 000007 173702                   MOV      #7,FUN       ;SET REWIND OP CODE
1525 005074 004767 003630                                     JSR      PC,EXEC       ;GO EXECUTE COMMAND
1526 005100 000240
1527 005102 032777 020000 173502 FT12A: BIT      #20000,ADS
1528 005110 001374                                     BNE     FT12A          ;AWAIT PIP
1529 005112 012767 016267 173556                   MOV      #MSFT12,EMADDR
1530 005120 004767 004034                                     JSR      PC,ERCHK     ;GO CHECK FOR ERROR
1531 005124 004767 004654                                     JSR      PC,ITER      ;GO SEE IF ITERATION
1532 005130 000167 175452                                     JMP      TSCD2        ;RETURN TO SCHEDULAR
1533

```

```

1534                                     ;WRITE/READ TEST*****
1535
1536 005134 000240 FT13: NOP
1537 005136 012767 000001 173560 MOV #1, RDYDX
1538 005144 012767 000001 173554 MOV #1, OPDYX
1539 005152 012767 000100 173532 MOV #100, RCNT ;SET RECORD COUNT
1540 005160 012767 016312 173510 MOV #MSFT13, EMADDR ;SET TEST HEADER
1541 005166 012767 000001 173612 MOV #1, PATRN
1542 005174 004767 004366 JSR PC, DSUP ;SET UP ALL ONES DATA PATTERN
1543 005200 012767 000300 173576 MOV #300, UDES ;REWIND TO BOT
1544 005206 004767 003650 FT13A: JSR PC, RWND ;SET 200 BPI, NORMAL
1545 005212 012767 177600 173466 MOV #-200, FCNT ;SET FC
1546 005220 012767 177700 173462 MOV #-100, WCNT ;SET WC
1547 005226 012767 016726 173450 MOV #WDATA, BADDR ;SET BA
1548 005234 012767 000061 173534 MOV #61, FUN ;SET WRITE OP-CODE
1549 005242 012767 014171 173444 MOV #MSG12, ERRP
1550 005250 004767 003454 FT13B: JSR PC, EXEC ;GO EXECUTE COMMAND
1551 005254 005067 173502 CLR SCOLP ;NO SCOPE LOOP
1552 005260 004767 003674 JSR PC, ERCHK ;GO CHECK ERROR
1553 005264 005367 173422 DEC RCNT ;SEE IF DONE ALL
1554 005270 001367 BNE FT13B ;IF NOT: BR
1555 005272 012767 000100 173412 MOV #100, RCNT ;SET RECORD COUNT
1556 005300 012767 020440 173376 MOV #RDATA, BADDR
1557 005306 062767 000200 173370 ADD #200, BADDR ;SET BA
1558 005314 012767 000077 173454 MOV #77, FUN ;SET READ REVERSE OP-CPDE
1559 005322 012767 014207 173364 MOV #MSG13, ERRP
1560 005330 004767 003374 FT13C: JSR PC, EXEC ;GO EXECUTE COMMAND
1561 005334 004767 003620 JSR PC, ERCHK ;GO CHECK ERROR
1562 005340 005367 173346 DEC RCNT ;SEE IF READ ALL
1563 005344 001371 BNE FT13C ;IF NOT: BR
1564 005346 162767 000200 173330 SUB #200, BADDR ;SET BA
1565 005354 012767 000071 173414 MOV #71, FUN ;SET READ FORWARD OP-CODE
1566 005362 012767 014234 173324 MOV #MSG14, ERRP
1567 005370 012767 000100 173314 MOV #100, RCNT ;SET RECORD COUNT
1568 005376 004767 003326 FT13D: JSR PC, EXEC ;GO EXECUTE COMMAND
1569 005402 004767 003552 JSR PC, ERCHK ;GO CHECK ERRORS
1570 005406 005367 173300 DEC RCNT ;SEE IF DONE ALL
1571 005412 001371 BNE FT13D ;IF NOT: BR
1572 005414 032767 062000 173362 BIT #2000, UDES ;SEE IF DONE PE
1573 005422 001017 BNE FT13X ;IF SO: BR
1574 005424 062767 000400 173352 ADD #400, UDES ;SELECT NEXT DENSITY
1575 005432 032767 002000 173344 BIT #2000, UDES ;SEE IF PE
1576 005440 001403 BEQ FT13E ;IF NOT: BR
1577 005442 005767 173344 TST NRZOF ;SEE IF NRZ ONLY
1578 005446 001005 BNE FT13X ;IF SO: BR
1579 005450 012767 000100 173234 FT13E: MOV #100, RCNT ;RESET RECORD COUNT
1580 005456 000167 177524 JMP FT13A ;GO DO NEXT DENSITY
1581 005462 000167 175120 FT13X: JMP TSCD2 ;RETURN TO SCHEDULAR
    
```

```

1582                                     ;SPACE TEST*****
1583
1584 005466 000240 FT14: NOP
1585 005470 012767 016341 173200 MOV #MSFT14,EMADDR ;SET TEST HEADER
1586 005476 012767 001700 173300 MOV #1700,UDES ;SET NRZ,NORMAL
1587 005504 004767 003352 FT14A1: JSR PC,RWIND ;GO INITIALIZE
1588 005510 012767 000100 173174 MOV #100,RCNT ;SET NUMBER OF RECORDER
1589 005516 012767 177777 011202 MOV #-1,WDATA ;SET DATA PATTERN
1590 005524 012767 177700 173154 MOV #-100,FCNT ;PRESET FRAME CNT
1591 005532 012767 177740 173150 MOV #-40,WCNT ;PRESET WORD CNT
1592 005540 004767 004310 FT14A: JSR PC,INIT1 ;GO REWIND
1593 005544 012767 001000 173154 MOV #1000,OPDYX
1594 005552 012767 040000 173144 MOV #40000,RDYDX
1595 005560 012767 000J61 173210 MOV #61,FUN ;SET WRITE OP-CODE
1596 005566 012767 102300 173152 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1597 005574 052777 000010 173006 BIS #10,JCS ;INHIBIT BUS ADDRESS INCREMENT
1598 005602 004767 003122 JSR PC,EXEC ;GO EXECUTE COMMAND
1599 005606 000240 NOP
1600 005610 012767 015244 173076 MOV #MSG46,ERRP ;SET ERROR CODE
1601 005616 004767 003336 JSR PC,ERCHK ;GO CHECK ERRORS
1602 005622 005767 173152 TST SERFL ;SEE IF ERROR
1603 005626 001402 BEQ FT14A2 ;IF NOT: BR
1604 005630 000167 000466 JMP FT14X ;ELSE EXIT
1605 005634 162767 000001 173044 FT14A2: SUB #1,FCNT ;BUMP FC
1606 005642 032767 000001 173036 BIT #1,FCNT ;SEE IF SHOULD BUMP WC
1607 005650 001403 BEQ FT14A3 ;IF NOT: BR
1608 005652 162767 000001 173030 SUB #1,WCNT ;BUMP WC
1609 005660 005367 173026 FT14A3: DEC RCNT ;SEE IF DONE ALL
1610 005664 001325 BNE FT14A ;WRITE ALL RECORDS
1611 005666 000240 NOP
1612 005670 012767 000100 173022 MOV #100,RRD ;PRESET RECORD POSITION
1613 005676 012767 000176 173016 MOV #176,RFD
1614 005704 000240 NOP
1615 005706 012767 177701 173014 MOV #-77,SCNT ;SET SPACE AMOUNT
1616 005714 012767 000033 173054 FT14B: MOV #33,FUN ;SET OP-CODE SPACE REVERSE
1617 005722 004767 003002 JSR PC,EXEC ;GO EXECUTE COMMAND
1618 005726 012767 015315 172760 MOV #MSG48,ERRP ;SET ERROR CODE
1619 005734 004767 003220 JSR PC,ERCHK ;GO CHECK ERRORS
1620 005740 005767 173034 TST SERFL ;SEE IF ERROR
1621 005744 001166 BNE FT14X ;IF SO: BR
1622 005746 004767 000070 JSR PC,FT14RR ;GO READ REVERSE + CHECK DATA
1623 005752 000240 NOP
1624 005754 012767 000031 173014 MOV #31,FUN ;SET SPACE FORWARD OP-CODE
1625 005762 005267 172742 INC SCNT ;SET SPACE AMOUNT
1626 005766 001555 BEQ FT14X ;IF DONE: BR
1627 005770 004767 002734 JSR PC,EXEC ;GO EXECUTE COMMAND
1628 005774 012767 015270 172712 MOV #MSG47,ERRP ;SET ERROR CODE
1629 006002 004767 003152 JSR PC,ERCHK ;GO CHECK ERROR
1630 006006 005767 172766 TST SERFL ;SEE IF ERROR FLAG
1631 006012 001143 BNE FT14X ;IF NO: BR
1632 006014 004767 000064 JSR PC,FT14RF ;GO READ FORWARD FOR POSITION CHECK
1633 006020 000240 NOP
1634 006022 005267 172702 INC SCNT ;DECREMENT SPACE AMOUNT
1635 006026 001535 BEQ FT14X ;IF DONE: BR
1636 006030 003267 172664 INC RRD ;BUMP DATA EXPT
1637 006034 005367 172662 DEC RFD ;BUMP DATA EXPT

```

1638	006040	000725			BR	FT14B	
1639	006042	000240			FT14RR: NOP		
1640	006044	012767	020440	172632	MOV	#RDATA, BADDR	;SET BA
1641	006052	012767	000077	172716	MOV	#77, FUN	;SET READ REVERSE OP-CODE
1642	006060	004767	002644		JSR	PC, EXEC	;GO EXECUTE COMMAND
1643	006064	000240			NOP		
1644	006066	016705	172626		MOV	RRO, R5	
1645	006072	020577	172510		CMP	R5, JFC	;SEE IF CORRECT RECORD
1646	006076	001020			BNE	FT14RER	;IF NOT: BR
1647	006100	000167	000026		JMP	FT14EC	;GO CLEAR RH11 ERROR BIT
1648	006104	000240			FT14RF: NOP		
1649	006106	012767	000071	172662	MOV	#71, FUN	;SET READ FORWARD OP-CODE
1650	006114	004767	002610		JSR	PC, EXEC	;GO EXECUTE COMMAND
1651	006120	016705	172576		MOV	RFD, R5	
1652	006124	020577	172456		CMP	R5, JFC	;SEE IF CORRECT RECORD
1653	006130	001003			BNE	FT14RER	;IF NOT: BR
1654	006132	004767	003716		FT14EC: JSR	PC, INIT1	;CLEAR RH
1655	006136	000207			RTS	PC	;RETURN
1656	006140	000240			FT14RER: NOP		
1657	006142	032777	020000	172466	BIT	#20000, JSWR	;SEE IF PRINT INHIBITED
1658	006150	001060			BNE	FT14R3	;IF SO: BR
1659	006152	012704	016341		MOV	#MSFT14, R4	
1660	006156	004767	004322		JSR	PC, TTOUT	;PRINT HEADER
1661	006162	012704	014107		MOV	#MSG9, R4	
1662	006166	004767	004312		JSR	PC, TTOUT	;PRINT ERROR TYPE
1663	006172	012704	014342		MOV	#MSG20, R4	;SET NRZ TAG POINTER
1664	006176	032767	002000	172600	BIT	#2000, UDES	;SEE IF PE
1665	006204	001402			BEQ	FT14R0	;IF NOT: BR
1666	006206	012704	014350		MOV	#MSG21, R4	;ELSE SET PE TAG POINTER
1667	006212	004767	004266		JSR	PC, TTOUT	;PRINT TAG
1668	006216	032767	000002	172552	FT14R0: BIT	#2, FUN	;SEE IF READ REVERSE
1669	006224	001003			BNE	FT14R1	;IF SO: BR
1670	006226	012704	014322		MOV	#MSG17, R4	
1671	006232	000402			BR	FT14R2	;GO PRINT
1672	006234	012704	014302		FT14R1: MOV	#MSG16, R4	
1673	006240	004767	004240		FT14R2: JSR	PC, TTOUT	;PRINT FRWD/REV
1674	006244	012704	014355		MOV	#MSG22, R4	
1675	006250	004767	004230		JSR	PC, TTOUT	;PRINT EXPT TAG
1676	006254	010503			MOV	R5, R3	
1677	006256	042703	177700		BIC	#177700, R3	;MASK RECORD NUMBER
1678	006262	004767	004362		JSR	PC, OCTP	;PRINT EXPT RECORD NUMBER
1679	006266	012704	014365		MOV	#MSG23, R4	
1680	006272	004767	004206		JSR	PC, TTOUT	;PRINT RCVD TAG
1681	006276	017703	172304		MOV	JFC, R3	
1682	006302	042703	177700		BIC	#177700, R3	;MASK RECORD NUMBER
1683	006306	004767	004336		JSR	PC, OCTP	;PRINT ACTUAL RECORD NUMBER
1684	006312	005777	172320		FT14R3: TST	JSWR	;SEE IF HALT ON ERROR
1685	006316	100001			BPL	FT14X	;IF NOT: BR
1686	006320	000000			HALT		
1687	006322	004767	004752		FT14X: JSR	PC, CKSWR	;CHECK FOR CNTL G
1688	006326	005767	172460		TST	NRZOF	;SEE IF NRZ ONLY
1689	006332	001011			BNE	FT14XX	;IF SO: BR
1690	006334	032767	002000	172442	BIT	#2000, UDES	;SEE IF DONE PE
1691	006342	001005			BNE	FT14XX	;IF SO: BR
1692	006344	012767	002300	172432	MOV	#2300, UDES	;SET TO PE
1693	006352	000167	177126		JMP	FT14A1	;DO IN PE

TMO2/TU16 BASIC FUNCTION TEST
DZTUBE.P11

MACY11 27(732) 11-OCT-76 13:04 PAGE 48

I04

1694 006356 000167 174224

FT14XX: JMP TSCD2

;RETURN TO SCHEDULAR


