

.REM •

IDENTIFICATION

PRODUCT CODE: AC-A797F-MC
PRODUCT NAME: CZTECFO TM03-TE16/TU77 BASIC FUNCTION TEST
DATE CREATED: 15 MARCH 1984
MAINTAINER: TAPE DIAGNOSTIC ENGINEERING
AUTHOR: J. HITT

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

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

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

COPYRIGHT (C) 1977,1984 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	3
3.	LOADING PROCEDURE	3
4.	STARTING PROCEDURE	3
5.	SWITCH SETTINGS	5
6.	ERROR PRINTOUTS	6
7.	OPERATION	7
8.	SUBTEST SUMMARIES	8
9.	LISTING	16

1. ABSTRACT

THIS PROGRAM IS INTENDED TO TEST ALL OF THE BASIC
FUNCTIONAL LEVEL OPERATIONS OF THE TMO3/TE16
MAG TAPE SYSTEM. ALL FUNCTIONS; WRITE, READ, SPACE,
ERASE, REWIND, ETC; WILL BE TESTED. IN ADDITION TO THE
TMO3/TE16 TESTS, THE RH WILL BE TESTED SEPARATELY IN
SO FAR AS IT IS POSSIBLE TO SEPARATE THE RH FROM THE
TMO3/TE16 ITSELF.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP11 PROCESSOR
B. 8K OF CORE
C. CONSOLE TTY
D. TMO3 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 CUR ENT 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 AVAIL-
ABLE TM03/TE16 COMBINATIONS ARE TESTED. ADDITIONALLY THE SOFTWARE
SWR IS INVOKED WITH A SWITCH SETTING OF 000000
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: IN ORDER TO CHANGE THE SETTING OF THE SOFTWARE SWR,
SET LOC: 176(SWREG:) TO THE DESIRED SETTING.

** 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).NON STANDARD
MODE FOR JUMPERS IS M8931 (W2-IN) ,M8937(W2-IN,W1-OUT).

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

TM03-TE16/TU77 BASIC FUNCTIONS TEST (DZTEC-B)
TYPE +C TO RESTART

REGISTER START: <172440> (CR) [REGS:]
VECTOR ADDRESS: <224> (CR) [VECT:]
IS CONTROLLER JUMPED IN NON-STANDARD MODE
TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD; <3> [JUMPER:]
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: RM 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 RESPONSES TO THE TTY REQUESTS

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

SINGLE TEST SELECTION: (SW0-SW4)

WHEN SW0-4 ARE SET TO ZERO (00) THE SCHEDULAR WILL EXECUTE ALL OF THE TESTS IN SEQUENCE . 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/TU77 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/TU77 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 TMO2 WILL ACCEPT AND EXECUTE ANOTHER COMMAND WHILE ITS SELECTED SLAVE IS REWINDING.

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

FT23: READ IN PRESET: THIS TEST WILL ASSURE THAT UNIT 0
IS 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,READPF;
THIS TEST ASSURES THAT AW 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.

FT27: 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)

542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580

```
.LIST BIN,LOC,SEQ
.TITLE CZTECFO TM03-TE16/TU77 BFT
;BASIC FUNCTION TEST
;AC-A797F-MC
;FEB 77
;J.G. ADAMS

;REVISED JUN 1977 BY J.G. ADAMS ;+B ADDED TU/7 CAPABILITY
;REVISED NOV 1978 BY MIKE PAGE ;+ DESIGNATES CODE ADDED FOR
;NON-STANDARD JUMPER CONFIG.
;REVISED MAY 1983 BY B. LEBLANC ;BL FIXED CC0000810,CC0001688
;REVISED SEP 1983 BY B. LEBLANC ;; FIXED CC0001619,CC0001743
;REVISED MARCH 1984 BY J. HITT ;ADDED XON/XOFF FUNCTIONALITY

.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=HALT AT END OF PASS
; 0=CONTINUOUS CYCLE
;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.
```



```
629 ;REGISTER EQUIVS*****
630
631 000000 R0=#0
632 000001 R1=#1
633 000002 R2=#2
634 000003 R3=#3
635 000004 R4=#4
636 000005 R5=#5
637 000006 SP=#6
638 000007 PC=#7
639
641
642
(1) ;ACT11 HOOK *****
(1) 000764 $SVPC=, ;SAVE CURRENT LOCATION CTR
(1) 000042 .=42
(1) 000042 000000 .WORD 0
(1) 000046 000046 .=46
(1) 000046 003226 .WORD $ENDAD ;SET LOCATION 46
(1) 000052 000052 .=52
(1) 000052 000000 .WORD 0 ;SET LOCATION 52 = 0
(1) 000764 .=$SVPC ;RESTORE LOCATION CTR
643 ;TTY INTERRUPT VECTOR*****
644
645 000060 .=60
646 000060 013576 .WORD TTINT ;TTY INTERRUPT HEADER ADDRESS
647 000062 000340 .WORD 340 ;PRIORITY LEVEL 7
648
649 ;SOFTWARE SWITCH REGISTER*****
650 ;USED IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
651
652 000176 000176 .=176
653 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
654
655 ;START ADDRESS*****
656
657
658 000200 000200 .=200
659 000200 000137 001600 JMP START ;PROGRAM START
660
661 ;RESTART ADDRESS*****
662 000210 000210 .=210
663 000210 000137 002540 JMP ST4
664
665 ;TMO3 INTERRUPT VECTOR*****
666
667 000224 000224 .=224
668 000224 013566 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
669 000226 000340 340
670
```

```
672
673          000510          .-510
674          ;MASS BUS REGISTER EQUIVS*****
675
676 000510 172440          C1: 172440
677 000512 172442          WC: 172442
678 000514 172444          BA: 172444
679 000516 172446          FC: 172446
680 000520 172450          CS: 172450
681 000522 172452          DS: 172452
682 000524 172454          ER: 172454
683 000526 172456          AS: 172456
684 000530 172460          CC: 172460
685 000532 172462          DB: 172462
686 000534 172464          MR: 172464
687 000536 172466          DT: 172466
688 000540 172470          SN: 172470
689 000542 172472          TC: 172472
690 000544 172474          BAE: 172474
691
692          ;CONSTANTS*****
693
694 000546 177776          PSW: 177776          ;PROCESSOR STATUS
695 000550 177570          SWR: 177570          ;SWITCH REGISTER
696 000552 177560          TKS: 177560          ;TTY READER STATUS
697 000554 177562          TKB: 177562          ;TTY READ BUFFER
698 000556 177564          TPS: 177564          ;TTY PUNCH STATUS
699 000560 177566          TPB: 177566          ;TTY PUNCH BUFFER
700 000562 177777          SERNUM: 177777          ;SERIAL NUMBER
701 000564 000011          DRVTP: 011          ;DRIVE TYPE
702 000566 000010          ITAMT: 10          ;ITERATION AMOUNT
703 000570 000224          VECT: 224          ;INTERRUPT VECTOR(RM)
704 000572 172440          REGS: 172440          ;STARTING REGISTER ADDRESS
705 000574 000004          BTRP: 4          ;BUS TRAP ADDRESS
706 000576 000006          BTRP2: 6          ;BUS TRAP PRIORITY LEVEL 7
```

```
708 ;FLAGS AND COUNTERS*****
709
710 000600 000000 TOB: 0
711 000602 000000 TIB: 0
712 000604 000000 RH17F: 0
713 000606 000000 HDRFL: 0
714 000610 000000 EMADDR: 0
715 000612 000000 DRVN: 0
716 000614 000000 SLVN: 0
717 000616 000000 BADDR: 0
718 000620 000000 FCNT: 0
719 000622 000000 WCNT: 0
720 000624 000000 RCNT: 0
721 000626 000000 ERRP: 0
722 000630 000000 ERRP1: 0
723 000632 000000 RRD: 0
724 000634 000000 RFD: 0
725 000636 000000 RDYDX: 0
726 000640 000000 OPDYX: 0
727 000642 000000 SCNT: 0
728 000644 000000 PFLG: 0
729 000646 000000 RTRN: 0
730 000650 000000 ERADD: 0
731 000652 000000 TEMP1: 0
732 000654 000000 TEMP2: 0
733 000656 000000 TEMP3: 0
734 000660 000000 STMSK: 0
735 000662 000000 ITCNT: 0
736 000664 000000 DSAV: 0
737 000666 000000 SAV1: 0
738 000670 000000 SAV2: 0
739 000672 000000 SAV3: 0
740 000674 000000 SCOLP: 0
741 000676 000000 ITRLP: 0
742 000700 000000 EXFL: 0
743 000702 000000 PEXFL: 0
744 000704 000000 STFLG: 0
745 000706 000000 LTADD: 0
746 000710 000000 FUN: 0
747 000712 000000 SERFL: 0
748 000714 000000 CRCNT: 0
749 000716 000000 UDES: 0
750 000720 000000 PATRN: 0
751 000722 000000 RHTF: 0
752 000724 000000 NRZOF: 0
753 000726 000000 RHOF: 0
754 000730 000000 PCNTR: 0
755 000732 000000 TEMPST: 0
756 000734 000000 COUNT: 0
757 000736 000000 RDSW: 0
758 000740 000000 NONSTD: 0
759 000742 000000 JUMPER: 0
760
```

