TM03/TE16

BASIC FUNCTION TESTS

MD-11-DZTEC-A

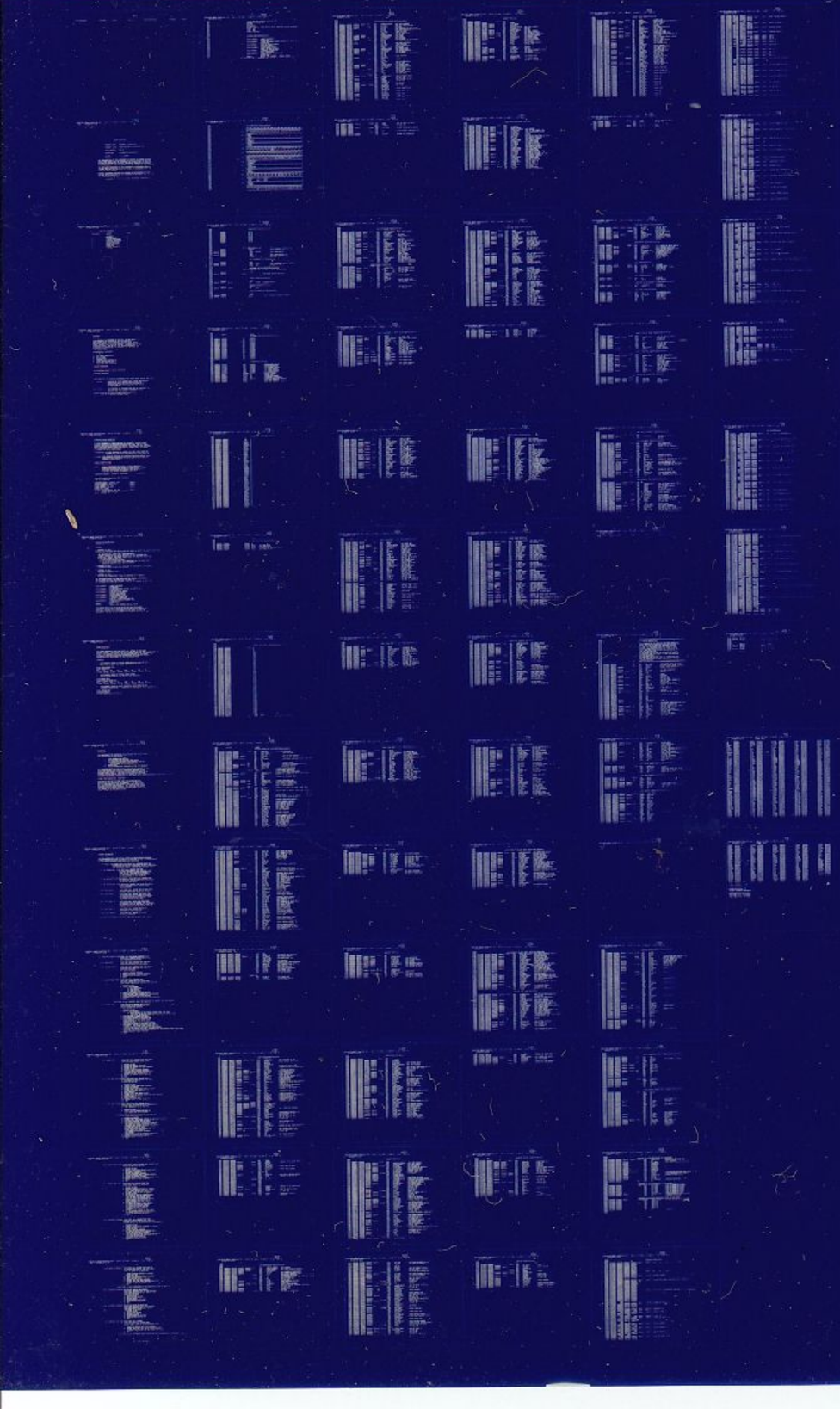
EP-DZTEC-A-DL-A

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



B01

EOF1DZR60ASE0	00010000	770526	PDP10 411	DRHDR1DZTECASE0	00010000	770526

```
.REM  %
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTEC-A-D

PRODUCT TITLE: TM03/TE16 BASIC FUNCTION TEST

DATE CREATED: FEB 77

MAINTAINER: DIAGNOSTIC ENGINEERING

AUTHOR: J. G. ADAMS

TABLE OF CONTENTS

1. ABSTRACT

THIS PROGRAM IS INTENDED TO TEST ALL OF THE BASIC FUNCTIONAL LEVEL OPERATIONS OF THE TM03/TE16 MAG TAPE SYSTEM. ALL FUNCTIONS; WRITE, READ, SPACE, ERASE, REWIND, ETC; WILL BE TESTED. IN ADDITION TO THE TM03/TE16 TESTS, THE RH WILL BE TESTED SEPARATELY IN SO FAR AS IT IS POSSIBLE TO SEPARATE THE RH FROM THE TM03/TE16 ITSELF.

2. REQUIREMENTS (HARDWARE)

A. ANY PDP11 PROCESSOR
B. 8K OF CORE
C. CONSOLE TTY
D. TM03 MAGTAPE CONTROLLER
E. MASS BUS CONTROLLER
F. TE16 MAG TAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD BINARY LOADING PROCEDURE

4. STARTING PROCEDURE

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.

**NOTE SEE ALSO SECTION 5-CONSOLE SWITCH SETTINGS
** TYPE ↑C TO RESTART PROGRAM (@200)

4.1 AUTOMATIC MODE OPERATION

IF THIS PROGRAM IS LOADED AND RUN IN AUTOMATIC (CHAIN) MODES DEFAULT RESPONSES TO OPERATOR REQUESTS ARE USED, AND ALL AVAILABLE TM03/TE16 COMBINATIONS ARE TESTED, ADDITIONALLY THE SOFTWARE SWR IS INVOKED WITH A SWITCH SETTING OF 100000 (HALT ON ERROR) IF LOADED VIA ACT11 CHAIN MODE.

**EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMDP CHAIN MODE THE PROGRAM WILL NOT TEST TM03 DRIVE #0, TE16 SLAVE #0.

** NOTE: THIS PROGRAM CONTAINS AN OPERATOR ASSISTED SUBTEST.THIS SUBTEST IS NOT EXECUTED IN CHAIN MODE.TO RUN LOAD THE PROGRAM IN DUMP MODE.

4.2 SAMPLE START AT 200

NOTE: DEFAULT RESPONSES ARE SHOWN IN ANGLE BRACKETS <>, OPERATOR RESPONSES ARE SHOWN IN PARENTHESES (), AND LOCATIONS CONTAINING THE DEFAULT ARE SHOWN IN []. TO INVOKE THE DEFAULT RESPONSE TYPE (CR).

```
PARAMETER REQUEST: <DEFAULT> (RESPONSE)          [LOCATION:]


TM03-TE16 BASIC FUNCTIONS TEST (DZTEC-A)
TYPE ↑C TO RESTART

REGISTER START: <172440> (CR)                    [REGS:]
VECTOR ADDRESS: <224> (CR)                       [VECT:]
DRIVE NUMBER: <0> (CR)                           [DRVN:]
SLAVE NUMBER: <0> (CR)                           [SLVN:]
SERIAL NO: 12345
RH ONLY (NO=0,YES=1): <0> (0)                    [RHOF:]
        IF THE SOFTWARE SWR IS INVOKED:
SWR = <000000> NEW  = (CR)
```

5. CONSOLE SWITCH SETTING

CONTROL:

1) CONTROL G (↑G):
SELECTS THE SOFTWARE SWR AND ALLOWS THE USER TO SELECT NEW SWITCH SETTINGS.

THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW=
WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWR.
AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
OF THE FOLLOWING AT THE TTY:
A) TYPE A NEW SWITCH SETTING
B) IF A (CR) IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
REGISTER CONTENTS WILL NOT BE CHANGED.

2) CONTROL A (↑A):
ALTERNATES USAGE OF SWR FROM HARDWARE TO SOFTWARE & VICE VERSA,

3) CONTROL C (↑C):
RESTARTS PROGRAM AT 200

4) CONTROL U (↑U):
DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

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.

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=CONTINUOUS CYCLE
	0=HALT AT END OF PASS
SW11(004000):	1=INHIBIT ITERATION
	0=DO ALL ITERATIONS PER TEST
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 SELECTING A TEST.

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 STEPS:

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE
   ****REFER TO SECTION 5 FOR DYNAMIC LOADING
   OF SOFTWARE SWITCH REGISTER.***
3. PRESS START
4. ENTER APPROPRIATE RESPONES 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 . IF SW0-4 IS
SET TO SOME SPECIFIC TEST NUMBER 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.

8. SUBTEST SUMMARIES

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 TM03/TE16 ITSELF. (SEE RH ONLY OPTION; PAR 7)

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

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

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

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

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

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

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

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

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

B. TM03/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 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 TM02 WILL ACCEPT AND EXECUTE ANOTHER COMMAND WHILE ITS SELECTED SLAVE IS REWINDING.

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

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

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

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

FT24: AUTOMATIC DENSITY SELECTION -WRITE NRZ,READPE:
THIS TEST ASSURES THAT AN NRZ WRITTEN
TAPE WHEN READ AS PE WILL SWITCH THE
SLAVE TO NRZ MODE.

1. REWIND SLAVE
2. WRITE AN NRZ RECORD
3. REWIND SLAVE
4. READ RECORD IN PE MODE
5. CHECK DS REG PES BIT=0
6. END

FT25: AUTOMATIC DENSITY SELECTION-WRITE PE,READ NRZ:
THIS TEST ASSURES THAT A PE WRITTEN
TAPE WHEN READ AS NRZ WILL SWITCH
THE SLAVE TO PE MODE.

1. REWIND SLAVE
2. WRITE A PE RECORD
3. REWIND A SLAVE
4. READ RECORD IN NRZ MODE
5. CHECK DS REG PES BIT=1
6. END.

FT26: 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 12 = 1)

%

519 .LIST BIN,LOC,SEQ

```
                                        .TITLE TM03/TE16 BASIC FUNCTION TEST
                                        ;MAINDEC-11-DZTEC-A-D
                                        ;FEB 77
                                        ;R. BARNES
                                        .MCALL  .$ACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE
                                        .NLIST  MC
                                        .LIST   ME
                                        .ENABLE ABS,AMA

                                        ;CONSOLE SWITCHES*********
                                        ;
                                        ;SW15(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=CONTINUOUS CYCLE
                                        ;               0=HALT AT END OF PASS
                                        ;SW11(40000):   1=INHIBIT ITERATIONS
                                        ;               0=DO ITERATIONS
                                        ;SW10(002000):  1=HALT AT END OF EACH TEST
                                        ;               0=CONTINUE
                                        ;SW0-4:         SELECT TEST NUMBER :: 00=ALL TESTS
                                ;USE SOFTWARE SWR IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
```

```
                                        ;TE16 REGISTER BITS**********

                                        ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;CS1:   ;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;FUN;GO ;
                                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;WC:    ; WORD                                                          ;
                                        ; COUNT                                                         ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;BA:    ; BUS                                                           ;
                                        ; ADDRESS                                                       ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;FC:    ; FRAME                                                         ;
                                        ; COUNT                                                         ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;CS2:   ;DLT;WCE;UPE;NED;NEM;PGE;MXF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
                                        ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;   ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;DS:    ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;TMK;BOT;SLA;
                                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;ER:    ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;PEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
                                        ;CRC;   ;   ;   ;   ;ITM;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;ATS:   ; ATTENTION                                                     ;
                                        ; SUMMARY                                                       ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;CC:    ; CHECK                                                         ;
                                        ; CHARACTER                                                     ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;DB:    ; DATA                                                          ;
                                        ; BUFFER                                                        ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;MR:    ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
                                        ; 0 ; 1 ; 2 ; 3 ; 4 ; 5 ; 6 ; 7 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ,DT:    ; DRIVE     ; 7 ;   ;SLV;                                       ;
                                        ; TYPE      ;CH ;   ;PR ;                                       ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;SN:    ; SERIAL                                                        ;
                                        ; NUMBER                                                        ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
                                ;TC:    ;ACL;FCS;SAC;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
                                        ;   ;   ;   ;DT ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
                                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
                                        ;REGISTER EQUIVS**********

            000000                      R0=%0
            000001                      R1=%1
            000002                      R2=%2
            000003                      R3=%3
            000004                      R4=%4
            000005                      R5=%5
            000006                      SP=%6
            000007                      PC=%7


                                ;ACT11 HOOK ***********
            000764                      $SVPC=.                 ;SAVE CURRENT LOCATION CTR
            000046                      .=46
000046      003154                      .WORD   $ENDAD          ;SET LOCATION 46
            000052                      .=52
000052      000000                      .WORD   0               ;SET LOCATION 52 = 0
            000764                      .=$SVPC                 ;RESTORE LOCATION CTR

                                        ;TTY INTERRUPT VECTOR**********

            000060                      .=60
000060      012656                      .WORD   TTINT           ;TTY INTERRUPT HEADER ADDRESS
000062      000340                      .WORD   340             ;PRIORITY LEVEL 7

                                        ;SOFTWARE SWITCH REGISTER**********
                                ;USED IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.

            000176                      .=176
000176      000000              SWREG:  0                       ;SOFTWARE SWITCH REGISTER


                                        ;START ADDRESS***********

            000200                      .=200
000200      000137  001600              JMP     START   ;PROGRAM START

                                        ;RESTART ADDRESS**********
            000210                      .=210
000210      000137  002542              JMP     ST4

                                        ;TM03 INTERRUPT VECTOR**********

            000224                      .=224
000224      012646                      MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
000226      000340                      340
```

```

              000510                    .=510
                                        ;MASS BUS REGISTER EQUIVS**********

     000510   172440          C1:       172440
     000512   172442          WC:       172442
     000514   172444          BA:       172444
     000516   172446          FC:       172446
     000520   172450          CS:       172450
     000522   172452          DS:       172452
     000524   172454          ER:       172454
     000526   172456          AS:       172456
     000530   172460          CC:       172460
     000532   172462          DB:       172462
     000534   172464          MR:       172464
     000536   172466          DT:       172466
     000540   172470          SN:       172470
     000542   172472          TC:       172472
     000544   172474          BAE:      172474

                                        ;CONSTANTS**********

     000546   177776          PSW:      177776          ;PROCESSOR STATUS
     000550   177570          SWR:      177570          ;SWITCH REGISTER
     000552   177560          TKS:      177560          ;TTY READER STATUS
     000554   177562          TKB:      177562          ;TTY READ BUFFER
     000556   177564          TPS:      177564          ;TTY PUNCH STATUS
     000560   177566          TPB:      177566          ;TTY PUNCH BUFFER
     000562   177777          SERNUM:   177777          ;SERIAL NUMBER
     000564   000011          DRVTP:    011             ;DRIVE TYPE
     000566   000010          ITAMT:    10              ;ITERATION AMOUNT
     000570   000224          VECT:     224             ;INTERRUPT VECTOR(RH)
     000572   172440          REGS:     172440          ;STARTING REGISTER ADDRESS
     000574   000004          BTRP:     4               ;BUS TRAP ADDRESS
     000576   000006          BTRP2:    6               ;BUS TRAP PRIORITY LEVEL 7
```

```
                                        ;FLAGS AND COUNTERS**********

000600  000000          TOB:    0
000602  000000          TIB:    0
000604  000000          RH17F:  0
000606  000000          HDRFL:  0
000610  000000          EMADDR: 0
000612  000000          DRVN:   0
000614  000000          SLVN:   0
000616  000000          BADDR:  0
000620  000000          FCNT:   0
000622  000000          WCNT:   0
000624  000000          RCNT:   0
000626  000000          ERRP:   0
000630  000000          ERRP1:  0
000632  000000          RRD:    0
000634  000000          RFD:    0
000636  000000          RDYDX:  0
000640  000000          OPDYX:  0
000642  000000          SCNT:   0
000644  000000          PFLG:   0
000646  000000          RTRN:   0
000650  000000          ERRDO:  0
000652  000000          TEMP1:  0
000654  000000          TEMP2:  0
000656  000000          TEMP3:  0
000660  000000          STMSK:  0
000662  000000          ITCNT:  0
000664  000000          DSAV:   0
000666  000000          SAV1:   0
000670  000000          SAV2:   0
000672  000000          SAV3:   0
000674  000000          SCOLP:  0
000676  000000          ITRLP:  0
000700  000000          EXFL:   0
000702  000000          PEXFL:  0
000704  000000          STFLG:  0
000706  000000          LTADD:  0
000710  000000          FUN:    0
000712  000000          SERFL:  0
000714  000000          CRCNT:  0
000716  000000          UDES:   0
000720  000000          PATRN:  0
000722  000000          RHTF:   0
000724  000000          NRZOF:  0
000726  000000          RHOF:   0
000730  000000          PCNTR:  0
000732  000000          TEMPST: 0
000734  000000          COUNT:  0
000736  000000          RDSW:   0
```

```
 727
 728                                        ;DATA PATTERN GENERATORS*****
 729
 730  000740  000000          DATBL:  0
 731  000742  012440          DATA0:  DAT1    ;ALL ONE BITS
 732  000744  012460          DATA1:  DAT2    ;ALL ZERO BITS
 733  000746  012464          DATA2:  DAT3    ;ALTERNATING ONE/ZERO BITS
 734  000750  012472          DATA3:  DAT4    ;ALL BITS 0-377
```

```
735
736                                             ;LOGIC TEST ENTRY TABLE**********
737
738  000752  000000                    TSTTBL: 0
739  000754  000000                            0
740  000756  003216                            FT1
741  000760  003216                            FT1
742  000762  003316                            FT2
743  000764  003316                            FT2
744  000766  003640                            FT3
745  000770  003640                            FT3
746  000772  004060                            FT4
747  000774  004060                            FT4
748  000776  004206                            FT5
749  001000  004206                            FT5
750  001002  004400                            FT6
751  001004  004400                            FT6
752  001006  004652                            FT7
753  001010  004652                            FT7
754  001012  004746                            FT10
755  001014  004746                            FT10
756  001016  005102                            FT11
757  001020  005102                            FT11
758  001022  005220                            FT12
759  001024  005220                            FT12
760  001026  005332                            FT13
761  001030  005332                            FT13
762  001032  005644                            FT14
763  001034  005644                            FT14
764  001036  006516                            FT15
765  001040  006516                            FT15
766  001042  006716                            FT16
767  001044  006716                            FT16
768  001046  007144                            FT17
769  001050  007144                            FT17
770  001052  007546                            FT20
771  001054  007546                            FT20
772  001056  007772                            FT21
773  001060  007772                            FT21
774  001062  010324                            FT22
775  001064  010324                            FT22
776  001066  010530                            FT23
777  001070  010530                            FT23
778  001072  010750                            FT24
779  001074  010750                            FT24
780  001076  011142                            FT25
781  001100  011142                            FT25
782  001102  011334                            FT26
783  001104  011334                            FT26
784  001106  003110                    .WORD   TEND
785  001110  000026            TLAST:  .WORD   26          ;CONTAINS # OF TESTS
```

```
                001600                          .=1600
                                                ;PROGRAM START AND HOUSEKEEPING**********