```

1695                                     ;ERASE TEST*****
1696
1697 006362 000240                                     FT15:  NOP
1698 006364 005067 172356                               CLR      STMSK
1699 006370 012767 000100 172326                     MOV      #100, RDYDX
1700 006376 012767 000010 172322                     MOV      #10, OPDYX
1701 006404 012767 016363 172264                     MOV      #MSFT15, EMADDR ;SET TEST HEADER
1702 006412 004767 002444                               JSR      PC, RWND ;REWIND
1703 006416 012767 020440 172260                     MOV      #RDATA, BADDR ;SET BA
1704 006424 012767 001700 172352                     MOV      #1700, UDES ;SET NRZ, NORMAL
1705 006432 012767 000025 172336 FT15A:  MOV      #25, FUN ;SET ERASE OP-CODE
1706 006440 012767 000200 172244                     MOV      #200, RCNT ;SET TO ERASE 128 TIMES
1707 006446 004767 002256 FT15B:  JSR      PC, EXEC ;GO EXECUTE COMMAND
1708 006452 012767 015244 172234                     MOV      #MSG46, ERRP ;SET ERROR CODE
1709 006460 004767 002474                               JSR      PC, ERCHK ;GO CHECK ERRORS
1710 006464 005767 172310                               TST      SEFL ;SEE IF ANY ERRORS
1711 006470 001032                               BNE      FT15X ;IF SO EXIT
1712 006472 005367 172214                               DEC      RCNT ;SEE IF DONE ERASING
1713 006476 001363                               BNE      FT15B ;IF NOT: BR
1714 006500 000240                               NOP
1715 006502 004767 002354                               JSR      PC, RWND ;REWIND
1716 006506 012767 177600 172174                     MOV      #-200, WCNT ;SET WC
1717 006514 012767 000071 172254                     MOV      #71, FUN ;SET READ FORWARD OP-CODE
1718 006522 012767 000040 172174                     MOV      #40, RDYDX ;SET DELAY
1719 006530 004767 002174                               JSR      PC, EXEC ;GO EXECUTE COMMAND
1720 006534 000240                               NOP
1721 006536 012767 015640 172150                     MOV      #MSG60, ERRP ;SET ERROR CODE
1722 006544 012767 020000 172174                     MOV      #20000, STMSK
1723 006552 004767 002402                               JSR      PC, ERCHK ;GO CHECK ERRORS
1724 006556 000167 174024 FT15X:  JMP      TSCD2 ;RETURN TO SCHEDULAR

```

; TAPE MARK WRITE/READ TEST*****

1725							
1726							
1727	006562	000240			FT16:	NOP	
1728	006564	012767	000001	172132		MOV	#1, RDYDX
1729	006572	012767	001000	172126		MOV	#1000, OPDYX
1730	006600	012767	016405	172070		MOV	#MSFT16, EMADDR ; SET HEADER
1731	006606	012767	001700	172170		MOV	#1700, UDES ; SET TO NRZ, NORMAL, ODD
1732	006614	004767	002242		FT16A:	JSR	PC, RWND ; REWIND
1733	006620	012767	177760	172060	FT16B:	MOV	#-20, FCNT ; FC=20
1734	006626	012767	177770	172054		MOV	#-10, WCNT ; WC=10
1735	006634	012767	000027	172134		MOV	#27, FUN ; SET WRITE TAPE MARK OP-CODE
1736	006642	004767	002062			JSR	PC, EXEC ; GO EXECUTE COMMAND
1737	006646	012767	001000	172072		MOV	#1000, STMSK ; SET FOR FCE MASK
1738	006654	012767	014171	172032		MOV	#MSG12, ERRP ; SET ERROR CODE
1739	006662	004767	002272			JSR	PC, ERCHK ; GO CHECK ERROR
1740	006666	004767	002634			JSR	PC, TMCHK ; GO SEE IF TM SET
1741	006672	012767	000077	172076		MOV	#77, FUN ; SET USED REVERSE OP-CODE
1742	006700	004767	002024			JSR	PC, EXEC ; GO EXECUTE COMMAND
1743	006704	012767	001000	172034		MOV	#1000, STMSK ; SET FCE ERROR MASK
1744	006712	012767	014207	171774		MOV	#MSG13, ERRP ; SET ERROR CODE
1745	006720	004767	002234			JSR	PC, ERCHK ; GO CHECK ERRORS
1746	006724	004767	002576			JSR	PC, TMCHK ; GO SEE IF TM SET
1747	006730	012767	000071	172040		MOV	#71, FUN ; SET READ FORWARD OP-CODE
1748	006736	004767	001766			JSR	PC, EXEC ; GO EXECUTE COMMAND
1749	006742	012767	014234	171744		MOV	#MSG14, ERRP ; SET ERROR CODE
1750	006750	004767	002204			JSR	PC, ERCHK ; TO CHECK ERRORS
1751	006754	004767	002546			JSR	PC, TMCHK ; GO SEE IF TM SET
1752	006760	032767	002000	172016		BIT	#2000, UDES ; SEE IF DONE PE
1753	006766	001012				BNE	FT16X ; IF SO: BR
1754	006770	005767	172016			TST	NRZOF ; SEE IF NRZ ONLY
1755	006774	001007				BNE	FT16X ; IF SO: BR
1756	006776	012767	002300	172000		MOV	#2300, UDES ; SET PE, NORMAL
1757	007004	004767	003044			JSR	PC, INIT1 ; INITIALIZE
1758	007010	000167	177604			JMP	FT16B ; DO IN PE
1759	007014	004767	002764		FT16X:	JSR	PC, ITER ; DO ITERATIONS
1760	007020	000167	173562			JMP	TSC02 ; RETURN TO SCHEDULAR
1761							

```

1762
1763
1764
1765 007024 005067 171662 FT17: CLR RCNT
1766 007030 012767 016446 171640 MOV #MSFT17,EMADDR ;SET HEADER
1767 007036 012767 001700 171740 MOV #1700,UDES ;SET TO NRZ
1758 007044 004767 002012 FT17A: JSR PC,RWHD ;REWIND TAPE
1769 007050 012767 000027 171720 FT17B: MOV #27,FUN
1770 007056 012767 040000 171640 MOV #40000,RDYDX ;SET DRY DELAY
1771 007064 012767 040000 171634 MOV #40000,OPDYX ;SET OP DELAY
1772 007072 004767 001632 JSR PC,EXEC ;GO WRITE TM
1773 007076 012767 102300 171642 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1774 007104 012767 014261 171602 MOV #MSG15,ERRP ;SET ERROR TYPE
1775 007112 004767 002042 JSR PC,ERCHK ;GO CHECK ERROR
1776 007116 005767 171656 TST SERFL ;SEE IF ERROR
1777 007122 001144 BNE FT17X ;IF SO: BR
1778 007124 004767 002376 JSR PC,TMCHK ;GO SEE IF TM SET
1779 007130 000240 NOP
1780 007132 000240 NOP
1781 007134 032767 000100 171550 BIT #100,RCNT ;SEE IF DONE PATTERN
1782 007142 001046 BNE FT17D ;IF SO: BR
1783 007144 062767 000020 171540 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1784 007152 016767 171534 171560 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1785 007160 012767 177600 171522 MOV #-200,WCNT ;WC=128
1786 007166 012767 177400 171512 MOV #-400,FCNT ;FC=256
1787 007174 012767 016726 171502 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1788 007202 012767 000061 171566 MOV #61,FUN ;SET WRITE OP CODE
1789 007210 000240 FT17C: NOP
1790 007212 000240 NOP
1791 007214 004767 001510 JSR PC,EXEC ;GO WRITE
1792 007220 012767 014171 171466 MOV #MSG12,ERRP ;SET ERROR CODE
1793 007226 012767 102300 171512 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1794 007234 004767 001720 JSR PC,ERCHK ;GO CHECK ERROR
1795 007240 005767 171534 TST SERFL ;SEE IF ERROR
1796 007244 001073 BNE FT17X ;IF SO: BR
1797 007246 005367 171466 DEC TEMP1 ;SEE IF DONE ALL
1798 007252 001356 BNE FT17C ;IF NOT: BR
1799 007254 000167 177570 JMP FT17B ;ELSE GO DO TM
1800 007260 000240 FT17D: NOP
1801 007262 012767 000033 171506 MOV #33,FUN ;SET SPACE REVERSE
1802 007270 012767 014302 171416 MOV #MSG16,ERRP ;SET ERROR CODE
1803 007276 012767 177600 171424 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1804 007304 012767 000005 171400 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1805 007312 004767 002536 FT17E: JSR PC,INIT1 ;GO INIT
1806 007316 004767 001406 JSR PC,EXEC ;GO SPACE
1807 007322 012767 001000 171416 MOV #1000,STMSK ;SET ERROR MASK
1808 007330 004767 001624 JSR PC,ERCHK ;GO CHECK ERROR
1809 007334 005767 171440 TST SERFL ;SEE IF ERROR
1810 007340 001035 BNE FT17X ;IF SO: BR
1811 007342 004767 002160 JSR PC,TMCHK ;GO SEE IF TM SET
1812 007346 005367 171340 DEC RCNT ;SEE IF DONE SPACES
1813 007352 001357 BNE FT17E ;IF NOT: BR
1814 007354 022767 000031 171414 CMP #31,FUN ;SEE IF DONE FORWARD
1815 007362 001410 BEQ FT17F ;IF SO: BR
1816 007364 012767 014322 171322 MOV #MSG17,ERRP ;SET ERROR CODE
1817 007372 012767 000031 171376 MOV #31,FUN ;SET TO SPACE FORWARD

```

1818	007400	000167	177672			JMP	FT17D1		: DO FORWARD
1819	007404	032767	002000	171372	FT17F:	BIT	#2000, UDES		: SEE IF DONE PE
1820	007412	001010				BNE	FT17X		: IF SO: BR
1821	007414	005767	171372			TST	NRZOF		: SEE IF NRZ ONLT
1822	007420	001005				BNE	FT17X		: IF SO: BR
1823	007422	012767	002300	171354		MOV	#2300, UDES		: SET TO PE
1824	007430	000167	177410			JMP	FT17A		: GO PE
1825	007434	000167	173146		FT17X:	JMP	TSCD2		: RETURN TO SCHEDULAR