762
763
764
765
766
767
768
769

000744 000000
000746 013262
000750 013302
000752 013306
000754 013314

;DATA PATTERN GENERATORS*****

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

771
772
773
774 000756 000000
775 000760 000000
776 000762 003270
777 000764 003270
778 000766 003376
779 000770 003376
780 000772 003720
781 000774 003720
782 000776 004140
783 001000 004140
784 001002 004266
785 001004 004266
786 001006 004460
787 001010 004460
788 001012 004732
789 001014 004732
790 001016 005026
791 001020 005026
792 001022 005162
793 001024 005162
794 001026 005300
795 001030 005300
796 001032 005412
797 001034 005412
798 001036 005724
799 001040 005724
800 001042 006576
801 001044 006576
802 001046 007210
803 001050 007210
804 001052 007436
805 001054 007436
806 001056 010040
807 001060 010040
808 001062 010264
809 001064 010264
810 001066 010616
811 001070 010616
812 001072 011022
813 001074 011022
814 001076 011242
815 001100 011242
816 001102 011434
817 001104 011434
818 001106 011626
819 001110 011626
820 001112 012064
821 001114 012064
822 001116 003162
823 001120 000027
824 001122 000000

;LOGIC TEST ENTRY TABLE*****

TSTTBL: 0
0
FT1
FT1
FT2
FT2
FT3
FT3
FT4
FT4
FT5
FT5
FT6
FT6
FT7
FT7
FT10
FT10
FT11
FT11
FT12
FT12
FT13
FT13
FT14
FT14
FT15
FT15
FT16
FT16
FT17
FT17
FT20
FT20
FT21
FT21
FT22
FT22
FT23
FT23
FT24
FT24
FT25
FT25
FT26
FT26
FT27
FT27

TLAST: .WORD TEND
\$CNTRLS: .WORD 27
0 ;CONTAINS # OF TESTS
;XON/XOFF FLAG

```

826          001600          .-1600
827          ;PROGRAM START AND HOUSEKEEPING*****
828
829 001600 012706 000500          START:  MOV    #500,SP          ;SET STACK POINTER
830 001604 013746 000006          MOV    @#6,-(SP)          ;SAVE VECTORS
831 001610 013746 000004          MOV    @#4,-(SP)
832 001614 012737 001640 000004          MOV    #1,@#4          ;SET UP FOR TIMEOUT
833 001622 005037 000006          CLR    @#6
834 001626 022777 177777 176714          CMP    #-1,@SWR          ;REFERENCE HARDWARE SWITCH REGISTER
835 001634 001402          BEQ    2$
836 001636 000404          BR     3$
837 001640 022626          1$:  CMP    (SP),-(SP)          ;ADJUST STACK
838 001642 012737 000176 000550          2$:  MOV    #SWREG,SWR          ;POINT TO SOFTWARE SWITCH REG
839 001650 012637 000004          3$:  MOV    (SP),@#4          ;RESTORE VECTORS
840 001654 012637 000006          MOV    (SP),@#6
841 001660 005027          CLR    (PC)          ;;CLEAR CHAIN INDICATOR
(1) 001662 000000          CHNFLG: .WORD 0          ;;CHAIN MODE INDICATOR
(1)          TST    @#42          ;;1/0 = CHAIN/NOT CHAIN MODE
(1) 001664 005737 000042          BEQ    50$          ;;BRANCH IF IN DUMP MODE
(1) 001670 001407          MOV    #SWREG,SWR          ;;INVOKE SOFTWARE SWR
(1) 001672 012737 000176 000550          INC    CHNFLG          ;;SET CHNFLG = CHAIN MODE
(1) 001700 005237 001662          JMP    SCHN          ;;GO TO CHAIN ADDRESS
(1) 001704 000137 001710          50$:
842 001710 000240          SCHN:  NOP
843 001712 122737 000006 000041          4$:  CMPB   @#6,@#41          ;BRANCH IF LOADED VIA TMDP (DUMP MODE)
844 001720 001005          BNE    5$
845 001722 012704 020214          MOV    #MSG69,R4          ;ADVISE USER TO REMOVE TMDP FROM UUT
846 001726 004737 014414          JSR    PC,TTOUT
847 001732 000000          HALT
848 001734 012704 015563          5$:  MOV    #MSG3,R4
849 001740 004737 014414          JSR    PC,TTOUT          ;PRINT TITLE
850 001744 005737 001662          TST    CHNFLG          ;SEE IF IN CHAIN MODE
851 001750 001402          BEQ    6$          ;IF NOT: BR
852 001752 000137 002554          JMP    TSCD          ;ELSE GO START TEST
853 001756 112737 000043 015363          6$:  MOVB   #'@,MSG3          ;DO NOT PRINT TITLE ON RESTART
854 001764 012704 015723          STOB:  MOV    #MSG4,R4
855 001770 004737 014414          JSR    PC,TTOUT          ;REQUEST REGISTER ADDRESS
856 001774 013703 000572          MOV    REGS,R3
857 002000 004737 014630          JSR    PC,OCTP          ;PRINT CURRENT ADDRESS
858 002004 012705 000572          MOV    #REGS,R5          ;SET ADDRESS SAVE LOC
859 002010 012701 000007          MOV    #7,R1          ;SET SIZE OF RESPONSE
860 002014 012702 176400          MOV    #176400,R2          ;SET UPPER LIMIT
861 002020 012703 172300          MOV    #172300,R3          ;SET LOWER LIMIT
862 002024 004737 014072          JSR    PC,TTR          ;GO GET RESPONSE
863 002030 012704 015746          MOV    #MSG5,R4
864 002034 004737 014414          JSR    PC,TTOUT          ;REQUEST VECTOR
865 002040 013703 000570          MOV    VECT,R3
866 002044 004737 014630          JSR    PC,OCTP          ;PRINT CURRENT VECTOR
867 002050 012705 000570          MOV    #VECT,R5          ;SET ADDRESS SAVE LOC
868 002054 012701 000004          MOV    #4,R1          ;SET SIZE OF RESPONSE
869 002060 012702 000224          MOV    #224,R2          ;SET UPPER LIMIT
870 002064 012703 000150          MOV    #150,R3          ;SET LOWER LIMIT
871 002070 004737 014072          JSR    PC,TTR          ;GO GET RESPONSE
872 002074 013700 000570          MOV    VECT,R0          ;GET VECTOR
873 002100 012720 013566          MOV    #MTINT,(R0)          ;LOAD INTERRUPT ADDRESS IN VECTOR
  