001600 012706 000500            START:  MOV     #500,SP         ;SET STACK POINTER
001604 013746 000006                    MOV     @#6,-(SP)               ;SAVE VECTORS
001610 013746 000004                    MOV     @#4,-(SP)
001614 012737 001640 000004             MOV     #1$,@#4                 ;SET UP FOR TIMEOUT
001622 005037 000006                    CLR     @#6
001626 022777 177777 176714             CMP     #-1,@SWR                ;REFERENCE HARDWARE SWITCH REGISTER
001634 001402                           BEQ     2$
001636 000404                           BR      3$
001640 022626                   1$:     CMP     (SP)+,(SP)+             ;ADJUST STACK
001642 012737 000176 000550     2$:     MOV     #SWREG,SWR              ;POINT TO SOFTWARE SWITCH REG
001650 012637 000004            3$:     MOV     (SP)+,@#4               ;RESTORE VECTORS
001654 012637 000006                    MOV     (SP)+,@#6
001660 005027                           CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
001662 000000                   CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
                                                                ;;1/0 = CHAIN/NOT CHAIN MODE
001664 022737 003154 000042             CMP     #SENDAD,@#42    ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
001672 001404                           BEQ     50$
001674 005737 000042                    TST     @#42            ;;BRANCH IF IN DUMP MODE
001700 001413                           BEQ     52$
001702 000406                           BR      51$
001704 012737 000176 000550     50$:    MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
001712 012777 100000 176630             MOV     #100000,@SWR    ;;WITH HALT ON ERROR SET
001720 005237 001662            51$:    INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
001724 000137 002556                    JMP     TSCD            ;;GO TO CHAIN ADDRESS
001730                          52$:
001730 122737 000006 000041     4$:     CMPB    #6,@#41         ;BRANCH IF LOADED VIA TMDP (DUMP MODE)
001736 001004                           BNE     5$
001740 012704 016675                    MOV     #MSG69,R4       ;ADVISE USER TO REMOVE TMDP FROM UUT
001744 004737 013442                    JSR     PC,TTOUT
001750 012704 014470            5$:     MOV     #MSG3,R4
001754 004737 013442                    JSR     PC,TTOUT        ;PRINT TITLE
001760 112737 000043 014470             MOVB    #'#,MSG3        ;DO NOT PRINT TITLE ON RESTART
001766 012704 014623            STOB:   MOV     #MSG4,R4
001772 004737 013442                    JSR     PC,TTOUT        ;REQUEST REGISTER ADDRESS
001776 013703 000572                    MOV     REGS,R3
002002 004737 013572                    JSR     PC,OCTP         ;PRINT CURRENT ADDRESS
002006 012705 000572                    MOV     #REGS,R5        ;SET ADDRESS SAVE LOC
002012 012701 000007                    MOV     #7,R1           ;SET SIZE OF RESPONSE
002016 012702 176400                    MOV     #176400,R2      ;SET UPPER LIMIT
002022 012703 172300                    MOV     #172300,R3      ;SET LOWER LIMIT
002026 004737 013120                    JSR     PC,TTR          ;GO GET RESPONSE
002032 012704 014646                    MOV     #MSG5,R4
002036 004737 013442                    JSR     PC,TTOUT        ;REQUEST VECTOR
002042 013703 000570                    MOV     VECT,R3
002046 004737 013572                    JSR     PC,OCTP         ;PRINT CURRENT VECTOR
002052 012705 000570                    MOV     #VECT,R5        ;SET ADDRESS SAVE LOC
002056 012701 000004                    MOV     #4,R1           ;SET SIZE OF RESPONSE
002062 012702 000224                    MOV     #224,R2         ;SET UPPER LIMIT
002066 012703 000150                    MOV     #150,R3         ;SET LOWER LIMIT
002072 004737 013120                    JSR     PC,TTR          ;GO GET RESPONSE
002076 013700 000570                    MOV     VECT,R0         ;GET VECTOR
002102 012720 012646                    MOV     #MTINT,(R0)+    ;LOAD INTERRUPT ADDRESS IN VECTOR
002106 012710 000340                    MOV     #340,(R0)       ;LOAD PRIORITY
```

```
842  002112  013700  000572          MOV     REGS,R0         ;GET START OF REGS
843  002116  012701  000017          MOV     #17,R1          ;SET NUMBER OF REGS
844  002122  012702  000510          MOV     #C1,R2          ;GET START OF TABLE
845  002126  010022          STO:    MOV     R0,(R2)+        ;BUILD TABLE
846  002130  062700  000002          ADD     #2,R0           ;BUMP ADDRESS
847  002134  005301                  DEC     R1              ;SEE IF DONE
848  002136  001373                  BNE     STO             ;IF NOT: BR
849  002140  012702  000600          MOV     #T0B,R2
850  002144  012700  000054          MOV     #54,R0
851  002150  005022          ST1:    CLR     (R2)+           ;CLEAR FLAGS + COUNTERS
852  002152  005300                  DEC     R0
853  002154  001375                  BNE     ST1
854  002156  012737  000001  000722  MOV     #1,RHTF         ;SET ADDRESS TEST FLAG
855  002164  000137  002750          JMP     TSAH            ;GO DO INITIAL ADDRESS TEST PASS
856  002170  012704  014725  ST1A:   MOV     #MSG10,R4
857  002174  004737  013442          JSR     PC,TTOUT        ;REQUEST DRIVE NUMBER
858  002200  013703  000612          MOV     DRVN,R3         ;GET CURRENT DRIVE #
859  002204  004737  013572          JSR     PC,OCTP         ;AND TYPE IT
860  002210  012705  000612          MOV     #DRVN,R5        ;SET ADDRESS OF DRIVE NUMBER SAVE
861  002214  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
862  002220  012702  000007          MOV     #7,R2           ;SET UPPER LIMIT
863  002224  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
864  002230  004737  013120          JSR     PC,TTR          ;GO GET RESPONSE
865  002234  012777  000040  176256  MOV     #40,@CS         ;SET INIT
866  002242  053777  000612  176250  BIS     DRVN,@CS        ;SET DRIVE NUMBER
867  002250  005777  176234          TST     @C1             ;ACCESS DRIVE
868  002254  032777  010000  176236  BIT     #10000,@CS      ;SEE IF NED
869  002262  001405                  BEQ     ST2             ;IF NOT: BR
870  002264  012704  015657          MOV     #MSG41,R4
871  002270  004737  013442          JSR     PC,TTOUT        ;PRINT NOT AVAIL
872  002274  000735                  BR      ST1A            ;REDO DRIVE REQUEST
873  002276  012704  014745  ST2:    MOV     #MSG11,R4
874  002302  004737  013442          JSR     PC,TTOUT        ;REQUEST SLAVE NUMBER
875  002306  013703  000614          MOV     SLVN,R3         ;GET CURRENT SLAVE #
876  002312  004737  013572          JSR     PC,OCTP         ;AND TYPE IT
877  002316  012705  000614          MOV     #SLVN,R5        ;SET ADDRESS OF SLAVE SAVE
878  002322  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
879  002326  012702  000007          MOV     #7,R2           ;SET UPPER LIMIT
880  002332  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
881  002336  004737  013120          JSR     PC,TTR          ;GO GET RESPONSE
882  002342  012777  000040  176150  MOV     #40,@CS         ;INIT
883  002350  053777  000612  176142  BIS     DRVN,@CS        ;SET DRIVE NUMBER
884  002356  013777  000614  176156  MOV     SLVN,@TC        ;LOAD SLAVE NUMBER
885  002364  032777  002000  176144  BIT     #2000,@DT       ;SEE IF SLAVE PRESENT
886  002372  001005                  BNE     ST3             ;IF SO: BR
887  002374  012704  015700          MOV     #MSG42,R4
888  002400  004737  013442          JSR     PC,TTOUT        ;PRINT NON-EXIST SLAVE
889  002404  000734                  BR      ST2             ;REDO SLAVE REQUEST
890  002406  012704  015721  ST3:    MOV     #MSG43,R4
891  002412  004737  013442          JSR     PC,TTOUT        ;PRINT SERIAL NUMBER TAG
892  002416  017703  176116          MOV     @SN,R3
893  002422  004737  014120          JSR     PC,SNPT         ;PRINT SERIAL NUMBER
894  002426  005037  000604          CLR     RH17F           ;SET RH INDICATOR = RH11
895  002432  013746  000004          MOV     @#4,-(SP)       ;SAVE ERROR TRAP VECTORS
896  002436  013746  000006          MOV     @#6,-(SP)       ;AND PRIORITY
897  002442  012737  002466  000004  MOV     #1$,@#4         ;SET TIME OUT TRAP TO 1$ BELOW
```

```
 898  002450  005037  000006                    CLR     @#6
 899  002454  005777  176064                    TST     @BAE            ;REFERENCE BAE REGISTER
 900  002460  012737  000001  000604            MOV     #1,RH17F        ;SET FLAG = RH70
 901  002466  012637  000006          1$:       MOV     (SP)+,@#6       ;RESTORE ERROR TRAP
 902  002472  012637  000004                    MOV     (SP)+,@#4
 903  002476  012704  016552                    MOV     #MSG62,R4       ;GET REQUEST
 904  002502  004737  013442                    JSR     PC,TTOUT        ;REQUEST RH11 ONLY RESPONSE
 905  002506  013703  000726                    MOV     RH0F,R3         ;GET CURRENT FLAG SETTING
 906  002512  004737  013572                    JSR     PC,OCTP         ;AND TYPE IT
 907  002516  012705  000726                    MOV     #RH0F,R5        ;SET FLAG ADDRESS
 908  002522  012701  000002                    MOV     #2,R1           ;SET SIZE OF RESPONSE
 909  002526  012702  000001                    MOV     #1,R2           ;SET UPPER LIMIT
 910  002532  012703  000000                    MOV     #0,R3           ;SET LOWER LIMIT
 911  002536  004737  013120                    JSR     PC,TTR          ;GO GET RESPONSE
 912
 913                              ;START 210
 914  002542  012706  000500          ST4:      MOV     #500,SP         ;SET STACK PTR
 915  002546  005037  000730                    CLR     PCNTR           ;CLEAR PASS COUNTER
 916  002552  004737  014222                    JSR     PC,GTSWR        ;GET SWITCHES
```

```
                                                ;TEST SCHEDULAR**********

002556  052777  000100  175766  TSCD:   BIS     #100,@TKS       ;SET KEYBOARD IE BIT
002564  005037  000704                  CLR     STFLG           ;CLEAR SINGLE TEST FLAG
002570  017700  175754                  MOV     @SWR,R0
002574  042700  177740                  BIC     #177740,R0
002600  001125                          BNE     STSCD           ;GO SELECT SINGLE TEST
002602  005737  001662                  TST     CHNFLG          ;;BRANCH IF NOT IN CHAIN MODE
002606  001457                          BEQ     TSCDA
002610  012737  177777  000612          MOV     #-1,DRVN        ;;INITIALIZE DRIVE #
002616  012737  177777  000614  NXTDRV: MOV     #-1,SLVN        ;;INITIALIZE SLAVE #
002624  012777  000040  175666  1$:     MOV     #40,@CS         ;;INIT CONTROLLER
002632  005237  000612                  INC     DRVN            ;;STEP DRIVE #
002636  022737  000010  000612          CMP     #10,DRVN        ;;EXIT IF ALL DRIVES TESTED
002644  001524                          BEQ     SDONE           ;;FOR AVAILABILITY
002646  013777  000612  175644          MOV     DRVN,@CS        ;;LOAD DRIVE #
002654  005777  175630                  TST     @CS             ;;ACCESS DRIVE
002660  032777  010000  175632          BIT     #10000,@CS      ;;BRANCH IF DRIVE NON EXISTANT
002666  001356                          BNE     1$              ;;(NED = 1)
002670  005237  000614          NXTSLV: INC     SLVN            ;;STEP SLAVE # AND BRANCH
002674  001011                          BNE     1$              ;;IF NOT SLAVE 0
002676  005737  000612                  TST     DRVN            ;;BRANCH IF NOT DRIVE # 0
002702  001006                          BNE     1$
002704  122737  000006  000041          CMPB    #6,@#41         ;;BRANCH IF NOT TMDP
002712  001002                          BNE     1$
002714  005237  000614                  INC     SLVN            ;;STEP TO SLAVE # 1
002720  022737  000010  000614  1$:     CMP     #10,SLVN        ;;BRANCH IF ALL SLAVES TESTED
002726  001733                          BEQ     NXTDRV          ;;FOR AVAILABILITY
002730  013777  000614  175604          MOV     SLVN,@TC        ;;LOAD SLAVE UNIT #
002736  032777  002000  175572          BIT     #2000,@DT       ;;BRANCH IF SLAVE NOT
002744  001751                          BEQ     NXTSLV          ;;PRESENT (SPR = 0)
002746  000240          TSCDA:  NOP
002750  012737  000752  000706  TSRH:   MOV     #TSTTBL,LTADD
002756  062737  000004  000706  TSCD0:  ADD     #4,LTADD
002764  013737  000706  000676  TSCD1:  MOV     LTADD,ITRLP
002772  062737  000002  000676          ADD     #2,ITRLP        ;SET ITERATION ADDRESS
003000  005037  000660                  CLR     STMSK
003004  005037  000626                  CLR     ERRP
003010  005037  000606                  CLR     HDRFL           ;CLEAR PRINT HEADER FLAG
003014  017700  175666                  MOV     @LTADD,R0       ;SET POINTER TO TEST
003020  000110                          JMP     (R0)            ;GO TO TEST
003022  032777  002000  175520  TSCD2:  BIT     #2000,@SWR      ;SEE IF HALT ON TEST
003030  001401                          BEQ     TSCD3           ;IF NOT: BR
003032  000000                          HALT
003034  005737  000704          TSCD3:  TST     STFLG           ;SE IF SINGLE TEST
003040  001746                          BEQ     TSCD0           ;IF NOT: BR
003042  017700  175502                  MOV     @SWR,R0
003046  042700  177740                  BIC     #177740,R0      ;BRANCH IF ALL TESTS SELECTED
003052  001641                          BEQ     TSCD
003054  012737  000001  000704  STSCD:  MOV     #1,STFLG        ;SET SINGLE TEST FLAG
003062  023700  001110                  CMP     TLAST,R0        ;SEE IF EXCEEDED TESTS
003066  002410                          BLT     TEND            ;IF SO: BR
003070  006300                          ASL     R0
003072  006100                          ROL     R0              ;SET TABLE MODIFIER
003074  012737  000752  000706          MOV     #TSTTBL,LTADD
003102  060037  000706                  ADD     R0,LTADD        ;SET TEST POINTER
```

```
 973  003106  000726                      BR      TSCD1
 974  003110  005737  001662      TEND:   TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
 975  003114  001265                      BNE     NXTSLV
 976  003116  012704  014661      SDONE:  MOV     #MSG6,R4
 977  003122  004737  013442              JSR     PC,TTOUT        ;PRINT END OF PASS
 978  003126  013703  000730              MOV     PCNTR,R3
 979  003132  004737  013572              JSR     PC,OCTP         ;PRINT PASS NUMBER
 980  003136  005000                      CLR     R0
 981  003140  005300              1$:     DEC     R0
 982  003142  001376                      BNE     1$
 983  003144  013700  000042              MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
 984  003150  001405                      BEQ     HERE            ;BRANCH IF NOT ACT11
 985  003152  000005                      RESET
 986  003154  004710              SENDAD: JSR     PC,(R0)
 987  003156  000240                      NOP
 988  003160  000240                      NOP
 989  003162  000240                      NOP
 990  003164  000240              HERE:   NOP
 991  003166  005737  001662              TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
 992  003172  001005                      BNE     TENDX
 993  003174  032777  010000  175346      BIT     #10000,@SWR     ;SEE IF HALT ON PASS
 994  003202  001001                      BNE     TENDX           ;IF NOT: BR
 995  003204  000000                      HALT
 996  003206  005237  000730      TENDX:  INC     PCNTR           ;BUMP PASS COUNTER
 997  003212  000137  002556              JMP     TSCD            ;RESTART
```

```
 998
 999                                            ;RH ADDRESSING TEST**********
1000
1001  003216  012737  016771  000610  FT1:     MOV     #MSFT1,EMADDR   ;SET HEADER
1002  003224  012777  013004  175342           MOV     #TRAP,@BTRP     ;SET TRAP HANDLER ADDRESS
1003  003232  012777  000340  175336           MOV     #340,@BTRP2
1004  003240  012700  000016                   MOV     #16,R0          ;SET NUMBER OF REGISTERS
1005  003244  013701  000510                   MOV     C1,R1           ;GET FIRST ADDRESS (CSI)
1006  003250  005711                  FT1A:    TST     (R1)            ;REFERENCE REGISTER
1007  003252  000240                           NOP                     ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
1008  003254  005300                  FT1B:    DEC     R0              ;SEE IF DONE ALL
1009  003256  001403                           BEQ     FT1X            ;IF SO: BR
1010  003260  062701  000002                   ADD     #2,R1           ;BUMP ADDRESS POINTER
1011  003264  000771                           BR      FT1A            ;CONTINUE
1012  003266  012777  000006  175300  FT1X:    MOV     #6,@BTRP        ;RESET TRAP CATCHER
1013  003274  005737  000722                   TST     RHTF            ;SEE IF INITIAL ADDRESS TEST PASS
1014  003300  001404                           BEQ     FT1XX           ;IF NOT: BR
1015  003302  005037  000722                   CLR     RHTF            ;CLEAR FLAG
1016  003306  000137  002170                   JMP     ST1A            ;RETURN
1017  003312  000137  003022          FT1XX:   JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1018
1019                                                ;RH REGISTER BITS READ/WRITE**********
1020
1021  003316  012737  017016  000610  FT2:    MOV     #MSFT2,EMADDR   ;SET TEST HEADER
1022  003324  012701  177777            FT2A:   MOV     #-1,R1          ;SET ALL ONES PATTERN
1023  003330  004737  012620                    JSR     PC,INIT1        ;GO INIT
1024  003334  013700  000512                    MOV     WC,R0           ;GET ADDRESS OF WORD COUNT
1025  003340  010102                            MOV     R1,R2           ;SET EXPT REGISTER BIT PATTERN
1026  003342  010110                            MOV     R1,(R0)         ;LOAD PATTERN
1027  003344  021002                            CMP     (R0),R2         ;SEE IF EXPT=RCVD
1028  003346  001410                            BEQ     FT2B            ;IF SO: BR
1029  003350  012737  015205  000650            MOV     #MSG25,ERADD    ;SET CODE
1030  003356  012737  003330  000674            MOV     #FT2A,$COLP     ;SET SCOPE
1031  003364  004737  003504                    JSR     PC,FT2ER        ;GO DO ERROR
1032  003370  013700  000514            FT2B:   MOV     BA,R0           ;GET ADDRESS OF BUS ADDRESS
1033  003374  010102                            MOV     R1,R2
1034  003376  042702  000001                    BIC     #1,R2           ;SET EXPT PATTERN
1035  003402  010110                            MOV     R1,(R0)         ;LOAD PATTERN
1036  003404  020210                            CMP     R2,(R0)         ;SEE IF EXPT=RCVD
1037  003406  001410                            BEQ     FT2C            ;IF SO:BR
1038  003410  012737  015213  000650            MOV     #MSG26,ERADD    ;SET ERROR CODE
1039  003416  012737  003370  000674            MOV     #FT2B,$COLP     ;SET SCOPE ADDRESS
1040  003424  004737  003504                    JSR     PC,FT2ER        ;GO DO ERROR
1041  003430  013700  000532            FT2C:   MOV     DB,R0           ;GET ADDRESS OF DATA BUFFER
1042  003434  010102                            MOV     R1,R2
1043  003436  010110                            MOV     R1,(R0)         ;LOAD PATTERN
1044  003440  012703  004000                    MOV     #4000,R3
1045  003444  005303            FT2D:   DEC     R3              ;DELAY
1046  003446  001376                    BNE     FT2D
1047  003450  020210                    CMP     R2,(R0)         ;SEE IF EXPT=RCVD
1048  003452  001410                    BEQ     FT2E            ;IF SO: BR
1049  003454  012737  015221  000650    MOV     #MSG27,ERADD    ;SET ERROR CODE
1050  003462  012737  003430  000674    MOV     #FT2C,$COLP     ;SET SCOPE ADDRESS
1051  003470  004737  003504            JSR     PC,FT2ER        ;GO DO ERROR
1052  003474  005701            FT2E:   TST     R1              ;SEE IF DONE RESET
1053  003476  001453                    BEQ     FT2X            ;IF SO: BR
1054  003500  005001                    CLR     R1              ;SET ZERO PATTERN
1055  003502  000712                    BR      FT2A            ;DO ZERO BITS
1056  003504  000240            FT2ER:  NOP
1057  003506  032777  020000  175034    BIT     #20000,@SWR     ;SEE IF PRINT ERROR
1058  003514  001034                    BNE     FT2ERB          ;IF NOT: BR
1059  003516  005737  000606            TST     HDRFL           ;SEE ID DONE HEADER
1060  003522  001004                    BNE     FT2ERA          ;IF SO: BR
1061  003524  013704  000610            MOV     EMADDR,R4
1062  003530  004737  013442            JSR     PC,TTOUT        ;DO HEADER
1063  003534  012737  000001  000606    MOV     #1,HDRFL        ;SET FLAG
1064  003542  013704  000650    FT2ERA: MOV     ERADD,R4
1065  003546  004737  013442            JSR     PC,TTOUT        ;PRINT ERROR CODE
1066  003552  012704  015151            MOV     #MSG22,R4
1067  003556  004737  013442            JSR     PC,TTOUT        ;PRINT EXPT TAG
1068  003562  010103                    MOV     R1,R3
1069  003564  004737  013560            JSR     PC,OCTPE        ;PRINT EXPT
1070  003570  012704  015161            MOV     #MSG23,R4
1071  003574  004737  013442            JSR     PC,TTOUT        ;PRINT RCVD TAG
1072  003600  011003                    MOV     (R0),R3
1073  003602  004737  013560            JSR     PC,OCTPE        ;PRINT RCVD
```