```

1826
1827                                     ;WRITE CHECK TEST*****
1828
1829 007440 000240 FT20: NOP
1830 007442 012767 016474 171226 MOV #MSFT20,EMADDR ;SET HEADER
1831 007450 004767 001406 JSR PC,RWND ;REWIND
1832 007454 012767 000003 171324 MOV #3,PATRN
1833 007462 004767 002100 JSR PC,DSUP ;GO SET PATTERN 3
1834 007466 012767 016726 171210 MOV #WDATA,BADDR ;SET BA
1835 007474 012767 177400 171204 MOV #-400,FCNT ;SET FC
1836 007502 012767 177600 171200 MOV #-200,WCNT ;SET WC
1837 007510 012767 001700 171266 MOV #1700,UDES ;SET NPZ NORMAL
1838 007516 012767 000061 171252 FT20A: MOV #61,FUN ;SET WRITE OP CODE
1839 007524 004767 001200 JSR PC,EXEC ;GO WRITE RECORD
1840 007530 012767 015244 171156 MOV #MSG46,ERRP ;SET ERROR CODE
1841 007536 004767 001416 JSR PC,ERCHK ;GO CHECK ERROR
1842 007542 005767 171232 TST SERFL ;SEE IF ERROR
1843 007546 001050 BNE FT20X ;IF SO: BR
1844 007550 012767 014302 171136 MOV #MSG16,ERRP ;SET REVERSE ERROR TAG
1845 007556 012767 000057 171212 MOV #57,FUN ;SET REVERSE WRITE CHECK OP-CODE
1846 007564 062767 000376 171112 ADD #376,BADDR ;SET BA FOR REVERSE CHECK
1847 007572 004767 001132 JSR PC,EXEC ;GO DO REVERSE CHECK
1848 007576 004767 001356 JSR PC,ERCHK ;GO CHECK ERROR
1849 007602 012767 014322 171104 FT20B: MOV #MSG17,ERRP ;SET FORWARD TAG
1850 007610 012767 000051 171160 MOV #51,FUN ;SET FORWARD CHECK OP CODE
1851 007616 162767 000376 171060 SUB #376,BADDR ;SET BA FOR FORWARD CHECK
1852 007624 004767 001100 JSR PC,EXEC ;GO DO FORWARD CHECK
1853 007630 004767 001324 JSR PC,ERCHK ;GO CHECK ERROR
1854 007634 032767 002000 171142 FT20C: BIT #2000,UDES ;SEE IF DONE PE
1855 007642 001012 BNE FT20X ;IF SO: BR
1856 007644 005767 171142 TST NRZOF ;SEE IF NRZ ONLY
1857 007650 001007 BNE FT20X ;IF SO: BR
1858 007652 012767 002300 171124 MOV #2300,UDES ;ELSE SET PE
1859 007660 004767 002170 JSR PC,INIT1 ;GO INIT
1860 007664 000167 177626 JMP FT20A ;DO IN PE
1861 007670 004767 002110 FT20X: JSR PC,ITER ;DO ITERATIONS
1862 007674 000167 172706 JMP TSCD2 ;RETURN TO SCHEDULAR
    
```

```

1863
1864
1865
1866 007700 012767 016525 170770 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1867 007706 004767 001150 JSR PC,RWIND ;GO REWIND
1868 007712 012767 000003 171066 MOV #3,PATRN
1869 007720 004767 001642 JSR PC,DSUP ;GO SET PATTERN 3
1870 007724 012767 016726 170752 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1871 007732 012767 176340 170746 MOV #-1440,FCNT ;SET FC=800
1872 007740 012767 177160 170742 MOV #-620,WCNT ;SET WC=400
1873 007746 012767 001700 171030 MOV #1700,UDES ;SET NRZ NORMAL
1874 007754 012767 000061 171014 MOV #61,FUN ;SET WRITE OP-CODE
1875 007762 004767 000742 JSR PC,EXEC ;GO DO WRITE 1
1876 007766 012767 014171 170720 MOV #MSG12,ERRP ;SET ERROR CODE
1877 007774 004767 001160 JSR PC,ERCHK ;GO CHECK FOR ERROR
1878 010000 004767 000724 JSR PC,EXEC ;YES DO WRITE 2
1879 010004 004767 001150 JSR PC,ERCHK ;YES CHECK FOR ERROR
1880 010010 000240 NOP
1881 010012 004767 001044 JSR PC,RWIND ;GO REWIND
1882 010016 012767 177160 170662 MOV #-620,FCNT ;SET FC=400
1883 010024 012767 177470 170656 MOV #-310,WCNT ;SET WC=200
1884 010032 004767 000672 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1885 010036 000240 FT21A: NOP
1886 010040 004767 001016 JSR PC,RWIND ;REWIND
1887 010044 012767 020440 170632 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1888 010052 012767 177160 170626 MOV #-620,FCNT ;SET FC=400
1889 010060 012767 177470 170622 MOV #-310,WCNT ;SET WC=200
1890 010066 012767 000071 170702 MOV #71,FUN ;SET READ OP-CODE
1891 010074 004767 000630 JSR PC,EXEC ;GO READ RECORD 1
1892 010100 012767 014234 170606 MOV #MSG14,ERRP ;SET ERROR CODE
1893 010106 004767 001046 JSR PC,ERCHK ;GO CHECK FOR ERROR
1894 010112 000240 NOP
1895 010114 052777 000010 170466 BIS #10,ACS ;INHIBIT BA INCREMENT
1896 010122 012767 176340 170556 MOV #-1440,FCNT ;SET FC=800
1897 010130 012767 177160 170552 MOV #-620,WCNT ;SET WC=400
1898 010136 004767 000566 JSR PC,EXEC ;GO READ RECORD 2
1899 010142 022777 001440 170436 CMP #1440,ACFC ;SEE IF READ RECORD 2
1900 010150 001410 BEQ FT21X ;IF SO: BR
1901 010152 012767 015142 170556 MOV #MSG44,ERADD ;SET ERROR CODE
1902 010160 012767 010036 170574 MOV #FT21A,COLP ;SET SCOPE ADDRESS
1903 010166 004767 173362 JSR PC,FT3ER ;GO PRINT ERROR
1904 010172 004767 001606 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1905 010176 000167 172404 JMP TSCD2 ;RETURN TO SCHEDULAR
1906
1907
    
```

```

1908                                     ;BUFFERED COMMAND TEST*****
1909
1910 010202 012767 016554 170466 FT22: MOV      #MSFT22,EMADDR ;SET TEST HEADER
1911 010210 004767 003646          JSR      PC,RWIND   ;GO REWIND
1912 010214 012700 000003          MOV      #3,R0     ;SET NUMBER OF WRITES
1913 010220 012767 001700 170556  MOV      #1700,UDES ;SET TO NRZ NORMAL
1914 010226 012767 016726 170450  MOV      #WDATA,BADDR ;SET BA=WRITE BUFFER
1915 010234 012767 177000 170444  MOV      #-1000,FCNT ;SET FC=1000
1916 010242 012767 177400 170440  MOV      #-400,WCNT  ;SET WC=400
1917 010250 012767 000061 170520  MOV      #61,FUN    ;SET WRITE OP-CODE
1918 010256 004767 000446          FT22A: JSR      PC,EXEC  ;GO DO WRITE
1919 010262 005300          DEC      R0        ;SEE IF DONE ALL
1920 010264 001374          SNE     FT22A     ;IF NOT: BR
1921 010266 000240          NOP
1922 010270 012777 000007 170302  MOV      #7,BC1    ;START REWIND
1923 010276 032777 000200 170306  FT22B: BIT      #200,BDS
1924 010304 001774          BEQ     FT22B
1925 010306 004767 001542          JSR      PC,INIT1  ;INITIALIZE
1926 010312 012767 000010 170404  MOV      #10,RDYDX ;SET LONG READY DELAY
1927 010320 004767 000404          JSR      PC,EXEC  ;ISSUE BUFFERED WRITE
1928 010324 000240          NOP
1929 010326 012767 015342 170360  MOV      #MSG49,ERRP ;SET ERROR CODE
1930 010334 012767 102300 170404  MOV      #102300,STMSK ;MARK DATA ERROR
1931 010342 004767 000612          JSR      PC,ERCHK ;GO CHECK ERROR
1932 010346 032777 000002 170236  BIT      #2,BDS    ;SEE IF BOT IS SET
1933 010354 001410          BEQ     FT22X     ;IF NOT: BR
1934 010356 012767 015370 170352  MOV      #MSG50,ERADD ;SET ERROR CODE
1935 010364 012767 010202 170370  MOV      #FT22,$COLP
1936 010372 004767 173156          JSR      PC,FT3ER  ;GO DO ERROR
1937 010376 004767 001402          FT22X: JSR      PC,ITER ;GO SEE IF ITERATION
1938 010402 000167 172200          JMP      TSC02    ;RETURN TO SCHEDULAR
1939
1940
    