```

874	002104	012710	000340		MOV	#340,(R0)	;LOAD PRIORITY
875	002110	013700	000572		MOV	REGS,R0	;GET START OF REGS
876	002114	012701	000017		MOV	#17,R1	;SET NUMBER OF REGS
877	002120	012702	000510		MOV	#C1,R2	;GET START OF TABLE
878	002124	010022		ST0:	MOV	R0,(R2)+	;BUILD TABLE
879	002126	062700	000002		ADD	#2,R0	;BUMP ADDRESS
880	002132	005301			DEC	R1	;SEE IF DONE
881	002134	001373			BNE	ST0	;IF NOT: BR
882	002136	012702	000600		MOV	#TOB,R2	
883	002142	012700	000054		MOV	#54,R0	
884	002146	005022		ST1:	CLR	(R2)+	;CLEAR FLAGS + COUNTERS
885	002150	005300			DEC	R0	
886	002152	001375			BNE	ST1	
887	002154	012737	000001	000722	MOV	#1,RHTF	;SET ADDRESS TEST FLAG
888	002162	000137	003022		JMP	TSRH	;GO DO INITIAL ADDRESS TEST PASS
889	002166	012704	016025	ST1A:	MOV	#MSG10A,R4	
890	002172	004737	014414		JSR	PC,TTOUT	;REQUEST JUMPER CONFIGURATION
891	002176	012705	000742		MOV	#JUMPER,R5	;GET ADDRESS OF RESPONSE
892	002202	012701	000002		MOV	#2,R1	;SET SIZE OF RESPONSE
893	002206	012702	000004		MOV	#4,R2	;SET RANGE
894	002212	012703	000000		MOV	#0,R3	;LOWER LIMIT
895	002216	004737	014072		JSR	PC,TTR	;GET RESPONSE
896	002222	022737	000002	000742	CMP	#2,JUMPER	;TEST FOR NON-STANDARD MODE
897	002230	001002			BNE	1\$	
898	002232	004737	013442		JSR	PC,NOST	;MODIFY TEST SCHEDULE
899	002236	012704	016165	1\$:	MOV	#MSG10,R4	
900	002242	004737	014414		JSR	PC,TTOUT	;REQUEST DRIVE NUMBER
901	002246	013703	000612		MOV	DRVN,R3	;GET CURRENT DRIVE #
902	002252	004737	014630		JSR	PC,OCTP	;AND TYPE IT
903	002256	012705	000612		MOV	#DRVN,R5	;SET ADDRESS OF DRIVE NUMBER SAVE
904	002262	012701	000002		MOV	#2,R1	;SET SIZE OF RESPONSE
905	002266	012702	000007		MOV	#7,R2	;SET UPPER LIMIT
906	002272	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
907	002276	004737	014072		JSR	PC,TTR	;GO GET RESPONSE
908	002302	012777	000040	176210	MOV	#40,@CS	;SET INIT
909	002310	053777	000612	176202	BIS	DRVN,@CS	;SET DRIVE NUMBER
910	002316	005777	176166		TST	@C1	;ACCESS DRIVE
911	002322	032777	010000	176170	BIT	#10000,@CS	;SEE IF WED
912	002330	001405			BEQ	ST2	;IF NOT: BR
913	002332	012704	017164		MOV	#MSG41,R4	
914	002336	004737	014414		JSR	PC,TTOUT	;PRINT NOT AVAIL
915	002342	000711			BR	ST1A	;REDO DRIVE REQUEST
916	002344	012704	016205	ST2:	MOV	#MSG11,R4	
917	002350	004737	014414		JSR	PC,TTOUT	;REQUEST SLAVE NUMBER
918	002354	013703	000614		MOV	SLVN,R3	;GET CURRENT SLAVE #
919	002360	004737	014630		JSR	PC,OCTP	;AND TYPE IT
920	002364	012705	000614		MOV	#SLVN,R5	;SET ADDRESS OF SLAVE SAVE
921	002370	012701	000002		MOV	#2,R1	;SET SIZE OF RESPONSE
922	002374	012702	000007		MOV	#7,R2	;SET UPPER LIMIT
923	002400	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
924	002404	004737	014072		JSR	PC,TTR	;GO GET RESPONSE
925	002410	012777	000040	176102	MOV	#40,@CS	;INIT
926	002416	053777	000612	176074	BIS	DRVN,@CS	;SET DRIVE NUMBER
927	002424	013777	000614	176110	MOV	SLVN,@TC	;LOAD SLAVE NUMBER
928	002432	032777	002000	176076	BIT	#2000,@DT	;SEE IF SLAVE PRESENT
929	002440	001005			BNE	ST3	;IF SO: BR

```
930 002442 012704 017205      MOV    #MSG42,R4
931 002446 004737 014414      JSR    PC,TTOUT      ;PRINT NON-EXIST SLAVE
932 002452 000734              BR     ST2           ;REDO SLAVE REQUEST
933 002454 012704 017226      ST3:  MOV    #MSG43,R4
934 002460 004737 014414      JSR    PC,TTOUT      ;PRINT SERIAL NUMBER TAG
935 002464 017703 176050      MOV    @SN,R3
936 002470 004737 015156      JSR    PC,SNPT       ;PRINT SERIAL NUMBER
937 002474 012704 020071      MOV    #MSG62,R4    ;GET REQUEST
938 002500 004737 014414      JSR    PC,TTOUT      ;REQUEST RH11 ONLY RESPONSE
939 002504 013703 000726      MOV    RHOF,R3      ;GET CURRENT FLAG SETTING
940 002510 004737 014630      JSR    PC,OCTP       ;AND TYPE IT
941 002514 012705 000726      MOV    @RHOF,R5     ;SET FLAG ADDRESS
942 002520 012701 000002      MOV    #2,R1        ;SET SIZE OF RESPONSE
943 002524 012702 000001      MOV    #1,R2        ;SET UPPER LIMIT
944 002530 012703 000000      MOV    #0,R3        ;SET LOWER LIMIT
