

TM02-TU16

TM02-TU16/TE16
MD-11-DZTUB-F

EP-DZTUB-F-DL-F
COPYRIGHT © 74-77
FICHE 1 OF 1

OCT 1977
digital
MADE IN USA

This microfiche card contains a grid of frames. The frames are arranged in approximately 10 rows and 6 columns. Each frame contains a small, high-contrast image of a document page, likely a technical drawing or data table. The images are very small and difficult to read, but they appear to be organized in a structured manner. The right half of the card is mostly blank, with some faint markings and a small grid of frames in the bottom right corner.

B01

EOF1020AEDS801

00010000

770920

PDP10 IDENTIFICATION BRIDZTUBFSEQ

00010000

770920
SEQ 0001

PRODUCT CODE: MAINDEC-11-DZTUB-F-D
PRODUCT TITLE: TMO2-TU16/TE16 BASIC FUNCTION TEST
DATE CREATED: 15 AUGUST 1977
MAINTAINER: DIAGNOSTIC ENGINEERING

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

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

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

COPYRIGHT (C) 1974, 1977 BY DIGITAL EQUIPMENT CORPORATION.

TABLE OF CONTENTS

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

1. ABSTRACT

THIS PROGRAM IS INTENDED TO TEST ALL OF THE BASIC FUNCTIONAL LEVEL OPERATIONS OF THE TMO2-TU16/TE16 MAG TAPE SYSTEM. ALL FUNCTIONS; WRITE, READ, SPACE, ERASE, REWIND, ETC; WILL BE TESTED. IN ADDITION TO THE TMO2-TU16/TE16 TESTS, THE RH WILL BE TESTED SEPARATELY IN SO FAR AS IT IS POSSIBLE TO SEPARATE THE RH FROM THE TMO2-TU16/TE16 ITSELF.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR - WITH OR WITHOUT A HARDWARE SWITCH REGISTER
- B. BK OF CORE
- C. CONSOLE TTY
- D. TMO2 MAGTAPE CONTROLLER
- E. MASS BUS CONTROLLER
- F. TU16 OR TE16 MAG TAPE TRANSPORT

3. LOADING PROCEEDURE

USE STANDARD BINARY LOADING PROCEEDURE

4. STARTING PROCEEDURE

***SOFTWARE SWITCH REGISTER IS LOCATED AT LOC. 176(REFER TO SECTION 5 FOR MORE DETAIL)

THERE ARE TWO (2) STARTING ADDRESSES THAT MAYBE USED: 200(8) AND 210(8)

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

E01

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

SEQ 0004

***SWR=XXXXXX NEW= WILL BE PRINTED FIRST IF SOFTWARE SWITCH REGISTER IS SELECTED.
(REFER TO SECTION 5 FOR OPERATOR ACTION)
TMO2-TU16/TE16 BASIC FUNCTION TEST
ENTER CONDITIONS IN OCTAL

REGISTER START: 172440 (CR)
VECTOR: 224 (CR)
DRIVE NUMBER: 0 (3)
SLAVE NUMBER: 0 (6) SERIAL NO: 200
RH11 OR RH70: (0)
RH ONLY: (0)
NRZ ONLY: (1) -NRZ (NON-RETURN-TO ZERO) IS THE METHOD OF RECORDING
ON MAGNETIC TAPE.

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

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

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 EVERY HALT.
- 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.

GO1

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.

SEQ 0006

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.

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

7. OPERATION

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

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE
****REFER TO SECTION 5 FOR DYNAMIC LOADING
OF SOFTWARE SWITCH REGISTER.***
3. PRESS START
4. ENTER APPROPRIATE RESPONSES TO THE TTY REQUESTS

ALL HARDWARE SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME. THE NORMAL, OR DEFAULT, OPERATING SEQUENCE IS ALL SWITCHES DOWN (ZERO). THE END OF EACH PASS IS NOTED BY A MESSAGE STATING END OF PASS AND THE NUMBER OF THAT PASS.
*****FOR THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER REFER TO SECTION 5 *****

SINGLE TEST SELECTION: (SW0-SW4)

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

RH11 OR RH70 OPTION:

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

RH ONLY OPTION:

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

B. 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-TUI16/TE16 ITSELF. (SEE RH ONLY OPTION; PAR 7)

FT1: RH ADDRESSING: THIS TEST WILL ASSURE THAT THE RH WILL RESPOND WITHOUT CAUSING A BUS TRAP TO ALL 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. ****

B. TMO2-TU16/TE16 BASIC FUNCTIONS: THE FOLLOWING FOURTEEN (14) TESTS WILL ASSURE OPERATION OF THE MAG TAPE BASIC FUNCTIONS.

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

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

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

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

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

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

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 RECORD SIZE.
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 COUNT EACH TIME) UNTIL ALL POSITIONS HAVE BEEN CHECKED. IF POSITION IS LOST; TEST ENDS.
8. REPEAT STEPS 1 THRU 7 FOR PE.
9. END

FT15: ERASE TEST: THIS TEST WILL ASSURE THAT THE ERASE FUNCTION WILL INDEED ERASE TAPES.

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

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

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

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

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

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

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

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

FT23: READ IN PRESET: THIS TEST WILL ASSURE THAT UNIT 0
IS REWIND 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 SELECTED: SW 11 = 1)

B02

9.

LISTING

SEQ 0014

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

.TITLE TM02-TU16/TE16 BASIC FUNCTION TEST
;MAINDEC-11-DZTUB-F-D
;15 AUGUST 1977
;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


```

72                                     ;REGISTER EQUIVS*****
73
74         000000                       RO=%0
75         000001                       R1=%1
76         000002                       R2=%2
77         000003                       R3=%3
78         000004                       R4=%4
79         000005                       R5=%5
80         000006                       SP=%6
81         000007                       PC=%7
82
83                                     ;TRAP CATCHERS*****
84
85         000000                       .=0
86         000200                       .REPT 200
87
88         .+2
89         HALT
90         .ENDR
91
92                                     ;TTY INTERRUPT VECTOR*****
93
94         000060 012144                 .=60
95         000062 000000                 TTINT          ;TTY INTERRUPT HEADER ADDRESS
96
97
98                                     ;SOFTWARE SWITCH REGISTER*****
99
100        000176 000000                 .=176
101
102                                     SWREG: 0          ;SOFTWARE SWITCH REGISTER
103
104
105                                     ;*****
106                                     ;THIS PROGRAM SUPPORTS THE SOFTWARE SWITCH REGISTER LOC.176.
107                                     ;REFER TO SECTION 5 OF DOCUMENT FOR DESCRIPTION
108                                     ;*****
109                                     ;START ADDRESS*****
110
111        000200 000200                 .=200
112        000200 005000                 CLR          RO
113        000202 000167 001372         JMP          START ;PROGRAM START
114
115        000210 000210                 .=210
116        000210 000240                 NOP
117        000212 012700 000001         MOV          #1,RO ;SET NO HEADER FLAG
118        000216 000167 001356         JMP          START
119
120                                     ;TM02 INTERRUPT VECTOR*****
121
122        000224 000224                 .=224
123        000224 012130                 MTINT
124        000226 000340                 340          ;TAPE INTERRUPT HANDLER ADDRESS
125

```

```

126
127          000600          .=-600
128                                     ;MASS BUS REGISTER EQUIVS*****
129
130 000600 172440  C1: 172440
131 000602 172442  WC: 172442
132 000604 172444  BA: 172444
133 000606 172446  FC: 172446
134 000610 172450  CS: 172450
135 000612 172452  DS: 172452
136 000614 172454  ER: 172454
137 000616 172456  AS: 172456
138 000620 172460  CC: 172460
139 000622 172462  DB: 172462
140 000624 172464  MR: 172464
141 000626 172466  DT: 172466
142 000630 172470  SN: 172470
143 000632 172472  TC: 172472
144
145                                     ;CONSTANTS*****
146
147 000634 177776  PSW: 177776          ;PROCESSOR STATUS
148 000636 177570  SWR: 177570          ;SWITCH REGISTER
149 000640 177560  TKS: 177560          ;TTY READER STATUS
150 000642 177562  TKB: 177562          ;TTY READ BUFFER
151 000644 177564  TPS: 177564          ;TTY PUNCH STATUS
152 000646 177566  TPB: 177566          ;TTY PUNCH BUFFER
153 000650 177777  SERNUM: 177777       ;SERIAL NUMBER
154 000652 000011  DRVTP: 011           ;DRIVE TYPE
155 000654 000010  ITAMT: 10            ;ITERATION AMOUNT
156 000656 000224  VECT: 224            ;INTERRUPT VECTOR(RH)
157 000660 172440  REGS: 172440         ;STARTING REGISTER ADDRESS
158 000662 000004  BTRP: 4              ;BUS TRAP ADDRESS
159 000664 000006  BTRP2: 6             ;BUS TRAP PRIORITY LEVEL 7

```

160		
161		
162	000666	000000
163	000670	000000
164	000672	000000
165	000674	000000
166	000676	000000
167	000700	000000
168	000702	000000
169	000704	000000
170	000706	000000
171	000710	000000
172	000712	000000
173	000714	000000
174	000716	000000
175	000720	000000
176	000722	000000
177	000724	000000
178	000726	000000
179	000730	000000
180	000732	000000
181	000734	000000
182	000736	000000
183	000740	000000
184	000742	000000
185	000744	000000
186	000746	000000
187	000750	000000
188	000752	000000
189	000754	000000
190	000756	000000
191	000760	000000
192	000762	000000
193	000764	000000
194	000766	000000
195	000770	000000
196	000772	000000
197	000774	000000
198	000776	000000
199	001000	000000
200	001002	000000
201	001004	000000
202	001006	000000
203	001010	000000
204	001012	000000
205	001014	000000
206	001016	000000
207	001020	000000
208	001022	000000
209	001024	000000

;FLAGS AND COUNTERS*****

TOB:	0
TIB:	0
RH17F:	0
HDRFL:	0
EMADDR:	0
DRVN:	0
SLVN:	0
BADDR:	0
FCNT:	0
WCNT:	0
RCNT:	0
ERRP:	0
ERRP1:	0
RRD:	0
RFD:	0
RDYDX:	0
OPDYX:	0
SCNT:	0
PFLG:	0
RTRN:	0
ERADD:	0
TEMP1:	0
TEMP2:	0
TEMP3:	0
STMSK:	0
ITCNT:	0
DSAV:	0
SAV1:	0
SAV2:	0
SAV3:	0
SCOLP:	0
ITRLP:	0
EXFL:	0
PEXFL:	0
STFLG:	0
LTADD:	0
FUN:	0
SERFL:	0
CRCNT:	0
UDES:	0
PATRN:	0
RHTF:	0
NRZOF:	0
RHOF:	0
PCNTR:	0
TEMPST:	0
COUNT:	0
RDSW:	0

;DATA PATTERN GENERATORS*****

210		
211		
212		
213	001026	000000
214	001030	011706
215	001032	011730

DATBL:	0	
DAT0:	DAT1	;ALL ONE BITS
DAT1:	DAT2	;ALL ZERO BITS

H02

TMO2-TU16/TE16 BASIC FUNCTION TEST
DZTUBF.P11 15-AUG-77 09:41

MACY11 30(1046) 15-AUG-77 10:47 PAGE 6

SEQ 0020

216 001034 011736
217 001036 011746

DATA2: DAT3 ;ALTERNATING ONE/ZERO BITS
DATA3: DAT4 ;ALL BITS 0-377

218
 219
 220
 221 001040 000000
 222 001042 000000
 223 001044 002774
 224 001046 002774
 225 001050 003074
 226 001052 003074
 227 001054 003420
 228 001056 003420
 229 001060 003636
 230 001062 003636
 231 001064 003764
 232 001066 003764
 233 001070 004166
 234 001072 004166
 235 001074 004454
 236 001076 004454
 237 001100 004550
 238 001102 004550
 239 001104 004704
 240 001106 004704
 241 001110 005022
 242 001112 005022
 243 001114 005134
 244 001116 005134
 245 001120 005466
 246 001122 005466
 247 001124 006362
 248 001126 006362
 249 001130 006562
 250 001132 006562
 251 001134 007024
 252 001136 007024
 253 001140 007440
 254 001142 007440
 255 001144 007700
 256 001146 007700
 257 001150 010230
 258 001152 010230
 259 001154 010434
 260 001156 010434
 261 001160 010654
 262 001162 010654
 263 001164 000000
 264 001166 000000
 265 001170 000000
 266 001172 000000

;LOGIC TEST ENTRY TABLE*****

TSTTBL: 0
 0
 FT1
 FT1
 FT2
 FT2
 FT3
 FT3
 FT4
 FT4
 FT5
 FT5
 FT6
 FT6
 FT7
 FT7
 FT10
 FT10
 FT11
 FT11
 FT12
 FT12
 FT13
 FT13
 FT14
 FT14
 FT15
 FT15
 FT16
 FT16
 FT17
 FT17
 FT20
 FT20
 FT21
 FT21
 FT22
 FT22
 FT23
 FT23
 FT24
 FT24
 0
 0
 0
 0

```

267          001600          . = 1600
268                                     ;PROGRAM START AND HOUSEKEEPING*****
269
270 001600 000240          START:  NOP
271 001602 012777 000340 177024  MOV    #340, @PSW          ;SET PRIORITY
272 001610 012706 000500          MOV    #500, SP          ;SET STACK POINTER
273
274 001614 013746 000006          SUSWR:  MOV    @#6, -(SP)          ;SAVE VECTORS
275 001620 013746 000004          MOV    @#4, -(SP)
276 001624 012737 001644 000004  MOV    #15, @#4          ;SET UP FOR TIMEOUT
277 001632 022777 177777 176776  CMP    #-1, @SWR          ;REFERENCE HARDWARE SWITCH REGISTER
278 001640 001402          BEQ    2$
279 001642 000404          BR     3$
280 001644 022626          1$:   CMP    (SP)+, (SP)+          ;ADJUST STACK
281 001646 012767 000176 176762  2$:   MOV    #SWREG, SWR          ;POINT TO SOFTWARE SWITCH REG
282 001654 012637 000004          3$:   MOV    (SP)+, @#4          ;RESTORE VECTORS
283 001660 012637 000006          MOV    (SP)+, @#6
284 001664 023727 000636 000176  CMP    @#SWR, #SWREG          ;IS SOFTWARE REG USED
285 001672 001002          BNE    4$          ;BRANCH IF NO
286 001674 004767 011476          JSR    PC, CNTLU          ;ALLOW SOFTWARE SWITCH REGISTER TO BE CHANGED
287 001700 005700          4$:   TST    R0          ;SEE IF PRINT HEADER
288 001702 001402          BEQ    STOA          ;IF SO: BR
289 001704 000167 000562          JMP    ST4          ;ELSE SKIP
290 001710 012704 013744          STOA:  MOV    #MSG3, R4
291 001714 004767 010610          JSR    PC, TTOUT          ;PRINT TITLE
292 001720 012704 014056          STOB:  MOV    #MSG4, R4
293 001724 004767 010600          JSR    PC, TTOUT          ;REQUEST REGISTER ADDRESS
294 001730 016703 176724          MOV    REGS, R3
295 001734 004767 010734          JSR    PC, OCTP          ;PRINT CURRENT ADDRESS
296 001740 012705 000660          MOV    #REGS, R5          ;SET ADDRESS SAVE LOC
297 001744 012701 000006          MOV    #6, R1          ;SET SIZE OF RESPONSE
298 001750 012702 176400          MOV    #176400, R2          ;SET UPPER LIMIT
299 001754 012703 172300          MOV    #172300, R3          ;SET LOWER LIMIT
300 001760 004767 010310          JSR    PC, TTR          ;GO GET RESPONSE
301 001764 012704 014101          MOV    #MSG5, R4
302 001770 004767 010534          JSR    PC, TTOUT          ;REQUEST VECTOR
303 001774 016703 176656          MOV    VECT, R3
304 002000 004767 010670          JSR    PC, OCTP          ;PRINT CURRENT VECTOR
305 002004 012705 000656          MOV    #VECT, R5          ;SET ADDRESS SAVE LOC
306 002010 012701 000003          MOV    #3, R1          ;SET SIZE OF RESPONSE
307 002014 012702 000224          MOV    #224, R2          ;SET UPPER LIMIT
308 002020 012703 000150          MOV    #150, R3          ;SET LOWER LIMIT
309 002024 004767 010244          JSR    PC, TTR          ;GO GET RESPONSE
310 002030 016700 176622          MOV    VECT, R0          ;GET VECTOR
311 002034 012720 012130          MOV    #MTINT, (R0)+          ;LOAD INTERRUPT ADDRESS IN VECTOR
312 002040 012710 000340          MOV    #340, (R0)          ;LOAD PRIORITY
313 002044 016700 176610          MOV    REGS, R0          ;GET START OF REGS
314 002050 012701 000016          MOV    #16, R1          ;SET NUMBER OF REGS
315 002054 012702 000600          MOV    #C1, R2          ;GET START OF TABLE
316 002060 010022          STO:  MOV    R0, (R2)+          ;BUILD TABLE
317 002062 062700 000002          ADD    #2, R0          ;BUMP ADDRESS
318 002066 005301          DEC    R1          ;SEE IF DONE
319 002070 001373          BNE    STO          ;IF NOT: BR
320 002072 012702 000666          MOV    #TOB, R2
321 002076 012700 000054          MOV    #54, R0
322 002102 005022          ST1:  CLR    (R2)+          ;CLEAR FLAGS + COUNTERS