```

```

1941                                     ;READ-IN PRESET TEST*****
1942
1943 010406 005767 170270          FT23: TST      SLVN          ;SEE IF SLAVE SELECT=0
1944 010412 001103                BNE      FT23X        ;IF NOT: BR
1945 010414 012767 016611 170254  MOV      #MSGFT23,EMADDR ;SET TEST HEADER
1946 010422 004767 001426                JSR      PC,INIT1     ;GO INIT
1947 010426 012767 001700 170350  MOV      #1700,UDES    ;SET TO NRZ NORMAL
1948 010434 012767 016726 170242  MOV      #WDATA,BAADR  ;SET BA=WRITE BUFFER
1949 010442 012767 177400 170236  MOV      #-400,FCNT    ;SET FC=400
1950 010450 012767 177600 170232  MOV      #-200,WCNT    ;SET WC=200
1951 010456 012767 000061 170312  MOV      #61,FUN       ;SET WRITE OP-CODE
1952 010464 004767 000240                JSR      PC,EXEC      ;GO DO WRITE
1953 010470 000240                NOP
1954 010472 004767 001356                JSR      PC,INIT1     ;INITIALIZE
1955 010476 012767 000021 170272  MOV      #21,FUN       ;SET READ-IN PRESET OP CODE
1956 010504 004767 000220                JSR      PC,EXEC      ;GO DO COMMAND
1957 010510 005000                CLR
1958 010512 012703 000004                MOV      #4,R3        ;SET MULT
1959 010516 032777 020000 170066  FT23A: BIT      #20000,ADS  ;SEE IF PIP RESET
1960 010524 001404                BEQ      FT23B        ;IF SO: BR
1961 010526 005300                DEC      R0
1962 010530 001372                BNE      FT23A        ;AWAIT PIP RESET
1963 010532 005303                DEC      R3
1964 010534 001370                BNE      FT23A
1965 010536 032777 000002 170046  FT23B: BIT      #2,ADS   ;DELAY
1966 010544 001010                BNE      FT23C        ;SEE IF BOT
1967 010546 012767 015426 170162  MOV      #MSG51,ERADD  ;IF SO: BR
1968 010554 012767 010406 170200  MOV      #FT23,SCOLP  ;SET ERROR CODE
1969 010562 004767 172766                JSR      PC,FT3ER     ;GO DO ERROR
1970 010566 012701 141000          FT23C: MOV      #141000,R1 ;SET EXPT TC
1971 010572 016700 170034                MOV      TC,R0        ;SET TC ADDRESS
1972 010576 020110                CMP      R1,(R0)      ;SEE IF EXPT=RCVD
1973 010600 001410                BEQ      FT23X        ;IF SO: BR
1974 010602 012767 015462 170126  MOV      #MSG52,ERADD  ;SET ERROR CODE
1975 010610 012767 010406 170144  MOV      #FT23,SCOLP  ;CLEAR SCOPE ADDRESS
1976 010616 004767 172442                JSR      PC,FT2ER     ;GO DO ERROR
1977 010622 000167 171760          FT23X: JMP      TSCD2   ;RETURN TO SCHEDULAR
1978
1979
    
```



```

1980                                     ;REWIND: OFF LINE TEST*****
1981
1982 010626 032777 004000 170002 FT24: BIT #4000,JSWR ;SEE IF IN CONTINUOUS MODE
1983 010634 0C1033 BNE FT24XX ;IF SO: BR
1984 010636 012767 016644 170032 MOV #MSFT24,EMADDR ;SET TEST HEADER
1985 010644 004767 001204 JSR PC,INIT1 ;GO INITIAIZE
1986 010650 012777 000003 167722 MOV #3,PC1 ;ISSUE REWIND: OFF LINE COMMAND
1987 010656 012700 004000 MOV #4000,R0
1988 010662 005300 FT24A: DEC R0 ;DELAY
1989 010664 0C1376 BNE FT24A
1990 010666 032777 010000 167716 BIT #10000,ADS ;SEE IF MOL IS RESET
1991 010674 001407 BEQ FT24X ;IF SO: BR
1992 010676 005067 170060 CLR SCOLP ;ASSURE NO SCOPE
1993 010702 012767 015501 170026 MOV #MSG53,ERADD ;SET ERROR CODE
1994 010710 004767 172640 JSR PC,FT3ER ;GO DO ERROR
1995 010714 012704 015525 FT24X: MOV #MSG54,R4
1996 010720 004767 001560 JSR PC,TTOUT ;PRINT ON LINE REQUEST
1997 010724 000167 171656 FT24XX: JMP TSCD2 ;RETURN TO SCHEDULAR
1998
1999

```

```

2000 ;COMMAND EXECUTE SUBROUTINE*****
2001
2002 010730 000240 EXEC: NOP
2003 010732 056777 170046 167672 BIS UDES,2TC ;LOAD TAPE CONT
2004 010740 016777 167744 167634 MOV WCNT,2WC ;LOAD WC
2005 010746 016777 167734 167632 MOV FCNT,2FC ;LOAD FC
2006 010754 016777 167724 167622 MOV BADDR,2BA ;LOAD BA
2007 010762 022767 000031 170006 CMP #31,FUN ;SEE IF SPACE FORWARD
2008 010770 001404 BEQ EXECA ;IF SO: BR
2009 010772 022767 000033 167776 CMP #33,FUN ;SEE IF SPACE REVERSE
2010 011000 001003 BNE EXECB ;IF NOT: BR
2011 011002 016777 167722 167576 EXECXA: MOV SCNT,2FC ;SET SPACE COUNT
2012 011010 000240 EXECB: NOP
2013 011012 016777 167760 167560 MOV FUN,2C1 ;LOAD OP-CODE + GO
2014 011020 000240 NOP
2015 011022 016703 167676 MOV RDYDX,R3 ;SET DELAY
2016 011026 005004 CLR R4
2017 011030 032777 000200 167554 EXECC: BIT #200,2DS ;SEE IF DRY
2018 011036 001004 BNE EXECX ;IF SO: BR
2019 011040 005304 DEC R4
2020 011042 001372 BNE EXECC
2021 011044 005303 DEC R3 ;DELAY FOR DRY
2022 011046 001370 BNE EXECC
2023 011050 016703 167652 EXECX: MOV OPDYX,R3
2024 011054 005303 EXECXA: DEC R3 ;DELAY
2025 011056 001376 BNE EXECXA
2026 011060 000207 EXECXX: RTS PC ;RETURN TO CALLER
2027
    
```

```

2028                                     ;REWIND SUBROUTINE*****
2029
2030 011062 000240                          RWND:  NOP
2031 011064 004767 000764                  JSR   PC,INIT1          ;INIT
2032 011070 012777 0C0007 167502          MOV   #7,PC1           ;START REWIND
2033 011076 012700 040000                  MOV   #40000,R0
2034 011102 005300                          RWNDA: DEC   R0
2035 011104 001376                          BNE   RWNDA           ;DELAY
2036 011106 032777 020000 167476          RWNDB: BIT   #20000,ADS
2037 011114 001374                          BNE   RWNDB           ;AWAIT PIP
2038 011116 032777 000002 167466          BIT   #2,ADS          ;SEE IF BOT
2039 011124 001012                          BNE   RWNDX           ;IF SO: BR
2040 011126 016704 167544                  MOV   EMADDR,R4
2041 011132 004767 001346                  JSR   PC,TTOUT         ;PRINT HEADER
2042 011136 012704 013702                  MOV   #MSG2,R4
2043 011142 004767 001336                  JSR   PC,TTOUT         ;PRINT REWIND ERROR
2044 011146 000167 171434                  JMP   TSC02           ;RETURN TO SCHEDULAR
2045 011152 004767 000676          RWNDX: JSR   PC,INIT1  ;INIT
2046 011156 000207                          RTS                    ;RETURN TO CALLER
2047

```

```

2048                                     ;ERROR CHECK SUBROUTINE*****
2049
2050 011160 005067 167614 ERCHK: CLR SERFL ;CLEAR FLAG
2051 011164 017767 167422 167560 MOV @DS,DSAV ;SAVE DRIVE STATUS REGISTER
2052 011172 032777 040000 167412 BIT #40000,@DS ;SEE IF ERROR
2053 011200 001001 BNE ERPT ;IF SO: BR
2054 011202 000207 RTS PC ;RETURN
2055 011204 017704 167404 ERPT: MOV @ER,R4 ;GET ERROR REGISTER
2056 011210 032767 002000 167566 BIT #2000,UDES ;SEE IF PE
2057 011216 001403 BEQ ERPTA1 ;IF SO: BR
2058 011220 042767 000200 167520 BIC #200,STMSK ;RESET PEF MASK
2059 011226 046704 167514 ERPTA1: BIC STMSK,R4 ;MASK DONT CARE BITS
2060 011232 001530 BEQ ERPTX ;IF NO UNEXPECTED ERRORS: BR
2061 011234 012767 000001 167536 ERPTG: MOV #1,SERFL ;SET FLAG
2062 011242 032777 020000 167366 BIT #20000,@SWR ;SEE IF SHOULD PRINT ERRORS
2063 011250 001115 BNE ERPTD ;IF NOT: BR
2064 011252 005767 167416 TST HDRFL ;SEE IF DONE HEADER
2065 011256 001006 BNE ERPTA ;IF SO: BR
2066 011260 005267 167410 INC HDRFL ;SET HEADER FLAG
2067 011264 016704 167406 MOV EMADDR,R4
2068 011270 004767 001210 JSR PC,TTOUT ;PRINT HEADER
2069 011274 016704 167414 ERPTA: MOV ERAP,R4 ;GET ERROR CODE
2070 011300 001114 BEQ ERPTB ;IF NONE: BR
2071 011302 004767 001176 JSR PC,TTOUT ;PRINT ERROR CODE
2072 011306 012704 014342 MOV #MSG20,R4 ;SET NRZ TAG
2073 011312 032777 002000 167312 BIT #2000,@TC ;SEE IF PE
2074 011320 001402 BEQ ERPT1A ;IF NOT: BR
2075 011322 012704 014350 MOV #MSG21,R4 ;ELSE SET PE TAG
2076 011326 004767 001152 ERPT1A: JSR PC,TTOUT ;PRINT TAG
2077 011332 016704 167360 ERPTB: MOV ERAP1,R4 ;SEE IF CODE 2
2078 011336 001402 BEQ ERPTB1 ;IF NOT: BR
2079 011340 004767 001140 JSR PC,TTOUT ;PRINT CODE 2
2080 011344 032777 010000 167264 ERPTB1: BIT #10000,@SWR ;SEE IF ITERATION
2081 011352 001010 BNE ERPTC ;IF NOT: BR
2082 011354 012704 015614 MOV #MSG56,R4
2083 011360 004767 001120 JSR PC,TTOUT ;PRINT ITER TAG
2084 011364 016703 167360 MOV ITCNT,R3
2085 011370 004767 001254 JSR PC,OC1P ;PRINT ITERATION
2086 011374 012704 013614 ERPTC: MOV #MSG1,R4
2087 011400 004767 001100 JSR PC,TTOUT ;PRINT REGISTER TAG
2088 011404 017703 167170 MOV @C1,R3
2089 011410 004767 001222 JSR PC,OC1PE ;PRINT C1
2090 011414 017703 167162 MOV @WC,R3
2091 011420 004767 001212 JSR PC,OC1PE ;PRINT WC
2092 011424 017703 167154 MOV @BA,R3
2093 011430 004767 001202 JSR PC,OC1PE ;PRINT BA
2094 011434 017703 167146 MOV @FC,R3
2095 011440 004767 001172 JSR PC,OC1PE ;PRINT FC
2096 011444 017703 167140 MOV @CS,R3
2097 011450 004767 001162 JSR PC,OC1PE ;PRINT CS2
2098 011454 017703 167132 MOV @DS,R3
2099 011460 004767 001152 JSR PC,OC1PE ;PRINT DS
2100 011464 017703 167124 MOV @ER,R3
2101 011470 004767 001142 JSR PC,OC1PE ;PRINT ER
2102 011474 017703 167132 MOV @TC,R3
2103 011500 004767 001132 JSR PC,OC1PE ;PRINT TC
    