```
1074   003606   005777   174736          FT2ERB: TST     @SWR            ;SEE IF HALT ON ERROR
1075   003612   100001                           BPL     FT2ERC          ;IF NOT: BR
1076   003614   000000                           HALT
1077   003616   004737   012512          FT2ERC: JSR     PC,SCOPE        ;GO SEE IF SCOPE ON ERROR
1078   003622   000240                           NOP
1079   003624   000207                           RTS     PC              ;IF NO SCOPE: CONTINUE TEST
1080   003626   000240                  FT2X:    NOP
1081   003630   004737   012546                  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1082   003634   000137   003022                  JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1083
1084                                                    ;RH INITIALIZE TEST**********
1085
1086  003640  012737  017053  000610  FT3:    MOV     #MSFT3,EMADDR   ;SET TEST HEADER
1087  003646  012737  003640  000674          MOV     #FT3,SCOLP
1088  003654  004737  012620                  JSR     PC,INIT1        ;GO INIT
1089  003660  052777  020000  174632          BIS     #20000,@CS      ;FORCE UPE =1
1090  003666  000240                          NOP
1091  003670  004737  012620                  JSR     PC,INIT1        ;GO INIT
1092  003674  005777  174610                  TST     @C1             ;SEE IF SC IS RESET
1093  003700  100005                          BPL     FT3A            ;IF SO: BR
1094  003702  012737  015257  000650          MOV     #MSG29,ERADD    ;SET ERROR CODE
1095  003710  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1096  003714  032777  040000  174566  FT3A:   BIT     #40000,@C1      ;SEE IF TRE IS RESET
1097  003722  001405                          BEQ     FT3B            ;IF SO: BR
1098  003724  012737  015306  000650          MOV     #MSG30,ERADD    ;SET ERROR CODE.
1099  003732  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1100  003736  017701  174556          FT3B:   MOV     @CS,R1          ;GET CS2
1101  003742  042701  000307                  BIC     #307,R1         ;MARK IR/OR
1102  003746  005701                          TST     R1              ;SEE IF RESET
1103  003750  001405                          BEQ     FT3X            ;IF SO: BR
1104  003752  012737  015336  000650          MOV     #MSG31,ERADD    ;SET ERROR CODE
1105  003760  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1106  003764  004737  012546          FT3X:   JSR     PC,ITER         ;GO SEE IF ITERATION
1107  003770  000137  003022                  JMP     TSCD2           ;RETURN TO SCHEDULAR
1108
1109                                          ;ERROR REPORT SUBROUTINE
1110  003774  000240          FT3ER:  NOP
1111  003776  032777  020000  174544          BIT     #20000,@SWR     ;SEE IF PRINT ERROR
1112  004004  001015                          BNE     2$              ;IF NOT: BR
1113  004006  005737  000606                  TST     HDRFL           ;SEE IF DONE HEADER
1114  004012  001006                          BNE     1$              ;IF SO: BR
1115  004014  013704  000610                  MOV     EMADDR,R4
1116  004020  004737  013442                  JSR     PC,TTOUT        ;PRINT HEADER
1117  004024  005237  000606                  INC     HDRFL
1118  004030  013704  000650          1$:     MOV     ERADD,R4
1119  004034  004737  013442                  JSR     PC,TTOUT        ;PRINT ERROR CODE
1120  004040  005777  174504          2$:     TST     @SWR            ;SEE IF HALT ON ERROR
1121  004044  100001                          BPL     3$              ;IF NOT: BR
1122  004046  000000                          HALT
1123  004050  000240          3$:     NOP
1124  004052  004737  012512                  JSR     PC,SCOPE        ;GO SEE IF SCOPE
1125  004056  000207                          RTS     PC              ;IF NOT: BR
```

```
1126
1127                                        ;RH11 SILO TEST 1: EPMTY SILO READ**********
1128
1129  004060  005737  000604        FT4:    TST     RH17F
1130  004064  001141                BNE     FT5X            ;IF RH70: BR
1131  004066  012737  017105  000610        MOV     #MSFT4,EMADDR   ;SET TEST TEST HEADER
1132  004074  012777  000040  174416        MOV     #40,@CS         ;INIT
1133  004102  017700  174424               MOV     @DB,R0          ;READ DB
1134  004106  005777  174406               TST     @CS             ;SEE IF DLT IS SET
1135  004112  100013                BPL     FT4ER           ;IF NOT: BR
1136  004114  005777  174370               TST     @C1             ;SEE IF SC IS SET
1137  004120  100014                BPL     FT4ERA          ;IF NOT: BR
1138  004122  032777  040000  174360        BIT     #40000,@C1      ;SEE IF TRE IS SET
1139  004130  001414                BEQ     FT4ERB          ;IF NOT: BR
1140  004132  004737  012546        FT4X:   JSR     PC,ITER         ;GO SEE IF ITERATION
1141  004136  000137  003022               JMP     TSCD2           ;RETURN TO SCHEDULAR
1142  004142  012737  015366  000650  FT4ER:  MOV     #MSG32,ERADD    ;SET ERROR CODE
1143  004150  000407                BR      FT4ERC
1144  004152  012737  015404  000650  FT4ERA: MOV     #MSG33,ERADD    ;SET ERROR CODE
1145  004160  000403                BR      FT4ERC
1146  004162  012737  015421  000650  FT4ERB: MOV     #MSG34,ERADD    ;SET ERROR CODE.
1147  004170  000240        FT4ERC: NOP
1148  004172  012737  004060  000674        MOV     #FT4,SCOLP      ;SET SCOPE ADDRESS
1149  004200  004737  003774               JSR     PC,FT3ER        ;GO PRINT ERROR
1150  004204  000752                BR      FT4X
```

```
1151
1152                                                ;RH11 SILO TEST 2: IR/OR CHECK**********
1153
1154  004206  005737  000604          FT5:    TST     @#RH17F         ;SEE IF RH70
1155  004212  001066                  BNE     FT5X            ;IF SO: BR
1156  004214  012737  017135  000610          MOV     #MSFT5,EMADDR   ;SET TEST HEADER
1157  004222  012737  004230  000674          MOV     #FT5A,$COLP     ;SET SCOPE ADDRESS
1158  004230  004737  012620          FT5A:   JSR     PC,INITI        ;GO INIT
1159  004234  032777  000100  174256          BIT     #100,@#CS       ;SEE IF IR IS SET
1160  004242  001005                  BNE     FT5B            ;IF SO: BR
1161  004244  012737  015437  000650          MOV     #MSG35,ERADD    ;SET ERROR CODE
1162  004252  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1163  004256  032777  000200  174234  FT5B:   BIT     #200,@#CS       ;SEE IF OR IS RESET
1164  004264  001405                  BEQ     FT5C            ;IF SO: BR
1165  004266  012737  015464  000650          MOV     #MSG36,ERADD    ;SET ERROR CODE
1166  004274  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1167  004300  012777  000000  174224  FT5C:   MOV     #0,@#RB         ;LOAD ZERO INTO SILO
1168  004306  032777  000200  174204          BIT     #200,@#CS       ;SEE THAT OR RESET
1169  004314  001405                  BEQ     FT5D            ;IF IT DOES: BR
1170  004316  012737  015513  000650          MOV     #MSG37,ERADD    ;SET ERROR CODE
1171  004324  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1172  004330  012777  177777  174174  FT5D:   MOV     #-1,@#RB        ;LOAD SILO WITH -1
1173  004336  012700  004000                  MOV     #4000,R0
1174  004342  032777  000200  174150  FT5E:   BIT     #200,@#CS       ;SEE IF OR IS SET
1175  004350  001007                  BNE     FT5X            ;IF SO: BR
1176  004352  005300                  DEC     R0
1177  004354  001372                  BNE     FT5E            ;AWAIT OR
1178  004356  012737  015513  000650          MOV     #MSG37,ERADD    ;SET ERROR CODE
1179  004364  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1180  004370  004737  012546          FT5X:   JSR     PC,ITER         ;GO SEE IF ITERATION
1181  004374  000137  003022                  JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1182
1183                                                ;RH11 SILO TEST 3: SILO DATA TEST**********
1184
1185  004400  005737  000604          FT6:    TST     RH17F
1186  004404  001052                          BNE     FT6X            ;IF RH70: BR
1187  004406  012737  017165  000610          MOV     #MSFT6,EMADDR   ;SET TEST HEADER
1188  004414  012737  004422  000674          MOV     #FT6A,$COLP     ;SET SCOPE ADDRESS
1189  004422  004737  012620          FT6A:   JSR     PC,INIT1        ;GO INIT
1190  004426  005000                          CLR     R0              ;PRESET DATA
1191  004430  010077  174076          FT6B:   MOV     R0,@DB          ;LOAD SILO
1192  004434  005200                          INC     R0              ;BUMP DATA
1193  004436  022700  000102                  CMP     #102,R0         ;SEE IF FILLED ALL
1194  004442  001372                          BNE     FT6B            ;IF NOT: BR
1195  004444  032777  000100  174046          BIT     #100,@CS        ;SEE IF IR IS RESET.
1196  004452  001405                          BEQ     FT6C            ;IF SO: BR
1197  004454  012737  015624  000650          MOV     #MSG40,ERADD    ;SET ERROR CODE
1198  004462  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1199  004466  032777  000200  174024  FT6C:   BIT     #200,@CS        ;SEE IF OR IS SET
1200  004474  001005                          BNE     FT6D            ;IF SO: BR
1201  004476  012737  015552  000650          MOV     #MSG38,ERADD    ;SET ERROR CODE
1202  004504  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1203  004510  005000          FT6D:   CLR     R0              ;PRESET DATA
1204  004512  017701  174014          FT6E:   MOV     @DB,R1          ;READ SILO
1205  004516  020001                          CMP     R0,R1           ;SEE IF EXPT=RCVD
1206  004520  001010                          BNE     FT6DE           ;IF NOT: BR
1207  004522  005200                          INC     R0              ;BUMP DATA
1208  004524  022700  000102                  CMP     #102,R0         ;SEE IF DONE ALL
1209  004530  001370                          BNE     FT6E            ;IF NOT: BR
1210  004532  004737  012546          FT6X:   JSR     PC,ITER         ;GO SEE IF ITERATION
1211  004536  000137  003022                  JMP     TSCD2           ;RETURN TO SCHEDULAR
1212
1213  004542  000240          FT6DE:  NOP
1214  004544  032777  020000  173776          BIT     #20000,@SWR     ;SEE IF PRINT ERROR
1215  004552  001032                          BNE     FT6DEB          ;IF NOT: BR
1216  004554  005737  000606                  TST     HDRFL           ;SEE IF DONE HEADER
1217  004560  013701  000610                  MOV     EMADDR,R1
1218  004564  004737  013442                  JSR     PC,TTOUT        ;PRINT HEADER
1219  004570  005237  000606                  INC     HDRFL           ;SET FLAG
1220  004574  012704  015604          FT6DEA: MOV     #MSG39,R4
1221  004600  004737  013442                  JSR     PC,TTOUT        ;PRINT SILO READ ERROR
1222  004604  012704  015151                  MOV     #MSG22,R4
1223  004610  004737  013442                  JSR     PC,TTOUT        ;PRINT EXPT TAG
1224  004614  010003                          MOV     R0,R3
1225  004616  004737  013572                  JSR     PC,OCTP         ;PRINT EXPT
1226  004622  012704  015161                  MOV     #MSG23,R4
1227  004626  004737  013442                  JSR     PC,TTOUT        ;PRINT RCVD TAG
1228  004632  010103                          MOV     R1,R3
1229  004634  004737  013572                  JSR     PC,OCTP         ;PRINT RCVD
1230  004640  005777  173704          FT6DEB: TST     @SWR            ;SEE IF HALT ON ERROR
1231  004644  100001                          BPL     FT6DEX          ;IF NOT: BR
1232  004646  000000                          HALT
1233  004650  000207          FT6DEX: RTS     PC              ;RETURN TO TEST
```

```
1234
1235                                                    ;RH11 SILO TEST 4: SILO OVERFLOW**********
1236
1237  004652  005737  000604          FT7:    TST     RH17F
1238  004656  001021                  BNE     FT7X            ;IF RH70: BR
1239  004660  012737  017215  000610  MOV     #MSFT7,EMADDR   ;SET TEST HEADER
1240  004666  012737  004652  000674  MOV     #FT7,SCOLP      ;SET SCOPE ADDRESS
1241  004674  004737  012620          JSR     PC,INIT1        ;GO INIT
1242  004700  012700  000103          MOV     #103,R0         ;SET SIZE OF SILO +1
1243  004704  010077  173622          FT7A:   MOV     R0,@DB  ;LOAD SILO
1244  004710  005300                  DEC     R0              ;SEE IF DONE
1245  004712  001374                  BNE     FT7A            ;IF NOT: BR
1246  004714  005777  173600          TST     @CS             ;SEE IF DLT IS SET
1247  004720  100004                  BPL     FT7ER           ;IF NOT: BR
1248  004722  004737  012546          FT7X:   JSR     PC,ITER ;GO SEE IF ITERATION
1249  004726  000137  003022          JMP     TSCD2           ;RETURN TO SCHEDULAR
1250  004732  012737  015366  000650  FT7ER:  MOV     #MSG32,ERADD    ;SET ERROR CODE
1251  004740  004737  003774          JSR     PC,FT3ER        ;GO DO ERROR
1252  004744  000766                  BR      FT7X
```

```
1253
1254                                                    ;RH11 SILO TEST 5: SILO RESET**********
1255
1256  004746  005737  000604          FT10:   TST     RH17F
1257  004752  001034                  BNE     FT10X           ;IF RH70: BR
1258  004754  012737  017245  000610          MOV     #MSFT10,EMADDR  ;SET TEST HEADER
1259  004762  012737  004746  000674          MOV     #FT10,SCOLP     ;SET SCOPE ADDRESS
1260  004770  012777  000040  173522          MOV     #40,@CS         ;INITIALIZE
1261  004776  012700  000004                  MOV     #4,R0           ;SET NUMBER OF SILO WRITER
1262  005002  010077  173524          FT10A:  MOV     R0,@DB          ;WRITE SILO
1263  005006  005300                  DEC     R0              ;SEE IF DONE
1264  005010  001374                  BNE     FT10A           ;IF NOT: BR
1265  005012  052777  000040  173500          BIS     #40,@CS         ;INITIALIZE
1266  005020  012777  177777  173504          MOV     #-1,@DB         ;WRITE SILO
1267  005026  017701  173500                  MOV     @DB,R1          ;READ SILO 1
1268  005032  017701  173474                  MOV     @DB,R1          ;READ SILO 2
1269  005036  005777  173456                  TST     @CS             ;SEE IF DLT IS SET
1270  005042  100011                  BPL     FT10ER          ;IF NOT: BR
1271  005044  004737  012546          FT10X:  JSR     PC,ITER         ;GO SEE IF ITERATION
1272  005050  005737  000726                  TST     RH0F            ;SEE IF RH11 ONLY
1273  005054  001402                  BEQ     FT10XX          ;IF NOT: BR
1274  005056  000137  003110                  JMP     TEND            ;ELSE GO TO END
1275  005062  000137  003022          FT10XX: JMP     TSCD2           ;RETURN TO SCHEDULAR
1276  005066  012737  015366  000650  FT10ER: MOV     #MSG32,ERADD    ;SET ERROR CODE
1277  005074  004737  003774                  JSR     PC,FT3ER        ;GO DO ERROR
1278  005100  000761                  BR      FT10X
```

```
1279                                                    ;NOP TEST*****
1280
1281  005102  000240                            FT11:   NOP
1282  005104  012737  005102  000674                    MOV     #FT11,SCOLP     ;SET SCOPE ADDRESS
1283  005112  004737  012620                            JSR     PC,INITI
1284  005116  012737  000300  000716                    MOV     #300,UDES       ;SET TC= ALL NRZ,NORM,ODD
1285  005124  012737  177777  000620                    MOV     #-1,FCNT        ;SET FC= ALL OVER
1286  005132  012737  177777  000622                    MOV     #-1,WCNT        ;SET WC= ALL OVER
1287  005140  012737  177777  000616                    MOV     #-1,BADDR       ;SET BA= ALL OVER
1288  005146  012737  000001  000636                    MOV     #1,RDYDX        ;SET DELAY
1289  005154  012737  000001  000640                    MOV     #1,OPDYX        ;SET OP DELAY
1290  005162  012737  000001  000710                    MOV     #1,FUN          ;SET NOP FUNCTIONS CODE
1291  005170  004737  011546                            JSR     PC,EXEC         ;GO EXECUTE COMMAND
1292  005174  000240                                    NOP
1293  005176  012737  017276  000610                    MOV     #MSFT11,EMADDR
1294  005204  004737  011776                            JSR     PC,ERCHK        ;GO CHECK REGISTER
1295  005210  004737  012546                            JSR     PC,ITER         ;GO SEE IF ITERATIONS
1296  005214  000137  003022                            JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1297                                                    ;REWIND TEST*****
1298
1299  005220  000240                          FT12:     NOP
1300  005222  012737  005220  000674                    MOV     #ET12,SCOLP
1301  005230  004737  012520                            JSR     PC,INITI        ;GO INITIALIZE
1302  005234  052777  001700  173300                    BIS     #1700,@TC       ;SET TO NRZ,NORMAL
1303  005242  012737  177760  000620                    MOV     #-20,FCNT       ;SET FC=20
1304  005250  012737  177770  000622                    MOV     #-10,WCNT       ;SET WC=10
1305  005256  012737  020112  000616                    MOV     #WDATA,BADDR    ;SET BA=WRITE BUFFER
1306  005264  012737  000007  000710                    MOV     #7,FUN          ;SET REWIND OP CODE
1307  005272  004737  011546                            JSR     PC,EXEC         ;GO EXECUTE COMMAND
1308  005276  000240                                    NOP
1309  005300  032777  020000  173214  FT12A:  BIT       #20000,@DS
1310  005306  001374                                    BNE     FT12A           ;AWAIT PIP
1311  005310  012737  017316  000610                    MOV     #MSFT12,EMADDR
1312  005316  004737  011776                            JSR     PC,ERCHK        ;GO CHECK FOR ERROR
1313  005322  004737  012546                            JSR     PC,ITER         ;GO SEE IF ITERATION
1314  005326  000137  003022                            JMP     TSCD2           ;RETURN TO SCHEDULAR
1315
```

```
1316                                                    ;WRITE/READ TEST*****
1317
1318  005332  000240                            FT13:   NOP
1319  005334  012737  000001  000636                    MOV     #1,RDYDX
1320  005342  012737  000001  000640                    MOV     #1,OPDYX
1321  005350  012737  000100  000624                    MOV     #100,RCNT       ;SET RECORD COUNT
1322  005356  012737  017341  000610                    MOV     #MSFT13,EMADDR  ;SET TEST HEADER
1323  005364  012737  000001  000720                    MOV     #1,PATRN
1324  005372  004737  012400                           JSR     PC,DSUP         ;SET UP ALL ONES DATA PATTERN
1325  005376  012737  001700  000716                    MOV     #1700,UDES      ;SET TO 800 BPI NORMAL
1326  005404  004737  011700                  FT13A:   JSR     PC,RWND         ;GO REWIND
1327  005410  012737  177600  000620                    MOV     #-200,FCNT      ;SET FC
1328  005416  012737  177700  000622                    MOV     #-100,WCNT      ;SET WC
1329  005424  012737  020112  000616                    MOV     #WDATA,BADDR    ;SET BA
1330  005432  012737  000061  000710                    MOV     #61,FUN         ;SET WRITE OP-CODE
1331  005440  012737  014765  000626                    MOV     #MSG12,ERRP
1332  005446  004737  011546                  FT13B:   JSR     PC,EXEC         ;GO EXECUTE COMMAND
1333  005452  005037  000674                           CLR     SCOLP           ;NO SCOPE LOOP
1334  005456  004737  011776                           JSR     PC,ERCHK        ;GO CHECK ERROR
1335  005462  005337  000624                           DEC     RCNT            ;SEE IF DONE ALL
1336  005466  001367                                   BNE     FT13B           ;IF NOT: BR
1337  005470  012737  000100  000624                    MOV     #100,RCNT       ;SET RECORD COUNT
1338  005476  012737  021624  000616                    MOV     #RDATA,BADDR
1339  005504  062737  000200  000616                    ADD     #200,BADDR      ;SET BA
1340  005512  012737  000077  000710                    MOV     #77,FUN         ;SET READ REVERSE OP-CPDE
1341  005520  012737  015003  000626                    MOV     #MSG13,ERRP
1342  005526  004737  011546                  FT13C:   JSR     PC,EXEC         ;GO EXECUTE COMMAND
1343  005532  004737  011776                           JSR     PC,ERCHK        ;GO CHECK ERROR
1344  005536  005337  000624                           DEC     RCNT            ;SEE IF READ ALL
1345  005542  001371                                   BNE     FT13C           ;IF NOT:BR
1346  005544  162737  000200  000616                    SUB     #200,BADDR      ;SET BA
1347  005552  012737  000071  000710                    MOV     #71,FUN         ;SET READ FORWARD OP-CODE
1348  005560  012737  015030  000626                    MOV     #MSG14,ERRP
1349  005566  012737  000100  000624                    MOV     #100,RCNT       ;SET RECORD COUNT
1350  005574  004737  011546                  FT13D:   JSR     PC,EXEC         ;GO EXECUTE COMMAND
1351  005600  004737  011776                           JSR     PC,ERCHK        ;GO CHECK ERRORS
1352  005604  005337  000624                           DEC     RCNT            ;SEE IF DONE ALL
1353  005610  001371                                   BNE     FT13D           ;IF NOT:BR
1354  005612  032737  002000  000716                    BIT     #2000,UDES      ;SEE IF DONE PE
1355  005620  001007                                   BNE     FT13X           ;IF SO: BR
1356  005622  012737  002300  000716                    MOV     #2300,UDES      ;SET PE MODE
1357  005630  012737  000100  000624                    MOV     #100,RCNT       ;RESET RECORD COUNT
1358  005636  000662                                   BR      FT13A           ;GO DO NEXT DENSITY
1359  005640  000137  003022                  FT13X:   JMP     TSCD2           ;RETURN TO SCHEDULAR
```

M03