945 002534 004737 014072      JSR    PC,TTR       ;GO GET RESPONSE
946
947                               ;START 210
948 002540 012706 000500      ST4:  MOV    #500,SP ;SET STACK PTR
949 002544 005037 000730      CLR    PCNTR        ;CLEAR PASS COUNTER
950 002550 004737 015260      JSR    PC,GTSWR     ;GET SWITCHES
```

```

952                                     ;TEST SCHEDULAR*****
953
954 002554 052777 000100 175770 TSCD: BIS      #100,@TKS      ;SET KEYBOARD IE BIT
955 002562 005037 000704          CLR      STFLG        ;CLEAR SINGLE TEST FLAG
956 002566 005037 000604          CLR      RH17F       ;SET RH INDICATOR = RH11
957 002572 013746 000004          MOV      @#4,-(SP)    ;SAVE ERROR TRAP VECTORS
958 002576 013746 000006          MOV      @#6,-(SP)    ;AND PRIORITY
959 002602 012737 002630 000004  MOV      @1,@#4       ;SET TIME OUT TRAP TO 1$ BELOW
960 002610 005037 000006          CLR      @#6
961 002614 005777 175724          TST     @BAE         ;REFERENCE BAE REGISTER
962 002620 012737 000001 000604  MOV      @1,RH17F     ;SET FLAG = RH70
963 002626 000401          BR      2$
964 002630 022626          1$:  CMP     (SP)+,(SP)+  ;RESET STACK
965 002632 012637 000006          2$:  MOV     (SP)+,@#6   ;RESTORE ERROR TRAP
966 002636 012637 000004          MOV     (SP)+,@#4
967 002642 017700 175702          MOV     @SWR,RO
968 002646 042700 177740          BIC     @177740,RO
969 002652 001125          BNE     STSCD        ;GO SELECT SINGLE TEST
970 002654 005737 001662          TST     CHNFLG       ;BRANCH IF NOT IN CHAIN MODE
(1) 002660 001457          BEQ     TSCDA
(1) 002662 012737 177777 000612  MOV     #-1,DRVN     ;;INITIALIZE DRIVE #
(1) 002670 012737 177777 000614  NXTDRV: MOV     #-1,SLVN  ;;INITIALIZE SLAVE #
(1) 002676 012777 000040 175614  1$:  MOV     @40,@CS    ;;INIT CONTROLLER
(1) 002704 005237 000612          INC     DRVN        ;;STEP DRIVE #
(1) 002710 022737 000010 000612  CMP     @10,DRVN    ;;EXIT IF ALL DRIVES TESTED
(1) 002716 001524          BEQ     $DONE       ;;FOR AVAILABILITY
(1) 002720 013777 000612 175572  MOV     DRVN,@CS    ;;LOAD DRIVE #
(1) 002726 005777 175556          TST     @C1         ;;ACCESS DRIVE
(1) 002732 032777 010000 175560  BIT     @10000,@CS  ;;BRANCH IF DRIVE NON EXISTANT
(1) 002740 001356          BNE     1$          ;;(NED = 1)
(1) 002742 005237 000614          NXTSLV: INC     SLVN  ;;STEP SLAVE # AND BRANCH
(1) 002746 001011          BNE     1$          ;;IF NOT SLAVE 0
(1) 002750 005737 000612          TST     DRVN        ;;BRANCH IF NOT DRIVE # 0
(1) 002754 001006          BNE     1$
(1) 002756 122737 000006 000041  CMPB   @6,@#41     ;;BRANCH IF NOT TMDP
(1) 002764 001002          BNE     1$
(1) 002766 005237 000614          INC     SLVN        ;;STEP TO SLAVE # 1
(1) 002772 022737 000010 000614  1$:  CMP     @10,SLVN   ;;BRANCH IF ALL SLAVES TESTED
(1) 003000 001733          BEQ     NXTDRV      ;;FOR AVAILABILITY
(1) 003002 013777 000614 175532  MOV     SLVN,@TC    ;;LOAD SLAVE UNIT #
(1) 003010 032777 002000 175520  BIT     @2000,@DT   ;;BRANCH IF SLAVE NOT
(1) 003016 001751          BEQ     NXTSLV     ;;PRESENT (SPR = 0)
971 003020 000240          TSCDA: NOP
972 003022 012737 000756 000706  TSRH:  MOV     @TSTTBL,LTADD
973 003030 062737 000004 000706  TSCD0: ADD     @4,LTADD
974 003036 013737 000706 000676  TSCD1: MOV     LTADD,IIRLF
975 003044 062737 000002 000676  ADD     @2,IIRLP    ;SET ITERATION ADDRESS
976 003052 005037 000660          CLR     STMSK
977 003056 005037 000626          CLR     ERRP
978 003062 005037 000606          CLR     HDRFL      ;CLEAR PRINT HEADER FLAG
979 003066 017700 175614          MOV     @LTADD,RO  ;SET POINTER TO TEST
980 003072 000110          JMP     (RO)        ;GO TO TEST
981 003074 032777 002000 175446  TSCD2: BIT     @2000,@SWR ;SEE IF HALT ON TEST
982 003102 001401          BEQ     TSCD3      ;IF NOT: BR
983 003104 000000          HALT
984 003106 005737 000704          TSCD3: TST     STFLG  ;SE IF SINGLE TEST
  