```

2104	011504	005777	167126
2105	011510	100001	
2106	011512	000000	
2107	011514	004767	001560
2108	011520	004767	000330
2109	011524	000207	
2110			
2111			

ERPTD:	TST	QSWR	;SEE IF HALT ON ERROR
	BPL	ERPTX	;IF NOT: BR
	HALT		
ERPTX:	JSR	PC,CKSWR	;CHECK FOR CNTL G
	JSR	PC,INITI	;INIT
ERPTXX:	RTS	PC	;RETURN

```

2112 ; TAPE MARK STATUS CHECK*****
2113
2114 011526 032767 000004 167216 TMCHK: BIT #4, DSAV ;SEE IF TM SET
2115 011534 001401 BEQ TMCHK1 ;IF NOT: BR
2116 011536 000207 TMCHK0: RTS PC ;ELSE RETURN
2117 011540 005767 167234 TMCHK1: TST SERFL ;SEE IF HAD ERROR
2118 011544 001374 BNE TMCHK0 ;IF SO: BR
2119 011546 012767 015624 167142 MOV #MSG57, ERRP1 ;SET ERROR CODE 2
2120 011554 004767 177454 JSR PC, ERRPG ;GO PRINT TM ERROR
2121 011560 005067 167132 CLR ERRP1 ;CLEAR CODE 2 FLAG
2122 011564 000207 RTS PC ;RETURN
2123
2124 ; DATA SETUP ROUTINE*****
2125
2126 011566 000240 DSUP: NOP
2127 011570 012703 DSO: MOV #WDATA, R3 ;R3 = ADDRS OF WRITE BUFFER
2128 011574 016701 MOV PATRN, R1 ;R1 = PATTERN SELECTOR
2129 011600 000241 CLC
2130 011602 006101 ROL R1 ;MAKE PATTERN SELECTOR EVEN
2131 011604 000171 JMP @DATBL(R1) ;GO GENERATE PATTERN
2132 011610 032777 010000 167010 DS1: BIT #10000, @DT ;SEE IF SEVEN TRACK
2133 011616 001410 BEQ DS3 ;IF NOT: BR
2134 011620 012702 000640 MOV #640, R2 ;SET BUFFER SIZE
2135 011624 012701 016726 MOV #WDATA, R1 ;SET START OF BUFFER
2136 011630 042721 140300 DS2: BIC #140300, (R1)+ ;MASK FOR 7 CH
2137 011634 005302 DEC R2 ;SEE IF DONE
2138 011636 001374 BNE DS2 ;IF NOT: BR
2139 011640 012702 000640 DS3: MOV #640, R2 ;R2=BUFFER SIZE +2
2140 011644 012701 020440 MOV #RDATA, R1 ;R1=READ DATA START
2141 011650 005021 DS4: CLR (R1)+ ;CLEAR BUFFER
2142 011652 005302 DEC R2 ;SEE IF DONE ALL
2143 011654 001375 BNE DS4 ;IF NOT: BR
2144 011656 000207 RTS PC ;EXIT
2145
2146 ; ALL ONES*****
2147
2148 011660 012701 177777 DAT1: MOV #-1, R1 ;R1=DATA
2149 011664 012702 000640 DAT1A: MOV #640, R2 ;R2=WORD COUNT +2
2150 011670 010123 DAT1B: MOV R1, (R3)+ ;LOAD BUFFER
2151 011672 005302 DEC R2 ;SEE IF DONE
2152 011674 001375 BNE DAT1B ;IF NOT: BR
2153 011676 000167 177706 JMP DS1 ;RETURN
2154
2155 ; ALL ZEROS*****
2156
2157 011702 005001 DAT2: CLR R1 ;R1=DATA
2158 011704 000167 177754 JMP DAT1A ;LOAD BUFFER
2159
    
```

```

2160                                     ;ONE/ZERO IN ALTERNATING CHARACTERS*****
2161
2162 011710 012701 125125          DAT3:  MOV    #125125,R1      ;R1=DATA
2163 011714 000167 177744          JMP     DAT1A          ;LOAD BUFFER
2164
2165                                     ;ALL BITS 0-377*****
2166
2167 011720 005001          DAT4:  CLR     R1          ;R1=STARTING DATA
2168 011722 012702 001500          MOV     #1500,R2      ;R2=CHARACTER COUNT
2169 011726 110123          DAT4A: MOVVB  R1,(R3)+    ;LOAD BUFFER
2170 011730 105201          INCB   R1            ;BUMP DATA
2171 011732 005302          DEC    R2            ;SEE IF DONE
2172 011734 001374          SNE    DAT4A        ;IF NOT: BR
2173 011736 000167 177646          JMP     DS1          ;RETURN
2174
2175
2176                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
2177
2178 011742 004767 001332          SCOPE: JSR    PC,CKSWR      ;CHECK FOR CNTL G
2179 011746 000240          NOP
2180 011750 032777 040000 166660          BIT    #40000,DSWR    ;SEE IF LOOP ON ERROR
2181 011756 001001          BNE    SCOPE1        ;IF SO: BR
2182 011760 000207          RTS    PC            ;ELSE EXIT
2183 011762 000240          SCOPE1: NOP
2184 011764 005767 166772          TST    SCOLP        ;SEE IF SCOPE ADDRESS
2185 011770 001001          BNE    SCOPE2        ;IF NOT: BR
2186 011772 000207          RTS    PC            ;ELSE EXIT
2187 011774 005726          SCOPE2: TST    (SP)+    ;RESET STACK
2188 011776 005726          TST    (SP)+
2189 012000 000177 166756          JMP     @SCOLP       ;LOOP ON ERROR
2190
2191                                     ;TEST ITERATION SUBROUTINE*****
2192
2193 012004 004767 001270          ITER:  JSR    PC,CKSWR      ;CHECK FOR CNTL G
2194 012010 000240          NOP
2195 012012 032777 010000 166616          BIT    #10000,DSWR    ;SEE IF ITERATIONS
2196 012020 001403          BEQ   ITER1          ;IF SO: BR
2197 012022 005067 166722          ITER0: CLR    ITCNT      ;CLEAR ITERATION COUNTER
2198 012026 000207          RTS    PC            ;ELSE EXIT
2199 012030 005267 166714          ITER1: INC    ITCNT      ;BUMP COUNTER
2200 012034 026767 166710 166612          CMP    ITCNT,ITAMT    ;SEE IF DONE ALL
2201 012042 001767          BEQ   ITER0          ;IF SO: BR
2202 012044 005726          TST    (SP)+        ;RESET STACK
2203 012046 017700 166712          MOV    @ITRLP,R0     ;SET ITERATION POINTER
2204 012052 000110          JMP    (R0)          ;GO ITERATE
2205
2206                                     ;INITIALIZE SUBROUTINE*****
2207
2208 012054 000240          INIT1: NOP
2209 012056 012777 000040 166524          MOV    #40,@CS       ;INIT
2210 012064 016777 166610 166516          INIT2: MOV    DRVN,@CS    ;SELECT DRIVE
2211 012072 016777 166604 166532          MOV    SLVN,@TC      ;SELECT SLAVE
2212 012100 000207          RTS    PC            ;RETURN
2213

```

```

2214                                     ;MAG TAPE INTERRUPT HANDLER*****
2215
2216 012102 000240          MTINT:  NOP
2217 012104 022626          CMP      (SP)+,(SP)+      ;RESET STACK POINTER
2218 012106 000240          NOP
2219 012110 000240          NOP
2220 012112 000177 166616  JMP      @RTRN          ;RETURN TO CALLER
2221
2222                                     ;TTY INTERRUPT HANDLER*****
2223
2224 012116 000240          TTINT:  NOP
2225 012120 000240          NOP
2226 012122 000240          NOP
2227 012124 000002          RTI
2228
2229                                     ;BUS ADDRESS TRAP HANDLER*****
2230
2231 012126 000240          TRAP:  NOP
2232 012130 032777 020000 166500 BIT      #20000,@SWR      ;SEE IF SHOULD PRINT ERRORS
2233 012136 001020          BNE     TRAP2          ;IF NOT: BR
2234 012140 005767 166530  TST     HDRFL          ;SEE IF DONE HEADER
2235 012144 001006          BNE     TRAP1          ;IF SO: BR
2236 012146 005267 166522  INC     HDRFL          ;ELSE SET HEADER FLAG
2237 012152 016704 166520  MOV     EMADDR,R4
2238 012156 004767 000322  JSR     PC,TTOUT       ;PRINT HEADER
2239 012162 012704 014375  TRAP1:  MOV     #MSG24,R4
2240 012166 004767 000312  JSR     PC,TTOUT       ;PRINT ERROR
2241 012172 010103          MOV     R1,R3
2242 012174 004767 000450  JSR     PC,OCTP        ;PRINT ADDRESS OF TRAP
2243 012200 005777 166432  TRAP2:  TST     @SWR      ;SEE IF HALT ON ERROR
2244 012204 100001          BPL     TRAPX          ;IF NOT: BR
2245 012206 000000          HALT
2246 012210 004767 001064  TRAPX:  JSR     PC,CKSWR      ;CHECK FOR CNTL G
2247 012214 022626          CMP     (SP)+,(SP)+    ;RESET STACK
2248 012216 012767 003026 166536  MOV     #FT1A,SCOLP    ;SET SCOPE ADDRESS
2249 012224 004767 177512  JSR     PC,SCOPE       ;GO SEE IF SCOPE LOOP
2250 012230 005767 166554  TST     RHTF           ;SEE IF INITIAL ADDRESS TEST
2251 012234 001402          BEQ     TRAPXX        ;IF NOT: BR
2252 012236 000167 167456  JMP     STOB           ;ELSE REDO ADDRESS REQUEST
2253 012242 000167 170564  TRAPXX: JMP     FT1B        ;RETURN TO TEST 1
2254

```



```

2255 ;*****
2256 ;TTY ENTRY SUBROUTINE:
2257 ;
2258 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
2259 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
2260 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
2261 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
2262 ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
2263 ;THE CALLING ROUTINE.
2264 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
2265 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
2266 ;MAY BE REENTERED.
2267 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
2268 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
2269 ;CARRIAGE RETURN
2270 ;*****
2271
2272 012246 005067 166466 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
2273 012252 005000 CLR RO
2274 012254 004767 000152 TTR0: JSR PC,TIN ;GO READ CHARACTER
2275 012260 122767 000215 166402 CMPB #215,TIB ;SEE IF CR
2276 012266 001005 BNE TTR1 ;IF NOT: BR
2277 012270 005767 166444 TST TEMP1 ;SEE IF FIRST CHARACTER
2278 012274 001446 BEQ TTR5 ;IF SO: BR
2279 012276 000167 000066 JMP TTR2 ;ELSE GO LOAD VALUE
2280 012302 122767 000260 166360 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0
2281 012310 101402 BLOS TTR1A ;IF NOT: BR
2282 012312 000167 000076 JMP TINER ;ELSE GO TO ERROR
2283 012316 122767 000270 166344 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7
2284 012324 101002 BHI TTR1B ;IF NOT: BR
2285 012326 000167 000062 JMP TINER ;ELSE GO TO ERROR
2286 012332 005267 166402 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
2287 012336 000241 CLC
2288 012340 006100 ROL RO
2289 012342 000241 CLC
2290 012344 006100 ROL RO ;SHIFT 3 LEFT
2291 012346 000241 CLC
2292 012350 006100 ROL RO
2293 012352 042767 177770 166310 BIC #177770,TIB ;STRIP ASCII
2294 012360 056700 166304 BIS TIB,RO ;LOAD CHARACTER
2295 012364 005301 DEC R1 ;SEE IF DONE
2296 012366 001332 BNE TTR0 ;IF NOT: BR
2297 012370 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
2298 012372 101402 BLOS TTR3 ;IF NOT: BR
2299 012374 000167 000014 JMP TINER ;ELSE GO TO ERROR
2300 012400 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
2301 012402 101402 BLOS TTR4 ;IF NOT: BR
2302 012404 000167 000004 JMP TINER ;ELSE GO TO ERROR
2303 012410 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
2304 012412 000207 TTR5: RTS PC ;EXIT
2305

```