```
                                                       ;SPACE TEST#####

005644  000240                            FT14:     NOP
005646  012737  017370  000610                      MOV     #MSFT14,EMADDR      ;SET TEST HEADER
005654  012737  001700  000716                      MOV     #1700,UDES          ;SET NRZ,NORMAL
005662  004737  011700                    FT14A1:   JSR     PC,RWND             ;GO INITIALIZE
005666  012737  000100  000624                      MOV     #100,RCNT           ;SET NUMBER OF RECORDER
005674  012737  177777  020112                      MOV     #-1,WDATA           ;SET DATA PATTERN
005702  012737  177700  000620                      MOV     #-100,FCNT          ;PRESET FRAME CNT
005710  012737  177740  000622                      MOV     #-40,WCNT           ;PRESET WORD CNT
005716  004737  012620                    FT14A:    JSR     PC,INIT1            ;GO REWIND
005722  012737  001000  000640                      MOV     #1000,OPDYX
005730  012737  040000  000636                      MOV     #40000,RDYDX
005736  012737  000061  000710                      MOV     #61,FUN             ;SET WRITE OP-CODE
005744  012737  102300  000660                      MOV     #102300,STMSK       ;MASK DATA RELATED ERRORS
005752  052777  000010  172540                      BIS     #10,@CS             ;INHIBIT BUS ADDRESS INCREMENT
005760  004737  011546                              JSR     PC,EXEC             ;GO EXECUTE COMMAND
005764  012737  016043  000626                      MOV     #MSG46,ERRP         ;SET ERROR CODE
005772  004737  011776                              JSR     PC,ERCHK            ;GO CHECK ERRORS
005776  005737  000712                              TST     SERFL               ;SEE IF ERROR
006002  001402                                      BEQ     FT14A2              ;IF NOT: BR
006004  000137  006470                              JMP     FT14X               ;ELSE EXIT
006010  005337  000620                    FT14A2:   DEC     FCNT                ;BUMP FC
006014  032737  000001  000620                      BIT     #1,FCNT             ;SEE IF SHOULD BUMP WC
006022  001403                                      BEQ     FT14A3              ;IF NOT: BR
006024  162737  000001  000622                      SUB     #1,WCNT             ;BUMP WC
006032  005337  000624                    FT14A3:   DEC     RCNT                ;SEE IF DONE ALL
006036  001327                                      BNE     FT14A               ;WRITE ALL RECORDS
006040  012737  000100  000632                      MOV     #100,RRD            ;PRESET RECORD POSITION
006046  012737  000176  000634                      MOV     #176,RFD
006054  012737  177701  000642                      MOV     #-77,SCNT           ;SET SPACE AMOUNT
006062  012737  000033  000710            FT14B:    MOV     #33,FUN             ;SET OP-CODE SPACE REVERSE
006070  004737  011546                              JSR     PC,EXEC             ;GO EXECUTE COMMAND
006074  012737  016114  000626                      MOV     #MSG48,ERRP         ;SET ERROR CODE
006102  004737  011776                              JSR     PC,ERCHK            ;GO CHECK ERRORS
006106  005737  000712                              TST     SERFL               ;SEE IF ERROR
006112  001166                                      BNE     FT14X               ;IF SO: BR
006114  004737  006210                              JSR     PC,FT14RR           ;GO READ REVERSE + CHECK DATA
006120  000240                                      NOP
006122  012737  000031  000710                      MOV     #31,FUN             ;SET SPACE FORWARD OP-CODE
006130  005237  000642                              INC     SCNT                ;SET SPACE AMOUNT
006134  001555                                      BEQ     FT14X               ;IF DONE: BR
006136  004737  011546                              JSR     PC,EXEC             ;GO EXECUTE COMMAND
006142  012737  016067  000626                      MOV     #MSG47,ERRP         ;SET ERROR CODE
006150  004737  011776                              JSR     PC,ERCHK            ;GO CHECK ERROR
006154  005737  000712                              TST     SERFL               ;SEE IF ERROR FLAG
006160  001143                                      BNE     FT14X               ;IF NO: BR
006162  004737  006252                              JSR     PC,FT14RF           ;GO READ FORWARD FOR POSITION CHECK
006166  000240                                      NOP
006170  005237  000642                              INC     SCNT                ;DECREMENT SPACE AMOUNT
006174  001535                                      BEQ     FT14X               ;IF DONE: BR
006176  005237  000632                              INC     RRD                 ;BUMP DATA EXPT
006202  005337  000634                              DEC     RFD                 ;BUMP DATA EXPT
006206  000725                                      BR      FT14B
006210  000240                            FT14RR:   NOP
006212  012737  021624  000616                      MOV     #RDATA,BADDR        ;SET BA
```

```
1416  006220  012737  000077  000710          MOV     #77,FUN         ;SET READ REVERSE OP-CODE
1417  006226  004737  011546                  JSR     PC,EXEC         ;GO EXECUTE COMMAND
1418  006232  000240                          NOP
1419  006234  013705  000632                  MOV     RRD,R5
1420  006240  020577  172252                  CMP     R5,@FC          ;SEE IF CORRECT RECORD
1421  006244  001020                          BNE     FT14RER         ;IF NOT: BR
1422  006246  000137  006300                  JMP     FT14EC          ;GO CLEAR RH11 ERROR BIT
1423  006252  000240          FT14RF: NOP
1424  006254  012737  000071  000710          MOV     #71,FUN         ;SET READ FORWARD OP-CODE
1425  006262  004737  011546                  JSR     PC,EXEC         ;GO EXECUTE COMMAND
1426  006266  013705  000634                  MOV     RFD,R5
1427  006272  020577  172220                  CMP     R5,@FC          ;SEE IF CORRECT RECORD
1428  006276  001003                          BNE     FT14RER         ;IF NOT: BR
1429  006300  004737  012620          FT14EC: JSR     PC,INIT1        ;CLEAR RH
1430  006304  000207                          RTS     PC              ;RETURN
1431  006306  000240          FT14RER: NOP
1432  006310  032777  020000  172232          BIT     #20000,@SWR     ;SEE IF PRINT INHIBITED
1433  006316  001060                          BNE     FT14R3          ;IF SO: BR
1434  006320  012704  017370                  MOV     #MSFT14,R4
1435  006324  004737  013442                  JSR     PC,TTOUT        ;PRINT HEADER
1436  006330  012704  014703                  MOV     #MSG9,R4
1437  006334  004737  013442                  JSR     PC,TTOUT        ;PRINT ERROR TYPE
1438  006340  012704  015136                  MOV     #MSG20,R4       ;SET NRZ TAG POINTER
1439  006344  032737  002000  000716          BIT     #2000,UDES      ;SEE IF PE
1440  006352  001402                          BEQ     FT14RD          ;IF NOT: BR
1441  006354  012704  015144                  MOV     #MSG21,R4       ;ELSE SET PE TAG POINTER
1442  006360  004737  013442          FT14RD: JSR     PC,TTOUT        ;PRINT TAG
1443  006364  032737  000002  000710          BIT     #2,FUN          ;SEE IF READ REVERSE
1444  006372  001003                          BNE     FT14R1          ;IF SO: BR
1445  006374  012704  015116                  MOV     #MSG17,R4
1446  006400  000402                          BR      FT14R2          ;GO PRINT
1447  006402  012704  015076          FT14R1: MOV     #MSG16,R4
1448  006406  004737  013442          FT14R2: JSR     PC,TTOUT        ;PRINT FRWD/REV
1449  006412  012704  015151                  MOV     #MSG22,R4
1450  006416  004737  013442                  JSR     PC,TTOUT        ;PRINT EXPT TAG
1451  006422  010503                          MOV     R5,R3
1452  006424  042703  177700                  BIC     #177700,R3      ;MASK RECORD NUMBER
1453  006430  004737  013572                  JSR     PC,OCTP         ;PRINT EXPT RECORD NUMBER
1454  006434  012704  015161                  MOV     #MSG23,R4
1455  006440  004737  013442                  JSR     PC,TTOUT        ;PRINT RCVD TAG
1456  006444  017703  172046                  MOV     @FC,R3
1457  006450  042703  177700                  BIC     #177700,R3      ;MASK RECORD NUMBER
1458  006454  004737  013572                  JSR     PC,OCTP         ;PRINT ACTUAL RECORD NUMBER
1459  006460  005777  172064          FT14R3: TST     @SWR            ;SEE IF HALT ON ERROR
1460  006464  100001                          BPL     FT14X           ;IF NOT: BR
1461  006466  000000                          HALT
1462  006470  032737  002000  000716  FT14X:  BIT     #2000,UDES      ;SEE IF DONE PE
1463  006476  001005                          BNE     FT14XX          ;IF SO: BR
1464  006500  012737  002300  000716          MOV     #2300,UDES      ;SET TO PE
1465  006506  000137  005662                  JMP     FT14A1          ;DO IN PE
1466  006512  000137  003022          FT14XX: JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1467                                            ;ERASE TEST*****
1468
1469  006516  000240                    FT15:   NOP
1470  006520  005037  000660                    CLR     STMSK
1471  006524  012737  000100  000636            MOV     #100,RDYDX
1472  006532  012737  000010  000640            MOV     #10,OPDYX
1473  006540  012737  017412  000610            MOV     #MSFT15,EMADDR  ;SET TEST HEADER
1474  006546  004737  011700                    JSR     PC,RWND         ;REWIND
1475  006552  012737  021624  000616            MOV     #RDATA,BADDR    ;SET BA
1476  006560  012737  001700  000716            MOV     #1700,UDES      ;SET NRZ, NORMAL
1477  006566  012737  000025  000710  FT15A:    MOV     #25,FUN         ;SET ERASE OP-CODE
1478  006574  012737  000200  000624            MOV     #200,RCNT       ;SET TO ERASE 128 TIMES
1479  006602  004737  011546          FT15B:    JSR     PC,EXEC         ;GO EXECUTE COMMAND
1480  006606  012737  016043  000626            MOV     #MSG46,ERRP     ;SET ERROR CODE
1481  006614  004737  011776                    JSR     PC,ERCHK        ;GO CHECK ERRORS
1482  006620  005737  000712                    TST     SERFL           ;SEE IF ANY ERRORS
1483  006624  001032                            BNE     FT15X           ;IF SO EXIT
1484  006626  005337  000624                    DEC     RCNT            ;SEE IF DONE ERASING
1485  006632  001363                            BNE     FT15B           ;IF NOT: BR
1486  006634  000240                            NOP
1487  006636  004737  011700                    JSR     PC,RWND         ;REWIND
1488  006642  012737  177600  000622            MOV     #-200,WCNT      ;SET WC
1489  006650  012737  000071  000710            MOV     #71,FUN         ;SET READ FORWARD OP-CODE
1490  006656  012737  000040  000636            MOV     #40,RDYDX       ;SET DELAY
1491  006664  004737  011546                    JSR     PC,EXEC         ;GO EXECUTE COMMAND
1492  006670  000240                            NOP
1493  006672  012737  016503  000626            MOV     #MSG60,ERRP     ;SET ERROR CODE
1494  006700  012737  020000  000660            MOV     #20000,STMSK
1495  006706  004737  011776                    JSR     PC,ERCHK        ;GO CHECK ERRORS
1496  006712  000137  003022          FT15X:    JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1497                                                ;TAPE MARK WRITE/READ TEST*****
1498
1499  006716  000240                        FT16:   NOP
1500  006720  012737  000001  000636              MOV     #1,RDYDX
1501  006726  012737  001000  000640              MOV     #1000,OPDYX
1502  006734  012737  017434  000610              MOV     #MSFT16,EMADDR  ;SET HEADER
1503  006742  012737  001700  000716              MOV     #1700,UDES      ;SET TO NRZ,NORMAL,ODD
1504  006750  004737  011700              FT16A:  JSR     PC,RWND         ;INIT AND REWIND SLAVE
1505  006754  012737  177760  000620      FT16B:  MOV     #-20,FCNT       ;FC=20
1506  006762  012737  177770  000622              MOV     #-10,WCNT       ;WC=10
1507  006770  012737  000027  000710              MOV     #27,FUN         ;SET WRITE TAPE MARK OP-CODE
1508  006776  004737  011546                      JSR     PC,EXEC         ;GO EXECUTE COMMAND
1509  007002  012737  001000  000660              MOV     #1000,STMSK     ;SET FOR FCE MASK
1510  007010  012737  014765  000626              MOV     #MSG12,ERRP     ;SET ERROR CODE
1511  007016  004737  011776                      JSR     PC,ERCHK        ;GO CHECK ERROR
1512  007022  004737  012340                      JSR     PC,TMCHK        ;GO SEE IF TM SET
1513  007026  012737  000077  000710              MOV     #77,FUN         ;SET USED REVERSE OP-CODE
1514  007034  004737  011546                      JSR     PC,EXEC         ;GO EXECUTE COMMAND
1515  007040  012737  001000  000660              MOV     #1000,STMSK     ;SET FCE ERROR MASK
1516  007046  012737  015003  000626              MOV     #MSG13,ERRP     ;SET ERROR CODE
1517  007054  004737  011776                      JSR     PC,ERCHK        ;GO CHECK ERRORS
1518  007060  004737  012340                      JSR     PC,TMCHK        ;GO SEE IF TM SET
1519  007064  012737  000071  000710              MOV     #71,FUN         ;SET READ  FORWARD OP-CODE
1520  007072  004737  011546                      JSR     PC,EXEC         ;GO EXECUTE COMMAND
1521  007076  012737  015030  000626              MOV     #MSG14,ERRP     ;SET ERROR CODE
1522  007104  004737  011776                      JSR     PC,ERCHK        ;TO CHECK ERRORS
1523  007110  004737  012340                      JSR     PC,TMCHK        ;GO SEE IF TM SET
1524  007114  032737  002000  000716              BIT     #2000,UDES      ;SEE IF DONE PE
1525  007122  001004                              BNE     FT16X           ;IF SO: BR
1526  007124  012737  002300  000716              MOV     #2300,UDES      ;SET PE, NORMAL
1527  007132  000706                              BR      FT16A           ;DO IN PE
1528  007134  004737  012546              FT16X:  JSR     PC,ITER         ;DO ITERATIONS
1529  007140  000137  003022                      JMP     TSCD2           ;RETURN TO SCHEDULAR
1530
```

```
1531
1532                                                ;TAPE MARK SPACE TEST**********
1533
1534  007144  005037  000624          FT17:   CLR     RCNT
1535  007150  012737  017475  000610          MOV     #MSFT17,EMADDR  ;SET HEADER
1536  007156  012737  001700  000716          MOV     #1700,UDES      ;SET TO NRZ
1537  007164  004737  011700          FT17A:  JSR     PC,RWND         ;REWIND TAPE
1538  007170  012737  000027  000710  FT17B:  MOV     #27,FUN
1539  007176  012737  040000  000636          MOV     #40000,RDYDX    ;SET DRY DELAY
1540  007204  012737  040000  000640          MOV     #40000,OPDYX    ;SET OP DELAY
1541  007212  004737  011546                  JSR     PC,EXEC         ;GO WRITE TM
1542  007216  012737  102300  000660          MOV     #102300,STMSK   ;MASK DATA RELATED ERRORS
1543  007224  012737  015055  000626          MOV     #MSG15,ERRP     ;SET ERROR TYPE
1544  007232  004737  011776                  JSR     PC,ERCHK        ;GO CHECK ERROR
1545  007236  005737  000712                  TST     SERFL           ;SEE IF ERROR
1546  007242  001137                          BNE     FT17X           ;IF SO: BR
1547  007244  004737  012340                  JSR     PC,TMCHK        ;GO SEE IF TM SET
1548  007250  000240                          NOP
1549  007252  000240                          NOP
1550  007254  032737  000100  000624          BIT     #100,RCNT       ;SEE IF DONE PATTERN
1551  007262  001045                          BNE     FT17D           ;IF SO: BR
1552  007264  062737  000020  000624          ADD     #20,RCNT        ;ADD 20 TO RECORD COUNT
1553  007272  013737  000624  000652          MOV     RCNT,TEMP1      ;SAVE RECORD COUNT
1554  007300  012737  177600  000622          MOV     #-200,WCNT      ;WC=128
1555  007306  012737  177400  000620          MOV     #-400,FCNT      ;FC=256
1556  007314  012737  020112  000616          MOV     #MDATA,BADDR    ;BA=WRITE BUFFER
1557  007322  012737  000061  000710          MOV     #61,FUN         ;SET WRITE OP CODE
1558  007330  000240                  FT17C:  NOP
1559  007332  000240                          NOP
1560  007334  004737  011546                  JSR     PC,EXEC         ;GO WRITE
1561  007340  012737  014765  000626          MOV     #MSG12,ERRP     ;SET ERROR CODE
1562  007346  012737  102300  000660          MOV     #102300,STMSK   ;MASK DATA RELATED ERRORS
1563  007354  004737  011776                  JSR     PC,ERCHK        ;GO CHECK ERROR
1564  007360  005737  000712                  TST     SERFL           ;SEE IF ERROR
1565  007364  001066                          BNE     FT17X           ;IF SO: BR
1566  007366  005337  000652                  DEC     TEMP1           ;SEE IF DONE ALL
1567  007372  001356                          BNE     FT17C           ;IF NOT: BR
1568  007374  000675                          BR      FT17B           ;ELSE GO DO TM
1569  007376  000240                  FT17D:  NOP
1570  007400  012737  000033  000710          MOV     #33,FUN         ;SET SPACE REVERSE
1571  007406  012737  015076  000626          MOV     #MSG16,ERRP     ;SET ERROR CODE
1572  007414  012737  177600  000642  FT17D1: MOV     #-200,SCNT      ;SET TO 200 RECORDS
1573  007422  012737  000005  000624          MOV     #5,RCNT         ;SET NUMBER OF OPS TO DO
1574  007430  004737  012620          FT17E:  JSR     PC,INIT1        ;GO INIT
1575  007434  004737  011546                  JSR     PC,EXEC         ;GO SPACE
1576  007440  012737  001000  000660          MOV     #1000,STMSK     ;SET ERROR MASK
1577  007446  004737  011776                  JSR     PC,ERCHK        ;GO CHECK ERROR
1578  007452  005737  000712                  TST     SERFL           ;SEE IF ERROR
1579  007456  001031                          BNE     FT17X           ;IF SO: BR
1580  007460  004737  012340                  JSR     PC,TMCHK        ;GO SEE IF TM SET
1581  007464  005337  000624                  DEC     RCNT            ;SEE IF DONE SPACES
1582  007470  001357                          BNE     FT17E           ;IF NOT: BR
1583  007472  022737  000031  000710          CMP     #31,FUN         ;SEE IF DONE FORWARD
1584  007500  001407                          BEQ     FT17F           ;IF SO: BR
1585  007502  012737  015116  000626          MOV     #MSG17,ERRP     ;SET ERROR CODE
1586  007510  012737  000031  000710          MOV     #31,FUN         ;SET TO SPACE FORWARD
```