```

323	002104	005300		DEC	RO	
324	002106	001375		BNE	ST1	
325	002110	012767	000001 176672	MOV	#1,RHTF	;SET ADDRESS TEST FLAG
326	002116	000167	000376	JMP	TSRH	;GO DO INITIAL ADDRESS TEST PASS
327	002122	012704	014162	ST1A: MOV	#MSG10,R4	
328	002126	004767	010376	JSR	PC,TTOUT	;REQUEST DRIVE NUMBER
329	002132	012705	000700	MOV	#DRVN,R5	;SET ADDRESS OF DRIVE NUMBER SAVE
330	002136	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
331	002142	012702	000007	MOV	#7,R2	;SET UPPER LIMIT
332	002146	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
333	002152	004767	010116	JSR	PC,TTR	;GO GET RESPONSE
334	002156	012777	000040 176424	MOV	#40,ACS	;SET INIT
335	002164	056777	176510 176416	BIS	DRVN,ACS	;SET DRIVE NUMBER
336	002172	005777	176402	TST	AC1	;ACCESS DRIVE
337	002176	032777	010000 176404	BIT	#10000,ACS	;SEE IF NED
338	002204	001405		BEQ	ST2	;IF NOT: BR
339	002206	012704	015114	MOV	#MSG41,R4	
340	002212	004767	010312	JSR	PC,TTOUT	;PRINT NOT AVAIL
341	002216	000741		BR	ST1A	;REDO DRIVE REQUEST
342	002220	012704	014202	ST2: MOV	#MSG11,R4	
343	002224	004767	010300	JSR	PC,TTOUT	;REQUEST SLAVE NUMBER
344	002230	012705	000702	MOV	#SLVN,R5	;SET ADDRESS OF SLAVE SAVE
345	002234	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
346	002240	012702	000007	MOV	#7,R2	;SET UPPER LIMIT
347	002244	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
348	002250	004767	010020	JSR	PC,TTR	;GO GET RESPONSE
349	002254	012777	000040 176326	MOV	#40,ACS	;INIT
350	002262	056777	176412 176320	BIS	DRVN,ACS	;SET DRIVE NUMBER
351	002270	016777	176406 176334	MOV	SLVN,ATC	;LOAD SLAVE NUMBER
352	002276	032777	002000 176322	BIT	#2000,ADT	;SEE IF SLAVE PRESENT
353	002304	001005		BNE	ST3	;IF SO: BR
354	002306	012704	015135	MOV	#MSG42,R4	
355	002312	004767	010212	JSR	PC,TTOUT	;PRINT NON-EXIST SLAVE
356	002316	000740		BR	ST2	;REDO SLAVE REQUEST
357	002320	012704	015156	ST3: MOV	#MSG43,R4	
358	002324	004767	010200	JSR	PC,TTOUT	;PRINT SERIAL NUMBER TAG
359	002330	017703	176274	MOV	ASN,R3	
360	002334	004767	010662	JSR	PC,SNPT	;PRINT SERIAL NUMBER
361	002340	012704	015740	MOV	#MSG61,R4	
362	002344	004767	010160	JSR	PC,TTOUT	;REQUEST RH11 OR RH70
363	002350	012705	000672	MOV	#RH17F,R5	;GET ADDRESS OF FLAG
364	002354	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
365	002360	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
366	002364	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
367	002370	004767	007700	JSR	PC,TTR	;GET RESPONSE
368	002374	012704	015760	MOV	#MSG62,R4	
369	002400	004767	010124	JSR	PC,TTOUT	;REQUEST RH11 ONLY RESPONSE
370	002404	012705	001014	MOV	#RHOF,R5	;SET FLAG ADDRESS
371	002410	012701	000001	MOV	#1,R1	;SET SIZE OF RESPONSE
372	002414	012702	000001	MOV	#1,R2	;SET UPPER LIMIT
373	002420	012703	000000	MOV	#0,R3	;SET LOWER LIMIT
374	002424	004767	007644	JSR	PC,TTR	;GO GET RESPONSE
375	002430	005767	176360	TST	RHOF	;SEE IF RH11 ONLY
376	002434	001016		BNE	ST4	;IF SO: BR
377	002436	012704	015631	MOV	#MSG55,R4	
378	002442	004767	010062	JSR	PC,TTOUT	;REQUEST NRZ ONLY RESPONSE

379	002446	012705	001012
380	002452	012701	000001
381	002456	012702	000001
382	002462	012703	000000
383	002466	004767	007602
384	002472	005067	176320

	MOV	#NRZOF,RS	;SET FLAG ADDRESS
	MOV	#1,R1	;SET SIZE OF RESPONSE
	MOV	#1,R2	;SET UPPER LIMIT
	MOV	#0,R3	;SET LOWER LIMIT
ST4:	JSR	PC,TTR	;GO GET RESPONSE
	CLR	PCNTR	;CLEAR PASS COUNTER

```

;TEST SCHEDULAR*****
385
386
387 002476 000240          TSCD:  NOP
388 002500 005067 176266  CLR      STFLG          ;CLEAR SINGLE TEST FLAG
389 002504 017700 176126  MOV      @SWR,RO
390 002510 042700 177740  BIC      #177740,RO
391 002514 005700          TST      RO
392 002516 001055          BNE      STSCD          ;GO SELECT SINGLE TEST
393 002520 012767 001040 176246  TSRH:  MOV      #TSTTBL,LTADD
394 002526 062767 000004 176240  TSCD0:  ADD      #4,LTADD
395 002534 016767 176234 176222  MOV      LTADD,ITRLP
396 002542 062767 000002 176214  ADD      #2,ITRLP          ;SET ITERATION ADDRESS
397 002550 005777 176220  TST      @LTADD
398 002554 001002          BNE      TSCD1
399 002556 000167 000144  JMP      TEND          ;GO TO END ROUTINE
400 002562 000240          TSCD1:  NOP
401 002564 005067 176156  CLR      STMSK
402 002570 005067 176120  CLR      ERRP
403 002574 005067 176074  CLR      HDRFL          ;CLEAR PRINT HEADER FLAG
404 002600 017700 176170  MOV      @LTADD,RO          ;SET POINTER TO TEST
405 002604 000110          JMP      (RO)          ;GO TO TEST
406 002606 000240          TSCD2:  NOP
407 002610 032777 002000 176020  BIT      #2000,@SWR          ;SEE IF HALT ON TEST
408 002616 001401          BEQ     TSCD3          ;IF NOT: BR
409 002620 000000          HALT
410 002622 004767 010476  TSCD3:  JSR     PC,CKSWR          ;CHECK FOR CNTL G
411 002626 000240          NOP
412 002630 005767 176136  TST      STFLG          ;SE IF SINGLE TEST
413 002634 001734          BEQ     TSCD0          ;IF NOT: BR
414 002636 017700 175774  MOV      @SWR,RO
415 002642 042700 177740  BIC      #177740,RO          ;MASK TEST NUMBER
416 002646 005700          TST      RO          ;SEE IF RETURN TO ALL
417 002650 001712          BEQ     TSCD          ;IF SO: BR
418 002652 000240          STSCD:  NOP
419 002654 012767 000001 176110  MOV      #1,STFLG          ;SET SINGLE TEST FLAG
420 002662 022700 000025  CMP      #25,RO          ;SEE IF EXCEEDED TESTS
421 002666 003417          BLE     TEND          ;IF SO: BR
422 002670 000241          CLC
423 002672 006100          ROL     RO
424 002674 006100          ROL     RO          ;SET TABLE MODIFIER
425 002676 012767 001040 176070  MOV      #TSTTBL,LTADD
426 002704 060067 176064          ADD     RO,LTADD          ;SET TEST POINTER
427 002710 016767 176060 176046  MOV      LTADD,ITRLP
428 002716 062767 000002 176040  ADD      #2,ITRLP          ;SET ITERATION POINTER
429 002724 000716          BR     TSCD1
430 002726 012704 014114  TEND:  MOV      #MSG6,R4
431 002732 004767 007572          JSR     PC,TTOUT          ;PRINT END OF PASS
432 002736 016703 176054          MOV     PCNTR,R3
433 002742 004767 007726          JSR     PC,OCIP
434 002746 032777 004000 175662  BIT      #4000,@SWR          ;PRINT PASS NUMBER
435 002754 001001          BNE     TENDX          ;SEE IF HALT ON PASS
436 002756 000000          HALT          ;IF NOT: BR
437 002760 004767 010340  TENDX:  JSR     PC,CKSWR          ;CHECK FOR CNTL G
438 002764 005267 176026          INC     PCNTR          ;BUMP PASS COUNTER
439 002770 000167 177502          JMP     TSCD          ;RESTART

```

```

440
441
442 ;RH ADDRESSING TEST*****
443 002774 012767 015773 175674 FT1: MOV #MSFT1,EMADDR ;SET HEADER
444 003002 012777 012154 175652 MOV #TRAP,#BTRP ;SET TRAP HANDLER ADDRESS
445 003010 012777 000340 175646 MOV #340,#BTRP2
446 003016 012700 000016 MOV #16,R0 ;SET NUMBER OF REGISTER
447 003022 016701 175552 MOV #C1,R1 ;GET FIRST ADDRESS (CS1)
448 003026 005711 FT1A: TST (R1) ;REFERENCE REGISTER
449 003030 000240 NOP ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
450 003032 005300 FT1B: DEC R0 ;SEE IF DONE ALL
451 003034 001403 BEQ FT1X ;IF SO: BR
452 003036 062701 000002 ADD #2,R1 ;BUMP ADDRESS POINTER
453 003042 000771 BR FT1A ;CONTINUE
454 003044 012777 000006 175610 FT1X: MOV #6,#BTRP ;RESET TRAP CATCHER
455 003052 005767 175732 TST RHTF ;SEE IF INITIAL ADDRESS TEST PASS
456 003056 001404 BEQ FT1XX ;IF NOT: BR
457 003060 005067 175724 CLR RHTF ;CLEAR FLAG
458 003064 000167 177032 JMP ST1A ;RETURN
459 003070 000167 177512 FT1XX: JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

460
461
462 ;RH REGISTER BITS READ/WRITE*****
463 003074 012767 016020 175574 FT2:  MOV #MSFT2,EMADDR ;SET TEST HEADER
464 003102 012701 177777          MOV #-1,R1 ;SET ALL ONES PATTERN
465 003106 004767 006770          FT2A: JSR PC,INIT1 ;GO INIT
466 003112 016700 175464          MOV WC,R0 ;GET ADDRESS OF WORD COUNT
467 003116 010102          MOV R1,R2 ;SET EXPT REGISTER BIT PATTERN
468 003120 010110          MOV R1,(R0) ;LOAD PATTERN
469 003122 021002          CMP (R0),R2 ;SEE IF EXPT=RCVD
470 003124 001410          BEQ FT2B ;IF SO: BR
471 003126 012767 014442 175602  MOV #MSG25,ERADD ;SET CODE
472 003134 012767 003106 175620  MOV #FT2A,SCOLP ;SET SCOPE
473 003142 004767 000116          JSR PC,FT2ER ;GO DO ERROR
474 003146 016700 175432          FT2B: MOV BA,R0 ;GET ADDRESS OF BUS ADDRESS
475 003152 010102          MOV R1,R2
476 003154 042702 000001          BIC #1,R2 ;SET EXPT PATTERN
477 003160 010110          MOV R1,(R0) ;LOAD PATTERN
478 003162 020210          CMP R2,(R0) ;SEE IF EXPT=RCVD
479 003164 001410          BEQ FT2C ;IF SO:BR
480 003166 012767 014450 175542  MOV #MSG26,ERADD ;SET ERROR CODE
481 003174 012767 003146 175560  MOV #FT2B,SCOLP ;SET SCOPE ADDRESS
482 003202 004767 000056          JSR PC,FT2ER ;GO DO ERROR
483 003206 016700 175410          FT2C: MOV DB,R0 ;GET ADDRESS OF DATA BUFFER
484 003212 010102          MOV R1,R2
485 003214 010110          MOV R1,(R0) ;LOAD PATTERN
486 003216 012703 004000          MOV #4000,R3
487 003222 005303          FT2D: DEC R3 ;DELAY
488 003224 001376          BNE FT2D
489 003226 020210          CMP R2,(R0) ;SEE IF EXPT=RCVD
490 003230 001410          BEQ FT2E ;IF SO: BR
491 003232 012767 014456 175476  MOV #MSG27,ERADD ;SET ERROR CODE
492 003240 012767 003206 175514  MOV #FT2C,SCOLP ;SET SCOPE ADDRESS
493 003246 004767 000012          JSR PC,FT2ER ;GO DO ERROR
494 003252 005701          FT2E: TST R1 ;SEE IF DONE RESET
495 003254 001454          BEQ FT2X ;IF SO: BR
496 003256 005001          CLR R1 ;SET ZERO PATTERN
497 003260 000167 177622          JMP FT2A ;DO ZERO BITS
498 003264 000240          FT2ER: NOP
499 003266 032777 020000 175342  BIT #20000,JSWR ;SEE IF PRINT ERROR
500 003274 001034          BNE FT2ERB ;IF NOT: BR
501 003276 005767 175372          TST HDRFL ;SEE IF DONE HEADER
502 003302 001004          BNE FT2ERA ;IF SO: BR
503 003304 016704 175366          MOV EMADDR,R4
504 003310 004767 007214          JSR PC,TTOUT ;DO HEADER
505 003314 012767 000001 175352  FT2ERA: MOV #1,HDRFL ;SET FLAG
506 003322 016704 175410          MOV ERADD,R4
507 003326 004767 007176          JSR PC,TTOUT ;PRINT ERROR CODE
508 003332 012704 014406          MOV #MSG22,R4
509 003336 004767 007166          JSR PC,TTOUT ;PRINT EXPT TAG
510 003342 010103          MOV R1,R3
511 003344 004767 007312          JSR PC,OCTPE ;PRINT EXPT
512 003350 012704 014416          MOV #MSG23,R4
513 003354 004767 007150          JSR PC,TTOUT ;PRINT RCVD TAG
514 003360 011003          MOV (R0),R3
515 003362 004767 007274          JSR PC,OCTPE ;PRINT RCVD

```

TM02-TU16/TE16 BASIC FUNCTION TEST
 DZTUBF.P11 15-AUG-77 09:41

MACY11 30(1046) 15-AUG-77 10:47 PAGE 14

SEQ 0028

516	003366	005777	175244
517	003372	100001	
518	003374	000000	
519	003376	004767	006366
520	003402	000240	
521	003404	000207	
522	003406	000240	
523	003410	004767	006416
524	003414	000167	177166

FT2ERB:	TST	@SWR	;SEE IF HALT ON ERROR
	BPL	FT2ERC	;IF NOT: BR
	HALT		
FT2ERC:	JSR	PC,SCOPE	;GO SEE IF SCOPE ON ERROR
	NOP		
	RTS	PC	;IF NO SCOPE: CONTINUE TEST
FT2X:	NOP		
	JSR	PC,ITER	;GO SEE IF ITERATIONS
	JMP	TSCD2	;RETURN TO SCHEDULAR

```

525
526
527
528 003420 012767 016055 175250 FT3:  MOV    #MSFT3,EMADDR ;SET TEST HEADER
529 003426 012767 003420 175326      MOV    #FT3,SCOLP
530 003434 004767 006442          JSR    PC,INIT1 ;GO INIT
531 003440 052777 020000 175142      BIS    #20000,@CS ;FORCE UPE =1
532 003446 000240          NOP
533 003450 004767 006426          JSR    PC,INIT1 ;GO INIT
534 003454 005777 175120          TST    @C1 ;SEE IF SC IS RESET
535 003460 100005          BPL    FT3A ;IF S0: BR
536 003462 012767 014514 175246      MOV    #MSG29,ERADD ;SET ERROR CODE
537 003470 004767 000060          JSR    PC,FT3ER ;GO DO ERROR
538 003474 032777 040000 175076 FT3A:  BIT    #40000,@C1 ;SEE IF TRE IS RESET
539 003502 001405          BEQ    FT3B ;IF S0: BR
540 003504 012767 014543 175224      MOV    #MSG30,ERADD ;SET ERROR CODE.
541 003512 004767 000036          JSR    PC,FT3ER ;GO DO ERROR
542 003516 017701 175066          FT3B:  MOV    @CS,R1 ;GET CS2
543 003522 042701 000307          BIC    #307,R1 ;MARK IR/OR
544 003526 005701          TST    R1 ;SEE IF RESET
545 003530 001405          BEQ    FT3X ;IF S0: BR
546 003532 012767 014573 175176      MOV    #MSG31,ERADD ;SET ERROR CODE
547 003540 004767 000010          JSR    PC,FT3ER ;GO DO ERROR
548 003544 004767 006262          FT3X:  JSR    PC,ITER ;GO SEE IF ITERATION
549 003550 000167 177032          JMP    TSCD2 ;RETURN TO SCHEDULAR
550 003554 032777 020000 175054 FT3ER:  BIT    #20000,@SWR ;SEE IF PRINT ERROR
551 003562 001015          BNE    FT3ERB ;IF NOT: BR
552 003564 005767 175104          TST    HDRFL ;SEE IF DONE HEADER
553 003570 001006          BNE    FT3ERA ;IF S0: BR
554 003572 016704 175100          MOV    EMADDR,R4
555 003576 004767 006726          JSR    PC,TTOUT ;PRINT HEADER
556 003602 005267 175066          INC    HDRFL
557 003606 016704 175124          FT3ERA: MOV    ERADD,R4
558 003612 004767 006712          JSR    PC,TTOUT ;PRINT ERROR CODE
559 003616 005777 175014          FT3ERB: TST    @SWR ;SEE IF HALT ON ERROR
560 003622 100001          BPL    FT3ERC ;IF NOT: BR
561 003624 000000          HALT
562 003626 000240          FT3ERC: NOP
563 003630 004767 006134          JSR    PC,SCOPE ;GO SEE IF SCOPE
564 003634 000207          RTS    PC ;IF NOT: BR