```

2306 ;TTY ENTRY ERROR SUBROUTINE*****
2307
2308 012414 012704 014101 TINER: MOV #MSG7,R4
2309 012420 004767 000060 JSR PC,TTOUT ;PRINT?
2310 012424 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
2311 012430 000207 RTS PC ;REDO VALUE ENTRY
2312
2313 ;TTY READ SUBROUTINE*****
2314
2315 012432 005077 166202 TTIN: CLR @TKS
2316 012436 005077 166200 CLR @TKB
2317 012442 005067 166222 CLR TIB
2318 012446 005277 166165 INC @TKS
2319 012452 105777 166162 TTIN1: TSTB @TKS
2320 012456 100375 BPL TTIN1
2321 012460 017767 166156 166202 MOV @TKB,TIB
2322 012466 105777 166152 TTIN2: TSTB @TPS
2323 012472 100375 BPL TTIN2
2324 012474 116777 166170 166144 MOVB TIB,@TPB
2325 012502 000207 RTS PC
2326
2327 ;TTY OUTPUT SUBROUTINE*****
2328
2329 012504 112467 166156 TTOUT: MOVB (R4)+,TOB
2330 012510 122767 000043 166150 CMPB #43,TOB
2331 012516 001446 BEQ TEX
2332 012520 122767 000045 166140 CMPB #45,TOB
2333 012526 001403 BEQ TCRLF
2334 012530 004767 000064 JSR PC,TOG
2335 012534 000763 BR TTOUT
2336 012536 112767 000015 166122 TCRLF: MOVB #15,TOB
2337 012544 004767 000050 JSR PC,TOG
2338 012550 012703 000004 MOV #4,R3
2339 012554 005067 166106 TCRLFA: CLR TOB
2340 012560 004767 000034 JSR PC,TOG
2341 012564 005303 DEC R3
2342 012566 001372 BNE TCRLFA ;DO FILLERS
2343 012570 112767 000012 166070 MOVB #12,TOB
2344 012576 004767 000016 JSR PC,TOG
2345 012602 105767 166216 TSTB RDSW
2346 012606 100401 BMI IS
2347 012610 000735 BR TTOUT
2348 012612 005067 166206 IS: CLR RDSW
2349 012616 000406 BR TEX
2350 012620 105777 166020 TOG: TSTB @TPS
2351 012624 100375 BPL TOG
2352 012626 116777 166034 166012 MOVB TOB,@TPB
2353 012634 000207 RTS PC
2354
2355
    
```

```

;OCTAL OUTPUT SUBROUTINE*****
2356
2357
2358 012636 012767 000001 000222 OCTPE: MOV #1,OFL
2359 012644 010304 MOV R3,R4
2360 012646 000410 BR OCTPO
2361 012650 005067 000212 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
2362 012654 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
2363 012656 001004 BNE OCTPO ;IF NOT ZERO: BR
2364 012660 004767 000162 JSR PC,OCTPG1 ;ELSE PRINT ZERO
2365 012664 000167 000120 JMP OCTP3 ;SPACE AND EXIT
2366 012670 032704 100000 OCTPO: BIT #100000,R4 ;SEE IF MSD = 1
2367 012674 001406 BEQ OCTP1 ;IF NOT: BR
2368 012676 012704 000001 MOV #1,R4
2369 012702 004767 000116 JSR PC,OCTPG ;PRINT 1
2370 012706 000167 000006 JMP OCTP2
2371 012712 005004 OCTP1: CLR R4
2372 012714 004767 000104 JSR PC,OCTPG ;PRINT 0
2373 012720 010304 OCTP2: MOV R3,R4
2374 012722 006004 ROR R4
2375 012724 006004 ROR R4
2376 012726 006004 ROR R4 ;POSITION DIGIT
2377 012730 006004 ROR R4
2378 012732 000304 SWAB R4
2379 012734 004767 000064 JSR PC,OCTPG ;PRINT DIGIT 2
2380 012740 010304 MOV R3,R4
2381 012742 006004 ROR R4
2382 012744 000304 SWAB R4
2383 012746 004767 000052 JSR PC,OCTPG ;PRINT DIGIT 3
2384 012752 010304 MOV R3,R4
2385 012754 006104 ROL R4
2386 012756 006104 ROL R4
2387 012760 000304 SWAB R4
2388 012762 004767 000036 JSR PC,OCTPG ;PRINT DIGIT 4
2389 012766 010304 MOV R3,R4
2390 012770 006004 ROR R4
2391 012772 006004 ROR R4
2392 012774 006004 ROR R4
2393 012776 004767 000022 JSR PC,OCTPG
2394 013002 010304 MOV R3,R4
2395 013004 004767 000014 JSR PC,OCTPG ;PRINT DIGIT 5
2396 013010 012767 000240 165650 OCTP3: MOV #240,T0B
2397 013016 004767 177576 JSR PC,T0B ;PRINT SPACE
2398 013022 000207 RTS PC ;EXIT
2399 013024 042704 177770 OCTPG: BIC #177770,R4
2400 013030 001004 BNE OCTPG0
2401 013032 005767 000030 TST OFL
2402 013036 001001 BNE OCTPG0
2403 013040 000207 RTS PC
2404 013042 005267 000020 OCTPG0: INC OFL
2405 013046 052704 000260 OCTPG1: BIS #260,R4
2406 013052 010467 165610 MOV R4,T0B
2407 013056 004767 177536 JSR PC,T0B
2408 013062 010304 MOV R3,R4
2409 013064 000207 RTS PC
2410 013066 000000 OFL: 0 ;FIRST CHAR FLAG
2411

```

;DATA CHARACTER OUTPUT SUBROUTINE*****

12									
13									
14	013070	005067	165572		DOUT:	CLR	TOB		
15	013074	012704	000010			MOV	#10,R4		;SET NUMBER TO PRINT
16	013100	110367	165562			MOVB	R3,TOB		
17	013104	105777	165534		DOUT1:	TSTB	ATPS		
18	013110	100375				BPL	DOUT1		
19	013112	132767	000200	165546		BITB	#200,TOB		
20	013120	001404				BEQ	DOUT2		
21	013122	012777	000061	165516		MOV	#061,ATPB		
22	013130	000403				BR	DOUT3		
23	013132	012777	000060	165506	DOUT2:	MOV	#060,ATPB		
24	013140	006167	165522		DOUT3:	ROL	TOB		
25	013144	005304				DEC	R4		
26	013146	001356				BNE	DOUT1		
27	013150	000207				RTS	PC		
28	013152	016703	165566		DOUTD:	MOV	TEMP3,R3		
29	013156	000303				SWAB	R3		
30	013160	004767	177704			JSR	PC,DOUT		
31	013164	016703	165554			MOV	TEMP3,R3		
32	013170	004767	177674			JSR	PC,DOUT		
33	013174	000207				RTS	PC		

;TU16 SERIAL NUMBER PRINT SUBROUTINE*****

34									
35									
36									
37	013176	010304			SNPT:	MOV	R3,R4		
38	013200	000304				SWAB	R4		
39	013202	006004				ROR	R4		
40	013204	006004				ROR	R4		
41	013206	006004				ROR	R4		
42	013210	006004				ROR	R4		
43	013212	004767	000036			JSR	PC,SNPG		;GET FIRST DIGIT
44	013216	010304				MOV	R3,R4		;GO PRINT
45	013220	000304				SWAB	R4		
46	013222	004767	000026			JSR	PC,SNPG		;GET SECOND DIGIT
47	013226	010304				MOV	R3,R4		;GO PRINT
48	013230	006004				ROR	R4		
49	013232	006004				ROR	R4		
50	013234	006004				ROR	R4		
51	013236	006004				ROR	R4		
52	013240	004767	000010			JSR	PC,SNPG		;GET THIRD DIGIT
53	013244	010304				MOV	R3,R4		;GO PRINT
54	013246	004767	000002			JSR	PC,SNPG		;GET FOURTH DIGIT
55	013252	000207				RTS	PC		;GO PRINT
56	013254	012767	000260	165404	SNPG:	MOV	#260,TOB		;EXIT
57	013262	042704	177760			BIC	#177760,R4		;SET BASE = 0
58	013266	050467	165374			BIS	R4,TOB		;MASK DIGIT
59	013272	004767	177322			JSR	PC,TOG		;SET ASCII
60	013276	000207				RTS	PC		;TYPE DIGIT
61									;RETURN

;CKSWR ROUTINE THAT ALLOWS THE LOADING OF LOC.176, SWREG*****
;FROM THE TTY AT SELECTED POINTS WITHIN THE PROGRAM*****

62									
63									
64									
65	013300	022767	000176	165330	CKSWR:	CMP	#SWREG,SWR		;SOFTWARE SWITCH REG PRESENT
66	013306	001041				BNE	OUT		;NO, GET OUT
67	013310	105777	165324			TSTB	ATKS		;YES, WAIT FOR