```
1587  007516  000736                          BR      FT17D1          ;DO FORWARD
1588  007520  032737  002000  000716  FT17F:  BIT     #2000,UDES      ;SEE IF DONE PE
1589  007526  001005                          BNE     FT17X           ;IF SO: BR
1590  007530  012737  002300  000716          MOV     #2300,UDES      ;SET TO PE
1591  007536  000137  007164                  JMP     FT17A           ;GO PE
1592  007542  000137  003022          FT17X:  JMP     TSCD2           ;RETURN  TO SCHEDULAR
```

```
1593
1594                                        ;WRITE CHECK TEST**********
1595
1596  007546  000240                FT20:   NOP
1597  007550  012737  017523  000610        MOV     #MSFT20,EMADDR  ;SET HEADER
1598  007556  012737  001700  000716        MOV     #1700,UDES      ;SET UNIT DESCRIPTION
1599  007564  004737  011700        FT20A:  JSR     PC,RWND         ;INIT AND REWIND SLAVE
1600  007570  012737  000003  000720        MOV     #3,PATRN
1601  007576  004737  012400                JSR     PC,DSUP         ;GO SET PATTERN 3
1602  007602  012737  020112  000616        MOV     #WDATA,BADDR    ;SET BA
1603  007610  012737  177400  000620        MOV     #-400,FCNT      ;SET FC
1604  007616  012737  177600  000622        MOV     #-200,WCNT      ;SET WC
1605  007624  012737  000061  000710        MOV     #61,FUN         ;SET WRITE OP CODE
1606  007632  004737  011546                JSR     PC,EXEC         ;GO WRITE RECORD
1607  007636  012737  016043  000626        MOV     #MSG46,ERRP     ;SET ERROR CODE
1608  007644  004737  011776                JSR     PC,ERCHK        ;GO CHECK ERROR
1609  007650  005737  000712                TST     SERFL           ;SEE IF ERORR
1610  007654  001042                        BNE     FT20X           ;IF SO: BR
1611  007656  012737  015076  000626        MOV     #MSG16,ERRP     ;SET REVERSE ERROR TAG
1612  007664  012737  000057  000710        MOV     #57,FUN         ;SET REVERSE WRITE CHECK OP-CODE
1613  007672  062737  000376  000616        ADD     #376,BADDR      ;SET BA FOR REVERSE CHECK
1614  007700  004737  011546                JSR     PC,EXEC         ;GO DO REVERSE CHECK
1615  007704  004737  011776                JSR     PC,ERCHK        ;GO CHECK ERROR
1616  007710  012737  015116  000626  FT20B:  MOV     #MSG17,ERRP     ;SET FORWARD TAG
1617  007716  012737  000051  000710        MOV     #51,FUN         ;SET FORWARD CHECK OP CODE
1618  007724  162737  000376  000616        SUB     #376,BADDR      ;SET BA FOR FORWARD CHECK
1619  007732  004737  011546                JSR     PC,EXEC         ;GO DO FORWARD CHECK
1620  007736  004737  011776                JSR     PC,ERCHK        ;GO CHECK ERROR
1621  007742  032737  002000  000716  FT20C:  BIT     #2000,UDES      ;SEE IF DONE PE
1622  007750  001004                        BNE     FT20X           ;IF SO: BR
1623  007752  012737  002300  000716        MOV     #2300,UDES      ;ELSE SET PE
1624  007760  000701                        BR      FT20A           ;DO IN PE
1625  007762  004737  012546        FT20X:  JSR     PC,ITER         ;DO ITERATIONS
1626  007766  000137  003022                JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1627
1628                                            ;ERASE HEAD TEST*****
1629
1630  007772  012737  017554  000610  FT21:     MOV     #MSFT21,EMADDR  ;SET TEST HEADER
1631  010000  004737  011700          FT21A:    JSR     PC,RWND         ;GO REWIND
1632  010004  012737  000003  000720            MOV     #3,PATRN
1633  010012  004737  012400                    JSR     PC,DSUP         ;GO SET PATTERN 3
1634  010016  012737  021112  000616            MOV     #WDATA,BADDR    ;SET BA=WRITE BUFFER
1635  010024  012737  176340  000620            MOV     #-800.,FCNT     ;SET FC=800(10)
1636  010032  012737  177160  000622            MOV     #-400.,WCNT     ;SET WC=400(10)
1637  010040  012737  001700  000716            MOV     #1700,UDES      ;SET NRZ, NORMAL
1638  010046  012737  000061  000710            MOV     #61,FUN         ;SET WRITE OP-CODE
1639  010054  004737  011546                    JSR     PC,EXEC         ;GO DO WRITE 1
1640  010060  012737  014765  000626            MOV     #MSG12,ERRP     ;SET ERROR CODE
1641  010066  004737  011776                    JSR     PC,ERCHK        ;GO CHECK FOR ERROR
1642  010072  004737  011546                    JSR     PC,EXEC         ;YES DO WRITE 2
1643  010076  004737  011776                    JSR     PC,ERCHK        ;YES CHECK FOR ERROR
1644  010102  000240                            NOP
1645  010104  004737  011700                    JSR     PC,RWND         ;GO REWIND
1646  010110  012737  177160  000620            MOV     #-400.,FCNT     ;SET FC=400(10)
1647  010116  012737  177470  000622            MOV     #-200.,WCNT     ;SET WC=200(10)
1648  010124  004737  011546                    JSR     PC,EXEC         ;GO REWRITE RECORD 1-WH TO EH
1649  010130  000240            FT21SCP:NOP
1650  010132  004737  011700                    JSR     PC,RWND         ;REWIND
1651  010136  012737  021624  000616            MOV     #RDATA,BADDR    ;SET BA=READ BUFFER
1652  010144  012737  177160  000620            MOV     #-400.,FCNT     ;SET FC=400
1653  010152  012737  177470  000622            MOV     #-200.,WCNT     ;SET WC=200
1654  010160  012737  000071  000710            MOV     #71,FUN         ;SET READ OP-CODE
1655  010166  004737  011546                    JSR     PC,EXEC         ;GO READ RECORD 1
1656  010172  012737  015030  000626            MOV     #MSG14,ERRP     ;SET ERROR CODE
1657  010200  004737  011776                    JSR     PC,ERCHK        ;GO CHECK FOR ERROR
1658  010204  000240                            NOP
1659  010206  052777  000010  170304            BIS     #10,@CS         ;INHIBIT BA INCREMENT
1660  010214  012737  176340  000620            MOV     #-800.,FCNT     ;SET FC=800(10)
1661  010222  012737  177160  000622            MOV     #-400.,WCNT     ;SET WC=400(10)
1662  010230  004737  011546                    JSR     PC,EXEC         ;GO READ RECORD 2
1663  010234  022777  001440  170254            CMP     #800.,@FC       ;SEE IF READ RECORD 2 OK
1664  010242  001424                            BEQ     FT21X           ;IF SO: BR
1665  010244  022777  001441  170244            CMP     #801.,@FC       ;BRANCH IF IN GREY AREA
1666  010252  001420                            BEQ     FT21X
1667  010254  022777  001440  170234  1$:       CMP     #800.,@FC       ;BRANCH IF ERASE HEAD REVERSED
1668  010262  101404                            BLOS    FT21B           ;IF SO: BR
1669  010264  012737  015736  000650            MOV     #MSG44,ERADD    ;SET ERASE HEAD INOPERATIVE ERROR CODE
1670  010272  000403                            BR      FT21C
1671  010274  012737  015766  000650  FT21B:    MOV     #MSG45,ERADD    ;SET ERASE HEAD REVERSED ERROR CODE
1672  010302  012737  010130  000674  FT21C:    MOV     #FT21SCP,SCOLP  ;SET SCOPE ADDRESS
1673  010310  004737  003774                    JSR     PC,FT3ER        ;GO PRINT ERROR
1674  010314  004737  012546          FT21X:    JSR     PC,ITER         ;GO SEE IF ITERATION
1675  010320  000137  003022                    JMP     TSCD2           ;RETURN TO SCHEDULAR
1676
1677
```

```
1678                                                    ;BUFFERED COMMAND TEST*****
1679
1680  010324  012737  017603  000610  FT22:   MOV     #MSFT22,EMADDR  ;SET TEST HEADER
1681  010332  004737  011700          JSR     PC,RWND         ;GO REWIND
1682  010336  012700  000003          MOV     #3,R0           ;SET NUMBER OF WRITES
1683  010342  012737  001700  000716  MOV     #1700,UDES      ;SET TO NRZ NORMAL
1684  010350  012737  020112  000616  MOV     #WDATA,BADDR    ;SET BA=WRITE BUFFER
1685  010356  012737  177000  000620  MOV     #-1000,FCNT     ;SET FC=1000
1686  010364  012737  177400  000622  MOV     #-400,WCNT      ;SET WC=400
1687  010372  012737  000061  000710  MOV     #61,FUN         ;SET WRITE OP-CODE
1688  010400  004737  011546          JSR     PC,EXEC         ;GO DO WRITE
1689  010404  005300          FT22A:  DEC     R0              ;SEE IF DONE ALL
1690  010406  001374                  BNE     FT22A           ;IF NOT: BR
1691  010410  000240                  NOP
1692  010412  012777  000007  170070  MOV     #7,@C1          ;START REWIND
1693  010420  032777  000200  170074  FT22B:  BIT     #200,@DS
1694  010426  001774                  BEQ     FT22B
1695  010430  004737  012620          JSR     PC,INIT1        ;INITIALIZE
1696  010434  012737  000010  000636  MOV     #10,RDYDX       ;SET LONG READY DELAY
1697  010442  004737  011546          JSR     PC,EXEC         ;ISSUE BUFFERED WRITE
1698  010446  000240                  NOP
1699  010450  012737  016141  000626  MOV     #MSG49,ERRP     ;SET ERROR CODE
1700  010456  012737  102300  000660  MOV     #102300,STMSK   ;MARK DATA ERROR
1701  010464  004737  011776          JSR     PC,ERCHK        ;GO CHECK ERROR
1702  010470  032777  000002  170024  BIT     #2,@DS          ;SEE IF BOT IS SET
1703  010476  001410                  BEQ     FT22X           ;IF NOT: BR
1704  010500  012737  016167  000650  MOV     #MSG50,ERADD    ;SET ERROR CODE
1705  010506  012737  010324  000674  MOV     #FT22,$COLP
1706  010514  004737  003774          JSR     PC,FT3ER        ;GO DO ERROR
1707  010520  004737  012546  FT22X:  JSR     PC,ITER         ;GO SEE IF ITERATION
1708  010524  000137  003022          JMP     TSCD2           ;RETURN TO SCHEDULAR
1709
1710
```

```
1711                                                    ;READ-IN PRESET TEST*****
1712
1713  010530  005737  000614          FT23:   TST     SLVN            ;SEE IF SLAVE SELECT=0
1714  010534  001103                  BNE     FT23X           ;IF NOT:BR
1715  010536  012737  017640  000610        MOV     #MSFT23,EMADDR  ;SET TEST HEADER
1716  010544  004737  012620              JSR     PC,INIT1        ;GO INIT
1717  010550  012737  001700  000716      MOV     #1700,UDES      ;SET TO NRZ NORMAL
1718  010556  012737  020112  000616      MOV     #WDATA,BADDR    ;SET BA=WRITE BUFFER
1719  010564  012737  177400  000620      MOV     #-400,FCNT      ;SET FC=400
1720  010572  012737  177600  000622      MOV     #-200,WCNT      ;SET WC=200
1721  010600  012737  000061  000710      MOV     #61,FUN         ;SET WRITE OP-CODE
1722  010606  004737  011546              JSR     PC,EXEC         ;GO DO WRITE
1723  010612  000240                      NOP
1724  010614  004737  012620              JSR     PC,INIT1        ;INITIALIZE
1725  010620  012737  000021  000710      MOV     #21,FUN         ;SET READ-IN PRESET OP CODE
1726  010626  004737  011546              JSR     PC,EXEC         ;GO DO COMMAND
1727  010632  005000                      CLR     R0
1728  010634  012703  000004              MOV     #4,R3           ;SET MULT
1729  010640  032777  020000  167654  FT23A:  BIT     #20000,@DS      ;SEE IF PIP RESET
1730  010646  001404                      BEQ     FT23B           ;IF SO: BR
1731  010650  005300                      DEC     R0
1732  010652  001372                      BNE     FT23A           ;AWAIT PIP RESET
1733  010654  005303                      DEC     R3
1734  010656  001370                      BNE     FT23A           ;DELAY
1735  010660  032777  000002  167634  FT23B:  BIT     #2,@DS          ;SEE IF BOT
1736  010666  001010                      BNE     FT23C           ;IF SO: BR
1737  010670  012737  016225  000650      MOV     #MSG51,ERADD    ;SET ERROR CODE
1738  010676  012737  010530  000674      MOV     #FT23,SCOLP
1739  010704  004737  003774              JSR     PC,FT3ER        ;GO DO ERROR
1740  010710  012701  141000          FT23C:  MOV     #141000,R1      ;SET EXPT TC
1741  010714  013700  000542              MOV     TC,R0           ;SET TC ADDRESS
1742  010720  020110                      CMP     R1,(R0)         ;SEE IF EXPT=RCVD
1743  010722  001410                      BEQ     FT23X           ;IF SO: BR
1744  010724  012737  016261  000650      MOV     #MSG52,ERADD    ;SET ERROR CODE
1745  010732  012737  010530  000674      MOV     #FT23,SCOLP     ;CLEAR SCOPE ADDRESS
1746  010740  004737  003504              JSR     PC,FT2ER        ;GO DO ERROR
1747  010744  000137  003022          FT23X:  JMP     TSCD2           ;RETURN TO SCHEDULAR
1748
1749
```

```
1750
1751                                    ;AUTO-DENSITY SELECT TEST: WRITE-NRZ,READ-PE
1752
1753  010750  012737  017727  000610  FT24:   MOV     #MSFT24,EMADDR  ;SET ERROR MSG HEADER
1754  010756  004737  011700            JSR     PC,RWND         ;REWIND SLAVE
1755  010762  012737  000001  000720    MOV     #1,PATRN        ;SELECT PATTERN
1756  010770  004737  012400            JSR     PC,DSUP         ;GO DO DATA SETUP
1757  010774  012737  020112  000616    MOV     #WDATA,BADDR    ;SET BUS ADDRESS,
1758  011002  012737  177400  000620    MOV     #-400,FCNT      ;FRAME COUNT,
1759  011010  012737  177600  000622    MOV     #-200,WCNT      ;WORD COUNT,
1760  011016  012737  001700  000716    MOV     #1700,UDES      ;& SLAVE DESC = NRZ NORMAL
1761  011024  012737  000061  000710    MOV     #61,FUN         ;LOAD OP CODE WRITE FWD
1762  011032  004737  011546            JSR     PC,EXEC         ;GO EXECUTE COMMAND
1763  011036  012737  016043  000626    MOV     #MSG46,ERRP     ;SET ERROR MSG ADDRESS
1764  011044  004737  011776            JSR     PC,ERCHK        ;GO CHECK ERRORS
1765  011050  005737  000712            TST     SERFL           ;BRANCH IF AN ERROR OCCURRED
1766  011054  001026                    BNE     FT24X
1767  011056  004737  011700            JSR     PC,RWND         ;REWIND SLAVE
1768  011062  012737  021624  000616    MOV     #RDATA,BADDR    ;SET BUS ADDRESS FOR READ
1769  011070  012737  002300  000716    MOV     #2300,UDES      ;SET SLAVE DESC = PE,NORMAL
1770  011076  012737  000071  000710    MOV     #71,FUN         ;SET OP CODE = READ FWD
1771  011104  004737  011546            JSR     PC,EXEC         ;GO READ RECORD
1772  011110  032777  000040  167404    BIT     #40,@DS         ;BRANCH ID PES BIT CLEARED
1773  011116  001405                    BEQ     FT24X
1774  011120  012737  016602  000650    MOV     #MSG63,ERADD
1775  011126  004737  003774            JSR     PC,FT3ER        ;GO PROCESS ERROR
1776  011132  004737  012546    FT24X:  JSR     PC,ITER
1777  011136  000137  003022            JMP     TSCD2           ;RETURN TO SCHEDULER
1778
```

```
1779
1780                                        ;AUTO-DENSITY SELECT TEST: WRITE-PE,READ-NRZ
1781  011142  012737  020005  000610  FT25: MOV   #MSFT25,EMADDR  ;SET ERROR MESSAGE ADDRESS
1782  011150  004737  011700          JSR   PC,RWND       ;REWIND SLAVE
1783  011154  012737  000001  000720  MOV   #1,PATRN      ;SELECT PATTERN
1784  011162  004737  012400          JSR   PC,DSUP       ;GO DO DATA SETUP
1785  011166  012737  020112  000616  MOV   #WDATA,BADDR  ;SET BUS ADDRESS
1786  011174  012737  177400  000620  MOV   #-400,FCNT    ;FRAME COUNT,
1787  011202  012737  177600  000622  MOV   #-200,WCNT    ;WORD COUNT,
1788  011210  012737  002300  000716  MOV   #2300,UDES    ;& SLAVE DESC = PE,NORMAL
1789  011216  012737  000061  000710  MOV   #61,FUN       ;LOAD WRITE OP CODE
1790  011224  004737  011546          JSR   PC,EXEC       ;GO EXECUTE WRITE
1791  011230  012737  016043  000626  MOV   #MSG46,ERRP    ;SET ERROR MSG ADR
1792  011236  004737  011776          JSR   PC,ERCHK      ;GO CHECK FOR ERRORS
1793  011242  005737  000712          TST   SERFL         ;BRANCH IF ERROR OCURRED
1794  011246  001026                  BNE   FT25X
1795  011250  004737  011700          JSR   PC,RWND       ;REWIND SLAVE
1796  011254  012737  021624  000616  MOV   #RDATA,BADDR  ;SET BUS ADDRESS FOR READ
1797  011262  012737  001700  000716  MOV   #1700,UDES    ;SET SLAVE DESC = NRZ,NORMAL
1798  011270  012737  000071  000710  MOV   #71,FUN       ;SET READ FWD OP CODE
1799  011276  004737  011546          JSR   PC,EXEC       ;GO EXECUTE
1800  011302  032777  000040  167212  BIT   #40,@DS       ;BRANCH ID PES BIT GOT SET
1801  011310  001005                  BNE   FT25X
1802  011312  012737  016633  000650  MOV   #MSG64,ERADD
1803  011320  004737  003774          JSR   PC,FT3ER      ;GO PROCESS ERROR
1804  011324  004737  012546  FT25X:  JSR   PC,ITER       ;ITERATION LOOP
1805  011330  000137  003022          JMP   TSC02         ;RETURN TO SCHEDULER
1806
1807                                        ;REWIND: OFF LINE TEST*****
1808
1809  011334  032777  010000  167206  FT26: BIT   #10000,@SWR    ;SEE IF IN CONTINUOUS MODE
1810  011342  001077                  BNE   FT26XX        ;IF SO: BR
1811  011344  005737  001662          TST   CHNFLG        ;BRANCH IF CHAIN MODE
1812  011350  001074                  BNE   FT26XX
1813  011352  012737  017673  000610  MOV   #MSFT26,EMADDR  ;SET TEST HEADER
1814  011360  004737  011700          JSR   PC,RWND       ;REWIND & SELECT SLAVE
1815  011364  012737  000001  000720  MOV   #1,PATRN      ;SELECT PATTERN (ALL 1'S)
1816  011372  004737  012400          JSR   PC,DSUP       ;FILL WRITE BUFFER
1817  011376  012737  020112  000616  MOV   #WDATA,BADDR  ;SET WRITE BUFFER BUS ADDRESS
1818  011404  012737  177400  000620  MOV   #-400,FCNT    ;SET FRAME COUNT
1819  011412  012737  177600  000622  MOV   #-200,WCNT    ;SET WORD COUNT
1820  011420  012737  001700  000716  MOV   #1700,UDES    ;SET UNIT DESCRIPTION = NRZ
1821  011426  012737  000061  000710  MOV   #61,FUN       ;SET WRITE COMMAND
1822  011434  004737  011546          JSR   PC,EXEC       ;GO WRITE A RECORD
1823  011440  012777  000003  167042  MOV   #3,@CI        ;ISSUE REWIND: OFF LINE COMMAND
1824  011446  005037  000674          CLR   SCOLP         ;CLEAR SCOPE LOOP
1825  011452  012700  004000          MOV   #4000,R0
1826  011456  005300          1$:     DEC   R0            ;DELAY
1827  011460  001376                  BNE   1$
1828  011462  032777  010000  167032  BIT   #10000,@DS    ;SEE IF MOL IS RESET
1829  011470  001406                  BEQ   2$            ;IF SO: BR
1830  011472  012737  016300  000650  MOV   #MSG53,ERADD   ;SET ERROR CODE
1831  011500  004737  003774          JSR   PC,FT3ER      ;GO DO ERROR
1832  011504  000412                  BR    FT26X
1833  011506  013700  000524    2$:   MOV   ER,R0         ;GET ADDRESS OF ERROR REG
1834  011512  005001                  CLR   R1            ;RESULT SHOULD BE 0
```