```

```

565
566
567
568 003636 005767 175030
569 003642 001141
570 003644 012767 016107 175024
571 003652 012777 000040 174730
572 003660 017700 174736
573 003664 005777 174720
574 003670 100013
575 003672 005777 174702
576 003676 100014
577 003700 032777 040000 174672
578 003706 001414
579 003710 004767 006116
580 003714 000167 176666
581 003720 012767 014623 175010
582 003726 000407
583 003730 012767 014641 175000
584 003736 000403
585 003740 012767 014656 174770
586 003746 000240
587 003750 012767 003636 175004
588 003756 004767 177572
589 003762 000752
;RH11 SILO TEST 1: EPMTY SILO READ*****
FT4: TST RH17F
      BNE FT5X ;IF RH70: BR
      MOV #MSFT4,EMADDR ;SET TEST TEST HEADER
      MOV #40,@CS ;INIT
      MOV @DB,@R0 ;READ DB
      TST @CS ;SEE IF DLT IS SET
      BPL FT4ER ;IF NOT: BR
      TST @C1 ;SEE IF SC IS SET
      BPL FT4ERA ;IF NOT: BR
      BIT #40000,@C1 ;SEE IF TRE IS SET
      BEQ FT4ERB ;IF NOT: BR
FT4X: JSR PC,ITER ;GO SEE IF ITERATION
      JMP TSCD2 ;RETURN TO SCHEDULAR
FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE
      BR FT4ERC
FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE
      BR FT4ERC
FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.
FT4ERC: NOP
      MOV #FT4,SCOLP ;SET SCOPE ADDRESS
      JSR PC,FT3ER ;GO PRINT ERROR
      BR FT4X

```

```

590
591
592
593 003764 005767 174702          FTS:  TST      RH17F          ;SEE IF RH70
594 003770 001066                    BNE      FT5X          ;IF S0: BR
595 003772 012767 016137 174676    MOV      #MSFT5,EMADDR ;SET TEST HEADER
596 004000 012767 004006 174754    MOV      #FT5A,SCOLP   ;SET SCOPE ADDRESS
597 004006 004767 006070          FTSA:  JSR      PC,INITI  ;GO INIT
598 004012 032777 000100 174570    BIT      #100,ACS      ;SEE IF IR IS SET
599 004020 001005                    BNE      FT5B          ;IF S0: BR
600 004022 012767 014674 174706    MOV      #MSG35,ERADD  ;SET ERROR CODE
601 004030 004767 000122          JSR      PC,FT5ER      ;GO DO ERROR
602 004034 032777 000200 174546    FT5B:  BIT      #200,ACS ;SEE IF OR IS RESET
603 004042 001405                    BEQ      FT5C          ;IF S0: BR
604 004044 012767 014721 174664    MOV      #MSG36,ERADD  ;SET ERROR CODE
605 004052 004767 000100          JSR      PC,FT5ER      ;GO DO ERROR
606 004056 012777 000000 174536    FT5C:  MOV      #0,ADB      ;LOAD ZERO INTO SILO
607 004064 032777 000200 174516    BIT      #200,ACS      ;SEE THAT OR RESET
608 004072 001405                    BEQ      FT5D          ;IF IT DOES: BR
609 004074 012767 014750 174634    MOV      #MSG37,ERADD  ;SET ERROR CODE
610 004102 004767 000050          JSR      PC,FT5ER      ;GO DO ERROR
611 004106 012777 177777 174506    FT5D:  MOV      #-1,ADB      ;LOAD SILO WITH -1
612 004114 012700 004000          MOV      #4000,RO
613 004120 032777 000200 174462    FT5E:  BIT      #200,ACS ;SEE IF OR IS SET
614 004126 001007                    BNE      FT5X          ;IF S0: BR
615 004130 005300                    DEC      RO
616 004132 001372                    BNE      FT5E          ;AWAIT OR
617 004134 012767 014750 174574    MOV      #MSG37,ERADD  ;SET ERROR CODE
618 004142 004767 000010          JSR      PC,FT5ER      ;GO DO ERROR
619 004146 004767 005660          FTSX:  JSR      PC,ITER  ;GO SEE IF ITERATION
620 004152 000167 176430          JMP      TSCD2         ;RETURN TO SCHEDULAR
621 004156 004767 177372          FT5ER: JSR      PC,FT3ER   ;GO PRINT ERROR
622 004162 000240                    NOP
623 004164 000207                    RTS      PC            ;CONTINUE TEST

```



```

624
625 ;RH11 SILO TEST 3: SILO DATA TEST*****
626
627 004166 005767 174500 FT6: TST RH17F
628 004172 001052 BNE FT6X ; IF RH70: BR
629 004174 012767 016167 174474 MOV #MSFT6,EMADDR ; SET TEST HEADER
630 004202 012767 004210 174552 MOV #FT6A,SCOLP ; SET SCOPE ADDRESS
631 004210 004767 005666 FT6A: JSR PC,INIT1 ; GO INIT
632 004214 005000 CLR RO ; PRESET DATA
633 004216 010077 174400 FT6B: MOV RO,20B ; LOAD SILO
634 004222 005200 INC RO ; BUMP DATA
635 004224 022700 000102 CMP #102,RO ; SEE IF FILLED ALL
636 004230 001372 BNE FT6B ; IF NOT: BR
637 004232 032777 000100 174350 BIT #100,2CS ; SEE IF IR IS RESET.
638 004240 001405 BEQ FT6C ; IF SO: BR
639 004242 012767 015061 174466 MOV #MSG40,ERADD ; SET ERROR CODE
640 004250 004767 000054 JSR PC,FT6ER ; GO DO ERROR
641 004254 032777 000200 174326 FT6C: BIT #200,2CS ; SEE IF OR IS SET
642 004262 001005 BNE FT6D ; IF SO: BR
643 004264 012767 015007 174444 MOV #MSG38,ERADD ; SET ERROR CODE
644 004272 004767 000032 JSR PC,FT6ER ; GO DO ERROR
645 004276 005000 FT6D: CLR RO ; PRESET DATA
646 004300 017701 174316 FT6E: MOV 20B,R1 ; READ SILO
647 004304 020001 CMP RO,R1 ; SEE IF EXPT=RCVD
648 004306 001014 BNE FT6DE ; IF NOT: BR
649 004310 005200 INC RO ; BUMP DATA
650 004312 022700 000102 CMP #102,RO ; SEE IF DONE ALL
651 004316 001370 BNE FT6E ; IF NOT: BR
652 004320 004767 005506 FT6X: JSR PC,ITER ; GO SEE IF ITERATION
653 004324 000167 176256 JMP TSCD2 ; RETURN TO SCHEDULAR
654 004330 000240 FT6ER: NOP
655 004332 004767 177216 JSR PC,FT3ER ; GO PRINT ERROR
656 004336 000207 RTS PC ; RETURN
657 004340 000240 FT6DE: NOP
658 004342 032777 020000 174266 BIT #20000,2SWR ; SEE IF PRINT ERROR
659 004350 001032 BNE FT6DEB ; IF NOT: BR
660 004352 005767 174316 TST HDRFL ; SEE IF DONE HEADER
661 004356 016701 174314 MOV EMADDR,R1
662 004362 004767 006142 JSR PC,TTOUT ; PRINT HEADER
663 004366 005267 174302 INC HDRFL ; SET FLAG
664 004372 012704 015041 FT6DEA: MOV #MSG39,R4 ; PRINT SILO READ ERROR
665 004376 004767 006126 JSR PC,TTOUT
666 004402 012704 014406 MOV #MSG22,R4 ; PRINT EXPT TAG
667 004406 004767 006116 JSR PC,TTOUT ; PRINT EXPT TAG
668 004412 010003 MOV RO,R3 ; PRINT EXPT
669 004414 004767 006254 JSR PC,OCTP ; PRINT RCVD TAG
670 004420 012704 014416 MOV #MSG23,R4 ; PRINT RCVD TAG
671 004424 004767 006100 JSR PC,TTOUT
672 004430 010103 MOV R1,R3 ; PRINT RCVD
673 004432 004767 006236 JSR PC,OCTP ; SEE IF HALT ON ERROR
674 004436 005777 174174 FT6DEB: TST 2SWR ; IF NOT: BR
675 004442 100001 BPL FT6DEX
676 004444 000000 HALT
677 004446 004767 006652 FT6DEX: JSR PC,CKSWR ; CHECK FOR CNTL G
678 004452 000207 RTS PC ; RETURN TO TEST

```

```

679
680
681
682 004454 005767 174212
683 004460 001021
684 004462 012767 016217 174206
685 004470 012767 004454 174264
686 004476 004767 005400
687 004502 012700 000103
688 004506 010077 174110
689 004512 005300
690 004514 001374
691 004516 005777 174066
692 004522 100004
693 004524 004767 005302
694 004530 000167 176052
695 004534 012767 014623 174174
696 004542 004767 177006
697 004546 000766

;RH11 SILO TEST 4: SILO OVERFLOW*****
FT7: TST RH17F
      BNE FT7X ; IF RH70: BR
      MOV #MSFT7,EMADDR ; SET TEST HEADER
      MOV #FT7,SCOLP ; SET SCOPE ADDRESS
      JSR PC,INIT1 ; GO INIT
      MOV #103,RO ; SET SIZE OF SILO +1
FT7A: MOV RO,20B ; LOAD SILO
      DEC RO ; SEE IF DONE
      BNE FT7A ; IF NOT: BR
      TST 2CS ; SEE IF DLT IS SET
      BPL FT7ER ; IF NOT: BR
FT7X: JSR PC,ITER ; GO SEE IF ITERATION
      JMP TSCD2 ; RETURN TO SCHEDULAR
FT7ER: MOV #MSG32,ERADD ; SET ERROR CODE
       JSR PC,FT3ER ; GO DO ERROR
       BR FT7X
    
```

```

698
699
700 ;RH11 SILO TEST 5: SILO RESET*****
701 004550 005767 174116 FT10: TST RH17F
702 004554 001034 BNE FT10X ;IF RH70: BR
703 004556 012767 016247 174112 MOV #MSFT10,EMADDR ;SET TEST HEADER
704 004564 012767 004550 174170 MOV #FT10,SCOLP ;SET SCOPE ADDRESS
705 004572 012777 000040 174010 MOV #40,ACS ;INITIALIZE
706 004600 012700 000004 MOV #4,RO ;SET NUMBER OF SILO WRITER
707 004604 010077 174012 FT10A: MOV RO,DOB ;WRITE SILO
708 004610 005300 DEC RO ;SEE IF DONE
709 004612 001374 BNE FT10A ;IF NOT: BR
710 004614 052777 000040 173766 BIS #40,ACS ;INITIALIZE
711 004622 012777 177777 173772 MOV #-1,DOB ;WRITE SILO
712 004630 017701 173766 MOV DOB,R1 ;READ SILO 1
713 004634 017701 173762 MOV DOB,R1 ;READ SILO 2
714 004640 005777 173744 TST ACS ;SEE IF DLT IS SET
715 004644 100011 BPL FT10ER ;IF NOT: BR
716 004646 004767 005160 FT10X: JSR PC,ITER ;GO SEE IF ITERATION
717 004652 005767 174136 TST RHOF ;SEE IF RH11 ONLY
718 004656 001402 BEQ FT10XX ;IF NOT: BR
719 004660 000167 176042 JMP TEND ;ELSE GO TO END
720 004664 000167 175716 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR
721 004670 012767 014623 174040 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE
722 004676 004767 176652 JSR PC,FT3ER ;GO DO ERROR
723 004702 000761 BR FT10X
    
```

```

724                                     ;NOP TEST*****
725
726 004704 000240                       FT11: NOP
727 004706 012767 004704 174046        MOV #FT11,SCOLP ;SET SCOPE ADDRESS
728 004714 004767 005162                JSR PC,INIT1
729 004720 012767 000300 174056        MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
730 004726 012767 177777 173752        MOV #-1,FCNT ;SET FC= ALL OVER
731 004734 012767 177777 173746        MOV #-1,WCNT ;SET WC= ALL OVER
732 004742 012767 177777 173734        MOV #-1,BADDR ;SET BA= ALL OVER
733 004750 012767 000001 173746        MOV #1,RDYDX ;SET DELAY
734 004756 012767 000001 173742        MOV #1,OPDYX ;SET OP DELAY
735 004764 012767 000001 174004        MOV #1,FUN ;SET NOP FUNCTIONS CODE
736 004772 004767 003760                JSR PC,EXEC ;GO EXECUTE COMMAND
737 004776 000240                       NOP
738 005000 012767 016300 173670        MOV #MSFT11,EMADDR
739 005006 004767 004177                JSR PC,ERCHK ;GO CHECK REGISTER
740 005012 004767 005014                JSR PC,ITER ;GO SEE IF ITERATIONS
741 005016 000167 175564                JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

742                                     ;REWIND TEST*****
743
744 005022 000240 FT12: NOP
745 005024 012767 005022 173730 MOV #FT12, SCOLP
746 005032 004767 005044 JSR PC, INIT1 ;GO INITIALIZE
747 005036 052777 001700 173566 BIS #1700, JTC ;SET TO NRZ, NORMAL
748 005044 012767 177760 173634 MOV #-20, FCNT ;SET FC=20
749 005052 012767 177770 173630 MOV #-10, WCNT ;SET WC=10
750 005060 012767 016760 173616 MOV #WDATA, BADDR ;SET BA=WRITE BUFFER
751 005066 012767 000007 173702 MOV #7, FUN ;SET REWIND OP CODE
752 005074 004767 003656 JSR PC, EXEC ;GO EXECUTE COMMAND
753 005100 000240 NOP
754 005102 032777 020000 173502 FT12A: BIT #20000, ADS
755 005110 001374 BNE FT12A ;AWAIT PIP
756 005112 012767 016320 173556 MOV #MSFT12, EMADDR
757 005120 004767 004062 JSR PC, ERCHK ;GO CHECK FOR ERROR
758 005124 004767 004702 JSR PC, ITER ;GO SEE IF ITERATION
759 005130 000167 175452 JMP TSCD2 ;RETURN TO SCHEDULAR
760