```

```

985 003112 001746          BEQ      TSCD0          ;IF NOT: BR
986 003114 017700 175430   MOV      @SWR,R0
987 003120 042700 177740   BIC      #177740,R0    ;BRANCH IF ALL TESTS SELECTED
988 003124 001613          BEQ      TSCD
989 003126 012737 000001 000704 STSCD: MOV      #1,STFLG      ;SET SINGLE TEST FLAG
990 003134 023700 C01120   CMP      TLAST,R0     ;SEE IF EXCEEDED TESTS
991 003140 002410          BLT      TEND          ;IF SO: BR
992 003142 006300          ASL      RO
993 003144 006100          ROL      RO           ;SET TABLE MODIFIER
994 003146 012737 000756 000706   MOV      #TSTTBL,LTADD
995 003154 060037 000706   ADD      RO,LTADD     ;SET TEST POINTER
996 003160 000726          BR       TSCD1
997 003162 005737 001662   TEND:   TST      CHNFLG ;BRANCH IF IN CHAIN MODE
998 003166 001265          BNE     NXTSLV
999 003170 012704 015761   $DONE:  MOV      #MSG6,R4
1000 003174 004737 014414   JSR     PC,TTOUT      ;PRINT END OF PASS
1001 003200 013703 000730   MOV     PCNTR,R3
1002 003204 004737 014630   JSR     PC,OCIP      ;PRINT PASS NUMBER
1003 003210 005000          CLR     RO
1004 003212 005300          1$:    DEC     RO
1005 003214 001376          BNE     1$
1006 003216 013700 000042   MOV     @#42,R0      ;GET ACT11 RETURN ADDRESS
(1) 003222 001405          BEQ     HERE         ;BRANCH IF NOT ACT11
(1) 003224 000005          RESET
(1) 003226 004710          $ENDAD: JSR     PC,(R0)
(1) 003230 000240          NOP
(1) 003232 000240          NOP
(1) 003234 000240          NOP
(1) 003236 000240          HERE:  NOP
1007 003240 005737 001662   TST     CHNFLG      ;BRANCH IF IN CHAIN MODE
1008 003244 001005          BNE     TENDX
1009 003246 032777 010000 175274   BIT     #10000,@SWR  ;SEE IF HALT ON PASS
1010 003254 001401          BEQ     TENDX        ;IF NOT: BR
1011 003256 000000          HALT
1012 003260 005237 000730   TENDX:  INC     PCNTR  ;BUMP PASS COUNTER
1013 003264 000137 002554   JMP     TSCD         ;RESTART

```

```
1015  
1016  
1017  
1018 003270 012737 020362 000610 FT1:  MOV    #MSFT1,EMADDR    ;SET HEADER  
1019 003276 012737 013756 000004      MOV    #TRAP,#04        ;SET TRAP HANDLER ADDRESS  
1020 003304 012737 000340 000006      MOV    #340,#06  
1021 003312 012700 000016      MOV    #16,R0          ;SET NUMBER OF REGISTERS  
1022 003316 013701 000510      MOV    C1,R1          ;GET FIRST ADDRESS (CS1)  
1023 003322 005711      FT1A:  TST    (R1)        ;REFERENCE REGISTER  
1024 003324 000240      NOP  
1025 003326 005300      FT1B:  DEC    R0          ;SEE IF DONE ALL  
1026 003330 001403      BEQ    FT1X          ;IF SO: BR  
1027 003332 062701 000002      ADD    #2,R1          ;BUMP ADDRESS POINTER  
1028 003336 000771      BR     FT1A          ;CONTINUE  
1029 003340 012737 000006 000004 FT1X:  MOV    #6,#04        ;RESET TRAP CATCHER  
1030 003346 012737 000000 000006      MOV    #HALT,#06  
1031 003354 005737 000722      TST    RHTF          ;SEE IF INITIAL ADDRESS TEST PASS  
1032 003360 001404      BEQ    FT1XX        ;IF NOT: BR  
1033 003362 005037 000722      CLR    RHTF         ;CLEAR FLAG  
1034 003366 000137 002166      JMP    ST1A         ;RETURN  
1035 003372 000137 003074      FT1XX: JMP    TSCD2        ;RETURN TO SCHEDULEAR
```

```

1037
1038
1039
1040 003376 012737 020407 000610 FT2:  MOV    #MSFT2,EMADDR  ;SET TEST HEADER
1041 003404 012701 177777          MOV    #-1,R1          ;SET ALL ONES PATTERN
1042 003410 004737 013540          FT2A: JSR    PC,INIT1      ;GO INIT
1043 003414 013700 000512          MOV    WC,R0          ;GET ADDRESS OF WORD COUNT
1044 003420 010102          MOV    R1,R2          ;SET EXPT REGISTER BIT PATTERN
1045 003422 010110          MOV    R1,(R0)        ;LOAD PATTERN
1046 003424 021002          CMP    (R0),R2        ;SEE IF EXPT=RCVD
1047 003426 J01410          BEQ    FT2B           ;IF SO: BR
1048 003430 012737 016512 000650          MOV    #MSG25,ERADD   ;SET CODE
1049 003436 012737 003410 000674          MOV    #FT2A,SCOLP    ;SET SCOPE
1050 003444 004737 003564          JSR    PC,FT2ER       ;GO DO ERROR
1051 003450 013700 000514          FT2B: MOV    BA,R0          ;GET ADDRESS OF BUS ADDRESS
1052 003454 010102          MOV    R1,R2
1053 003456 042702 000001          BIC    #1,R2          ;SET EXPT PATTERN
1054 003462 010110          MOV    R1,(R0)        ;LOAD PATTERN
1055 003464 020210          CMP    R2,(R0)        ;SEE IF EXPT=RCVD
1056 003466 001410          BEQ    FT2C           ;IF SO:BR
1057 003470 J12737 016520 000650          MOV    #MSG26,ERADD   ;SET ERROR CODE
1058 003476 012737 003450 000674          MOV    #FT2B,SCOLP    ;SET SCOPE ADDRESS
1059 003504 004737 003564          JSR    PC,FT2ER       ;GO DO ERROR
1060 003510 013700 000532          FT2C: MOV    DB,R0          ;GET ADDRESS OF DATA BUFFER
1061 003514 010102          MOV    R1,R2
1062 003516 010110          MOV    R1,(R0)        ;LOAD PATTERN
1063 003520 012703 004000          MOV    #4000,R3
1064 003524 005303          FT2D: DEC    R3          ;DELAY
1065 003526 001376          BNE    FT2D
1066 003530 020210          CMP    R2,(R0)        ;SEE IF EXPT=RCVD
1067 003532 001410          BEQ    FT2E           ;IF SO: BR
1068 003534 012737 016526 000650          MOV    #MSG27,ERADD   ;SET ERROR CODE
1069 003542 012737 003510 000674          MOV    #FT2C,SCOLP    ;SET SCOPE ADDRESS
1070 003550 004737 003564          JSR    PC,FT2ER       ;GO DO ERROR
1071 003554 005701          FT2E: TST    R1          ;SEE IF DONE RESET
1072 003556 001453          BEQ    FT2X           ;IF SO: BR
1073 003560 005001          CLR    R1             ;SET ZERO PATTERN
1074 003562 000712          BR     FT2A           ;DO ZERO BITS
1075 003564 000240          FT2ER: NOP
1076 003566 032777 020000 174754          BIT    #20000,@SWR    ;SEE IF PRINT ERROR
1077 003574 001034          BNE    FT2ERB        ;IF NOT: BR
1078 003576 005737 000606          TST    HDRFL          ;SEE ID DONE HEADER
1079 003602 001004          BNE    FT2ERA        ;IF SO: BR
1080 003604 013704 000610          MOV    EMADDR,R4
1081 003610 004737 014414          JSR    PC,TTOUT      ;DO HEADER
1082 003614 012737 000001 000606          FT2ERA: MOV    #1,HDRFL     ;SET FLAG
1083 003622 013704 000650          MOV    ERADD,R4
1084 003626 004737 014414          JSR    PC,TTOUT      ;PRINT ERROR CODE
1085 003632 012704 016456          MOV    #MSG22,R4
1086 003636 004737 014414          JSR    PC,TTOUT      ;PRINT EXPT TAG
1087 003642 010103          MOV    R1,R3
1088 003644 004737 014616          JSR    PC,OCTPE      ;PRINT EXPT
1089 003650 012704 016466          MOV    #MSG23,R4
1090 003654 004737 014414          JSR    PC,TTOUT      ;PRINT RCVD TAG
1091 003660 011003          MOV    (R0),R3
1092 003662 004737 014616          JSR    PC,OCTPE      ;PRINT RCVD