```
1835  011514  020110                        CMP     R1,(R0)         ;BRANCH IF ERROR REG = 0
1836  011516  001405                        BEQ     FT26X
1837  011520  012737  016667  000650        MOV     #MSG67,ERADD    ;SET ERROR MSG HEADER
1838  011526  004737  003504                JSR     PC,FT2ER        ;GO TYPE ERROR
1839  011532  012704  016325        FT26X:  MOV     #MSG54,R4
1840  011536  004737  013442                JSR     PC,TTOUT        ;PRINT ON LINE REQUEST
1841  011542  000137  003022        FT26XX: JMP     TSCD2           ;RETURN TO SCHEDULER
1842
```

```
1843                                                    ;COMMAND EXECUTE SUBROUTINE*****
1844
1845  011546  000240                          EXEC:    NOP
1846  011550  053777  000716  166764                   BIS     UDES,@TC        ;LOAD TAPE CONT
1847  011556  013777  000622  166726                   MOV     WCNT,@WC        ;LOAD WC
1848  011564  013777  000620  166724                   MOV     FCNT,@FC        ;LOAD FC
1849  011572  013777  000616  166714                   MOV     BADDR,@BA       ;LOAD BA
1850  011600  022737  000031  000710                   CMP     #31,FUN         ;SEE IF SPACE FORWARD
1851  011606  001404                                   BEQ     EXECA           ;IF SO: BR
1852  011610  022737  000033  000710                   CMP     #33,FUN         ;SEE IF SPACE REVERSE
1853  011616  001003                                   BNE     EXECB           ;IF NOT: BR
1854  011620  013777  000642  166670          EXECA:   MOV     SCNT,@FC        ;SET SPACE COUNT
1855  011626  000240                          EXECB:   NOP
1856  011630  013777  000710  166652                   MOV     FUN,@C1         ;LOAD OP-CODE + GO
1857  011636  000240                                   NOP
1858  011640  013703  000636                          MOV     RDYDX,R3        ;SET DELAY
1859  011644  005004                                   CLR     R4
1860  011646  032777  000200  166646          EXECC:   BIT     #200,@DS        ;SEE IF DRY
1861  011654  001004                                   BNE     EXECX           ;IF SO: BR
1862  011656  005304                                   DEC     R4
1863  011660  001372                                   BNE     EXECC
1864  011662  005303                                   DEC     R3              ;DELAY FOR DRY
1865  011664  001370                                   BNE     EXECC
1866  011666  013703  000640                 EXECX:   MOV     OPDYX,R3
1867  011672  005303                 EXECXA:  DEC     R3              ;DELAY
1868  011674  001376                                   BNE     EXECXA
1869  011676  000207                 EXECXX:  RTS     PC              ;RETURN TO CALLER
1870
```

```
1871                                                    ;REWIND SUBROUTINE*****
1872
1873  011700  000240                           RWND:   NOP
1874  011702  004737  012620                           JSR     PC,INIT1        ;INIT
1875  011706  012777  000007  166574                   MOV     #7,@C1          ;START REWIND
1876  011714  012700  040000                           MOV     #40000,R0
1877  011720  005300                           RWNDA:  DEC     R0
1878  011722  001376                                   BNE     RWNDA           ;DELAY
1879  011724  032777  020000  166570           RWNDB:  BIT     #20000,@DS
1880  011732  001374                                   BNE     RWNDB           ;AWAIT PIP
1881  011734  032777  000002  166560                   BIT     #2,@DS          ;SEE IF BOT
1882  011742  001012                                   BNE     RWNDX           ;IF SO: BR
1883  011744  013704  000610                           MOV     EMADDR,R4
1884  011750  004737  013442                           JSR     PC,TTOUT        ;PRINT HEADER
1885  011754  012704  014452                           MOV     #MSG2,R4
1886  011760  004737  013442                           JSR     PC,TTOUT        ;PRINT REWIND ERROR
1887  011764  000137  003022                           JMP     TSCD2           ;RETURN TO SECHEDULAR
1888  011770  004737  012620                   RWNDX:  JSR     PC,INIT1        ;INIT
1889  011774  000207                                   RTS     PC              ;RETURN TO CALLER
1890
```

```
1891                                                    ;ERROR CHECK SUBROUTINE*****
1892
1893  011776  005037  000712                ERCHK:  CLR     SERFL           ;CLEAR FLAG
1894  012002  017737  166514  000664                MOV     @DS,DSAV        ;SAVE DRIVE STATUS REGISTER
1895  012010  032777  040000  166504                BIT     #40000,@DS      ;SEE IF ERROR
1896  012016  001001                                BNE     ERPT            ;IF SO: BR
1897  012020  000207                                RTS     PC              ;RETURN
1898  012022  017704  166476                ERPT:   MOV     @ER,R4          ;GET ERROR REGISTER
1899  012026  032737  002000  000716                BIT     #2000,UDES      ;SEE IF PE
1900  012034  001403                                BEQ     ERPTA1          ;IF SO: BR
1901  012036  042737  000200  000660                BIC     #200,STMSK      ;RESET PEF MASK
1902  012044  043704  000660                ERPTA1: BIC     STMSK,R4        ;MASK DONT CARE BITS
1903  012050  001530                                BEQ     ERPTX           ;IF NO UNEXPECTED ERRORS: BR
1904  012052  012737  000001  000712        ERPTG:  MOV     #1,SERFL        ;SET FLAG
1905  012060  032777  020000  166462                BIT     #20000,@SWR     ;SEE IF SHOULD PRINT ERRORS
1906  012066  001115                                BNE     ERPTD           ;IF NOT: BR
1907  012070  005737  000606                        TST     HDRFL           ;SEE IF DONE HEADER
1908  012074  001006                                BNE     ERPTA           ;IF SO: BR
1909  012076  005237  000606                        INC     HDRFL           ;SET HEADER FLAG
1910  012102  013704  000610                        MOV     EMADDR,R4
1911  012106  004737  013442                        JSR     PC,TTOUT        ;PRINT HEADER
1912  012112  013704  000626                ERPTA:  MOV     ERRP,R4         ;GET ERROR CODE
1913  012116  001414                                BEQ     ERPTB           ;IF NONE: BR
1914  012120  004737  013442                        JSR     PC,TTOUT        ;PRINT ERROR CODE
1915  012124  012704  015136                        MOV     #MSG20,R4       ;SET NRZ TAG
1916  012130  032777  002000  166404                BIT     #2000,@TC       ;SEE IF PE
1917  012136  001402                                BEQ     ERPT1A          ;IF NOT: BR
1918  012140  012704  015144                        MOV     #MSG21,R4       ;ELSE SET PE TAG
1919  012144  004737  013442                ERPT1A: JSR     PC,TTOUT        ;PRINT TAG
1920  012150  013704  000630                ERPTB:  MOV     ERRP1,R4        ;SEE IF CODE 2
1921  012154  001402                                BEQ     ERPTB1          ;IF NOT: BR
1922  012156  004737  013442                        JSR     PC,TTOUT        ;PRINT CODE 2
1923  012162  032777  004000  166360        ERPTB1: BIT     #4000,@SWR      ;SEE IF ITERATION
1924  012170  001010                                BNE     ERPTC           ;IF NOT: BR
1925  012172  012704  016457                        MOV     #MSG56,R4
1926  012176  004737  013442                        JSR     PC,TTOUT        ;PRINT ITER TAG
1927  012202  013703  000662                        MOV     ITCNT,R3
1928  012206  004737  013572                        JSR     PC,OCTP         ;PRINT ITERATION
1929  012212  012704  014364                ERPTC:  MOV     #MSG1,R4
1930  012216  004737  013442                        JSR     PC,TTOUT        ;PRINT REGISTER TAG
1931  012222  017703  166262                        MOV     @CS,R3
1932  012226  004737  013560                        JSR     PC,OCTPE        ;PRINT CS1
1933  012232  017703  166254                        MOV     @WC,R3
1934  012236  004737  013560                        JSR     PC,OCTPE        ;PRINT WC
1935  012242  017703  166246                        MOV     @BA,R3
1936  012246  004737  013560                        JSR     PC,OCTPE        ;PRINT BA
1937  012252  017703  166240                        MOV     @FC,R3
1938  012256  004737  013560                        JSR     PC,OCTPE        ;PRINT FC
1939  012262  017703  166232                        MOV     @CS,R3
1940  012266  004737  013560                        JSR     PC,OCTPE        ;PRINT CS2
1941  012272  017703  166224                        MOV     @DS,R3
1942  012276  004737  013560                        JSR     PC,OCTPE        ;PRINT DS
1943  012302  017703  166216                        MOV     @ER,R3
1944  012306  004737  013560                        JSR     PC,OCTPE        ;PRINT ER
1945  012312  017703  166224                        MOV     @TC,R3
1946  012316  004737  013560                        JSR     PC,OCTPE        ;PRINT TC
```

```
1947  012322  005777  166222          ERPTD:  TST     @SWR            ;SEE IF HALT ON ERROR
1948  012326  100001                          BPL     ERPTX           ;IF NOT: BR
1949  012330  000000                          HALT
1950  012332  004737  012620          ERPTX:  JSR     PC,INIT1        ;INIT
1951  012336  000207                  ERPTXX: RTS     PC              ;RETURN
1952
1953
```

```
                                                ;TAPE MARK STATUS CHECK**********
                                                
012340  032737  000004  000664  TMCHK:  BIT     #4,DSAV         ;SEE IF TM SET
012346  001401                          BEQ     TMCHK1          ;IF NOT: BR
012350  000207                  TMCHK0: RTS     PC              ;ELSE RETURN
012352  005737  000712          TMCHK1: TST     SERFL           ;SEE IF HAD ERROR
012356  001374                          BNE     TMCHK0          ;IF SO: BR
012360  012737  016467  000630          MOV     #MSG57,ERRP1    ;SET ERROR CODE 2
012366  004737  012052                  JSR     PC,ERPTG        ;GO PRINT TM ERROR
012372  005037  000630                  CLR     ERRP1           ;CLEAR CODE 2 FLAG
012376  000207                          RTS     PC              ;RETURN
                                                
                                                ;DATA SETUP ROUTINE**********
                                                
012400  000240                  DSUP:   NOP
012402  012703  020112          DSO:    MOV     #WDATA,R3       ;R3 = ADDRS OF WRITE BUFFER
012406  013701  000720                  MOV     PATRN,R1        ;R1 = PATTERN SELECTOR
012412  006301                          ASL     R1              ;MAKE PATTERN SELECTOR EVEN
012414  004771  000740                  JSR     PC,@DATBL(R1)   ;GO GENERATE PATTERN
012420  012702  000640                  MOV     #640,R2         ;R2=BUFFER SIZE +2
012424  012701  021624                  MOV     #RDATA,R1       ;R1=READ DATA START
012430  005021                  1$:     CLR     (R1)+           ;CLEAR BUFFER
012432  005302                          DEC     R2              ;SEE IF DONE ALL
012434  001375                          BNE     1$              ;IF NOT: BR
012436  000207                          RTS     PC              ;EXIT
                                                
                                                ;ALL ONES**********
                                                
012440  012701  177777          DAT1:   MOV     #-1,R1          ;R1=DATA
012444  012702  000640          DAT1A:  MOV     #640,R2         ;R2=WORD COUNT +2
012450  010123                  1$:     MOV     R1,(R3)+        ;LOAD BUFFER
012452  005302                          DEC     R2              ;SEE IF DONE
012454  001375                          BNE     1$              ;IF NOT: BR
012456  000207                          RTS     PC
                                                
                                                ;ALL ZEROS**********
                                                
012460  005001                  DAT2:   CLR     R1              ;R1=DATA
012462  000770                          BR      DAT1A           ;LOAD BUFFER
                                                
                                                ;ONE/ZERO IN ALTERNATING CHARACTERS**********
                                                
012464  012701  125125          DAT3:   MOV     #125125,R1      ;R1=DATA
012470  000765                          BR      DAT1A           ;LOAD BUFFER
                                                
                                                ;ALL BITS 0-377**********
                                                
012472  005001                  DAT4:   CLR     R1              ;R1=STARTING DATA
012474  012702  001500                  MOV     #1500,R2        ;R2=CHARACTER COUNT
012500  110123                  1$:     MOVB    R1,(R3)+        ;LOAD BUFFER
012502  105201                          INCB    R1              ;BUMP DATA
012504  005302                          DEC     R2              ;SEE IF DONE
012506  001374                          BNE     1$              ;IF NOT: BR
012510  000207                          RTS     PC
```

```
2009
2010
2011
2012                                        ;SCOPE LOOP ON ERROR SUBROUTINE**********
2013
2014  012512  000240                SCOPE:  NOP
2015  012514  032777  040000  166026        BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
2016  012522  001001                        BNE     1$              ;IF SO: BR
2017  012524  000207                        RTS     PC              ;ELSE EXIT
2018  012526  000240                1$:     NOP
2019  012530  005737  000674                TST     @SCOLP          ;SEE IF SCOPE ADDRESS
2020  012534  001001                        BNE     2$              ;IF NOT: BR
2021  012536  000207                        RTS     PC              ;ELSE EXIT
2022  012540  022626                2$:     CMP     (SP)+,(SP)+     ;RESET STACK
2023  012542  000177  166126                JMP     @SCOLP          ;LOOP ON ERROR
2024
2025                                        ;TEST ITERATION SUBROUTINE**********
2026
2027  012546  000240                ITER:   NOP
2028  012550  032777  004000  165772        BIT     #4000,@SWR      ;SEE IF ITERATIONS
2029  012556  001403                        BEQ     2$              ;IF SO: BR
2030  012560  005037  000662        1$:     CLR     ITCNT           ;CLEAR ITERATION COUNTER
2031  012564  000207                        RTS     PC              ;ELSE EXIT
2032  012566  005737  000730        2$:     TST     PCNTR           ;DO SINGLE SUBTEST ITERATION
2033  012572  001772                        BEQ     1$              ;ON FIRST PASS
2034  012574  005237  000662                INC     ITCNT           ;BUMP COUNTER
2035  012600  023737  000662  000566        CMP     ITCNT,ITAMT     ;SEE IF DONE ALL
2036  012606  001764                        BEQ     1$              ;IF SO: BR
2037  012610  005726                        TST     (SP)+           ;RESET STACK
2038  012612  017700  166060                MOV     @ITRLP,R0       ;SET ITERATION POINTER
2039  012616  000110                        JMP     (R0)            ;GO ITERATE
2040
2041                                        ;INITIALIZE SUBROUTINE**********
2042
2043  012620  000240                INIT1:  NOP
2044  012622  012777  000040  165670        MOV     #40,@CS         ;INIT
2045  012630  013777  000612  165662INIT2:  MOV     DRVN,@CS        ;SELECT DRIVE
2046  012636  013777  000614  165676        MOV     SLVN,@TC        ;SELECT SLAVE
2047  012644  000207                        RTS     PC              ;RETURN
2048
```

```
2049                                                    ;MAG TAPE INTERRUPT HANDLER**********
2050
2051  012646  000240                            MTINT:  NOP
2052  012650  013716  000646                    MOV     RTRN,(SP)       ;RETURN TO (RTRN)
2053  012654  000002                            RTI                     ;RETURN
2054
2055                                                    ;TTY INTERRUPT HANDLER**********
2056
2057  012656  017746  165672                    TTINT:  MOV     @TKB,-(SP)      ;GET CHARACTER
2058  012662  042716  000200                    BIC     #200,(SP)       ;CLEAR PARITY BIT
2059  012666  122716  000003                    CMPB    #3,(SP)         ;BRANCH IF NOT CONTROL C
2060  012672  001010                            BNE     1$
2061  012674  005737  001662                    TST     CHNFLG          ;INHIBIT ↑C IF CHAIN MODE
2062  012700  001005                            BNE     1$
2063  012702  005077  165640                    CLR     @PSW
2064  012706  000005                            RESET
2065  012710  000137  000200                    JMP     @#200           ;RESTART PROGRAM
2066  012714  122716  000001            1$:     CMPB    #1,(SP)         ;BRANCH IF NOT ↑A
2067  012720  001017                            BNE     2$
2068  012722  022737  000176  000550            CMP     #SWREG,SWR      ;BRANCH IF HARDWARE SWR IS INVOKED
2069  012730  001016                            BNE     3$
2070  012732  012737  177570  000550            MOV     #177570,SWR     ;INVOKE HARDWARE SWR
2071  012740  004737  014320                    JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
2072  012744  012704  016743                    MOV     #MSG70,R4       ;TYPE 'HARDWARE SWR IN USE'
2073  012750  004737  013442                    JSR     PC,TTOUT
2074  012754  004737  014342                    JSR     PC,.RESTORE
2075  012760  122716  000007            2$:     CMPB    #7,(SP)         ;BRANCH IF NOT ↑G
2076  012764  001005                            BNE     4$
2077  012766  012737  000176  000550    3$:     MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
2078  012774  004737  014222                    JSR     PC,GTSWR        ;GET SOFTWARE SWITCHES
2079  013000  005726                    4$:     TST     (SP)+           ;POP CHARACTER OFF THE STACK
2080  013002  000002                            RTI
2081
2082                                                    ;BUS ADDRESS TRAP HANDLER**********
2083
2084  013004  000240                    TRAP:   NOP
2085  013006  032777  020000  165534            BIT     #20000,@SWR     ;SEE IF SHOULD PRINT ERRORS
2086  013014  001020                            BNE     TRAP2           ;IF NOT: BR
2087  013016  005737  000606                    TST     HDRFL           ;SEE IF DONE HEADER
2088  013022  001006                            BNE     TRAP1           ;IF SO: BR
2089  013024  005237  000606                    INC     HDRFL           ;ELSE SET HEADER FLAG
2090  013030  013704  000610                    MOV     EMADDR,R4
2091  013034  004737  013442                    JSR     PC,TTOUT        ;PRINT HEADER
2092  013040  012704  015171            TRAP1:  MOV     #MSG24,R4
2093  013044  004737  013442                    JSR     PC,TTOUT        ;PRINT ERROR
2094  013050  010103                            MOV     R1,R3           ;GET ADDRESS THAT CAUSED THE TRAP
2095  013052  004737  013572                    JSR     PC,OCTP         ;PRINT ADDRESS OF TRAP
2096  013056  005777  165466            TRAP2:  TST     @SWR            ;SEE IF HALT ON ERROR
2097  013062  100001                            BPL     TRAPX           ;IF NOT: BR
2098  013064  000000                            HALT
2099  013066  022626                    TRAPX:  CMP     (SP)+,(SP)+     ;RESET STACK
2100  013070  012737  003250  000674            MOV     #FT1A,SCOLP     ;SET SCOPE ADDRESS
2101  013076  004737  012512                    JSR     PC,SCOPE        ;GO SEE IF SCOPE LOOP
2102  013102  005737  000722                    TST     RHTF            ;SEE IF INITIAL ADDRESS TEST
2103  013106  001402                            BEQ     TRAPXX          ;IF NOT: BR
2104  013110  000137  001766                    JMP     STOB            ;ELSE REDO ADDRESS REQUEST
```