```

```

;WRITE/READ TEST*****
761
762
763 005134 000240 FT13: NOP
764 005136 012767 000001 173560 MOV #1, RDYDX
765 005144 012767 000001 173554 MOV #1, OPDYX
766 005152 012767 000100 173532 MOV #100, RCNT ;SET RECORD COUNT
767 005160 012767 016343 173510 MOV #MSFT13, EMADDR ;SET TEST HEADER
768 005166 012767 000001 173612 MOV #1, PATRN
769 005174 004767 004414 JSR PC, DSUP ;SET UP ALL ONES DATA PATTERN
770 005200 012767 000300 173576 MOV #300, UDES ;REWIND TO BOT
771 005206 004767 003676 FT13A: JSR PC, REND ;SET 200 BPI, NORMAL
772 005212 012767 177600 173466 MOV #-200, FCNT ;SET FC
773 005220 012767 177700 173462 MOV #-100, WCNT ;SET WC
774 005226 012767 016760 173450 MOV #WDATA, BADDR ;SET BA
775 005234 012767 000061 173534 MOV #61, FUN ;SET WRITE OP-CODE
776 005242 012767 014222 173444 MOV #MSG12, ERRP
777 005250 004767 003502 FT13B: JSR PC, EXEC ;GO EXECUTE COMMAND
778 005254 005067 173502 CLR SCOLP ;NO SCOPE LOOP
779 005260 004767 003722 JSR PC, ERCHK ;GO CHECK ERROR
780 005264 005367 173422 DEC RCNT ;SEE IF DONE ALL
781 005270 001367 BNE FT13B ;IF NOT: BR
782 005272 012767 000100 173412 MOV #100, RCNT ;SET RECORD COUNT
783 005300 012767 020472 173376 MOV #RDATA, BADDR
784 005306 062767 000200 173370 ADD #200, BADDR ;SET BA
785 005314 012767 000077 173454 MOV #77, FUN ;SET READ REVERSE OP-CPDE
786 005322 012767 014240 173364 MOV #MSG13, ERRP
787 005330 004767 003422 FT13C: JSR PC, EXEC ;GO EXECUTE COMMAND
788 005334 004767 003646 JSR PC, ERCHK ;GO CHECK ERROR
789 005340 005367 173346 DEC RCNT ;SEE IF READ ALL
790 005344 001371 BNE FT13C ;IF NOT: BR
791 005346 162767 000200 173330 SUB #200, BADDR ;SET BA
792 005354 012767 000071 173414 MOV #71, FUN ;SET READ FORWARD OP-CODE
793 005362 012767 014265 173324 MOV #MSG14, ERRP
794 005370 012767 000100 173314 MOV #100, RCNT ;SET RECORD COUNT
795 005376 004767 003354 FT13D: JSR PC, EXEC ;GO EXECUTE COMMAND
796 005402 004767 003600 JSR PC, ERCHK ;GO CHECK ERRORS
797 005406 005367 173300 DEC RCNT ;SEE IF DONE ALL
798 005412 001371 BNE FT13D ;IF NOT: BR
799 005414 032767 002000 173362 BIT #2000, UDES ;SEE IF DONE PE
800 005422 001017 BNE FT13X ;IF SO: BR
801 005424 062767 000400 173352 ADD #400, UDES ;SELECT NEXT DENSITY
802 005432 032767 002000 173344 BIT #2000, UDES ;SEE IF PE
803 005440 001403 BEQ FT13E ;IF NOT: BR
804 005442 005767 173344 TST NRZOF ;SEE IF NRZ ONLY
805 005446 001005 BNE FT13X ;IF SO: BR
806 005450 012767 000100 173234 FT13E: MOV #100, RCNT ;RESET RECORD COUNT
807 005456 000167 177524 JMP FT13A ;GO DO NEXT DENSITY
808 005462 000167 175120 FT13X: JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

;SPACE TEST*****
809
810
811 005466 000240 FT14: NOP
812 005470 012767 016372 173200 MOV #MSFT14,EMADDR ;SET TEST HEADER
813 005476 012767 001700 173300 MOV #1700,UDES ;SET NRZ,NORMAL
814 005504 004767 003400 FT14A1: JSR PC,RWIND ;GO INITIALIZE
815 005510 012767 000100 173174 MOV #100,RCNT ;SET NUMBER OF RECORDER
816 005516 012767 177777 011234 MOV #-1,WDATA ;SET DATA PATTERN
817 005524 012767 177700 173154 MOV #-100,FCNT ;PRESET FRAME CNT
818 005532 012767 177740 173150 MOV #-40,WCNT ;PRESET WORD CNT
819 005540 004767 004336 FT14A: JSR PC,INIT1 ;GO REWIND
820 005544 012767 001000 173154 MOV #1000,OPDYX
821 005552 012767 040000 173144 MOV #40000,RDYDX
822 005560 012767 000061 173210 MOV #61,FUN ;SET WRITE OP-CODE
823 005566 012767 102300 173152 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
824 005574 052777 000010 173006 BIS #10,ACS ;INHIBIT BUS ADDRESS INCREMENT
825 005602 004767 003150 JSR PC,EXEC ;GO EXECUTE COMMAND
826 005606 000240 NOP
827 005610 012767 015275 173076 MOV #MSG46,ERRP ;SET ERROR CODE
828 005616 004767 003364 JSR PC,ERCHK ;GO CHECK ERRORS
829 005622 005767 173152 TST SERFL ;SEE IF ERROR
830 005626 001402 BEQ FT14A2 ;IF NOT: BR
831 005630 000167 000466 JMP FT14X ;ELSE EXIT
832 005634 162767 000001 173044 FT14A2: SUB #1,FCNT ;BUMP FC
833 005642 032767 000001 173036 BIT #1,FCNT ;SEE IF SHOULD BUMP WC
834 005650 001403 BEQ FT14A3 ;IF NOT: BR
835 005652 162767 000001 173030 SUB #1,WCNT ;BUMP WC
836 005660 005367 173026 FT14A3: DEC RCNT ;SEE IF DONE ALL
837 005664 001325 BNE FT14A ;WRITE ALL RECORDS
838 005666 000240 NOP
839 005670 012767 000100 173022 MOV #100,RRD ;PRESET RECORD POSITION
840 005676 012767 000176 173016 MOV #176,RFD
841 005704 000240 NOP
842 005706 012767 177701 173014 MOV #-77,SCNT ;SET SPACE AMOUNT
843 005714 012767 000033 173054 FT14B: MOV #33,FUN ;SET OP-CODE SPACE REVERSE
844 005722 004767 003030 JSR PC,EXEC ;GO EXECUTE COMMAND
845 005726 012767 015346 172760 MOV #MSG48,ERRP ;SET ERROR CODE
846 005734 004767 003246 JSR PC,ERCHK ;GO CHECK ERRORS
847 005740 005767 173034 TST SERFL ;SEE IF ERROR
848 005744 001166 BNE FT14X ;IF SO: BR
849 005746 004767 000070 JSR PC,FT14RR ;GO READ REVERSE + CHECK DATA
850 005752 000240 NOP
851 005754 012767 000031 173014 MOV #31,FUN ;SET SPACE FORWARD OP-CODE
852 005762 005267 172742 INC SCNT ;SET SPACE AMOUNT
853 005766 001555 BEQ FT14X ;IF DONE: BR
854 005770 004767 002762 JSR PC,EXEC ;GO EXECUTE COMMAND
855 005774 012767 015321 172712 MOV #MSG47,ERRP ;SET ERROR CODE
856 006002 004767 003200 JSR PC,ERCHK ;GO CHECK ERROR
857 006006 005767 172766 TST SERFL ;SEE IF ERROR FLAG
858 006012 001143 BNE FT14X ;IF NO: BR
859 006014 004767 000064 JSR PC,FT14RF ;GO READ FORWARD FOR POSITION CHECK
860 006020 000240 NOP
861 006022 005267 172702 INC SCNT ;DECREMENT SPACE AMOUNT
862 006026 001535 BEQ FT14X ;IF DONE: BR
863 006030 005267 172664 INC RRD ;BUMP DATA EXPT
864 006034 005367 172662 DEC RFD ;BUMP DATA EXPT

```

865	006040	000725				BR	FT14B		
866	006042	000240				FT14RR: NOP			
867	006044	012767	020472	172632		MOV	#RDATA, BADDR	; SET BA	
868	006052	012767	000077	172716		MOV	#77, FUN	; SET READ REVERSE OP-CODE	
869	006060	004767	002672			JSR	PC, EXEC	; GO EXECUTE COMMAND	
870	006064	000240				NOP			
871	006066	016705	172626			MOV	RRO, R5		
872	006072	020577	172510			CMP	R5, JFC	; SEE IF CORRECT RECORD	
873	006076	001020				BNE	FT14RER	; IF NOT: BR	
874	006100	000167	000026			JMP	FT14EC	; GO CLEAR RH11 ERROR BIT	
875	006104	000240				FT14RF: NOP			
876	006106	012767	000071	172662		MOV	#71, FUN	; SET READ FORWARD OP-CODE	
877	006114	004767	002636			JSR	PC, EXEC	; GO EXECUTE COMMAND	
878	006120	016705	172576			MOV	RFD, R5		
879	006124	020577	172456			CMP	R5, JFC	; SEE IF CORRECT RECORD	
880	006130	001003				BNE	FT14RER	; IF NOT: BR	
881	006132	004767	003744			FT14EC: JSR	PC, INIT1	; CLEAR RH	
882	006136	000207				RTS	PC	; RETURN	
883	006140	000240				FT14RER: NOP			
884	006142	032777	020000	172466		BIT	#2000, JSWR	; SEE IF PRINT INHIBITED	
885	006150	001060				BNE	FT14R3	; IF SO: BR	
886	006152	012704	016372			MOV	#MSFT14, R4		
887	006156	004767	004346			JSR	PC, TOUT	; PRINT HEADER	
888	006162	012704	014140			MOV	#MSG9, R4		
889	006166	004767	004336			JSR	PC, TOUT	; PRINT ERROR TYPE	
890	006172	012704	014373			MOV	#MSG20, R4	; SET NRZ TAG POINTER	
891	006176	032767	002000	172600		BIT	#2000, UDES	; SEE IF PE	
892	006204	001402				BEQ	FT14R0	; IF NOT: BR	
893	006206	012704	014401			MOV	#MSG21, R4	; ELSE SET PE TAG POINTER	
894	006212	004767	004312			FT14R0: JSR	PC, TOUT	; PRINT TAG	
895	006216	032767	000002	172552		BIT	#2, FUN	; SEE IF READ REVERSE	
896	006224	001003				BNE	FT14R1	; IF SO: BR	
897	006226	012704	014353			MOV	#MSG17, R4		
898	006232	000402				BR	FT14R2	; GO PRINT	
899	006234	012704	014333			FT14R1: MOV	#MSG16, R4		
900	006240	004767	004264			FT14R2: JSR	PC, TOUT	; PRINT FRWD/REV	
901	006244	012704	014406			MOV	#MSG22, R4		
902	006250	004767	004254			JSR	PC, TOUT	; PRINT EXPT TAG	
903	006254	010503				MOV	R5, R3		
904	006256	042703	177700			BIC	#177700, R3	; MASK RECORD NUMBER	
905	006262	004767	004406			JSR	PC, OCTP	; PRINT EXPT RECORD NUMBER	
906	006266	012704	014416			MOV	#MSG23, R4		
907	006272	004767	004232			JSR	PC, TOUT	; PRINT RCVD TAG	
908	006276	017703	172304			MOV	JFC, R3		
909	006302	042703	177700			BIC	#177700, R3	; MASK RECORD NUMBER	
910	006306	004767	004362			JSR	PC, OCTP	; PRINT ACTUAL RECORD NUMBER	
911	006312	005777	172320			FT14R3: TST	JSWR	; SEE IF HALT ON ERROR	
912	006316	100001				BPL	FT14X	; IF NOT: BR	
913	006320	000000				HALT			
914	006322	004767	004776			FT14X: JSR	PC, CKSWR	; CHECK FOR CNTL G	
915	006326	005767	172460			TST	NRZOF	; SEE IF NRZ ONLY	
916	006332	001011				BNE	FT14XX	; IF SO: BR	
917	006334	032767	002000	172442		BIT	#2000, UDES	; SEE IF DONE PE	
918	006342	001005				BNE	FT14XX	; IF SO: BR	
919	006344	012767	002300	172432		MOV	#2300, UDES	; SET TO PE	
920	006352	000167	177126			JMP	FT14A1	; DO IN PE	

B04

TM02-TU16/TE16 BASIC FUNCTION TEST
DZTUBF.P11 15-AUG-77 09:41

MACY11 30(1046) 15-AUG-77 10:47 PAGE 26

SEQ 0040

921 006356 000167 174224

FT14XX: JMP TSCD2 ;RETURN TO SCHEDULAR

```

;ERASE TEST*****
922
923
924 006362 000240          FT15:  NOP
925 006364 005067 172356  CLR      STMSK
926 006370 012767 000100 172326  MOV      #100,RDYDX
927 006376 012767 000010 172322  MOV      #10,OPDYX
928 006404 012767 016414 172264  MOV      #MSFT15,EMADDR ;SET TEST HEADER
929 006412 004767 002472      JSR      PC,RWIND ;REWIND
930 006416 012767 020472 172260  MOV      #RDATA,BADDR ;SET BA
931 006424 012767 001700 172352  MOV      #1700,ODES ;SET NRZ, NORMAL
932 006432 012767 000025 172336  FT15A: MOV      #25,FUN ;SET ERASE OP-CODE
933 006440 012767 000200 172244  MOV      #200,RCNT ;SET TO ERASE 128 TIMES
934 006446 004767 002304      FT15B: JSR      PC,EXEC ;GO EXECUTE COMMAND
935 006452 012767 015275 172234  MOV      #MSG46,ERRP ;SET ERROR CODE
936 006460 004767 002522      JSR      PC,ERCHK ;GO CHECK ERRORS
937 006464 005767 172310      TST      SERFL ;SEE IF ANY ERRORS
938 006470 001032      BNE      FT15X ;IF SO EXIT
939 006472 005367 172214      DEC      RCNT ;SEE IF DONE ERASING
940 006476 001363      BNE      FT15B ;IF NOT: BR
941 006500 000240      NOP
942 006502 004767 002402      JSR      PC,RWIND ;REWIND
943 006506 012767 177600 172174  MOV      #-200,WCNT ;SET WC
944 006514 012767 000071 172254  MOV      #71,FUN ;SET READ FORWARD OP-CODE
945 006522 012767 000040 172174  MOV      #40,RDYDX ;SET DELAY
946 006530 004767 002222      JSR      PC,EXEC ;GO EXECUTE COMMAND
947 006534 000240      NOP
948 006536 012767 015671 172150  MOV      #MSG60,ERRP ;SET ERROR CODE
949 006544 012767 020000 172174  MOV      #20000,STMSK
950 006552 004767 002430      JSR      PC,ERCHK ;GO CHECK ERRORS
951 006556 000167 174024      FT15X: JMP      TSCD2 ;RETURN TO SCHEDULAR

```

```

;TAPE MARK WRITE/READ TEST*****
952
953
954 006562 000240
955 006564 012767 000001 172132
956 006572 012767 001000 172126
957 006600 012767 016436 172070
958 006606 012767 001700 172170
959 006614 004767 002270
960 006620 012767 177760 172060
961 006626 012767 177770 172054
962 006634 012767 000027 172134
963 006642 004767 002110
964 006646 012767 001000 172072
965 006654 012767 014222 172032
966 006662 004767 002320
967 006666 004767 002662
968 006672 012767 000077 172076
969 006700 004767 002052
970 006704 012767 001000 172034
971 006712 012767 014240 171774
972 006720 004767 002262
973 006724 004767 002624
974 006730 012767 000071 172040
975 006736 004767 002014
976 006742 012767 014265 171744
977 006750 004767 002232
978 006754 004767 002574
979 006760 032767 002000 172016
980 006766 001012
981 006770 005767 172016
982 006774 001007
983 006776 012767 002300 172000
984 007004 004767 003072
985 007010 000167 177604
986 007014 004767 003012
987 007020 000167 173562
988

FT16:  NOP
      MOV #1, RDYDX
      MOV #1000, OPDYX
      MOV #MSFT16, EMADDR ; SET HEADER
      MOV #1700, UDES ; SET TO NRZ, NORMAL, ODD
FT16A: JSR PC, RWD ; REWIND
FT16B: MOV #-20, FCNT ; FC=20
      MOV #-10, WCNT ; WC=10
      MOV #27, FUN ; SET WRITE TAPE MARK OP-CODE
      JSR PC, EXEC ; GO EXECUTE COMMAND
      MOV #1000, STMSK ; SET FOR FCE MASK
      MOV #MSG12, ERRP ; SET ERROR CODE
      JSR PC, ERCHK ; GO CHECK ERROR
      JSR PC, TMCHK ; GO SEE IF TM SET
      MOV #77, FUN ; SET USED REVERSE OP-CODE
      JSR PC, EXEC ; GO EXECUTE COMMAND
      MOV #1000, STMSK ; SET FCE ERROR MASK
      MOV #MSG13, ERRP ; SET ERROR CODE
      JSR PC, ERCHK ; GO CHECK ERRORS
      JSR PC, TMCHK ; GO SEE IF TM SET
      MOV #71, FUN ; SET READ FORWARD OP-CODE
      JSR PC, EXEC ; GO EXECUTE COMMAND
      MOV #MSG14, ERRP ; SET ERROR CODE
      JSR PC, ERCHK ; TO CHECK ERRORS
      JSR PC, TMCHK ; GO SEE IF TM SET
      BIT #2000, UDES ; SEE IF DONE PE
      BNE FT16X ; IF SO: BR
      TST NRZOF ; SEE IF NRZ ONLY
      BNE FT16X ; IF SO: BR
      MOV #2300, UDES ; SET PE, NORMAL
      JSR PC, INIT1 ; INITIALIZE
      JMP FT16B ; DO IN PE
FT16X: JSR PC, ITER ; DO ITERATIONS
      JMP TSCD2 ; RETURN TO SCHEDULAR