```

1093	003666	005777	174656
1094	003672	100001	
1095	003674	000000	
1096	003676	004737	013334
1097	003702	000240	
1098	003704	000207	
1099	003706	000240	
1100	003710	004737	013370
1101	003714	000137	003074

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

```

1103
1104
1105
1106 003720 012737 020444 000610 FT3:  MOV    #MSFT3,EMADDR  ;SET TEST HEADER
1107 003726 012737 003720 000674      MOV    #FT3,SCOLP
1108 003734 004737 013540      JSR    PC,INIT1      ;GO INIT
1109 003740 052777 020000 174552      BIS    #20000,@CS    ;FORCE UPE =1
1110 003746 000240      NOP
1111 003750 004737 013540      JSR    PC,INIT1      ;GO INIT
1112 003754 005777 174530      TST    @C1           ;SEE IF SC IS RESET
1113 003760 100005      BPL    FT3A          ;IF SO: BR
1114 003762 012737 016564 000650      MOV    #MSG29,ERADD  ;SET ERROR CODE
1115 003770 004737 004054      JSR    PC,FT3ER      ;GO DO ERROR
1116 003774 032777 040000 174506 FT3A:  BIT    #40000,@C1    ;SEE IF TRE IS RESET
1117 004002 001405      BEQ    FT3B          ;IF SO: BR
1118 004004 012737 016613 000650      MOV    #MSG30,ERADD  ;SET ERROR CODE.
1119 004012 004737 004054      JSR    PC,FT3ER      ;GO DO ERROR
1120 004016 017701 174476      FT3B:  MOV    @CS,R1       ;GET CS2
1121 004022 042701 000307      BIC    #307,R1       ;MARK IR/OR
1122 004026 005701      TST    R1           ;SEE IF RESET
1123 004030 001405      BEQ    FT3X          ;IF SO: BR
1124 004032 012737 016643 000650      MOV    #MSG31,ERADD  ;SET ERROR CODE
1125 004040 004737 004054      JSR    PC,FT3ER      ;GO DO ERROR
1126 004044 004737 013370 FT3X:  JSR    PC,ITER      ;GO SEE IF ITERATION
1127 004050 000137 003074      JMP    TSCD2         ;RETURN TO SCHEDULAR
1128
1129
1130 004054 000240      ;ERROR REPORT SUBROUTINE
1131 004056 032777 020000 174464 FT3ER:  NOP
1132 004064 001015      BIT    #20000,@SWR   ;SEE IF PRINT ERROR
1133 004066 005737 000606      BNE    2$           ;IF NOT: BR
1134 004072 001006      TST    HDRFL        ;SEE IF DONE HEADER
1135 004074 013704 000610      BNE    1$           ;IF SO: BR
1136 004100 004737 014414      MOV    EMADDR,R4
1137 004104 005237 000606      JSR    PC,TTOUT      ;PRINT HEADER
1138 004110 013704 000650      INC    HDRFL
1139 004114 004737 014414      1$:  MOV    ERADD,R4
1140 004120 005777 174424      2$:  JSR    PC,TTOUT      ;PRINT ERROR CODE
1141 004124 100001      TST    @SWR         ;SEE IF HALT ON ERROR
1142 004126 000000      BPL    3$           ;IF NOT: BR
1143 004130 000240      HALT
1144 004132 004737 013334      3$:  NOP
1145 004136 000207      JSR    PC,SCOPE     ;GO SEE IF SCOPE
                          RTS    PC                       ;IF NOT: BR