```
 2105  013114  000137  003254          TRAPXX: JMP     FT1B            ;RETURN TO TEST 1
 2106
```

```
2107                                    ;*************************************************************
2108                                    ;TTY ENTRY SUBROUTINE:
2109                                    ;
2110                                    ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
2111                                    ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
2112                                    ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
2113                                    ;LIMITS.  ALL RESPONSE MUST BE TYPED IN OCTAL
2114                                    ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
2115                                    ;THE CALLING ROUTINE.
2116                                    ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
2117                                    ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
2118                                    ;MAY BE REENTERED.
2119                                    ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
2120                                    ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
2121                                    ;CARRIAGE RETURN
2122                                    ;*************************************************************
2123
2124  013120  010146                TTR:  MOV     R1,-(SP)          ;SAVE CHAR COUNT ON STACK
2125  013122  011601                10$:  MOV     (SP),R1           ;RESTORE CHAR COUNT (FOR ↑U)
2126  013124  005037  000652              CLR     TEMP1             ;CLEAR FIRST CHARACTER FLAG
2127  013130  005000                       CLR     R0
2128  013132  004737  013400        1$:   JSR     PC,TTIN           ;GO READ CHARACTER
2129  013136  122737  000003  000602      CMPB    #3,TIB            ;BRANCH IF NOT ↑C
2130  013144  001003                       BNE     11$
2131  013146  000005                       RESET                     ;RESET
2132  013150  000137  000200              JMP     @#200             ;RESTART
2133  013154  122737  000015  000602  11$:  CMPB    #15,TIB           ;SEE IF CR
2134  013162  001004                       BNE     2$                ;IF NOT: BR
2135  013164  005737  000652              TST     TEMP1             ;SEE IF FIRST CHARACTER
2136  013170  001471                       BEQ     9$                ;IF SO: BR
2137  013172  000457                       BR      6$                ;ELSE GO LOAD VALUE
2138  013174  122737  000025  000602  2$:   CMPB    #25,TIB           ;BRANCH IF NOT CONTROL U
2139  013202  001005                       BNE     21$
2140  013204  012704  016663              MOV     #MSG65,R4         ;TYPE <CR><LF>
2141  013210  004737  013442              JSR     PC,TTOUT
2142  013214  000742                       BR      10$               ;RESTART
2143  013216  122737  000177  000602  21$:  CMPB    #177,TIB          ;BRANCH IF NOT 'RUBOUT'
2144  013224  001012                       BNE     3$
2145  013226  000241                       CLC                       ;REMOVE LAST CHARACTER
2146  013230  006000                       ROR     R0
2147  013232  006200                       ASR     R0
2148  013234  006200                       ASR     R0
2149  013236  012704  016665              MOV     #MSG66,R4         ;TYPE '\'
2150  013242  004737  013442              JSR     PC,TTOUT
2151  013246  005201                       INC     R1                ;DECREMENT CHAR RECEIVED COUNT
2152  013250  000730                       BR      1$                ;GET NEXT CHARACTER
2153  013252  122737  000060  000602  3$:   CMPB    #60,TIB           ;SEE IF CHAR IS LESS THAN 0
2154  013260  101402                       BLOS    4$                ;IF NOT: BR
2155  013262  000137  013360              JMP     TIMER             ;ELSE GO TO ERROR
2156  013266  122737  000070  000602  4$:   CMPB    #70,TIB           ;SEE IF CHAR IS GREATER THAN 7
2157  013274  101002                       BHI     5$                ;IF NOT: BR
2158  013276  000137  013360              JMP     TIMER             ;ELSE GO TO ERROR
2159  013302  005237  000652        5$:   INC     TEMP1             ;SET FIRST CHARACTER FLAG
2160  013306  006300                       ASL     R0
2161  013310  006300                       ASL     R0                ;SHIFT 3 LEFT
2162  013312  006300                       ASL     R0
```

```
2163  013314  042737  177770  000602          BIC     #177770,TIB     ;STRIP ASCII
2164  013322  053700  000602                  BIS     TIB,R0          ;LOAD CHARACTER
2165  013326  005301                          DEC     R1              ;SEE IF DONE
2166  013330  001300                          BNE     1$              ;IF NOT: BR
2167  013332  020002                  6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
2168  013334  101402                          BLOS    7$              ;IF NOT: BR
2169  013336  000137  013360                  JMP     TINER           ;ELSE GO TO ERROR
2170  013342  020300                  7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
2171  013344  101402                          BLOS    8$              ;IF NOT: BR
2172  013346  000137  013360                  JMP     TINER           ;ELSE GO TO ERROR
2173  013352  010015                  8$:     MOV     R0,(R5)         ;LOAD VALUE
2174  013354  005726                  9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
2175  013356  000207                          RTS     PC              ;EXIT
2176
2177                                          ;TTY ENTRY ERROR SUBROUTINE**********
2178
2179  013360  012704  014677          TINER:  MOV     #MSG7,R4
2180  013364  004737  013442                  JSR     PC,TTOUT        ;PRINT?
2181  013370  005726                          TST     (SP)+           ;POP CHAR COUNT OFF STACK
2182  013372  162716  000020                  SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
2183  013376  000207                          RTS     PC              ;REDO VALUE ENTRY
2184
2185                                          ;TTY READ SUBROUTINE**********
2186
2187  013400  005277  165146          TTIN:   INC     @TKS
2188  013404  105777  165142          1$:     TSTB    @TKS
2189  013410  100375                          BPL     1$
2190  013412  117737  165136  000602          MOVB    @TKB,TIB
2191  013420  042737  000200  000602          BIC     #200,TIB        ;STRIP PARITY BIT
2192  013426  013737  000602  000600          MOV     TIB,TOB         ;MOVE CHAR TO OUTPUT BFR
2193  013434  004737  013542                  JSR     PC,TOG          ;AND TYPE IT
2194  013440  000207                          RTS     PC
2195
2196                                          ;TTY OUTPUT SUBROUTINE************
2197
2198  013442  112437  000600          TTOUT:  MOVB    (R4)+,TOB
2199  013446  122737  000043  000600          CMPB    #43,TOB
2200  013454  001440                          BEQ     TEX
2201  013456  122737  000045  000600          CMPB    #45,TOB
2202  013464  001403                          BEQ     1$
2203  013466  004737  013542                  JSR     PC,TOG
2204  013472  000763                          BR      TTOUT
2205  013474  112737  000015  000600  1$:     MOVB    #15,TOB
2206  013502  004737  013542                  JSR     PC,TOG
2207  013506  012703  000004                  MOV     #4,R3
2208  013512  005037  000600          2$:     CLR     TOB
2209  013516  004737  013542                  JSR     PC,TOG
2210  013522  005303                          DEC     R3
2211  013524  001372                          BNE     2$              ;DO FILLERS
2212  013526  112737  000012  000600          MOVB    #12,TOB
2213  013534  004737  013542                  JSR     PC,TOG
2214  013540  000740                          BR      TTOUT
2215  013542  105777  165010          TOG:    TSTB    @TPS
2216  013546  100375                          BPL     TOG
2217  013550  113777  000600  165002          MOVB    TOB,@TPB
2218  013556  000207                  TEX:    RTS     PC
```

```
2219
2220
```

```
                                                 ;OCTAL OUTPUT SUBROUTINE***********

013560  012737  000001  014010  OCTPE:   MOV     #1,OFL
013566  010304                           MOV     R3,R4
013570  000410                           BR      OCTPO
013572  005037  014010          OCTP:    CLR     OFL              ;CLEAR FLAG FOR LEADING ZERO
013576  010304                  OCTPE1:  MOV     R3,R4            ;SEE IF NUMBER IS ZERO
013600  001004                           BNE     OCTPO            ;IF NOT ZERO: BR
013602  004737  013770                   JSR     PC,OCTPG1        ;ELSE PRINT ZERO
013606  000137  013732                   JMP     OCTP3            ;SPACE AND EXIT
013612  032704  100000          OCTPO:   BIT     #100000,R4       ;SEE IF MSD = 1
013616  001406                           BEQ     OCTP1            ;IF NOT: BR
013620  012704  000001                   MOV     #1,R4
013624  004737  013746                   JSR     PC,OCTPG         ;PRINT 1
013630  000137  013642                   JMP     OCTP2
013634  005004                  OCTP1:   CLR     R4
013636  004737  013746                   JSR     PC,OCTPG         ;PRINT 0
013642  010304                  OCTP2:   MOV     R3,R4
013644  006004                           ROR     R4
013646  006004                           ROR     R4
013650  006004                           ROR     R4               ;POSITION DIGIT
013652  006004                           ROR     R4
013654  000304                           SWAB    R4
013656  004737  013746                   JSR     PC,OCTPG         ;PRINT DIGIT 2
013662  010304                           MOV     R3,R4
013664  006004                           ROR     R4
013666  000304                           SWAB    R4
013670  004737  013746                   JSR     PC,OCTPG         ;PRINT DIGIT 3
013674  010304                           MOV     R3,R4
013676  006104                           ROL     R4
013700  006104                           ROL     R4
013702  000304                           SWAB    R4
013704  004737  013746                   JSR     PC,OCTPG         ;PRINT DIGIT 4
013710  010304                           MOV     R3,R4
013712  006004                           ROR     R4
013714  006004                           ROR     R4
013716  006004                           ROR     R4
013720  004737  013746                   JSR     PC,OCTPG
013724  010304                           MOV     R3,R4
013726  004737  013746                   JSR     PC,OCTPG         ;PRINT DIGIT 5
013732  012737  000240  000600  OCTP3:   MOV     #240,TOB
013740  004737  013542                   JSR     PC,TOG           ;PRINT SPACE
013744  000207                           RTS     PC               ;EXIT
013746  042704  177770          OCTPG:   BIC     #177770,R4
013752  001004                           BNE     OCTPGO
013754  005737  014010                   TST     OFL
013760  001001                           BNE     OCTPGO
013762  000207                           RTS     PC

013764  005237  014010          OCTPGO:  INC     OFL
013770  052704  000260          OCTPG1:  BIS     #260,R4
013774  010437  000600                   MOV     R4,TOB
014000  004737  013542                   JSR     PC,TOG
014004  010304                           MOV     R3,R4
014006  000207                           RTS     PC
014010  000000                  OFL:     0                        ;FIRST CHAR FLAG
```

```
                                                ;DATA CHARACTER OUTPUT SUBROUTINE***********
014012  005037  000600          DOUT:   CLR     TOB
014016  012704  000010                  MOV     #10,R4          ;SET NUMBER TO PRINT
014022  110337  000600                  MOVB    R3,TOB
014026  105777  164524          1$:     TSTB    @TPS
014032  100375                          BPL     1$
014034  132737  000200  000600        BITB    #200,TOB
014042  001404                          BEQ     2$
014044  012777  000061  164506        MOV     #061,@TPB
014052  000403                          BR      3$
014054  012777  000060  164476  2$:     MOV     #060,@TPB
014062  006137  000600          3$:     ROL     TOB
014066  005304                          DEC     R4
014070  001356                          BNE     1$
014072  000207                          RTS     PC

014074  013703  000656          DOUTD:  MOV     TEMP3,R3
014100  000303                          SWAB    R3
014102  004737  014012                  JSR     PC,DOUT
014106  013703  000656                  MOV     TEMP3,R3
014112  004737  014012                  JSR     PC,DOUT
014116  000207                          RTS     PC

                                                ;TE16 SERIAL NUMBER PRINT SUBROUTINE**********
014120  010304          SNPT:   MOV     R3,R4
014122  000304                  SWAB    R4
014124  006004                  ROR     R4
014126  006004                  ROR     R4
014130  006004                  ROR     R4
014132  006004                  ROR     R4              ;GET FIRST DIGIT
014134  004737  014176          JSR     PC,SNPG         ;GO PRINT
014140  010304                  MOV     R3,R4
014142  000304                  SWAB    R4              ;GET SECOND DIGIT
014144  004737  014176          JSR     PC,SNPG         ;GO PRINT
014150  010304                  MOV     R3,R4
014152  006004                  ROR     R4
014154  006004                  ROR     R4
014156  006004                  ROR     R4
014160  006004                  ROR     R4              ;GET THIRD DIGIT
014162  004737  014176          JSR     PC,SNPG         ;GO PRINT
014166  010304                  MOV     R3,R4           ;GET FOURTH DIGIT
014170  004737  014176          JSR     PC,SNPG         ;GO PRINT
014174  000207                  RTS     PC              ;EXIT
014176  012737  000260  000600  SNPG:   MOV     #260,TOB        ;SET BASE = 0
014204  042704  177760          BIC     #177760,R4      ;MASK DIGIT
014210  050437  000600          BIS     R4,TOB          ;SET ASCII
014214  004737  013542          JSR     PC,TOG          ;TYPE DIGIT
014220  000207                  RTS     PC              ;RETURN
```

```
2329
2330                                    ;ROUTINE TO LOAD NEW VALUE INTO SWITCHES
2331  014222  022737  000176  000550  GTSWR:  CMP     @#SWREG,SWR     ;BRANCH IF SOFTWARE SWR
2332  014230  001032                          BNE     1$              ;NOT INVOKED
2333  014232  004737  014320                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
2334  014236  012704  020067                  MOV     #$MSWR,R4
2335  014242  004737  013442                  JSR     PC,TTOUT
2336  014246  017703  164276                  MOV     @SWR,R3
2337  014252  004737  013560                  JSR     PC,OCTPE
2338  014256  012704  020076                  MOV     #$MNEW,R4
2339  014262  004737  013442                  JSR     PC,TTOUT
2340  014266  013705  000550                  MOV     SWR,R5          ;TTR ROUTINE RETURNS NEW VALUE TO (R5)
2341  014272  012701  000007                  MOV     #7,R1           ;LIMIT RESPONSE TO 7 CHARS
2342  014276  012702  177777                  MOV     #177777,R2      ;BETWEEN 0 AND 177777
2343  014302  012703  000000                  MOV     #0,R3
2344  014306  004737  013120                  JSR     PC,TTR
2345  014312  004737  014342                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
2346  014316  000207                  1$:     RTS     PC
2347
2348                                    ;;ROUTINE TO SAVE REGISTERS ON THE STACK
2349  014320  010546                  .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
2350  014322  010446                          MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
2351  014324  010346                          MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
2352  014326  010246                          MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
2353  014330  010146                          MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
2354  014332  010046                          MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
2355  014334  016646  000014                  MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
2356  014340  000207                          RTS     PC              ;;RETURN TO CALLER
2357
2358                                    ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
2359  014342  012666  000014          .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
2360  014346  012600                          MOV     (SP)+,%0
2361  014350  012601                          MOV     (SP)+,%1
2362  014352  012602                          MOV     (SP)+,%2
2363  014354  012603                          MOV     (SP)+,%3
2364  014356  012604                          MOV     (SP)+,%4
2365  014360  012605                          MOV     (SP)+,%5
2366  014362  000207                          RTS     PC              ;;RETURN
2367
2368
```

```
2369                                                    ;MESSAGE TABLE*****
2370
2371  014364  041445  030523  020040  MSG1:   .ASCII  /%CS1    WC     BA     FC     CS2    /
2372  014372  020040  041527  020040
2373  014400  020040  041040  020101
2374  014406  020040  020040  041506
2375  014414  020040  020040  041440
2376  014422  031123  020040  020040
2377  014430  051504  020040  020040          .ASCII  /DS     ER     TC%#/
2378  014436  042440  020122  020040
2379  014444  020040  041524  021445
2380  014452  051105  053505  047111  MSG2:   .ASCII  /%REWIND ERROR#/
2381  014460  020104  051105  047522
2382  014466  021522
2383  014470  020440  046524  031460  MSG3:   .ASCII  /%%TM03-TE16 BASIC FUNCTION TEST (DZTEC-A)%/
2384  014476  052055  030505  020066
2385  014504  040503  044523  020103
2386  014512  052506  041516  044524
2387  014520  047117  052040  051505
2388  014526  020124  042050  052132
2389  014534  041505  040455  022451
2390  014542  054524  042520  036040          .ASCII  /TYPE <CR> TO TERMINATE RESPONSE & ↑C TO RESTART%#/
2391  014550  051103  020076  047524
2392  014556  052040  051105  044515
2393  014564  040516  042524  051040
2394  014572  051505  047520  051516
2395  014600  020105  020046  041536
2396  014606  052040  020117  042522
2397  014614  052123  051101  022524
2398  014622      043
2399  014623      045 042522  044507  MSG4:   .ASCII  /%REGISTER START = #/
2400  014630  052123  051105  051440
2401  014636  040524  052122  036440
2402  014644  021440
2403  014646  053045  041505  047524  MSG5:   .ASCII  /%VECTOR = #/
2404  014654  020122  020075      043
2405  014661      045 047105  020104  MSG6:   .ASCII  /%END OF PASS #/
2406  014666  043117  050040  051501
2407  014674  020123      043
2408  014677      040 020077      043 MSG7:   .ASCII  / ? #/
2409  014703      045 047520  044523  MSG9:   .ASCII  /%POSITION ERROR: #/
2410  014710  044524  047117  042440
2411  014716  051122  051117  020072
2412  014724      043
2413  014725      045 051104  053111  MSG10:  .ASCII  /%DRIVE NUMBER: #/
2414  014732  020105  052516  041115
2415  014740  051105  020072      043
2416  014745      045 046123  053101  MSG11:  .ASCII  /%SLAVE NUMBER: #/
2417  014752  020105  052516  041115
2418  014760  051105  020072      043
2419  014765      045 051127  052111  MSG12:  .ASCII  /%WRITE ERROR #/
2420  014772  020105  051105  047522
2421  015000  020122      043
2422  015003      045 042522  042101  MSG13:  .ASCII  /%READ REVERSE ERROR #/
2423  015010  051040  053105  051105
2424  015016  042523  042440  051122
```

```
2425  015024  051117  021440
2426  015030  051045  040505  020104  MSG14:  .ASCII  /%READ FORWARD ERROR #/
2427  015036  047506  053522  051101
2428  015044  020104  051105  047522
2429  015052  020122     043
2430  015055     045  051127  052111  MSG15:  .ASCII  /%WRITE TM ERROR #/
2431  015062  020105  046524  042440
2432  015070  051122  051117  021440
2433  015076  051045  053105  051105  MSG16:  .ASCII  /%REVERSE ERROR #/
2434  015104  042523  042440  051122
2435  015112  051117  021440
2436  015116  043045  051117  040527  MSG17:  .ASCII  /%FORWARD ERROR #/
2437  015124  042122  042440  051122
2438  015132  051117  021440
2439  015136  047040  055122  021440  MSG20:  .ASCII  / NRZ #/
2440  015144  050040  020105     043  MSG21:  .ASCII  / PE #/
2441  015151     040  054105  052120  MSG22:  .ASCII  / EXPT: #/
2442  015156  020072     043
2443  015161     040  041522  042126  MSG23:  .ASCII  / RCVD: #/
2444  015166  020072     043
2445  015171     045  052502  020123  MSG24:  .ASCII  /%BUS TRAP: #/
2446  015176  051124  050101  020072
2447  015204     043
2448  015205     045  041527  020072  MSG25:  .ASCII  /%WC: #/
2449  015212     043
2450  015213     045  040502  020072  MSG26:  .ASCII  /%BA: #/
2451  015220     043
2452  015221     045  041104  020072  MSG27:  .ASCII  /%DB: #/
2453  015226     043
2454  015227     045  047111  052111  MSG28:  .ASCII  /%INIT DID NOT CLEAR RH #/
2455  015234  042040  042111  047040
2456  015242  052117  041440  042514
2457  015250  051101  051040  020110
2458  015256     043
2459  015257     045  041523  047040  MSG29:  .ASCII  /%SC NOT RESET BY INIT #/
2460  015264  052117  051040  051505
2461  015272  052105  041040  020131
2462  015300  047111  052111  021440
2463  015306  052045  042522  047040  MSG30:  .ASCII  /%TRE NOT RESET BY INIT #/
2464  015314  052117  051040  051505
2465  015322  052105  041040  020131
2466  015330  047111  052111  021440
2467  015336  041445  031123  047040  MSG31:  .ASCII  /%CS2 NOT RESET BY INIT #/
2468  015344  052117  051040  051505
2469  015352  052105  041040  020131
2470  015360  047111  052111  021440
2471  015366  042045  052114  047040  MSG32:  .ASCII  /%DLT NOT SET #/
2472  015374  052117  051440  052105
2473  015402  021440
2474  015404  051445  020103  047516  MSG33:  .ASCII  /%SC NOT SET #/
2475  015412  020124  042523  020124
2476  015420     043
2477  015421     045  051124  020105  MSG34:  .ASCII  /%TRE NOT SET #/
2478  015426  047516  020124  042523
2479  015434  020124     043
2480  015437     045  051111  047040  MSG35:  .ASCII  /%IR NOT SET BY INIT #/
```