```

```

989
990
991
992 007024 005067 171662 FT17: CLR RCNT
993 007030 012767 016477 171640 MOV #MSFT17,EMADDR ;SET HEADER
994 007036 012767 001700 171740 MOV #1700,UDES ;SET TO NRZ
995 007044 004767 002040 FT17A: JSR PC,RWIND ;REWIND TAPE
996 007050 012767 000027 171720 FT17B: MOV #27,FUN
997 007056 012767 040000 171640 MOV #40000,RDYDX ;SET DRY DELAY
998 007064 012767 040000 171634 MOV #40000,OPDYX ;SET OP DELAY
999 007072 004767 001660 JSR PC,EXEC ;GO WRITE TM
1000 007076 012767 102300 171642 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1001 007104 012767 014312 171602 MOV #MSG15,ERRP ;SET ERROR TYPE
1002 007112 004767 002070 JSR PC,ERCHK ;GO CHECK ERROR
1003 007116 005767 171656 TST SERFL ;SEE IF ERROR
1004 007122 001144 BNE FT17X ;IF SO: BR
1005 007124 004767 002424 JSR PC,TMCHK ;GO SEE IF TM SET
1006 007130 000240 NOP
1007 007132 000240 NOP
1008 007134 032767 000100 171550 BIT #100,RCNT ;SEE IF DONE PATTERN
1009 007142 001046 BNE FT17D ;IF SO: BR
1010 007144 062767 000020 171540 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1011 007152 016767 171534 171560 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1012 007160 012767 177600 171522 MOV #-200,WCNT ;WC=128
1013 007166 012767 177400 171512 MOV #-400,FCNT ;FC=256
1014 007174 012767 016760 171502 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1015 007202 012767 000061 171566 MOV #61,FUN ;SET WRITE OP CODE
1016 007210 000240 FT17C: NOP
1017 007212 000240 NOP
1018 007214 004767 001536 JSR PC,EXEC ;GO WRITE
1019 007220 012767 014222 171466 MOV #MSG12,ERRP ;SET ERROR CODE
1020 007226 012767 102300 171512 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1021 007234 004767 001746 JSR PC,ERCHK ;GO CHECK ERROR
1022 007240 005767 171534 TST SERFL ;SEE IF ERROR
1023 007244 001073 BNE FT17X ;IF SO: BR
1024 007246 005367 171466 DEC TEMP1 ;SEE IF DONE ALL
1025 007252 001356 BNE FT17C ;IF NOT: BR
1026 007254 000167 177570 JMP FT17B ;ELSE GO DO TM
1027 007260 000240 FT17D: NOP
1028 007262 012767 000033 171506 MOV #33,FUN ;SET SPACE REVERSE
1029 007270 012767 014333 171416 MOV #MSG16,ERRP ;SET ERROR CODE
1030 007276 012767 177600 171424 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1031 007304 012767 000005 171400 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1032 007312 004767 002564 FT17E: JSR PC,INIT1 ;GO INIT
1033 007316 004767 001434 JSR PC,EXEC ;GO SPACE
1034 007322 012767 001000 171416 MOV #1000,STMSK ;SET ERROR MASK
1035 007330 004767 001652 JSR PC,ERCHK ;GO CHECK ERROR
1036 007334 005767 171440 TST SERFL ;SEE IF ERROR
1037 007340 001035 BNE FT17X ;IF SO: BR
1038 007342 004767 002206 JSR PC,TMCHK ;GO SEE IF TM SET
1039 007346 005367 171340 DEC RCNT ;SEE IF DONE SPACES
1040 007352 001357 BNE FT17E ;IF NOT: BR
1041 007354 022767 000031 171414 CMP #31,FUN ;SEE IF DONE FORWARD
1042 007362 001410 BEQ FT17F ;IF SO: BR
1043 007364 012767 014353 171322 MOV #MSG17,ERRP ;SET ERROR CODE
1044 007372 012767 000031 171376 MOV #31,FUN ;SET TO SPACE FORWARD

```

F04

TM02-TU16/TE16 BASIC FUNCTION TEST
DZTUBF.P11 15-AUG-77 09:41

MACY11 30(1046) 15-AUG-77 10:47 PAGE 30

SEQ 0044

1045	007400	000167	177672			JMP	FT17D1		;DO FORWARD
1046	007404	032767	002000	171372	FT17F:	BIT	#2000,UDES		;SEE IF DONE PE
1047	007412	001010				BNE	FT17X		;IF SO: BR
1048	007414	005767	171372			TST	NR20F		;SEE IF NRZ ONLT
1049	007420	001005				BNE	FT17X		;IF SO: BR
1050	007422	012767	002300	171354		MOV	#2300,UDES		;SET TO PE
1051	007430	000167	177410			JMP	FT17A		;GO PE
1052	007434	000167	173146		FT17X:	JMP	TSCD2		;RETURN TO SCHEDULAR

```

1053
1054
1055
1056 007440 000240
1057 007442 012767 016525 171226
1058 007450 004767 001434
1059 007454 012767 000003 171324
1060 007462 004767 002126
1061 007466 012767 016760 171210
1062 007474 012767 177400 171204
1063 007502 012767 177600 171200
1064 007510 012767 001700 171266
1065 007516 012767 000061 171252
1066 007524 004767 001226
1067 007530 012767 015275 171156
1068 007536 004767 001444
1069 007542 005767 171232
1070 007546 001050
1071 007550 012767 014333 171136
1072 007556 012767 000057 171212
1073 007564 062767 000376 171112
1074 007572 004767 001160
1075 007576 004767 001404
1076 007602 012767 014353 171104
1077 007610 012767 000051 171160
1078 007616 162767 000376 171060
1079 007624 004767 001126
1080 007630 004767 001352
1081 007634 032767 002000 171142
1082 007642 001012
1083 007644 005767 171142
1084 007650 001007
1085 007652 012767 002300 171124
1086 007660 004767 002216
1087 007664 000167 177626
1088 007670 004767 002136
1089 007674 000167 172706

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

```

```

1090
1091
1092
1093 007700 012767 016556 170770 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1094 007706 004767 001176 JSR PC,RWND ;GO REWIND
1095 007712 012767 000003 171066 MOV #3,PATRN
1096 007720 004767 001670 JSR PC,DSUP ;GO SET PATTERN 3
1097 007724 012767 016760 170752 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1098 007732 012767 176340 170746 MOV #-1440,FCNT ;SET FC=800
1099 007740 012767 177160 170742 MOV #-620,WCNT ;SET WC=400
1100 007746 012767 001700 171030 MOV #1700,UDES ;SET NRZ, NORMAL
1101 007754 012767 000061 171014 MOV #61,FUN ;SET WRITE OP-CODE
1102 007762 004767 000770 JSR PC,EXEC ;GO DO WRITE 1
1103 007766 012767 014222 170720 MOV #MSG12,ERRP ;SET ERROR CODE
1104 007774 004767 001206 JSR PC,ERCHK ;GO CHECK FOR ERROR
1105 010000 004767 000752 JSR PC,EXEC ;YES DO WRITE 2
1106 010004 004767 001176 JSR PC,ERCHK ;YES CHECK FOR ERROR
1107 010010 000240 NOP
1108 010012 004767 001072 JSR PC,RWND ;GO REWIND
1109 010016 012767 177160 170662 MOV #-620,FCNT ;SET FC=400
1110 010024 012767 177470 170656 MOV #-310,WCNT ;SET WC=200
1111 010032 004767 000720 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1112 010036 000240 FT21A: NOP
1113 010040 004767 001044 JSR PC,RWND ;REWIND
1114 010044 012767 020472 170632 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1115 010052 012767 177160 170626 MOV #-620,FCNT ;SET FC=400
1116 010060 012767 177470 170622 MOV #-310,WCNT ;SET WC=200
1117 010066 012767 000071 170702 MOV #71,FUN ;SET READ OP-CODE
1118 010074 004767 000656 JSR PC,EXEC ;GO READ RECORD 1
1119 010100 012767 014265 170606 MOV #MSG14,ERRP ;SET ERROR CODE
1120 010106 004767 001074 JSR PC,ERCHK ;GO CHECK FOR ERROR
1121 010112 000240 NOP
1122 010114 052777 000010 170466 BIS #10,ACS ;INHIBIT BA INCREMENT
1123 010122 012767 176340 170556 MOV #-1440,FCNT ;SET FC=800
1124 010130 012767 177160 170552 MOV #-620,WCNT ;SET WC=400
1125 010136 004767 000614 JSR PC,EXEC ;GO READ RECORD 2
1126 010142 022777 001440 170436 CMP #1440,AF C ;SEE IF READ RECORD 2
1127 010150 001423 BEQ FT21X ;IF SO: BR
1128 010152 022777 001441 170426 CMP #1441,AF C ;++F CHECK FOR 801 FRAMES
1129 010160 001417 BEQ FT21X ;++F IF SO: BR
1130 010162 012767 015242 170546 MOV #MSG45,ERADD ;++F SET ERROR CODE
1131 010170 022777 001440 170410 CMP #1440,AF C ;++F MORE THAN 801 FRAMES ?
1132 010176 101403 BLOS 1$ ;++F IF SO: BR
1133 010200 012767 015173 170530 MOV #MSG44,ERADD ;++F SET ERROR CODE
1134 010206 012767 010036 170546 1$: MOV #FT21A,SCOLP ;SET SCOPE ADDRESS
1135 010214 004767 173334 JSR PC,FT3ER ;GO PRINT ERROR
1136 010220 004767 001606 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1137 010224 000167 172356 JMP TSCD2 ;RETURN TO SCHEDULAR
1138
1139

```

```

;BUFFERED COMMAND TEST*****
1140
1141
1142 010230 012767 016605 170440 FT22: MOV #MSFT22,EMADDR ;SET TEST HEADER
1143 010236 004767 000646 JSR PC,RWIND ;GO REWIND
1144 010242 012700 000003 MOV #3,RO ;SET NUMBER OF WRITES
1145 010246 012767 001700 170530 MOV #1700,UDES ;SET TO NRZ NORMAL
1146 010254 012767 016760 170422 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1147 010262 012767 177000 170416 MOV #-1000,FCNT ;SET FC=1000
1148 010270 012767 177400 170412 MOV #-400,WCNT ;SET WC=400
1149 010276 012767 000061 170472 MOV #61,FUN ;SET WRITE OP-CODE
1150 010304 004767 000446 FT22A: JSR PC,EXEC ;GO DO WRITE
1151 010310 005300 DEC RO ;SEE IF DONE ALL
1152 010312 001374 BNE FT22A ;IF NOT: BR
1153 010314 000240 NOP
1154 010316 012777 000007 170254 MOV #7,AC1 ;START REWIND
1155 010324 032777 000200 170260 FT22B: BIT #200,ADS
1156 010332 001774 BEQ FT22B
1157 010334 004767 001542 JSR PC,INIT1 ;INITIALIZE
1158 010340 012767 000010 170356 MOV #10,RDYDX ;SET LONG READY DELAY
1159 010346 004767 000404 JSR PC,EXEC ;ISSUE BUFFERED WRITE
1160 010352 000240 NOP
1161 010354 012767 015373 170332 MOV #MSG49,ERRP ;SET ERROR CODE
1162 010362 012767 102300 170356 MOV #102300,STMSK ;MARK DATA ERROR
1163 010370 004767 000612 JSR PC,ERCHK ;GO CHECK ERROR
1164 010374 032777 000002 170210 BIT #2,ADS ;SEE IF BOT IS SET
1165 010402 001410 BEQ FT22X ;IF NOT: BR
1166 010404 012767 015421 170324 MOV #MSG50,ERADD ;SET ERROR CODE
1167 010412 012767 010230 170342 MOV #FT22,SCOLP
1168 010420 004767 173130 JSR PC,FT3ER ;GO DO ERROR
1169 010424 004767 001402 FT22X: JSR PC,ITER ;GO SEE IF ITERATION
1170 010430 000167 172152 JMP TSCD2 ;RETURN TO SCHEDULAR
1171
1172

```



```

1173                                     ;READ-IN PRESET TEST*****
1174
1175 010434 005767 170242 FT23: TST SLVN ;SEE IF SLAVE SELECT=0
1176 010440 001103 BNE FT23X ;IF NOT: BR
1177 010442 012767 016642 170226 MOV #MSFT23,EMADDR ;SET TEST HEADER
1178 010450 004767 001426 JSR PC,INIT1 ;GO INIT
1179 010454 012767 001700 170322 MOV #1700,UDES ;SET TO NRZ NORMAL
1180 010462 012767 016760 170214 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1181 010470 012767 177400 170210 MOV #-400,FCNT ;SET FC=400
1182 010476 012767 177600 170204 MOV #-200,WCNT ;SET WC=200
1183 010504 012767 000061 170264 MOV #61,FUN ;SET WRITE OP-CODE
1184 010512 004767 000240 JSR PC,EXEC ;GO DO WRITE
1185 010516 000240 NOP
1186 010520 004767 001356 JSR PC,INIT1 ;INITIALIZE
1187 010524 012767 000021 170244 MOV #21,FUN ;SET READ-IN PRESET OP CODE
1188 010532 004767 000220 JSR PC,EXEC ;GO DO COMMAND
1189 010536 005000 CLR R0
1190 010540 012703 000004 MOV #4,R3 ;SET MULT
1191 010544 032777 020000 170040 FT23A: BIT #20000,ADS ;SEE IF PIP RESET
1192 010552 001404 BEQ FT23B ;IF SO: BR
1193 010554 005300 DEC R0
1194 010556 001372 BNE FT23A ;AWAIT PIP RESET
1195 010560 005303 DEC R3
1196 010562 001370 BNE FT23A ;DELAY
1197 010564 032777 000002 170020 FT23B: BIT #2,ADS ;SEE IF BOT
1198 010572 001010 BNE FT23C ;IF SO: BR
1199 010574 012767 015457 170134 MOV #MSG51,ERADD ;SET ERROR CODE
1200 010602 012767 010434 170152 MOV #FT23,SCOLP
1201 010610 004767 172740 JSR PC,FT3ER ;GO DO ERROR
1202 010614 012701 141000 FT23C: MOV #141000,R1 ;SET EXPT TC
1203 010620 016700 170006 MOV TC,R0 ;SET TC ADDRESS
1204 010624 020110 CMP R1,(R0) ;SEE IF EXPT=RCVD
1205 010626 001410 BEQ FT23X ;IF SO: BR
1206 010630 012767 015513 170100 MOV #MSG52,ERADD ;SET ERROR CODE
1207 010636 012767 010434 170116 MOV #FT23,SCOLP ;CLEAR SCOPE ADDRESS
1208 010644 004767 172414 JSR PC,FT2ER ;GO DO ERROR
1209 010650 000167 171732 FT23X: JMP TSCD2 ;RETURN TO SCHEDULAR
1210
1211

```

K04

TM02-TU16/TE16 BASIC FUNCTION TEST
DZTUBF.P11 15-AUG-77 09:41

MACY11 30(1046) 15-AUG-77 10:47 PAGE 35

SEQ 0049

```

1212                                     ;REWIND: OFF LINE TEST*****
1213
1214 010654 032777 004000 167754 FT24: BIT      #4000, @SWR      ;SEE IF IN CONTINUOUS MODE
1215 010662 001033                                     BNE      FT24XX      ;IF SO: BR
1216 010664 012767 016675 170004     MOV      #MSFT24, EMADDR ;SET TEST HEADER
1217 010672 004767 001204                                     JSR      PC, INIT1    ;GO INITIAIZE
1218 010676 012777 000003 167674     MOV      #3, @C1      ;ISSUE REWIND: OFF LINE COMMAND
1219 010704 012700 004000                                     MOV      #4000, R0
1220 010710 005300                                     FT24A: DEC      R0      ;DELAY
1221 010712 001376                                     BNE      FT24A
1222 010714 032777 010000 167670     BIT      #10000, @DS   ;SEE IF MOL IS RESET
1223 010722 001407                                     BEQ      FT24X      ;IF SO: BR
1224 010724 005067 170032                                     CLR      SCOLP      ;ASSURE NO SCOPE
1225 010730 012767 015532 170000     MOV      #MSG53, ERADD ;SET ERROR CODE
1226 010736 004767 172612                                     JSR      PC, FT3ER    ;GO DO ERROR
1227 010742 012704 015556                                     FT24X: MOV      #MSG54, R4
1228 010746 004767 001556                                     JSR      PC, TTOUT    ;PRINT ON LINE REQUEST
1229 010752 000167 171630                                     FT24XX: JMP      TSCD2 ;RETURN TO SCHEDULAR
1230
1231

```

```

1232                                     ;COMMAND EXECUTE SUBROUTINE*****
1233
1234 010756 000240 EXEC: NOP
1235 010760 056777 170020 167644 BIS UDES, JTC ;LOAD TAPE CONT
1236 010766 016777 167716 167606 MOV WCNT, JWC ;LOAD WC
1237 010774 016777 167706 167604 MOV FCNT, JFC ;LOAD FC
1238 011002 016777 167676 167574 MOV BADDR, JBA ;LOAD BA
1239 011010 022767 000031 167760 CMP #31, FUN ;SEE IF SPACE FORWARD
1240 011016 001404 BEQ EXECA ;IF SO: BR
1241 011020 022767 000033 167750 CMP #33, FUN ;SEE IF SPACE REVERSE
1242 011026 001003 BNE EXECB ;IF NOT: BR
1243 011030 016777 167674 167550 EXECA: MOV SCNT, JFC ;SET SPACE COUNT
1244 011036 000240 EXECB: NOP
1245 011040 016777 167732 167532 MOV FUN, JCI ;LOAD OP-CODE + GO
1246 011046 000240 NOP
1247 011050 016703 167650 MOV RDYDX, R3 ;SET DELAY
1248 011054 005004 CLR R4
1249 011056 032777 000200 167526 EXECC: BIT #200, JDS ;SEE IF DRY
1250 011064 001004 BNE EXECX ;IF SO: BR
1251 011066 005304 DEC R4
1252 011070 001372 BNE EXECC
1253 011072 005303 DEC R3 ;DELAY FOR DRY
1254 011074 001370 BNE EXECC
1255 011076 016703 167624 EXECX: MOV OPDYX, R3
1256 011102 005303 EXECXA: DEC R3 ;DELAY
1257 011104 001376 BNE EXECXA
1258 011106 000207 EXECXX: RTS PC ;RETURN TO CALLER
1259