```

```
1147
1148
1149
1150 004140 005737 000604          FT4:  TST      RH17F
1151 004144 001141                BNE      FT5X      ;IF RH70: BR
1152 004146 012737 020476 000610  MOV      @MSFT4,EMADDR ;SET TEST TEST HEADER
1153 004154 012777 000040 174336  MOV      @40,@CS      ;INIT
1154 004162 017700 174344                MOV      @DB,R0      ;READ DB
1155 004166 005777 174326                TST      @CS      ;SEE IF DLT IS SET
1156 004172 100013                BPL      FT4ER      ;IF NOT: BR
1157 004174 005777 174310                TST      @C1      ;SEE IF SC IS SET
1158 004200 100014                BPL      FT4ERA      ;IF NOT: BR
1159 004202 032777 040000 174300  BIT      @40000,@C1  ;SEE IF TRE IS SET
1160 004210 001414                BEQ      FT4ERB      ;IF NOT: BR
1161 004212 004737 013370          FT4X:  JSR      PC,ITER   ;GO SEE IF ITERATION
1162 004216 000137 003074                JMP      TSCD2      ;RETURN TO SCHEDULAR
1163 004222 012737 016673 000650  FT4ER:  MOV      @MSG32,ERADD ;SET ERROR CODE
1164 004230 000407                BR      FT4ERC
1165 004232 012737 016711 000650  FT4ERA: MOV      @MSG33,ERADD ;SET ERROR CODE
1166 004240 000403                BR      FT4ERC
1167 004242 012737 016726 000650  FT4ERB: MOV      @MSG34,ERADD ;SET ERROR CODE.
1168 004250 000240          FT4ERC: NOP
1169 004252 012737 004140 000674  MOV      @FT4,SCOLP  ;SET SCOPE ADDRESS
1170 004260 004737 004054                JSR      PC,FT3ER   ;GO PRINT ERROR
1171 004264 000752                BR      FT4X
```

```
1173
1174
1175
1176 004266 005737 000604 FT5: TST RH17F ;SEE IF RH70
1177 004272 001066 BNE FT5X ;IF SO: BR
1178 004274 012737 020526 000610 MOV #MSG5,EMADDR ;SET TEST HEADER
1179 004302 012737 004310 000674 MOV #FT5A,SCOLP ;SET SCOPE ADDRESS
1180 004310 004737 013540 FT5A: JSR PC,INIT1 ;GO INIT
1181 004314 032777 000100 174176 BIT #100,@CS ;SEE IF IR IS SET
1182 004322 001005 BNE FT5B ;IF SO: BR
1183 004324 012737 016744 000650 MOV #MSG35,ERADD ;SET ERROR CODE
1184 004332 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1185 004336 032777 000200 174154 FT5B: BIT #200,@CS ;SEE IF OR IS RESET
1186 004344 001405 BEQ FT5C ;IF SO: BR
1187 004346 012737 016771 000650 MOV #MSG36,ERADD ;SET ERROR CODE
1188 004354 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1189 004360 012777 000000 174144 FT5C: MOV #0,@DB ;LOAD ZERO INTO SILO
1190 004366 032777 000200 174124 BIT #200,@CS ;SEE THAT OR RESET
1191 004374 001405 BEQ FT5D ;IF IT DOES: BR
1192 004376 012737 017020 000650 MOV #MSG37,ERALD ;SET ERROR CODE
1193 004404 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1194 004410 012777 177777 174114 FT5D: MOV #-1,@DB ;LOAD SILO WITH -1
1195 004416 012700 004000 MOV #4000,R0
1196 004422 032777 000200 174070 FT5E: BIT #200,@CS ;SEE IF OR IS SET
1197 004430 001007 BNE FT5X ;IF SO: BR
1198 004432 005300 DEC R0
1199 004434 001372 BNE FT5E ;AWAIT OR
1200 004436 012737 017020 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1201 004444 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1202 004450 004737 013370 FT5X: JSR PC,ITER ;GO SEE IF ITERATION
1203 004454 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

1205
1206
1207
1208 004460 005737 000604 FT6: TST RH17F
1209 004464 001052 BNE FT6X ;IF RH70: BR
1210 004466 012737 020556 000610 MOV #MSGT6,EMADDR ;SET TEST HEADER
1211 004474 012737 004502 000674 MOV #FT6A,SCOLP ;SET SCOPE ADDRESS
1212 004502 004737 013540 FT6A: JSR PC,INIT1 ;GO INIT
1213 004506 005000 CLR RO ;PRESET DATA
1214 004510 010077 174016 FT6B: MOV RO,8DB ;LOAD SILO
1215 004514 005200 INC RO ;BUMP DATA
1216 004516 022700 000102 CMP #102,RO ;SEE IF FILLED ALL
1217 004522 001372 BNE FT6B ;IF NOT: BR
1218 004524 032777 000100 173766 BIT #100,8CS ;SEE IF IR IS RESET.
1219 004532 001405 BEQ FT6C ;IF SO: BR
1220 004534 012737 017131 000650 MOV #MSG40,ERADD ;SET ERROR CODE
1221 004542 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1222 004546 032777 000200 173744 FT6C: BIT #200,8CS ;SEE IF OR IS SET
1223 004554 001005 BNE FT6D ;IF SO: BR
1224 004556 012737 017057 000650 MOV #MSG38,ERADD ;SET ERROR CODE
1225 004564 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1226 004570 005000 FT6D: CLR RO ;PRESET DATA
1227 004572 017701 173734 FT6E: MOV 8DB,R1 ;READ SILO
1228 004576 020001 CMP RO,R1 ;SEE IF EXPT-RCVD
1229 004600 001010 BNE FT6DE ;IF NOT: BR
1230 004602 005200 INC RO ;BUMP DATA
1231 004604 022700 000102 CMP #102,RO ;SEE IF DONE ALL
1232 004610 001370 BNE FT6E ;IF NOT: BR
1233 004612 004737 013370 FT6X: JSR PC,ITER ;GO SEE IF ITERATION
1234 004616 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
1235
1236 004622 000240 FT6DE: NOP
1237 004624 032777 020000 173716 BIT #20000,8SWR ;SEE IF PRINT ERROR
1238 004632 001032 BNE FT6DEB ;IF NOT: BR
1239 004634 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1240 004640 013701 000610 MOV EMADDR,R1
1241 004644 004737 014414 JSR PC,TTOUT ;PRINT HEADER
1242 004650 005237 000606 INC HDRFL ;SET FLAG
1243 004654 012704 017111 FT6DEA: MOV #MSG39,R4
1244 004660 004737 014414 JSR PC,TTOUT ;PRINT SILO READ ERROR
1245 004664 012704 016456 MOV #MSG22,R4
1246 004670 004737 014414 JSR PC,TTOUT ;PRINT EXPT TAG
1247 004674 010003 MOV RO,R3
1248 004676 004737 014630 JSR PC,OCTP ;PRINT EXPT
1249 004702 012704 016466 MOV #MSG23,R4
1250 004706 004737 014414 JSR PC,TTOUT ;PRINT RCVD TAG
1251 004712 010103 MOV R1,R3
1252 004714 004737 014630 JSR PC,OCTP ;PRINT RCVD
1253 004720 005777 173624 FT6DEB: TST 8SWR ;SEE IF HALT ON ERROR
1254 004724 100001 BPL FT6DEX ;IF NOT: BR
1255 004726 000000 HALT
1256 004730 000207 FT6DEX: RTS PC ;RETURN TO TEST
  