```
2481  015444  052117  051440  052105
2482  015452  041040  020131  047111
2483  015460  052111  021440
2484  015466  047516  020122  047516  MSG36:  .ASCII  /%OR NOT RESET BY INIT #/
2485  015474  020124  042523  042523
2486  015502  020124  054502  044440
2487  015510  044516  020124      043
2488  015516      045  051117  047040  MSG37:  .ASCII  /%OR NOT RESET BY 1 SILO ENTRY #/
2489  015524  052111  051040  051505
2490  015532  021105  041040  020131
2491  015540  020124  044523  047514
2492  015546  047440  052116  054522
2493  015554  021440
2494  015560  047516  020122  047516  MSG38:  .ASCII  /%OR NOT SET BY SILO FULL #/
2495  015566  020124  042523  020124
2496  015574  054502  051440  046111
2497  015602  020117  052506  046114
2498  015610  021440
2499  015614  041045  042101  051440  MSG39:  .ASCII  /%BAD SILO READ #/
2500  015622  046111  020117  042522
2501  015630  042101  021440
2502  015636  044445  020122  047516  MSG40:  .ASCII  /%IR NOT RESET BY SILO FULL#/
2503  015644  020124  042522  042523
2504  015652  020124  054502  051440
2505  015660  046111  020117  052506
2506  015666  046114      043
2507  015672      045  047516  026516  MSG41:  .ASCII  /%NON-EXIST DRIVE#/
2508  015700  041040  051511  020124
2509  015706  051040  053111  021505
2510  015712  047045  047117  042455  MSG42:  .ASCII  /%NON-EXIST SLAVE#/
2511  015720  044530  052123  051440
2512  015726  040514  042523      043
2513  015733      045  042523  044522  MSG43:  .ASCII  /%SERIAL NO: #/
2514  015740  046101  047040  035117
2515  015746  021440
2516  015752  042455  040522  042523  MSG44:  .ASCII  /%ERASE HEAD INOPERATIVE#/
2517  015760  044040  040505  020104
2518  015766  047111  050117  051105
2519  015774  052101  053111  021505
2520  016000  050045  051517  044523  MSG45:  .ASCII  /%POSSIBLE ERASE HEAD PROBLEM: /
2521  016006  046102  020105  051105
2522  016014  051501  020105  042510
2523  016022  042101  050040  047522
2524  016030  046102  046505  020072
2525  016036  044103  041505  020113          .ASCII  /CHECK POLARITY#/
2526  016044  047520  040514  044522
2527  016052  054524      043
2528  016057      045  042523  026524  MSG46:  .ASCII  /%SET-UP WRITE ERROR#/
2529  016065  050123  053440  044522
2530  016073  042524  042440  051122
2531  016101  051117      043
2532  016104      045  050123  041501  MSG47:  .ASCII  /%SPACE FORWARD ERROR#/
2533  016112  020105  047506  053522
2534  016120  051101  020104  051105
2535  016126  047522  021522
2536  016132  051445  040520  042503  MSG48:  .ASCII  /%SPACE REVERSE ERROR#/
```

```
016122  051040  053105  051105
016130  042523  042440  051122
016136  051117     043
016141     045  052502  043106  MSG49:  .ASCII  /%BUFFERED WRITE ERROR#/
016146  051105  042105  053440
016154  044522  042524  042440
016162  051122  051117     043
016167     045  047502  020124  MSG50:  .ASCII  /%BOT SET AFTER BUFFERED WRITE#/
016174  042523  020124  043101
016202  042524  020122  052502
016210  043106  051105  042105
016216  053440  044522  042524
016224     043
016225     045  047516  041040  MSG51:  .ASCII  /%NO BOT FROM READ IN PRESET#/
016232  052117  043040  047522
016240  020115  042522  042101
016246  044440  020116  051120
016254  051505  052105     043
016261     045  041524  044440  MSG52:  .ASCII  /%TC INCORRECT #/
016266  041516  051117  042522
016274  052103  021440
016300  046445  046117  043040  MSG53:  .ASCII  /%MOL FAILED TO CLEAR#/
016306  044501  042514  020104
016314  047524  041440  042514
016322  051101     043
016325     045  051045  051505  MSG54:  .ASCII  /%%RESET SLAVE TO ON LINE BEFORE CONTINUING/
016332  052105  051440  040514
016340  042526  052040  020117
016346  047117  046040  047111
016354  020105  042502  047506
016362  042522  041440  047117
016370  044524  052516  047111
016376     107
016377     045  042523  020124          .ASCII  /%SET SW12=1 IF YOU DOT WISH TO REPEAT THIS TEST#/
016404  053523  031061  030475
016412  044440  020106  047531
016420  020125  047504  020124
016426  044527  044123  052040
016434  020117  042522  042520
016442  052101  052040  044510
016450  020123  042524  052123
016456     043
016457     040  052111  051105  MSG56:  .ASCII  / ITER: #/
016464  020072     043
016467     045  046524  047040  MSG57:  .ASCII  /%TM NOT SET#/
016474  052117  051440  052105
016502     043
016503     045  044505  044124  MSG60:  .ASCII  /%EITHER TAPE NOT ERASED OR OPI PROBLEM#/
016510  051105  052040  050101
016516  020105  047516  020124
016524  051105  051501  042105
016532  047440  020122  050117
016540  020111  051120  041117
016546  042514  021515
016552  051045  020110  047117  MSG62:  .ASCII  /%RH ONLY (NO=0,YES=1): #/
016560  054514  024040  047516
```

```
2593  016566  030075  054454  051505
2594  016572  030075  035061  021440
2595  016600  042045  042111  047040  MSG63:  .ASCII  /%DID NOT AUTO SELECT NRZ%/
2596  016610  052117  040440  052125
2597  016616  020117  042523  042514
2598  016624  052103  047040  055122
2599  016632      043
2600  016633      045  044504  020104  MSG64:  .ASCII  /%DID NOT AUTO SELECT PE%/
2601  016640  047516  020124  052501
2602  016646  047524  051440  046105
2603  016654  041505  020124  042520
2604  016662      043
2605  016663      045      043        MSG65:  .ASCII  /%%/
2606  016665      134      043        MSG66:  .ASCII  /\%/
2607  016667      045  051105  020072  MSG67:  .ASCII  /%ER: %/
2608  016674      043
2609  016675      045  042522  047515  MSG69:  .ASCII  /%REMOVE TMDP FROM SLAVE TO BE TESTED%%/
2610  016702  042526  052040  042115
2611  016710  020120  051106  046517
2612  016716  051440  040514  042526
2613  016724  052040  020117  042502
2614  016732  052040  051505  042524
2615  016740  022504      043
2616  016743      045  040510  042122  MSG70:  .ASCII  /%HARDWARE SWR IN USE%%/
2617  016750  040527  042522  051440
2618  016756  051127  044440  020116
2619  016764  051525  022505      043
2620
```

```
2621                                                    ;TEST HEADERS*****
2622
2623  016771  000045  043045  030524  MSFT1:  .ASCII  /%%FT1:RH ADDRESSING #/
2624  016776  051072  020110  042101
2625  017004  051104  051505  044523
2626  017012  043516  021440
2627  017016  022445  052106  035062  MSFT2:  .ASCII  /%%FT2:RH REGISTER BITS TEST #/
2630  017024  044122  051040  043505
2631  017032  051511  047524  020123
2632  017040  044502  051524  052040
2633  017046  051505  020124  000043
2634  017053  000045  043045  031524  MSFT3:  .ASCII  /%%FT3:RH INITIALIZE TEST #/
2635  017061  051072  020110  047111
2636  017067  052111  040511  044514
2637  017074  042522  052040  051505
2640  017102  020124  000043
2641  017105  000045  043045  032124  MSFT4:  .ASCII  /%%FT4:RH11 SILO TEST 1 #/
2642  017112  051072  030510  020061
2643  017120  044523  047514  052040
2644  017126  051505  020124  020061
2645  017134  000043
2646  017135  000045  043045  032524  MSFT5:  .ASCII  /%%FT5:RH11 SILO TEST 2 #/
2647  017142  051072  030510  020061
2650  017150  044523  047514  052040
2651  017156  051505  020124  020062
2652  017164  000043
2653  017165  000045  043045  033124  MSFT6:  .ASCII  /%%FT6:RH11 SILO TEST 3 #/
2654  017172  051072  030510  020061
2655  017200  044523  047514  052040
2656  017206  051505  020124  020063
2657  017214  000043
2660  017215  000045  043045  033524  MSFT7:  .ASCII  /%%FT7:RH11 SILO TEST 4 #/
2661  017222  051072  030510  020061
2662  017230  044523  047514  052040
2663  017236  051505  020124  020064
2664  017244  000043
2665  017245  000045  043045  030524  MSFT10: .ASCII  /%%FT10:RH11 SILO TEST 5 #/
2666  017253  035063  044122  030461
2667  017260  051440  046111  020117
2670  017266  042524  052123  032440
2671  017274  021440
2672  017276  022445  052106  030461  MSFT11: .ASCII  /%%FT11:NOP TEST#/
2673  017304  047072  050117  052040
2674  017312  051505  021524
2675  017316  022445  052106  031061  MSFT12: .ASCII  /%%FT12:REWIND TEST#/
2676  017324  051072  053505  047111
2677  017332  020104  042524  052123
2700  017340  000043
2701  017341  000045  043045  030524  MSFT13: .ASCII  /%%FT13:WRITE-READ TEST#/
2702  017346  035063  051127  052111
2703  017354  026505  042522  042101
2704  017362  052040  051505  021524
2705  017370  022445  052106  032061  MSFT14: .ASCII  /%%FT14:SPACE TEST#/
2706  017376  051472  040520  042503
2707  017404  052040  051505  021524
2710  017412  022445  052106  032461  MSFT15: .ASCII  /%%FT15:ERASE TEST#/
```

```
2677  017420  042472  040522  042523
2678  017426  052040  051505  021524
2679  017434  022445  052106  033061  MSFT16: .ASCII  /%%FT16:TAPE MARK WRITE-READ TEST#/
2680  017442  052072  050101  020105
2681  017450  040515  042523  053440
2682  017456  044522  042524  051055
2683  017464  040505  020104  042524
2684  017472  052123      043
2685  017475      045  043045  030524  MSFT17: .ASCII  /%%FT17:TM SPACE TEST #/
2686  017502  035067  046524  051440
2687  017510  040520  042503  052040
2688  017516  051505  020124      043
2689  017523      045  043045  031124  MSFT20: .ASCII  /%%FT20:WRITE CHECK TEST #/
2690  017530  035064  051127  052111
2691  017536  020106  044103  041505
2692  017544  020113  042524  052123
2693  017552  021440
2694  017554  022445  052106  030462  MSFT21: .ASCII  /%%FT21:ERASE HEAD TEST#/
2695  017562  042472  040522  042523
2696  017570  044040  040505  020104
2697  017576  042524  052123      043
2698  017603      045  043045  031124  MSFT22: .ASCII  /%%FT22:BUFFERED COMMAND TEST#/
2699  017610  035062  052502  043106
2700  017616  051105  042105  041440
2701  017624  046517  040515  042116
2702  017632  052040  051505  021524
2703  017640  022445  052106  031462  MSFT23: .ASCII  /%%FT23:READ IN PRESET TEST#/
2704  017646  051072  040505  020104
2705  017654  047111  050040  042522
2706  017662  042523  020124  042524
2707  017670  052123      043
2708  017673      045  043045  031124  MSFT26: .ASCII  /%%FT26:REWIND-OFF LINE TEST#/
2709  017700  035066  042522  044527
2710  017706  042116  047455  043106
2711  017714  046040  047111  020105
2712  017722  042524  052123      043
2713  017727      045  043045  031124  MSFT24: .ASCII  /%%FT24:AUTO DENSITY SELECT: WRITE-NRZ,READ-PE#/
2714  017734  035064  052501  047524
2715  017742  042040  047105  044523
2716  017750  054524  051440  046105
2717  017756  041505  036124  053440
2718  017764  044522  042524  047055
2719  017772  055122  051054  040505
2720  020000  026504  042520      043
2721  020005      045  043045  031124  MSFT25: .ASCII  /%%FT25:AUTO DENSITY SELECT: WRITE-PE,READ-NRZ#/
2722  020012  035065  052501  047524
2723  020020  042040  047105  044523
2724  020026  054524  051440  046105
2725  020034  041505  036124  053440
2726  020042  044522  042524  050055
2727  020050  026105  042522  042101
2728  020056  047055  055122      043
2729  020063      045  043536      043  $CNTG:  .ASCII  /%↑G#/
2730  020067      045  053523  036522  $MSWR:  .ASCII  /%SWR= #/
2731  020074  021440
2732  020076  020040  042516  036527  $MNEW:  .ASCII  /  NEW= #/
```

```
 2733  020104  021440
 2734  020106  022477     043          $QUEST: .ASCII  /?%#/
 2735
 2736
 2737                  020112                  .EVEN
 2738  020112          000000          WDATA:  0
 2739                  021624                  .=.+1510
 2740  021624          000000          RDATA:  0
 2741
 2742                  000001                  .END
```

```
AS      000526
BA      000514
BADDR   000616
BAE     000544
BTRP    000574
BTRP2   000576
CC      000530
CHNFLG  001662
COUNT   000734
CRCNT   000714
CS      000520
C1      000510
DATA0   000742
DATA1   000744
DATA2   000746
DATA3   000750
DATBL   000740
DAT1    012440
DAT1A   012444
DAT2    012460
DAT3    012464
DAT4    012472
DB      000532
DOUT    014012
DOUTD   014074
DRVN    000612
DRVTP   000564
DS      000522
DSAV    000664
DSLP    012400
DSO     012402
DT      000536
EMADDR  000610
ER      000524
ERADD   000650
ERCHK   011776
ERPT    012022
ERPTA   012112
ERPTA1  012044
ERPTB   012150
ERPTB1  012162
ERPTC   012212
ERPTD   012322
ERPTG   012052
ERPTX   012332
ERPTXX  012336
ERPT1A  012144
ERRP    000626
ERRP1   000630
EXEC    011546
EXECA   011620
EXECB   011626
EXECC   011646
EXECX   011666
EXECXA  011672
EXECXX  011676
EXFL    000700
FC      000516
FCNT    000620
FT1     003216
FT1A    003250
FT1B    003254
FT1X    003266
FT1XX   003312
FT10    004746
FT10A   005002
FT10ER  005066
FT10X   005044
FT10XX  005062
FT11    005102
FT12    005220
FT12A   005300
FT13    005332
FT13A   005404
FT13B   005446
FT13C   005526
FT13D   005574
FT13X   005640
FT14    005644
FT14A   005716
FT14A1  005662
FT14A2  006010
FT14A3  006032
FT14B   006062
FT14EC  006300
FT14RE  006306
FT14RF  006252
FT14RR  006210
FT14RO  006360
FT14R1  006402
FT14R2  006406
FT14R3  006460
FT14X   006470
FT14XX  006512
FT15    006516
FT15A   006566
FT15B   006602
FT15X   006712
FT16    006716
FT16A   006750
FT16B   006754
FT16X   007134
FT17    007144
FT17A   007164
FT17B   007170
FT17C   007330
FT17D   007376
FT17D1  007414
FT17E   007430
FT17F   007520
FT17X   007542
FT2     003316
FT2A    003330
FT2B    003370
FT2C    003430
FT2D    003444
FT2E    003474
FT2ER   003504
FT2ERA  003534
FT2ERB  003606
FT2ERC  003616
FT2X    003626
FT20    007546
FT20A   007564
FT20B   007710
FT20C   007742
FT20X   007762
FT21    007772
FT21A   010000
FT21B   010274
FT21C   010302
FT21SC  010130
FT21X   010314
FT22    010324
FT22A   010400
FT22B   010420
FT22X   010520
FT23    010530
FT23A   010640
FT23B   010660
FT23C   010710
FT23X   010744
FT24    010750
FT24X   011132
FT25    011142
FT25X   011324
FT26    011334
FT26X   011532
FT26XX  011542
FT3     003640
FT3A    003714
FT3B    003736
FT3ER   003774
FT3X    003764
FT4     004060
FT4ER   004142
FT4ERA  004152
FT4ERB  004162
FT4ERC  004170
FT4X    004132
FT5     004206
FT5A    004230
FT5B    004256
FT5C    004300
FT5D    004330
FT5E    004342
FT5X    004370
FT6     004400
FT6A    004422
FT6B    004430
FT6C    004466
FT6D    004510
FT6DE   004542
FT6DEA  004574
FT6DEB  004640
FT6DEX  004650
FT6E    004512
FT6X    004532
FT7     004652
FT7A    004704
FT7ER   004732
FT7X    004722
FUN     000710
GTSWR   014222
HDRFL   000606
HERE    003164
INIT1   012620
INIT2   012630
ITAMT   000566
ITCNT   000662
ITER    012546
ITRLP   000676
LTADD   000706
MR      000534
MSFT1   016771
MSFT10  017245
MSFT11  017276
MSFT12  017316
MSFT13  017341
MSFT14  017370
MSFT15  017412
MSFT16  017434
MSFT17  017475
MSFT2   017016
MSFT20  017523
MSFT21  017554
MSFT22  017603
MSFT23  017640
MSFT24  017727
MSFT25  020005
MSFT26  017673
MSFT3   017053
MSFT4   017105
MSFT5   017135
MSFT6   017165
MSFT7   017215
MSG1    014364
MSG10   014725
MSG11   014745
MSG12   014765
MSG13   015003
MSG14   015030
MSG15   015055
MSG16   015076
MSG17   015116
MSG2    014452
MSG20   015136
MSG21   015144
MSG22   015151
MSG23   015161
MSG24   015171
MSG25   015205
MSG26   015213
MSG27   015221
MSG28   015227
MSG29   015257
MSG3    014470
MSG30   015306
MSG31   015336
MSG32   015366
MSG33   015404
MSG34   015421
MSG35   015437
MSG36   015464
MSG37   015513
MSG38   015552
MSG39   015604
MSG4    014623
MSG40   015624
MSG41   015657
MSG42   015700
MSG43   015721
MSG44   015736
MSG45   015766
MSG46   016043
MSG47   016067
MSG48   016114
MSG49   016141
MSG5    014646
MSG50   016167
MSG51   016225
MSG52   016261
MSG53   016300
MSG54   016325
MSG56   016457
```

TM03/TE16 BASIC FUNCTION TEST MACY11 27(1006) 31-MAR-77 17:23 PAGE 75
C 29-MAR-77 09:54 SYMBOL TABLE

```
MSG57   016467      OCTP3   013732      SCNT    000642      TEMP3   000656      TSCD3   003034
MSG6    014661      OFL     014010      SCOLP   000674      TEND    003110      TSRH    002750
MSG60   016503      OPDYX   000640      SCOPE   012512      TENDX   003206      TSTTBL  000752
MSG62   016552      PATRN   000720      SERFL   000712      TEX     013556      TTIN    013400
MSG63   016602      PCNTR   000730      SERNUM  000562      TIB     000602      TTINT   012656
MSG64   016633      PEXFL   000702      SLVN    000614      TINER   013360      TTOUT   013442
MSG65   016663      PFLG    000644      SN      000540      TKB     000554      TTR     013120
MSG66   016665      PSW     000546      SNPG    014176      TKS     000552      UDES    000716
MSG67   016667      RCNT    000624      SNPT    014120      TLAST   001110      VECT    000570
MSG69   016675      RDATA   021624      START   001600      TMCHK   012340      WC      000512
MSG7    014677      RDSW    000736      STFLG   000704      TMCHKO  012350      WCNT    000622
MSG70   016743      RDYDX   000636      STMSK   000660      TMCHKY1 012352      WDATA   020112
MSG9    014703      REGS    000572      STSCD   003054      TO8     000600      $CNTG   020063
MTINT   012646      RFD     000634      STO     002126      TOG     013542      $DONE   003116
NRZOF   000724      RHOF    000726      STO8    001766      TPB     000560      $ENDAD  003154
NXTDRV  002616      RHTF    000722      ST1     002150      TPS     000556      $MNEM   020076
NXTSLV  002670      RH17F   000604      ST1A    002170      TRAP    013004      $MSWR   020067
OCTP    013572      RRD     000632      ST2     002276      TRAPX   013066      $QUEST  020106
OCTPE   013560      RTRN    000646      ST3     002406      TRAPXX  013114      $SVPC =  000764
OCTPE1  013576      RWND    011700      ST4     002542      TRAP1   013040      .     =  021626
OCTPG   013746      RWNDA   011720      SWR     000550      TRAP2   013056      .RESTO  014342
OCTPGO  013764      RWNDB   011724      SWREG   000176      TSCD    002556      .SAVE   014320
OCTPG1  013770      RWNDX   011770      TC      000542      TSCDA   002746
OCTPO   013612      SAV1    000666      TEMPST  000732      TSCDO   002756
OCTP1   013634      SAV2    000670      TEMP1   000652      TSCD1   002764
OCTP2   013642      SAV3    000672      TEMP2   000654      TSCD2   003022

. ABS.  021626      000
```

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DZTECA,DZTECA/SOL+C.DOC,C
RUN-TIME: 3 5 .3 SECONDS
RUN-TIME RATIO: 36/8=4.1
CORE USED: 6K (11 PAGES)