```

```

1260                                     ;REWIND SUBROUTINE*****
1261
1262 011110 000240          RWND:  NOP
1263 011112 004767 000764          JSR  PC, INIT1          ;INIT
1264 011116 012777 000007 167454  MOV  #7, R0          ;START REWIND
1265 011124 012700 040000          MOV  #40000, R0
1266 011130 005300          RWNDA: DEC  R0
1267 011132 001376          BNE  RWNDA          ;DELAY
1268 011134 032777 020000 167450  RWNDB: BIT  #20000, R0S
1269 011142 001374          BNE  RWNDB          ;AWAIT PIP
1270 011144 032777 000002 167440  BIT  #2, R0S          ;SEE IF BOT
1271 011152 001012          BNE  RWNDX          ;IF SO: BR
1272 011154 016704 167516          MOV  EMADDR, R4
1273 011160 004767 001344          JSR  PC, TTOUT        ;PRINT HEADER
1274 011164 012704 013726          MOV  #MSG2, R4
1275 011170 004767 001334          JSR  PC, TTOUT        ;PRINT REWIND ERROR
1276 011174 000167 171406          JMP  TSCD2           ;RETURN TO SECHEDULAR
1277 011200 004767 000676          RWNDX: JSR PC, INIT1 ;INIT
1278 011204 000207          RTS  PC              ;RETURN TO CALLER
1279

```

```

;ERROR CHECK SUBROUTINE*****
1280
1281
1282 011206 005067 167566 ERCHK: CLR SERFL ;CLEAR FLAG
1283 011212 012767 167374 167532 MOV ADS,DSAV ;SAVE DRIVE STATUS REGISTER
1284 011220 032777 040000 167364 BIT #40000,ADS ;SEE IF ERROR
1285 011226 001001 BNE ERPT ;IF SO: BR
1286 011230 000207 RTS PC ;RETURN
1287 011232 017704 167356 ERPT: MOV @ER,R4 ;GET ERROR REGISTER
1288 011236 032767 002000 167540 BIT #2000,UDES ;SEE IF PE
1289 011244 001403 BEQ ERPTA1 ;IF SO: BR
1290 011246 042767 000200 167472 BIC #200,STMSK ;RESET PEF MASK
1291 011254 046704 167466 ERPTA1: BIC STMSK,R4 ;MASK DONT CARE BITS
1292 011260 001530 BEQ ERPTX ;IF NO UNEXPECTED ERRORS: BR
1293 011262 012767 000001 167510 ERPTG: MOV #1,SERFL ;SET FLAG
1294 011270 032777 020000 167340 BIT #20000,JSWR ;SEE IF SHOULD PRINT ERRORS
1295 011276 001115 BNE ERPTD ;IF NOT: BR
1296 011300 005767 167370 TST HDRFL ;SEE IF DONE HEADER
1297 011304 001006 BNE ERPTA ;IF SO: BR
1298 011306 005267 167362 INC HDRFL ;SET HEADER FLAG
1299 011312 016704 167360 MOV EMADDR,R4
1300 011316 004767 001206 JSR PC,TTOUT ;PRINT HEADER
1301 011322 016704 167366 ERPTA: MOV ERAP,R4 ;GET ERROR CODE
1302 011326 001414 BEQ ERPTB ;IF NONE: BR
1303 011330 004767 001174 JSR PC,TTOUT ;PRINT ERROR CODE
1304 011334 012704 014373 MOV #MSG20,R4 ;SET NRZ TAG
1305 011340 032777 002000 167264 BIT #2000,ATC ;SEE IF PE
1306 011346 001402 BEQ ERPT1A ;IF NOT: BR
1307 011350 012704 014401 MOV #MSG21,R4 ;ELSE SET PE TAG
1308 011354 004767 001150 ERPT1A: JSR PC,TTOUT ;PRINT TAG
1309 011360 016704 167332 ERPTB: MOV ERAP1,R4 ;SEE IF CODE 2
1310 011364 001402 BEQ ERPTB1 ;IF NOT: BR
1311 011366 004767 001136 JSR PC,TTOUT ;PRINT CODE 2
1312 011372 032777 010000 167236 ERPTB1: BIT #10000,JSWR ;SEE IF ITERATION
1313 011400 001010 BNE ERPTC ;IF NOT: BR
1314 011402 012704 015645 MOV #MSG56,R4
1315 011406 004767 001116 JSR PC,TTOUT ;PRINT ITER TAG
1316 011412 016703 167332 MOV ITCNT,R3
1317 011416 004767 001252 JSR PC,OC1P ;PRINT ITERATION
1318 011422 012704 013640 ERPTC: MOV #MSG1,R4 ;PRINT REGISTER TAG
1319 011426 004767 001076 JSR PC,TTOUT ;PRINT REGISTER TAG
1320 011432 017703 167142 MOV @C1,R3
1321 011436 004767 001220 JSR PC,OC1PE ;PRINT CS1
1322 011442 017703 167134 MOV @WC,R3
1323 011446 004767 001210 JSR PC,OC1PE ;PRINT WC
1324 011452 017703 167126 MOV @BA,R3
1325 011456 004767 001200 JSR PC,OC1PE ;PRINT BA
1326 011462 017703 167120 MOV @FC,R3
1327 011466 004767 001170 JSR PC,OC1PE ;PRINT FC
1328 011472 017703 167112 MOV @CS,R3
1329 011476 004767 001160 JSR PC,OC1PE ;PRINT CS2
1330 011502 017703 167104 MOV @DS,R3
1331 011506 004767 001150 JSR PC,OC1PE ;PRINT DS
1332 011512 017703 167076 MOV @ER,R3
1333 011516 004767 001140 JSR PC,OC1PE ;PRINT ER
1334 011522 017703 167104 MOV @TC,R3
1335 011526 004767 001130 JSR PC,OC1PE ;PRINT TC

```

TMO2-TU16/TE16 BASIC FUNCTION TEST
DZTUBF.P11 15-AUG-77 09:41

MACY11 30(1046) 15-AUG-77 10:47 PAGE 39

SEQ 0053

1336	011532	005777	167100
1337	011536	100001	
1338	011540	000000	
1339	011542	004767	001556
1340	011546	004767	000330
1341	011552	000207	
1342			
1343			

ERPTD:	TST	@SWR	;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

```

1344                                     ;TAPE MARK STATUS CHECK*****
1345
1346 011554 032767 000004 167170 TMCHK: BIT      #4, DSAV      ;SEE IF TM SET
1347 011562 001401          BEQ      TMCHK1      ;IF NOT: BR
1348 011564 000207          TMCHK0: RTS      PC          ;ELSE RETURN
1349 011566 005767 167206          TMCHK1: TST      SERFL      ;SEE IF HAD ERROR
1350 011572 001374          BNE      TMCHK0      ;IF SO: BR
1351 011574 012767 015655 167114          MOV      #MSG57, ERRP1    ;SET ERROR CODE 2
1352 011602 004767 177454          JSR      PC, ERPTG      ;GO PRINT TM ERROR
1353 011606 005067 167104          CLR      ERRP1        ;CLEAR CODE 2 FLAG
1354 011612 000207          RTS      PC          ;RETURN
1355
1356                                     ;DATA SETUP ROUTINE*****
1357
1358 011614 000240          DSUP:  NOP
1359 011616 012703 016760          DSO:  MOV      #WDATA, R3    ;R3 = ADDRS OF WRITE BUFFER
1360 011622 016701 167160          MOV      PATRN, R1      ;R1 = PATTERN SELECTOR
1361 011626 000241          CLC
1362 011630 006101          ROL      R1            ;MAKE PATTERN SELECTOR EVEN
1363 011632 000171 001026          JMP      @DATBL(R1)     ;GO GENERATE PATTERN
1364 011636 032777 010000 166762          DS1:  BIT      #10000, @DT ;SEE IF SEVEN TRACK
1365 011644 001410          BEQ      DS3            ;IF NOT: BR
1366 011646 012702 000640          MOV      #640, R2      ;SET BUFFER SIZE
1367 011652 012701 016760          MOV      #WDATA, R1    ;SET START OF BUFFER
1368 011656 042721 140300          DS2:  BIC      #140300, (R1)+ ;MASK FOR 7 CH
1369 011662 005302          DEC      R2            ;SEE IF DONE
1370 011664 001374          BNE      DS2            ;IF NOT: BR
1371 011666 012702 000640          DS3:  MOV      #640, R2  ;R2=BUFFER SIZE +2
1372 011672 012701 020472          MOV      #RDATA, R1    ;R1=READ DATA START
1373 011676 005021          DS4:  CLR      (R1)+    ;CLEAR BUFFER
1374 011700 005302          DEC      R2            ;SEE IF DONE ALL
1375 011702 001375          BNE      DS4            ;IF NOT: BR
1376 011704 000207          RTS      PC          ;EXIT
1377
1378                                     ;ALL ONES*****
1379
1380 011706 012701 177777          DAT1:  MOV      #-1, R1   ;R1=DATA
1381 011712 012702 000640          DAT1A: MOV      #640, R2  ;R2=WORD COUNT +2
1382 011716 010123          DAT1B: MOV      R1, (R3)+ ;LOAD BUFFER
1383 011720 005302          DEC      R2            ;SEE IF DONE
1384 011722 001375          BNE      DAT1B         ;IF NOT: BR
1385 011724 000167 177706          JMP      DS1          ;RETURN
1386
1387                                     ;ALL ZEROS*****
1388
1389 011730 005001          DAT2:  CLR      R1      ;R1=DATA
1390 011732 000167 177754          JMP      DAT1A        ;LOAD BUFFER
1391

```

```

1392                                     ;ONE/ZERO IN ALTERNATING CHARACTERS*****
1393
1394 011736 012701 125125 DAT3: MOV #125125,R1 ;R1=DATA
1395 011742 000167 177744 JMP DAT1A ;LOAD BUFFER
1396
1397                                     ;ALL BITS 0-377*****
1398
1399 011746 005001 DAT4: CLR R1 ;R1=STARTING DATA
1400 011750 012702 001500 MOV #1500,R2 ;R2=CHARACTER COUNT
1401 011754 110123 DAT4A: MOV R1,(R3)+ ;LOAD BUFFER
1402 011756 105201 INCB R1 ;BUMP DATA
1403 011760 005302 DEC R2 ;SEE IF DONE
1404 011762 001374 BNE DAT4A ;IF NOT: BR
1405 011764 000167 177646 JMP DS1 ;RETURN
1406
1407
1408                                     ;SCOPE LOOP ON ERROR SUBROUTINE*****
1409
1410 011770 004767 001330 SCOPE: JSR PC,CKSWR ;CHECK FOR CNTL G
1411 011774 000240 NOP
1412 011776 032777 040000 166632 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
1413 012004 001001 BNE SCOPE1 ;IF SO: BR
1414 012006 000207 RTS PC ;ELSE EXIT
1415 012010 000240 SCOPE1: NOP
1416 012012 005767 166744 TST SCOLP ;SEE IF SCOPE ADDRESS
1417 012016 001001 BNE SCOPE2 ;IF NOT: BR
1418 012020 000207 RTS PC ;ELSE EXIT
1419 012022 005726 SCOPE2: TST (SP)+ ;RESET STACK
1420 012024 005726 TST (SP)+
1421 012026 000177 166730 JMP @SCOLP ;LOOP ON ERROR
1422
1423                                     ;TEST ITERATION SUBROUTINE*****
1424
1425 012032 004767 001266 ITER: JSR PC,CKSWR ;CHECK FOR CNTL G
1426 012036 000240 NOP
1427 012040 032777 010000 166570 BIT #10000,@SWR ;SEE IF ITERATIONS
1428 012046 001403 BEQ ITER1 ;IF SO: BR
1429 012050 005067 166674 ITER0: CLR ITCNT ;CLEAR ITERATION COUNTER
1430 012054 000207 RTS PC ;ELSE EXIT
1431 012056 005267 166666 ITER1: INC ITCNT ;BUMP COUNTER
1432 012062 026767 166662 166561 CMP ITCNT,ITAMT ;SEE IF DONE ALL
1433 012070 001767 BEQ ITER0 ;IF SO: BR
1434 012072 005726 TST (SP)+ ;RESET STACK
1435 012074 017700 166664 MOV @ITRLP,R0 ;SET ITERATION POINTER
1436 012100 000110 JMP (R0) ;GO ITERATE
1437
1438                                     ;INITIALIZE SUBROUTINE*****
1439
1440 012102 000240 INIT1: NOP
1441 012104 012777 000040 166476 MOV #40,@CS ;INIT
1442 012112 016777 166562 166470 INIT2: MOV DRVN,@CS ;SELECT DRIVE
1443 012120 016777 166556 166504 MOV SLVN,@TC ;SELECT SLAVE
1444 012126 000207 RTS PC ;RETURN
1445

```



```

1446                                     ;MAG TAPE INTERRUPT HANDLER*****
1447
1448 012130 000240 MTINT: NOP
1449 012132 022626          CMP      (SP)+,(SP)+      ;RESET STACK POINTER
1450 012134 000240          NOP
1451 012136 000240          NOP
1452 012140 000177 166570  JMP      @RTRN      ;RETURN TO CALLER
1453
1454                                     ;TTY INTERRUPT HANDLER*****
1455
1456 012144 000240 TTINT: NOP
1457 012146 000240          NOP
1458 012150 000240          NOP
1459 012152 000002          RTI
1460
1461                                     ;BUS ADDRESS TRAP HANDLER*****
1462
1463 012154 000240 TRAP:  NOP
1464 012156 032777 020000 166452 BIT      #20000,@SWR      ;SEE IF SHOULD PRINT ERRORS
1465 012164 001020          BNE      TRAP2      ;IF NOT: BR
1466 012166 005767 166502 TST      HDRFL      ;SEE IF DONE HEADER
1467 012172 001006          BNE      TRAP1      ;IF SO: BR
1468 012174 005267 166474 INC      HDRFL      ;ELSE SET HEADER FLAG
1469 012200 016704 166472 MOV      EMADDR,R4
1470 012204 004767 000320 JSR      PC,TTOUT      ;PRINT HEADER
1471 012210 012704 014426 TRAP1:  MOV      #MSG24,R4
1472 012214 004767 000310 JSR      PC,TTOUT      ;PRINT ERROR
1473 012220 010103          MOV      R1,R3
1474 012222 004767 000446 JSR      PC,OCTP      ;PRINT ADDRESS OF TRAP
1475 012226 005777 166404 TRAP2:  TST      @SWR      ;SEE IF HALT ON ERROR
1476 012232 100001          BPL      TRAPX      ;IF NOT: BR
1477 012234 000000          HALT
1478 012236 004767 001062 TRAPX:  JSR      PC,CKSWR      ;CHECK FOR CNTL G
1479 012242 022626          CMP      (SP)+,(SP)+      ;RESET STACK
1480 012244 012767 003026 166510 MOV      #FT1A,SCOLP      ;SET SCOPE ADDRESS
1481 012252 004767 177512 JSR      PC,SCOPE      ;GO SEE IF SCOPE LOOP
1482 012256 005767 166526 TST      RHTF      ;SEE IF INITIAL ADDRESS TEST
1483 012262 001402          BEQ      TRAPXX      ;IF NOT: BR
1484 012264 000167 167430 JMP      STOB      ;ELSE REDO ADDRESS REQUEST
1485 012270 000167 170536 TRAPXX: JMP      FT1B      ;RETURN TO TEST 1
1486

```

```

1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504 012274 005067 166440      TTR:  CLR      TEMP1      ;CLEAR FIRST CHARACTER FLAG
1505 012300 005000                CLR      RO
1506 012302 004767 000152      TTR0:  JSR      PC,TTIN    ;GO READ CHARACTER
1507 012306 122767 000015 166354  CMPB    #15,TIB        ;SEE IF CR
1508 012314 001005                BNE     TTR1          ;IF NOT: BR
1509 012316 005767 166416      TST     TEMP1        ;SEE IF FIRST CHARACTER
1510 012322 001446                BEQ     TTR5          ;IF SO: BR
1511 012324 000167 000066      JMP     TTR2          ;ELSE GO LOAD VALUE
1512 012330 122767 000060 166332  TTR1:  CMPB    #60,TIB        ;SEE IF CHAR IS LESS THAN 0
1513 012336 101402                BLOS   TTR1A         ;IF NOT: BR
1514 012340 000167 000076      JMP     TTR1A        ;ELSE GO TO ERROR
1515 012344 122767 000070 166316  TTR1A: CMPB    #70,TIB        ;SEE IF CHAR IS GREATER THAN 7
1516 012352 101002                BHI    TTR1B         ;IF NOT: BR
1517 012354 000167 000062      JMP     TTR1B        ;ELSE GO TO ERROR
1518 012360 005267 166354      TTR1B: INC     TEMP1      ;SET FIRST CHARACTER FLAG
1519 012364 000241                CLC
1520 012366 006100                ROL    RO
1521 012370 000241                CLC
1522 012372 006100                ROL    RO      ;SHIFT 3 LEFT
1523 012374 000241                CLC
1524 012376 006100                ROL    RO
1525 012400 042767 177770 166262  BIC    #177770,TIB    ;STRIP ASCII
1526 012406 056700 166256      BIS    TIB,RO        ;LOAD CHARACTER
1527 012412 005301                DEC    R1            ;SEE IF DONE
1528 012414 001332                BNE    TTR0          ;IF NOT: BR
1529 012416 020002                CMP    RO,R2         ;SEE IF EXCEEDED MAXIMUM LIMIT
1530 012420 101402                BLOS   TTR3          ;IF NOT: BR
1531 012422 000167 000014      JMP     TTR3          ;ELSE GO TO ERROR
1532 012426 020300                CMP    R3,RO         ;SEE IF BELOW MINIMUM LIMIT
1533 012430 101402                BLOS   TTR4          ;IF NOT: BR
1534 012432 000167 000004      JMP     TTR4          ;ELSE GO TO ERROR
1535 012436 010015                TTR4:  MOV    RO,(R5)   ;LOAD VALUE
1536 012440 000207                TTR5:  RTS    PC      ;EXIT
1537