```

;MESSAGE TABLE*****
2519
2520
2521 013614 041445 030523 020040 MSG1: .ASCII /%CS1 WC BA FC CS2 /
2522 013622 020040 041527 020040
2523 013630 020040 041040 020101
2524 013636 020040 020040 041506
2525 013644 020040 020040 041440
2526 013652 031123 020040 020040
2527 013660 051504 020040 020040 .ASCII /DS ER TC%/
2528 013666 042440 020122 020040
2529 013674 020040 041524 021445
2530 013702 051045 053505 047111 MSG2: .ASCII /%REWIND ERROR%/
2531 013710 020104 051105 047522
2532 013716 021522
2533 013720 022445 046524 031060 MSG3: .ASCII /%TMO2-TU16 BASIC FUNCTION TEST (DZTUB-E)%/
2534 013726 052055 030525 020066
2535 013734 040502 044523 020103
2536 013742 052506 041516 044524
2537 013750 047117 052040 051505
2538 013756 020124 042050 052132
2539 013764 041125 042455 022451
2540 013772 047105 042524 020122 .ASCII /ENTER CONDITIONS IN OCTAL%/
2541 014000 047503 042116 052111
2542 014006 047511 051516 044440
2543 014014 020116 041517 040524
2544 014022 022514 043
2545 014025 045 042522 044507 MSG4: .ASCII /%REGISTER START = %/
2546 014032 052123 051105 051440
2547 014040 040524 052122 036440
2548 014046 021440
2549 014050 053045 041505 047524 MSG5: .ASCII /%VECTOR = %/
2550 014056 020122 020075 043
2551 014063 045 047105 020104 MSG6: .ASCII /%END OF PASS %/
2552 014070 043117 050040 051501
2553 014076 020123 043
2554 014101 040 020077 043 MSG7: .ASCII / ? %/
2555 014105 045 043 MSG8: .ASCII /%#/
2556 014107 045 047520 044523 MSG9: .ASCII /%POSITION ERROR: %/
2557 014114 044524 047117 042440
2558 014122 051122 051117 020072
2559 014130 043
2560 014131 045 051104 053111 MSG10: .ASCII /%DRIVE NUMBER: %/
2561 014136 020105 052516 041115
2562 014144 051105 020072 043
2563 014151 045 046123 053101 MSG11: .ASCII /%SLAVE NUMBER: %/
2564 014156 020105 052516 041115
2565 014164 051105 020072 043
2566 014171 045 051127 052111 MSG12: .ASCII /%WRITE ERROR %/
2567 014176 020105 051105 047522
2568 014204 020122 043
2569 014207 045 042522 042101 MSG13: .ASCII /%READ REVERSE ERROR %/
2570 014214 051040 053105 051105
2571 014222 042523 042440 051122
2572 014230 051117 021440
2573 014234 051045 040505 020104 MSG14: .ASCII /%READ FORWARD ERROR %/
2574 014242 047506 053522 051101

```

2575	014250	020104	051105	047522			
2576	014256	020122	043				
2577	014261	045	051127	052111	MSG15:	.ASCII	/%WRITE TM ERROR #/
2578	014266	020105	046524	042440			
2579	014274	051123	051117	021440			
2580	014302	051045	053105	051105	MSG16:	.ASCII	/%REVERSE ERROR #/
2581	014310	04252	042440	051122			
2582	014316	051117	021440				
2583	014322	043045	051117	040527	MSG17:	.ASCII	/%FORWARD ERROR #/
2594	014330	042122	042440	051122			
2585	014336	051117	021440				
2586	014342	047040	055122	021440	MSG20:	.ASCII	/ NRZ #/
2587	014350	050040	020105	043	MSG21:	.ASCII	/ PE #/
2588	014355	040	054105	052120	MSG22:	.ASCII	/ EXPT: #/
2589	014362	020072	043				
2590	014365	040	041522	042126	MSG23:	.ASCII	/ RCVD: #/
2591	014372	020072	043				
2592	014375	045	052502	020123	MSG24:	.ASCII	/%BUS TRAP: #/
2593	014402	051124	050101	020072			
2594	014410	043					
2595	014411	045	041527	020072	MSG25:	.ASCII	/%MC: #/
2596	014416	043					
2597	014417	045	040502	020072	MSG26:	.ASCII	/%BA: #/
2598	014424	043					
2599	014425	045	041104	020072	MSG27:	.ASCII	/%DB: #/
2600	014432	043					
2601	014433	045	047111	052111	MSG28:	.ASCII	/%INIT DID NOT CLEAR RH #/
2602	014440	042040	042111	047040			
2603	014446	052117	041440	042514			
2604	014454	051101	051040	020110			
2605	014462	043					
2606	014463	045	041523	047040	MSG29:	.ASCII	/%SC NOT RESET BY INIT #/
2607	014470	052117	051040	051505			
2608	014476	052105	041040	020131			
2609	014504	047111	052111	021440			
2610	014512	052045	042522	047040	MSG30:	.ASCII	/%TRE NOT RESET BY INIT #/
2611	014520	052117	051040	051505			
2612	014526	052105	041040	020131			
2613	014534	047111	052111	021440			
2614	014542	041445	031123	047040	MSG31:	.ASCII	/%CS2 NOT RESET BY INIT #/
2615	014550	052117	051040	051505			
2616	014556	052105	041040	020131			
2617	014564	047111	052111	021440			
2618	014572	042045	052114	047040	MSG32:	.ASCII	/%DLT NOT SET #/
2619	014600	052117	051440	052105			
2620	014606	021440					
2621	014610	051445	020103	047516	MSG33:	.ASCII	/%SC NOT SET #/
2622	014616	020124	042523	020124			
2623	014624	043					
2624	014625	045	051124	020105	MSG34:	.ASCII	/%TRE NOT SET #/
2625	014632	047516	020124	042523			
2626	014640	020124	043				
2627	014643	045	051111	047040	MSG35:	.ASCII	/%IR NOT SET BY INIT #/
2628	014650	052117	051440	052105			
2629	014656	041040	020131	047111			
2630	014664	052111	021440				

2631	014670	047445	020122	047516	MSG36: .ASCII	/%OR NOT RESET BY INIT #/
2632	014676	020124	042522	042523		
2633	014704	020124	054502	044440		
2634	014712	044516	020124	043		
2635	014717	045	051117	047040	MSG37: .ASCII	/%OR NOT RESET BY 1 SILO ENTRY #/
2636	014724	052117	051040	051505		
2637	014732	052105	041040	020131		
2638	014740	020061	044523	047514		
2639	014746	042440	052116	054522		
2640	014754	021440				
2641	014756	047445	020122	047516	MSG38: .ASCII	/%OR NOT SET BY SILO FULL #/
2642	014764	020124	042523	020124		
2643	014772	054502	051440	046111		
2644	015000	020117	052506	046114		
2645	015006	021440				
2646	015010	041045	042101	051440	MSG39: .ASCII	/%BAD SILO READ #/
2647	015016	046111	020117	042522		
2648	015024	042101	021440			
2649	015030	044445	020122	047516	MSG40: .ASCII	/%IR NOT RESET BY SILO FULL#/
2650	015036	020124	042522	042523		
2651	015044	020124	054502	051440		
2652	015052	046111	020117	052506		
2653	015060	046114	043			
2654	015063	040	047516	026516	MSG41: .ASCII	/ NON-EXIST DRIVE#/
2655	015070	054105	051511	020124		
2656	015076	051104	053111	021505		
2657	015104	047040	047117	042455	MSG42: .ASCII	/ NON-EXIST SLAVE#/
2658	015112	044530	052123	051440		
2659	015120	040514	042526	043		
2660	015125	040	042523	044522	MSG43: .ASCII	/ SERIAL NO: #/
2661	015132	046101	047040	035117		
2662	015140	021440				
2663	015142	042445	040522	042523	MSG44: .ASCII	/%ERASE HEAD INOPERATIVE/
2664	015150	044040	040505	020104		
2665	015156	047111	050117	051105		
2666	015164	052101	053111	105		
2667	015171	045	044103	041505	.ASCII	/%CHECK POLARITY#/
2668	015176	020113	047520	040514		
2669	015204	044522	054524	043		
2670	015211	045	051105	051501	MSG45: .ASCII	/%ERASE HEAD POLARITY WRONG#/
2671	015216	020105	042510	042101		
2672	015224	050040	046117	051101		
2673	015232	052111	020131	051127		
2674	015240	047117	021507			
2675	015244	051445	052105	052455	MSG46: .ASCII	/%SET-UP WRITE ERROR#/
2676	015252	020120	051127	052111		
2677	015260	020105	051105	047522		
2678	015266	021522				
2679	015270	051445	040520	042503	MSG47: .ASCII	/%SPACE FORWARD ERROR#/
2680	015276	043040	051117	040527		
2681	015304	042122	042440	051122		
2682	015312	051117	043			
2683	015315	045	050123	041501	MSG48: .ASCII	/%SPACE REVERSE ERROR#/
2684	015322	020105	042522	042526		
2685	015330	051522	020105	051105		
2686	015336	047522	021522			

2687	015342	041045	043125	042506	MSG49: .ASCII	/%BUFFERED WRITE ERROR#/ 051127
2688	015350	042522	020104	051127		
2689	015356	052111	020105	051105		
2690	015364	047522	021522			
2691	015370	041045	052117	051440	MSG50: .ASCII	/%BOT SET AFTER BUFFERED WRITE#/ 052106
2692	015376	052105	040440	052106		
2693	015404	051105	041040	043125		
2694	015412	042506	042522	020104		
2695	015420	051127	052111	021505		
2696	015426	047045	020117	047502	MSG51: .ASCII	/%NO BOT FROM READ IN PRESET#/ 046517
2697	015434	020124	051106	046517		
2698	015442	051040	040505	020104		
2699	015450	047111	050040	042522		
2700	015456	042523	021524			
2701	015462	052045	020103	047111	MSG52: .ASCII	/%TC INCORRECT #/ 041505
2702	015470	047503	051122	041505		
2703	015476	020124	043			
2704	015501	045	046123	053101	MSG53: .ASCII	/%SLAVE NOT OFF LINE#/ 020124
2705	015506	020105	047516	020124		
2706	015514	043117	020106	044514		
2707	015522	042516	043			
2708	015525	045	051045	051505	MSG54: .ASCII	/%%RESET SLAVE TO ON LINE BEFORE CONTINUING#/ 040514
2709	015532	052105	051440	040514		
2710	015540	042526	052040	020117		
2711	015546	047117	046040	047111		
2712	015554	020105	042502	047506		
2713	015562	042522	041440	047117		
2714	015570	044524	052516	047111		
2715	015576	021507				
2716	015600	047045	055122	047440	MSG55: .ASCII	/%NRZ ONLY: #/ 021440
2717	015606	046116	035131	021440		
2718	015614	044440	042524	035122	MSG56: .ASCII	/ ITER: #/ 042520
2719	015622	021440				
2720	015624	052045	020115	047516	MSG57: .ASCII	/%TM NOT SET#/ 021524
2721	015632	020124	042523	021524		
2722	015640	042445	052111	042510	MSG60: .ASCII	/%EITHER TAPE NOT ERASED OR OPI PROBLEM#/ 042520
2723	015646	020122	040524	042520		
2724	015654	047040	052117	042440		
2725	015662	040522	042523	020104		
2726	015670	051117	047440	044520		
2727	015676	050040	047522	046102		
2728	015704	046505	043			
2729	015707	045	044122	030461	MSG61: .ASCII	/%RH11 OR RH70: #/ 044122
2730	015714	047440	020122	044122		
2731	015722	030067	020072	043		
2732	015727	045	044122	047440	MSG62: .ASCII	/%RH ONLY: #/ 021440
2733	015734	046116	035131	021440		
2734						