```

1538
1539
1540 012442 012704 014132
1541 012446 004767 000056
1542 012452 162716 000020
1543 012456 000207
1544
1545
1546
1547 012460 005077 166154
1548 012464 005077 166152
1549 012470 105777 166144
1550 012474 100375
1551 012476 017767 166140 166164
1552 012504 042767 177600 166156
1553 012512 105777 166126
1554 012516 100375
1555 012520 116777 166144 166120
1556 012526 000207
1557
1558
1559
1560 012530 112467 166132
1561 012534 122767 000043 166124
1562 012542 001446
1563 012544 122767 000045 166114
1564 012552 001403
1565 012554 004767 000064
1566 012560 000763
1567 012562 112767 000015 166076
1568 012570 004767 000050
1569 012574 012703 000004
1570 012600 005067 166062
1571 012604 004767 000034
1572 012610 005303
1573 012612 001372
1574 012614 112767 000012 166044
1575 012622 004767 000016
1576 012626 105767 166172
1577 012632 100401
1578 012634 000735
1579 012636 005067 166162
1580 012642 000406
1581 012644 105777 165774
1582 012650 100375
1583 012652 116777 166010 165766
1584 012660 000207
1585
1586

```

;TTY ENTRY ERROR SUBROUTINE*****
TINER: MOV    #MSG7,R4
        JSR    PC,TTOUT      ;PRINT?
        SUB    #20,(SP)      ;RESET SP TO START OF VALUE ROUTINE
        RTS    PC           ;REDO VALUE ENTRY

;TTY READ SUBROUTINE*****
TTIN:  CLR    @TKS
        CLR    @TKB
        TSTB   @TKS
        BPL    .-4
        MOV    @TKB,TIB
        BIC    #177600,TIB
TTIN2: TSTB   @TPS
        BPL    TTIN2
        MOVB   TIB,@TPB
        RTS    PC

;TTY OUTPUT SUBROUTINE*****
TTOUT: MOVB   (R4)+,TOB
        CMPB   #43,TOB
        BEQ    TEX
        CMPB   #45,TOB
        BEQ    TCRLF
        JSR    PC,TOG
        BR     TTOUT
TCRLF: MOVB   #15,TOB
        JSR    PC,TOG
        MOV    #4,R3
TCRLFA: CLR    TOB
        JSR    PC,TOG
        DEC    R3
        BNE    TCRLFA      ;DO FILLERS
        MOVB   #12,TOB
        JSR    PC,TOG
        TSTB   RDSW
        BMI    1$
        BR     TTOUT
1$:     CLR    RDSW
        BR     TEX
TOG:    TSTB   @TPS
        BPL    TOG
        MOVB   TOB,@TPB
        RTS    PC

```

```

;OCTAL OUTPUT SUBROUTINE*****
1587
1588
1589 012662 012767 000001 000222 OCTPE: MOV #1,OFL
1590 012670 010304 MOV R3,R4
1591 012672 000410 BR OCTPO
1592 012674 005067 000212 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
1593 012700 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
1594 012702 001004 BNE OCTPO ;IF NOT ZERO: BR
1595 012704 004767 000162 JSR PC,OCTPG1 ;ELSE PRINT ZERO
1596 012710 000167 000120 JMP OCTP3 ;SPACE AND EXIT
1597 012714 032704 100000 OCTPO: BIT #100000,R4 ;SEE IF MSD = 1
1598 012720 001406 BEQ OCTP1 ;IF NOT: BR
1599 012722 012704 000001 MOV #1,R4
1600 012726 004767 000116 JSR PC,OCTPG ;PRINT 1
1601 012732 000167 000006 JMP OCTP2
1602 012736 005004 OCTP1: CLR R4
1603 012740 004767 000104 JSR PC,OCTPG ;PRINT 0
1604 012744 010304 OCTP2: MOV R3,R4
1605 012746 006004 ROR R4
1606 012750 006004 ROR R4
1607 012752 006004 ROR R4 ;POSITION DIGIT
1608 012754 006004 ROR R4
1609 012756 000304 SWAB R4
1610 012760 004767 000064 JSR PC,OCTPG ;PRINT DIGIT 2
1611 012764 010304 MOV R3,R4
1612 012766 006004 ROR R4
1613 012770 000304 SWAB R4
1614 012772 004767 000052 JSR PC,OCTPG ;PRINT DIGIT 3
1615 012776 010304 MOV R3,R4
1616 013000 006104 ROL R4
1617 013002 006104 ROL R4
1618 013004 000304 SWAB R4
1619 013006 004767 000036 JSR PC,OCTPG ;PRINT DIGIT 4
1620 013012 010304 MOV R3,R4
1621 013014 006004 ROR R4
1622 013016 006004 ROR R4
1623 013020 006004 ROR R4
1624 013022 004767 000022 JSR PC,OCTPG
1625 013026 010304 MOV R3,R4
1626 013030 004767 000014 JSR PC,OCTPG ;PRINT DIGIT 5
1627 013034 012767 000240 165624 OCTP3: MOV #240,T08
1628 013042 004767 177576 JSR PC,T08 ;PRINT SPACE
1629 013046 000207 RTS PC ;EXIT
1630 013050 042704 177770 OCTPG: BIC #177770,R4
1631 013054 001004 BNE OCTPG0
1632 013056 005767 000030 TST OFL
1633 013062 001001 BNE OCTPG0
1634 013064 000207 RTS PC
1635 013066 005267 000020 OCTPG0: INC OFL
1636 013072 052704 000260 OCTPG1: BIS #260,R4
1637 013076 010467 165564 MOV R4,T08
1638 013102 004767 177536 JSR PC,T08
1639 013106 010304 MOV R3,R4
1640 013110 000207 RTS PC
1641 013112 000000 OFL: 0 ;FIRST CHAR FLAG
1642

```

```

1643                                     ;DATA CHARACTER OUTPUT SUBROUTINE*****
1644
1645 013114 005067 165546 DOUT: CLR TOB
1646 013120 012704 000010 MOV #10,R4 ;SET NUMBER TO PRINT
1647 013124 110367 165536 MOVB R3,TOB
1648 013130 105777 165510 DOUT1: TSTB @TPS
1649 013134 100375 BPL DOUT1
1650 013136 132767 000200 165522 BITB #200,TOB
1651 013144 001404 BEQ DOUT2
1652 013146 012777 000061 165472 MOV #061,@TPB
1653 013154 000403 BR DOUT3
1654 013156 012777 000060 165462 DOUT2: MOV #060,@TPB
1655 013164 006167 165476 DOUT3: ROL TOB
1656 013170 005304 DEC R4
1657 013172 001356 BNE DOUT1
1658 013174 000207 RTS PC
1659 013176 016703 165542 DOUTD: MOV TEMP3,R3
1660 013202 000303 SWAB R3
1661 013204 004767 177704 JSR PC,DOUT
1662 013210 016703 165530 MOV TEMP3,R3
1663 013214 004767 177674 JSR PC,DOUT
1664 013220 000207 RTS PC

```

```

1665                                     ;TU16/TE16 SERIAL NUMBER PRINT SUBROUTINE*****
1666
1667
1668 013222 010304 SNPT: MOV R3,R4
1669 013224 000304 SWAB R4
1670 013226 006004 ROR R4
1671 013230 006004 ROR R4
1672 013232 006004 ROR R4
1673 013234 006004 ROR R4 ;GET FIRST DIGIT
1674 013236 004767 000036 JSR PC,SNPG ;GO PRINT
1675 013242 010304 MOV R3,R4
1676 013244 000304 SWAB R4 ;GET SECOND DIGIT
1677 013246 004767 000026 JSR PC,SNPG ;GO PRINT
1678 013252 010304 MOV R3,R4
1679 013254 006004 ROR R4
1680 013256 006004 ROR R4
1681 013260 006004 ROR R4
1682 013262 006004 ROR R4 ;GET THIRD DIGIT
1683 013264 004767 000010 JSR PC,SNPG ;GO PRINT
1684 013270 010304 MOV R3,R4 ;GET FOURTH DIGIT
1685 013272 004767 000002 JSR PC,SNPG ;GO PRINT
1686 013276 000207 RTS PC ;EXIT
1687 013300 012767 000260 165360 SNPG: MOV #260,TOB ;SET BASE = 0
1688 013306 042704 177760 BIC #177760,R4 ;MASK DIGIT
1689 013312 050467 165350 BIS R4,TOB ;SET ASCII
1690 013316 004767 177322 JSR PC,TOG ;TYPE DIGIT
1691 013322 000207 RTS PC ;RETURN

```

```

1692
1693                                     ;CKSWR ROUTINE THAT ALLOWS THE LOADING OF LOC.176, SWREG*****
1694                                     ;FROM THE TTY AT SELECTED POINTS WITHIN THE PROGRAM*****
1695
1696 013324 022767 000176 165304 CKSWR: CMP #SWREG,SWR ;SOFTWARE SWITCH REG PRESENT
1697 013332 001041 BNE OUT ;NO, GET OUT
1698 013334 105777 165300 TSTB @TKS ;YES, WAIT FOR

```



```

1750                                     ;MESSAGE TABLE*****
1751
1752 013640 041445 030523 020040 MSG1: .ASCII /%CSI WC BA FC CS2 /
1753 013646 020040 041527 020040
1754 013654 020040 041040 020101
1755 013662 020040 020040 041506
1756 013670 020040 020040 041440
1757 013676 031123 020040 020040
1758 013704 051504 020040 020040 .ASCII /DS ER TC%#/
1759 013712 042440 020122 020040
1760 013720 020040 041524 021445
1761 013726 051045 053505 047111 MSG2: .ASCII /%REWIND ERROR#/
1762 013734 020104 051105 047522
1763 013742 021522
1764 013744 022445 046524 031060 MSG3: .ASCII /%TMO2-TU16-TE16 BASIC FUNCTION TEST (DZTUB-F)%/
1765 013752 052055 030525 026466
1766 013760 042524 033061 041040
1767 013766 051501 041511 043040
1768 013774 047125 052103 047511
1769 014002 020116 042524 052123
1770 014010 024040 055104 052524
1771 014016 026502 024506 045
1772 014023 105 052116 051105 .ASCII /ENTER CONDITIONS IN OCTAL%#/
1773 014030 041440 047117 044504
1774 014036 044524 047117 020123
1775 014044 047111 047440 052103
1776 014052 046101 021445
1777 014056 051045 043505 051511 MSG4: .ASCII /%REGISTER START = #/
1778 014064 042524 020122 052123
1779 014072 051101 020124 020075
1780 014100 043
1781 014101 045 042526 052103 MSG5: .ASCII /%VECTOR = #/
1782 014106 051117 036440 021440
1783 014114 042445 042116 047440 MSG6: .ASCII /%END OF PASS #/
1784 014122 020106 040520 051523
1785 014130 021440
1786 014132 037440 021440 MSG7: .ASCII / ? #/
1787 014136 021445 MSG8: .ASCII /%#/
1788 014140 050045 051517 052111 MSG9: .ASCII /%POSITION ERROR: #/
1789 014146 047511 020116 051105
1790 014154 047522 035122 021440
1791 014162 042045 044522 042526 MSG10: .ASCII /%DRIVE NUMBER: #/
1792 014170 047040 046525 042502
1793 014176 035122 021440
1794 014202 051445 040514 042526 MSG11: .ASCII /%SLAVE NUMBER: #/
1795 014210 047040 046525 042502
1796 014216 035122 021440
1797 014222 053445 044522 042524 MSG12: .ASCII /%WRITE ERROR #/
1798 014230 042440 051122 051117
1799 014236 021440
1800 014240 051045 040505 020104 MSG13: .ASCII /%READ REVERSE ERROR #/
1801 014246 042522 042526 051522
1802 014254 020105 051105 047522
1803 014262 020122 043
1804 014265 045 042522 042101 MSG14: .ASCII /%READ FORWARD ERROR #/
1805 014272 043040 051117 040527

```

1806	014300	042122	042440	051122				
1807	014306	051117	021440					
1808	014312	053445	044522	042524	MSG15:	.ASCII	/%WRITE TM ERROR #/	
1809	014320	052040	020115	051105				
1810	014326	047522	020122	043				
1811	014333	045	042522	042526	MSG16:	.ASCII	/%REVERSE ERROR #/	
1812	014340	051522	020105	051105				
1813	014346	047522	020122	043				
1814	014353	045	047506	053522	MSG17:	.ASCII	/%FORWARD ERROR #/	
1815	014360	051101	020104	051105				
1816	014366	047522	020122	043				
1817	014373	040	051116	020132	MSG20:	.ASCII	/ NRZ #/	
1818	014400	043						
1819	014401	040	042520	021440	MSG21:	.ASCII	/ PE #/	
1820	014406	042440	050130	035124	MSG22:	.ASCII	/ EXPT: #/	
1821	014414	021440						
1822	014416	051040	053103	035104	MSG23:	.ASCII	/ RCVD: #/	
1823	014424	021440						
1824	014426	041045	051525	052040	MSG24:	.ASCII	/%BUS TRAP: #/	
1825	014434	040522	035120	021440				
1826	014442	053445	035103	021440	MSG25:	.ASCII	/%WC: #/	
1827	014450	041045	035101	021440	MSG26:	.ASCII	/%BA: #/	
1828	014456	042045	035102	021440	MSG27:	.ASCII	/%DB: #/	
1829	014464	044445	044516	020124	MSG28:	.ASCII	/%INIT DID NOT CLEAR RH #/	
1830	014472	044504	020104	047516				
1831	014500	020124	046103	040505				
1832	014506	020122	044122	021440				
1833	014514	051445	020103	047516	MSG29:	.ASCII	/%SC NOT RESET BY INIT #/	
1834	014522	020124	042522	042523				
1835	014530	020124	054502	044440				
1836	014536	044516	020124	043				
1837	014543	045	051124	020105	MSG30:	.ASCII	/%TRE NOT RESET BY INIT #/	
1838	014550	047516	020124	042522				
1839	014556	042523	020124	054502				
1840	014564	044440	044516	020124				
1841	014572	043						
1842	014573	045	051503	020062	MSG31:	.ASCII	/%CS2 NOT RESET BY INIT #/	
1843	014600	047516	020124	042522				
1844	014606	042523	020124	054502				
1845	014614	044440	044516	020124				
1846	014622	043						
1847	014623	045	046104	020124	MSG32:	.ASCII	/%DLT NOT SET #/	
1848	014630	047516	020124	042523				
1849	014636	020124	043					
1850	014641	045	041523	047040	MSG33:	.ASCII	/%SC NOT SET #/	
1851	014646	052117	051440	052105				
1852	014654	021440						
1853	014656	052045	042522	047040	MSG34:	.ASCII	/%TRE NOT SET #/	
1854	014664	052117	051440	052105				
1855	014672	021440						
1856	014674	044445	020122	047516	MSG35:	.ASCII	/%IR NOT SET BY INIT #/	
1857	014702	020124	042523	020124				
1858	014710	054502	044440	044516				
1859	014716	020124	043					
1860	014721	045	051117	047040	MSG36:	.ASCII	/%OR NOT RESET BY INIT #/	
1861	014726	052117	051040	051505				

1862	014734	052105	041040	020131	
1863	014742	047111	052111	021440	
1864	014750	047445	020122	047516	MSG37: .ASCII /%OR NOT RESET BY 1 SILO ENTRY #/
1865	014756	020124	042522	042523	
1866	014764	020124	054502	030440	
1867	014772	051440	046111	020117	
1868	015000	047105	051124	020131	
1869	015006	043			
1870	015007	045	051117	047040	MSG38: .ASCII /%OR NOT SET BY SILO FULL #/
1871	015014	052117	051440	052105	
1872	015022	041040	020131	044523	
1873	015030	047514	043040	046125	
1874	015036	020114	043		
1875	015041	045	040502	020104	MSG39: .ASCII /%BAD SILO READ #/
1876	015046	044523	047514	051040	
1877	015054	040505	020104	043	
1878	015061	045	051111	047040	MSG40: .ASCII /%IR NOT RESET BY SILO FULL#/
1879	015066	052117	051040	051505	
1880	015074	052105	041040	020131	
1881	015102	044523	047514	043040	
1882	015110	046125	021514		
1883	015114	047040	047117	042455	MSG41: .ASCII / NON-EXIST DRIVE#/
1884	015122	044530	052123	042040	
1885	015130	044522	042526	043	
1886	015135	040	047516	026516	MSG42: .ASCII / NON-EXIST SLAVE#/
1887	015142	054105	051511	020124	
1888	015150	046123	053101	021505	
1889	015156	051440	051105	040511	MSG43: .ASCII / SERIAL NO: #/
1890	015164	020114	047516	020072	
1891	015172	043			
1892	015173	045	051105	051501	MSG44: .ASCII /%ERASE HEAD INOPERATIVE/
1893	015200	020105	042510	042101	
1894	015206	044440	047516	042520	
1895	015214	040522	044524	042526	
1896	015222	041445	042510	045503	.ASCII /%CHECK POLARITY#/
1897	015230	050040	046117	051101	
1898	015236	052111	021531		
1899	015242	042445	040522	042523	MSG45: .ASCII /%ERASE HEAD POLARITY WRONG#/
1900	015250	044040	040505	020104	
1901	015256	047520	040514	044522	
1902	015264	054524	053440	047522	
1903	015272	043516	043		
1904	015275	045	042523	026524	MSG46: .ASCII /%SET-UP WRITE ERROR#/
1905	015302	050125	053440	044522	
1906	015310	042524	042440	051122	
1907	015316	051117	043		
1908	015321	045	050123	041501	MSG47: .ASCII /%SPACE FORWARD ERROR#/
1909	015326	020105	047506	053522	
1910	015334	051101	020104	051105	
1911	015342	047522	021522		
1912	015346	051445	040520	042503	MSG48: .ASCII /%SPACE REVERSE ERROR#/
1913	015354	051040	053105	051105	
1914	015362	042523	042440	051122	
1915	015370	051117	043		
1916	015373	045	052502	043106	MSG49: .ASCII /%BUFFERED WRITE ERROR#/
1917	015400	051105	042105	053440	

1918	015406	044522	042524	042440	
1919	015414	051122	051117	043	
1920	015421	045	047502	020124	MSG50: .ASCII /*BOT SET AFTER BUFFERED WRITE*/
1921	015426	042523	020124	043101	
1922	015434	042524	020122	052502	
1923	015442	043106	051105	042105	
1924	015450	053440	044522	042524	
1925	015456	043			
1926	015457	045	047516	041040	MSG51: .ASCII /*NO BOT FROM READ IN PRESET*/
1927	015464	052117	043040	047522	
1928	015472	020115	042522	042101	
1929	015500	044440	020116	051120	
1930	015506	051505	052105	043	
1931	015513	045	041524	044440	MSG52: .ASCII /*TC INCORRECT */
1932	015520	041516	051117	042522	
1933	015526	052103	021440		
1934	015532	051445	040514	042526	MSG53: .ASCII /*SLAVE NOT OFF LINE*/
1935	015540	047040	052117	047440	
1936	015546	043106	046040	047111	
1937	015554	021505			
1938	015556	022445	042522	042523	MSG54: .ASCII /*%RESET SLAVE TO ON LINE BEFORE CONTINUING*/
1939	015564	020124	046123	053101	
1940	015572	020105	047524	047440	
1941	015600	020116	044514	042516	
1942	015606	041040	043105	051117	
1943	015614	020105	047503	052116	
1944	015622	047111	044525	043516	
1945	015630	043			
1946	015631	045	051116	020132	MSG55: .ASCII /*NRZ ONLY: */
1947	015636	047117	054514	020072	
1948	015644	043			
1949	015645	040	052111	051105	MSG56: .ASCII / ITER: */
1950	015652	020072	043		
1951	015655	045	046524	047040	MSG57: .ASCII /*TM NOT SET*/
1952	015662	052117	051440	052105	
1953	015670	043			
1954	015671	045	044505	044124	MSG60: .ASCII /*EITHER TAPE NOT ERASED OR OPI PROBLEM*/
1955	015676	051105	052040	050101	
1956	015704	020105	047516	020124	
1957	015712	051105	051501	042105	
1958	015720	047440	020122	050117	
1959	015726	020111	051120	041117	
1960	015734	042514	021515		
1961	015740	051045	030510	020061	MSG61: .ASCII /*RH11 OR RH70: */
1962	015746	051117	051040	033510	
1963	015754	035060	021440		
1964	015760	051045	020110	047117	MSG62: .ASCII /*RH ONLY: */
1965	015766	054514	020072	043	
1966					

```

;TEST HEADERS*****
1967
1968
1969 015773      045 043045 030524 MSFT1: .ASCII /%%FT1:RH ADDRESSING #/
1970 016000 051072 020110 042101
1971 016006 051104 051505 044523
1972 016014 043516 021440
1973 016020 022445 052106 035062 MSFT2: .ASCII /%%FT2:RH REGISTER BITS TEST #/
1974 016026 044122 051040 043505
1975 016034 051511 042524 020122
1976 016042 044502 051524 052040
1977 016050 051505 020124 043
1978 016055      045 043045 031524 MSFT3: .ASCII /%%FT3:RH INITIALIZE TEST #/
1979 016062 051072 020110 047111
1980 016070 052111 040511 044514
1981 016076 042532 052040 051505
1982 016104 020124 043
1983 016107      045 043045 032124 MSFT4: .ASCII /%%FT4:RH11 SILO TEST 1 #/
1984 016114 051072 030510 020061
1985 016122 044523 047514 052040
1986 016130 051505 020124 020061
1987 016136      043
1988 016137      045 043045 032524 MSFT5: .ASCII /%%FT5:RH11 SILO TEST 2 #/
1989 016144 051072 030510 020061
1990 016152 044523 047514 052040
1991 016160 051505 020124 020062
1992 016166      043
1993 016167      045 043045 033124 MSFT6: .ASCII /%%FT6:RH11 SILO TEST 3 #/
1994 016174 051072 030510 020061
1995 016202 044523 047514 052040
1996 016210 051505 020124 020063
1997 016216      043
1998 016217      045 043045 033524 MSFT7: .ASCII /%%FT7:RH11 SILO TEST 4 #/
1999 016224 051072 030510 020061
2000 016232 044523 047514 052040
2001 016240 051505 020124 020064
2002 016246      043
2003 016247      045 043045 030524 MSFT10: .ASCII /%%FT10:RH11 SILO TEST 5 #/
2004 016254 035060 044122 030461
2005 016262 051440 046111 020117
2006 016270 042524 052123 032440
2007 016276 021440
2008 016300 022445 052106 030461 MSFT11: .ASCII /%%FT11:NOP TEST#/
2009 016306 047072 050117 052040
2010 016314 051505 021524
2011 016320 022445 052106 031061 MSFT12: .ASCII /%%FT12:REWIND TEST#/
2012 016326 051072 053505 047111
2013 016334 020104 042524 052123
2014 016342      043
2015 016343      045 043045 030524 MSFT13: .ASCII /%%FT13:WRITE-READ TEST#/
2016 016350 035063 051127 052111
2017 016356 026505 042522 042101
2018 016364 052040 051505 021524
2019 016372 022445 052106 032061 MSFT14: .ASCII /%%FT14:SPACE TEST#/
2020 016400 051472 040520 042503
2021 016406 052040 051505 021524
2022 016414 022445 052106 032461 MSFT15: .ASCII /%%FT15:ERASE TEST#/

```

2023	016422	042472	040522	042523	
2024	016430	052040	051505	021524	
2025	016436	022445	052106	033061	MSFT16: .ASCII /%FT16:TAPE MARK WRITE-READ TEST#/
2026	016444	052072	050101	020105	
2027	016452	040515	045522	053440	
2028	016460	044522	042524	051055	
2029	016466	040505	020104	042524	
2030	016474	052123	043		
2031	016477	045	043045	030524	MSFT17: .ASCII /%FT17:TM SPACE TEST #/
2032	016504	035067	046524	051440	
2033	016512	040520	042503	052040	
2034	016520	051505	020124	043	
2035	016525	045	043045	031124	MSFT20: .ASCII /%FT20:WRITE CHECK TEST #/
2036	016532	035060	051127	052111	
2037	016540	020105	044103	041505	
2038	016546	020113	042524	052123	
2039	016554	021440			
2040	016556	022445	052106	030462	MSFT21: .ASCII /%FT21:ERASE HEAD TEST#/
2041	016564	042472	040522	042523	
2042	016572	044040	040505	020104	
2043	016600	042524	052123	043	
2044	016605	045	043045	031124	MSFT22: .ASCII /%FT22:BUFFERED COMMAND TEST#/
2045	016612	035062	052502	043106	
2046	016620	051105	042105	041440	
2047	016626	046517	040515	042116	
2048	016634	052040	051505	021524	
2049	016642	022445	052106	031462	MSFT23: .ASCII /%FT23:READ IN PRESET TEST#/
2050	016650	051072	040505	020104	
2051	016656	047111	050040	042522	
2052	016664	042523	020124	042524	
2053	016672	052123	043		
2054	016675	045	043045	031124	MSFT24: .ASCII /%FT24:REWIND-OFF LINE TEST#/
2055	016702	035064	042522	044527	
2056	016710	042116	047455	043106	
2057	016716	046040	047111	020105	
2058	016724	042524	052123	043	
2059	016731	045	043536	043	SCNTG: .ASCII /%TG#/
2060	016735	045	053523	036522	SMSWR: .ASCII /%SWR= #/
2061	016742	021440			
2062	016744	020040	042516	036527	SMNEW: .ASCII / NEW= #/
2063	016752	021440			
2064	016754	022477	043		SQUEST: .ASCII /?%#/
2065					
2066					
2067		016760			.EVEN
2068	016760	000000			WDATA: 0
2069		020472			.+.1510
2070	020472	000000			RDATA: 0
2071					
2072		000001			.END

MSFT16	016436	957	2025#			
MSFT17	016477	993	2031#			
MSFT2	016020	463	1973#			
MSFT20	016525	1057	2035#			
MSFT21	016556	1093	2040#			
MSFT22	016605	1142	2044#			
MSFT23	016642	1177	2049#			
MSFT24	016675	1216	2054#			
MSFT3	016055	528	1978#			
MSFT4	016107	570	1983#			
MSFT5	016137	595	1988#			
MSFT6	016167	629	1993#			
MSFT7	016217	684	1998#			
MSG1	013640	1318	1752#			
MSG10	014162	327	1791#			
MSG11	014202	342	1794#			
MSG12	014222	776	965	1019	1103	1797#
MSG13	014240	786	971	1800#		
MSG14	014265	793	976	1119	1804#	
MSG15	014312	1001	1808#			
MSG16	014333	899	1029	1071	1811#	
MSG17	014353	897	1043	1076	1814#	
MSG2	013726	1274	1761#			
MSG20	014373	890	1304	1817#		
MSG21	014401	893	1307	1819#		
MSG22	014406	508	666	901	1820#	
MSG23	014416	512	670	906	1822#	
MSG24	014426	1471	1824#			
MSG25	014442	471	1826#			
MSG26	014450	480	1827#			
MSG27	014456	491	1828#			
MSG28	014464	1829#				
MSG29	014514	536	1833#			
MSG3	013744	290	1764#			
MSG30	014543	540	1837#			
MSG31	014573	546	1842#			
MSG32	014623	581	695	721	1847#	
MSG33	014641	583	1850#			
MSG34	014656	585	1853#			
MSG35	014674	600	1856#			
MSG36	014721	604	1860#			
MSG37	014750	609	617	1864#		
MSG38	015007	643	1870#			
MSG39	015041	664	1875#			
MSG4	014056	292	1777#			
MSG40	015061	639	1878#			
MSG41	015114	339	1883#			
MSG42	015135	354	1886#			
MSG43	015156	357	1889#			
MSG44	015173	1133	1892#			
MSG45	015242	1130	1899#			
MSG46	015275	827	935	1067	1904#	
MSG47	015321	855	1908#			
MSG48	015346	845	1912#			
MSG49	015373	1161	1916#			
MSG5	014101	301	1781#			

MSG50	015421	1166	1920#															
MSG51	015457	1199	1926#															
MSG52	015513	1206	1931#															
MSG53	015532	1225	1934#															
MSG54	015556	1227	1938#															
MSG55	015631	377	1946#															
MSG56	015645	1314	1949#															
MSG57	015655	1351	1951#															
MSG6	014114	430	1783#															
MSG60	015671	948	1954#															
MSG61	015740	361	1961#															
MSG62	015760	368	1964#															
MSG7	014132	1540	1786#															
MSG8	014136	1787#																
MSG9	014140	888	1788#															
MTINT	012130	123	311	1448#														
NRZOF	001012	204#	379	804	915	981	1048	1083										
OCTP	012674	295	304	433	669	673	905	910	1317	1474	1592#							
OCTPE	012662	511	515	1321	1323	1325	1327	1329	1331	1333	1335	1589#	1709					
OCTPE1	012700	1593#																
OCTPG	013050	1600	1603	1610	1614	1619	1624	1626	1630#									
OCTPG0	013066	1631	1633	1635#														
OCTPG1	013072	1595	1636#															
OCTP0	012714	1591	1594	1597#														
OCTP1	012736	1598	1602#															
OCTP2	012744	1601	1604#															
OCTP3	013034	1596	1627#															
OFL	013112	1589*	1592*	1632	1635*	1641#												
OPDYX	000726	178#	734*	765*	820*	927*	956*	998*	1255									
OUT	013436	1697	1699	1703	1714#	1731												
PATRN	001006	202#	768*	1059*	1095*	1360												
PCNTR	001016	206#	384*	432	438*													
PEXFL	000770	195#																
PFLG	000732	180#																
PSW	000634	147#	271*															
RCNT	000712	172#	766*	780*	782*	789*	794*	797*	806*	815*	836*	933*	939*	992*				
RDATA	020472	1008	1010#	1011	1031*	1039*												
RDSW	001024	783	867	930	1114	1372	2070#											
RDYDX	000724	209#	1576	1579*	1726*													
REGS	000660	177#	733*	764*	821*	926*	945*	955*	997*	1158*	1247							
RFD	000722	157#	294	296	313													
RHOF	001014	176#	840*	864*	878													
RHTF	001010	205#	370	375	717													
RH17F	000672	203#	325*	455	457*	1482												
RRD	000720	164#	363	568	593	627	682	701										
RTRN	000734	175#	839*	863*	871													
RWND	011110	181#	1452															
RWINDA	011130	771	814	929	942	959	995	1058	1094	1108	1113	1143	1262#					
RWINDB	011134	1266#	1267															
RWINDX	011200	1268#	1269															
SAV1	000754	1271	1277#															
SAV2	000756	189#																
SAV3	000760	190#																
SCNT	000730	191#																
SCOLP	000762	179#	842*	852*	861*	1030*	1243											
		192#	472*	481*	492*	529*	587*	596*	630*	685*	704*	727*	745*	778*				

TSC00	002526	394#	413												
TSC01	002562	398	400#	429											
TSC02	002606	406#	459	524	549	580	620	653	694	720	741	759	808	921	
		951	987	1052	1089	1137	1170	1209	1229	1276					
TSC03	002622	408	410#												
TSRH	002520	326	393#												
TSTTBL	001040	221#	393	425											
TTIN	012460	1506	1547#	1718											
TTINT	012144	94	1456#												
TTIN2	012512	1553#	1554												
TTOUT	012530	291	293	302	328	340	343	355	358	362	369	378	431	504	
		507	509	513	555	558	662	665	667	671	887	889	894	900	
		902	907	1228	1273	1275	1300	1303	1308	1311	1315	1319	1470	1472	
		1541	1560#	1566	1578	1705	1707	1711	1737						
TTR	012274	300	309	333	348	367	374	383	1504#						
TTR0	012302	1506#	1528												
TTR1	012330	1508	1512#												
TTR1A	012344	1513	1515#												
TTR1B	012360	1516	1518#												
TTR2	012416	1511	1529#												
TTR3	012426	1530	1532#												
TTR4	012436	1533	1535#												
TTR5	012440	1510	1536#												
UDES	001004	201#	729*	770*	799	801*	802	813*	891	717	919*	931*	958*	979	
		983*	994*	1046	1050*	1064*	1081	1085*	1100*	1145*	1179*	1235	1288		
VECT	000656	156#	303	305	310										
WC	000602	131#	466	1236*	1322										
WCNT	000710	171#	731*	749*	773*	818*	835*	943*	961*	1012*	1063*	1099*	1110*	1116*	
		1124*	1148*	1182*	1236										
WDATA	016760	750	774	816*	1014	1061	1097	1146	1180	1359	1367	2068#			
\$CNTG	016731	1704	2059#												
\$MNEW	016744	1710	2062#												
\$MSWR	016735	1706	2060#												
\$QUEST	016754	1736	2064#												
\$READ	013440	1713	1716#												
.	= 020474	85#	90	93#	99#	111#	115#	122#	127#	267#	1550	2067#	2069#		

. ABS. 020474 000

ERRORS DETECTED: 0

DZTUBF.BIN,DZTUBF.LST/CRF/SOL/NL:TOC=DZTUBF.P11
RUN-TIME: 24.8 SECONDS
RUN-TIME RATIO: 63/7=8.5
CORE USED: 9K (17 PAGES)

L06