```

; TEST HEADERS*****
2735
2736
2737 015742 022445 052106 035061 MSFT1: .ASCII /%%FT1:RH ADDRESSING #/
2738 015750 044122 040440 042104
2739 015756 042522 051523 047111
2740 015764 020107 043
2741 015767 045 043045 031124 MSFT2: .ASCII /%%FT2:RH REGISTER BITS TEST #/
2742 015774 051072 020110 042522
2743 016002 044507 052123 051105
2744 016010 041040 052111 020123
2745 016016 042524 052123 021440
2746 016024 022445 052106 035063 MSFT3: .ASCII /%%FT3:RH INITIALIZE TEST #/
2747 016032 044122 044440 044516
2748 016040 044524 046101 055111
2749 016046 020105 042524 052123
2750 016054 021440
2751 016056 022445 052106 035064 MSFT4: .ASCII /%%FT4:RH11 SILO TEST 1 #/
2752 016064 044122 030461 051440
2753 016072 046111 020117 042524
2754 016100 052123 030440 021440
2755 016106 022445 052106 035065 MSFT5: .ASCII /%%FT5:RH11 SILO TEST 2 #/
2756 016114 044122 030461 051440
2757 016122 046111 020117 042524
2758 016130 052123 031040 021440
2759 016136 022445 052106 035066 MSFT6: .ASCII /%%FT6:RH11 SILO TEST 3 #/
2760 016144 044122 030461 051440
2761 016152 046111 020117 042524
2762 016160 052123 031440 021440
2763 016166 022445 052106 035067 MSFT7: .ASCII /%%FT7:RH11 SILO TEST 4 #/
2764 016174 044122 030461 051440
2765 016202 046111 020117 042524
2766 016210 052123 032040 021440
2767 016216 022445 052106 030061 MSFT10: .ASCII /%%FT10:RH11 SILO TEST 5 #/
2768 016224 051072 030510 020061
2769 016232 044523 047514 052040
2770 016240 051505 020124 020065
2771 016246 043
2772 016247 045 043045 030524 MSFT11: .ASCII /%%FT11:NOP TEST#/
2773 016254 035061 047516 020120
2774 016262 042524 052123 043
2775 016267 045 043045 030524 MSFT12: .ASCII /%%FT12:REWIND TEST#/
2776 016274 035062 042522 044527
2777 016302 042116 052040 051505
2778 016310 021524
2779 016312 022445 052106 031461 MSFT13: .ASCII /%%FT13:WRITE-READ TEST#/
2780 016320 053472 044522 042524
2781 016326 051055 040505 020104
2782 016334 042524 052123 043
2783 016341 045 043045 030524 MSFT14: .ASCII /%%FT14:SPACE TEST#/
2784 016346 035064 050123 041501
2785 016354 020105 042524 052123
2786 016362 043
2787 016363 045 043045 030524 MSFT15: .ASCII /%%FT15:ERASE TEST#/
2788 016370 035065 051105 051501
2789 016376 020105 042524 052123
2790 016404 043

```

2791	016405	045	043045	030524	MSFT16: .ASCII	/%FT16:TAPE MARK WRITE-READ TEST#/ #/
2792	016412	035066	040524	042520		
2793	016420	046440	051101	020113		
2794	016426	051127	052111	026505		
2795	016434	042522	042101	052040		
2796	016442	051505	021524			
2797	016446	022445	052106	033461	MSFT17: .ASCII	/%FT17:TM SPACE TEST #/ #/
2798	016454	052072	020115	050123		
2799	016462	041501	020105	042524		
2800	016470	052123	021440			
2801	016474	022445	052106	030062	MSFT20: .ASCII	/%FT20:WRITE CHECK TEST #/ #/
2802	016502	053472	044522	042524		
2803	016510	041440	042510	045503		
2804	016516	052040	051505	020124		
2805	016524	043				
2806	016525	045	043045	031124	MSFT21: .ASCII	/%FT21:ERASE HEAD TEST#/ #/
2807	016532	035061	051105	051501		
2808	016540	020105	042510	042101		
2809	016546	052040	051505	021524		
2810	016554	022445	052106	031062	MSFT22: .ASCII	/%FT22:BUFFERED COMMAND TEST#/ #/
2811	016562	041072	043125	042506		
2812	016570	042522	020104	047503		
2813	016576	046515	047101	020104		
2814	016604	042524	052123	043		
2815	016611	045	043045	031124	MSFT23: .ASCII	/%FT23:READ IN PRESET TEST#/ #/
2816	016616	035063	042522	042101		
2817	016624	044440	020116	051120		
2818	016632	051505	052105	052040		
2819	016640	051505	021524			
2820	016644	022445	052106	032062	MSFT24: .ASCII	/%FT24:REWIND-OFF LINE TEST#/ #/
2821	016652	051072	053505	047111		
2822	016660	026504	043117	020106		
2823	016666	044514	042516	052040		
2824	016674	051505	021524			
2825	016700	057045	021507		\$CNTG: .ASCII	/%G#/ #/
2826	016704	051445	051127	020075	\$MSWR: .ASCII	/%SWR= #/ #/
2827	016712	043				
2828	016713	040	047040	053505	\$MNEW: .ASCII	/ NEW= #/ #/
2829	016720	020075	043			
2830	016723	077	021445		\$QUEST: .ASCII	/?%#/ #/
2831						
2832						
2833						
2834	016726	000000			WDATA: 0	.EVEN
2835		020440				.+.1510
2836	020440	000000			RDATA: 0	
2837						
2838		000001				.END

MSFT16	016405	1730	2791#			
MSFT17	016446	1766	2797#			
MSFT2	015767	1236	2741#			
MSFT20	016474	1830	2801#			
MSFT21	016525	1866	2806#			
MSFT22	016554	1910	2810#			
MSFT23	016611	1945	2815#			
MSFT24	016644	1984	2820#			
MSFT3	016024	1301	2746#			
MSFT4	016056	1343	2751#			
MSFT5	016106	1368	2755#			
MSFT6	016136	1402	2759#			
MSFT7	016166	1457	2763#			
MSG1	013614	2086	2521#			
MSG10	014131	1100	2560#			
MSG11	014151	1115	2563#			
MSG12	014171	1549	1738	1792	1876	2566#
MSG13	014207	1559	1744	2569#		
MSG14	014234	1566	1749	1892	2573#	
MSG15	014261	1774	2577#			
MSG16	014302	1672	1802	1844	2580#	
MSG17	014322	1670	1816	1849	2583#	
MSG2	013702	2042	2530#			
MSG20	014342	1663	2072	2586#		
MSG21	014350	1666	2075	2587#		
MSG22	014355	1281	1439	1674	2588#	
MSG23	014365	1285	1443	1679	2590#	
MSG24	014375	2239	2592#			
MSG25	014411	1244	2595#			
MSG26	014417	1253	2597#			
MSG27	014425	1264	2599#			
MSG28	014433	2601#				
MSG29	014463	1309	2606#			
MSG3	013720	1063	2533#			
MSG30	014512	1313	2610#			
MSG31	014542	1319	2614#			
MSG32	014572	1354	1468	1494	2618#	
MSG33	014610	1356	2621#			
MSG34	014625	1358	2624#			
MSG35	014643	1373	2627#			
MSG36	014670	1377	2631#			
MSG37	014717	1382	1390	2635#		
MSG38	014756	1416	2641#			
MSG39	015010	1437	2646#			
MSG4	014025	1065	2545#			
MSG40	015030	1412	2649#			
MSG41	015063	1112	2654#			
MSG42	015104	1127	2657#			
MSG43	015125	1130	2660#			
MSG44	015142	1901	2663#			
MSG45	015211	2670#				
MSG46	015244	1600	1708	1840	2675#	
MSG47	015270	1628	2679#			
MSG48	015315	1618	2683#			
MSG49	015342	1929	2687#			
MSG5	014050	1074	2549#			

TSRH	002520	1099	1166#											
TSTBL	001040	994#	1166	1198										
TTIN	012432	2274	2315#	2487										
TTINT	012116	867	2224#											
TTINI	012452	2319#	2320											
TTIN2	012466	2322#	2323											
TTOUT	012504	1064	1066	1075	1101	1113	1116	1128	1131	1135	1142	1151	1204	1277
		1280	1282	1286	1328	1331	1435	1438	1440	1444	1660	1662	1667	1673
		1675	1680	1996	2041	2043	2068	2071	2076	2079	2083	2087	2238	2240
		2309	2329#	2335	2347	2474	2476	2480	2506					
TTR	012246	1073	1082	1106	1121	1140	1147	1156	2272#					
TTR0	012254	2274#	2296											
TTR1	012302	2276	2280#											
TTR1A	012316	2281	2283#											
TTR1B	012332	2284	2286#											
TTR2	012370	2279	2297#											
TTR3	012400	2298	2300#											
TTR4	012410	2301	2303#											
TTR5	012412	2278	2304#											
WDES	001004	974#	1502*	1543*	1572	1574*	1575	1586*	1664	1690	1692*	1704*	1731*	1752
		1756#	1767*	1819	1823*	1837*	1854	1858*	1873*	1913*	1947*	2003	2056	
VECT	000656	929#	1076	1078	1083									
WC	000602	904#	1239	2004*	2090									
WCNT	000710	944#	1504*	1522*	1546*	1591*	1608*	1716*	1734*	1785*	1836*	1872*	1883*	1889*
		1897*	1916*	1950*	2004									
WDATA	016726	1523	1547	1589*	1787	1834	1870	1914	1948	2127	2135	2834#		
SCNTG	016700	2473	2825#											
SMNEW	016713	2479	2828#											
SMSWR	016704	2475	2826#											
SQUEST	016723	2505	2830#											
SREAD	013414	2482	2485#											
.	= 020442	606#	607	609	611	613	615	617	619	621	623	625	627	629
		631	633	635	637	639	641	643	645	647	649	651	653	655
		657	659	661	663	665	667	669	671	673	675	677	679	681
		683	685	687	689	691	693	695	697	699	701	703	705	707
		709	711	713	715	717	719	721	723	725	727	729	731	733
		735	737	739	741	743	745	747	749	751	753	755	757	759
		761	763	765	767	769	771	773	775	777	779	781	783	785
		787	789	791	793	795	797	799	801	803	805	807	809	811
		813	815	817	819	821	823	825	827	829	831	833	835	837
		839	841	843	845	847	849	851	853	855	857	859	861	866#
		872#	884#	888#	895#	900#	1040#	2835#						

COMMEN	1#
ENDCOM	1#
ESCAPE	1#
GETPRI	1#
GETSWR	1#
MULT	1#
NEWTST	1#
POP	1#
PUSH	1#
REPORT	1#
SETPRI	1#
SETUP	1#
SKIP	1#
SLASH	1#
STARS	1#
SWRSU	1#
TYPBIN	1#
TYPDEC	1#
TYPNAM	1#
TYPNUM	1#
TYPOCS	1#
TYPOCT	1#
TYPTXT	1#
SSESCA	1#
SSNEWT	1#
SSSKIP	1#
.EQUAT	1#
.HEADE	1#
.KT11	1#
.SETUP	1#
.SWRHI	1#
.SACT1	1#
.SAPT8	1#
.SAPT1	1#
.SAPTY	1#
.SASTA	1#
.SCATC	1#
.SCMTA	1#
.SOB2D	1#
.SOB20	1#
.SDIV	1#
.SEOP	1#
.SERRO	1#
.SERRT	1#
.SMULT	1#
.SPOWE	1#
.SRAND	1#
.SRDDE	1#
.SRDOC	1#
.SREAD	1#
.SR2AZ	1#
.SSAVE	1#
.SSB2D	1#
.SSB20	1#
.SSCOP	1#
.SSIZE	1#

H07

TM02 TU16 BASIC FUNCTION TEST MACY11 27(732) 11-OCT-76 13:04 PAGE 88
DZTUBE.P11 CROSS REFERENCE TABLE -- MACRO NAMES

.SSUPR 1#
.STRAP 1#
.STYPB 1#
.STYPD 1#
.STYPE 1#
.STYPO 1#
.S4OCA 1#
.1170 1#