```



```
1278  
1279  
1280  
1281 005026 005737 000604 FT10: TST RH17F  
1282 005032 001034 BNE FT10X ;IF RH70: BR  
1283 005034 012737 020636 000610 MOV #MSFT10,EMADDR ;SET TEST HEADER  
1284 005042 012737 005026 000674 MOV #FT10,SCOLP ;SET SCOPE ADDRESS  
1285 005050 012777 000040 173442 MOV #40,@CS ;INITIALIZE  
1286 005056 012700 000004 MOV #4,RO ;SET NUMBER OF SILO WRITER  
1287 005062 010077 173444 FT10A: MOV RO,@DB ;WRITE SILO  
1288 005066 005300 DEC RO ;SEE IF DONE  
1289 005070 001374 BNE FT10A ;IF NOT: BR  
1290 005072 052777 000040 173420 BIS #40,@CS ;INITIALIZE  
1291 005100 012777 177777 173424 MOV #-1,@DB ;WRITE SILO  
1292 005106 017701 173420 MOV @DB,R1 ;READ SILO 1  
1293 005112 017701 173414 MOV @DB,R1 ;READ SILO 2  
1294 005116 005777 173376 TST @CS ;SEE IF DLT IS SET  
1295 005122 100011 BPL FT10ER ;IF NOT: BR  
1296 005124 004737 013370 FT10X: JSR PC,ITER ;GO SEE IF ITERATION  
1297 005130 005737 000726 TST RHOF ;SEE IF RH11 ONLY  
1298 005134 001402 BEQ FT10XX ;IF NOT: BR  
1299 005136 000137 003162 JMP TEND ;ELSE GO TO END  
1300 005142 000137 003074 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR  
1301 005146 012737 016673 000650 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE  
1302 005154 004737 004054 JSR PC,FT3ER ;GO DO ERROR  
1303 005160 000761 BR FT10X
```

```
1305 ;NOP TEST*****
1306
1307 005162 000240 FT11: NOP
1308 005164 012737 005162 000674 MOV #FT11,SCOLP ;SET SCOPE ADDRESS
1309 005172 004737 013540 JSR PC,INIT1
1310 005176 012737 000300 000716 MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
1311 005204 012737 177777 000620 MOV #-1,FCNT ;SET FC= ALL OVER
1312 005212 012737 177777 000622 MOV #-1,WCNT ;SET WC= ALL OVER
1313 005220 012737 177777 000616 MOV #-1,BADDR ;SET BA= ALL OVER
1314 005226 012737 000001 000636 MOV #1,RDYDX ;SET DELAY
1315 005234 012737 000001 000640 MOV #1,OPDYX ;SET OP DELAY
1316 005242 012737 000001 000710 MOV #1,FUN ;SET NOP FUNCTIONS CODE
1317 005250 004737 012310 JSR PC,EXEC ;GO EXECUTE COMMAND
1318 005254 000240 NOP
1319 005256 012737 020667 000610 MOV #MSFT11,EMADDR
1320 005264 004737 012510 JSR PC,ERCHK ;GO CHECK REGISTER
1321 005270 004737 013370 JSR PC,ITER ;GO SEE IF ITERATIONS
1322 005274 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
1324                                     ;REWIND TEST*****
1325
1326 005300 000240                      FT12:  NOP
1327 005302 012737 005300 000674      MOV    #FT12,SCOLP
1328 005310 004737 013540                JSR    PC,INIT1          ;GO INITIALIZE
1329 005314 052777 001700 173220      BIS    #1700,STC        ;SET TO NRZ,NORMAL
1330 005322 012737 177760 000620      MOV    #-20,FCNT       ;SET FC=20
1331 005330 012737 177770 000622      MOV    #-10,WCNT       ;SET WC=10
1332 005336 012737 021544 000616      MOV    #WDATA,BADDR    ;SET BA=WRITE BUFFER
1333 005344 012737 000007 000710      MOV    #7,FUN          ;SET REWIND OP CODE
1334 005352 004737 012310                JSR    PC,EXEC          ;GO EXECUTE COMMAND
1335 005356 000240                      NOP
1336 005360 032777 020000 173134      FT12A: BIT    #20000,ADS
1337 005366 001374                      BNE    FT12A           ;AWAIT PIP
1338 005370 012737 020707 000610      MOV    #MSFT12,EMADDR
1339 005376 004737 012510                JSR    PC,ERCHK        ;GO CHECK FOR ERROR
1340 005402 004737 013370                JSR    PC,ITER         ;GO SEE IF ITERATION
1341 005406 000137 003074                JMP    TSCD2           ;RETURN TO SCHEDULAR
1342
```

```

1344                                     ;WRITE/READ TEST*****
1345
1346 005412 000240                      FT13:  NOP
1347 005414 012737 000001 000636      MOV     #1,RDYDX
1348 005422 012737 000001 000640      MOV     #1,OPDYX
1349 005430 012737 000100 000624      MOV     #100,RCNT      ;SET RECORD COUNT
1350 005436 012737 020732 000610      MOV     #MSFT13,EMADDR ;SET TEST HEADER
1351 005444 012737 000001 000720      MOV     #1,PATRN
1352 005452 004737 013222                JSR     PC,DSUP        ;SET UP ALL ONES DATA PATTERN
1353 005456 012737 001700 000716      MOV     #1700,UDES    ;SET TO 800 BPI NORMAL
1354 005464 004737 012442                FT13A: JSR     PC,RWND      ;GO REWIND
1355 005470 012737 177600 000620      MOV     #-200,FCNT    ;SET FC
1356 005476 012737 177700 000622      MOV     #-100,WCNT   ;SET WC
1357 005504 012737 021544 000616      MOV     #WDATA,BADDR ;SET BA
1358 005512 012737 000061 000710      MOV     #61,FUN      ;SET WRITE OP-CODE
1359 005520 012737 016225 000626      MOV     #MSG12,ERRP
1360 005526 004737 012310                FT13B: JSR     PC,EXEC    ;GO EXECUTE COMMAND
1361 005532 005037 000674                CLR     SCOLP        ;NO SCOPE LOOP
1362 005536 004737 012510                JSR     PC,ERCHK     ;GO CHECK ERROR
1363 005542 005337 000624                DEC     RCNT         ;SEE IF DONE ALL
1364 005546 001367                BNE     FT13B        ;IF NOT: BR
1365 005550 012737 000100 000624      MOV     #100,RCNT    ;SET RECORD COUNT
1366 005556 012737 023256 000616      MOV     #RDATA,BADDR
1367 005564 062737 000200 000616      ADD     #200,BADDR   ;SET BA
1368 005572 012737 000077 000710      MOV     #77,FUN      ;SET READ REVERSE OP-CPDE
1369 005600 012737 016243 000626      MOV     #MSG13,ERRP
1370 005606 004737 012310                FT13C: JSR     PC,EXEC    ;GO EXECUTE COMMAND
1371 005612 004737 012510                JSR     PC,ERCHK     ;GO CHECK ERROR
1372 005616 005337 000624                DEC     RCNT         ;SEE IF READ ALL
1373 005622 001371                BNE     FT13C        ;IF NOT:BR
1374 005624 162737 000200 000616      SUB     #200,BADDR   ;SET BA
1375 005632 012737 000071 000710      MOV     #71,FUN      ;SET READ FORWARD OP-CODE
1376 005640 012737 016270 000626      MOV     #MSG14,ERRP
1377 005646 012737 000100 000624      MOV     #100,RCNT    ;SET RECORD COUNT
1378 005654 004737 012310                FT13D: JSR     PC,EXEC    ;GO EXECUTE COMMAND
1379 005660 004737 012510                JSR     PC,ERCHK     ;GO CHECK ERRORS
1380 005664 005337 000624                DEC     RCNT         ;SEE IF DONE ALL
1381 005670 001371                BNE     FT13D        ;IF NOT:BR
1382 005672 032737 002000 000716      BIT     #2000,UDES   ;SEE IF DONE PE
1383 005700 001007                BNE     FT13X        ;IF SO: BR
1384 005702 012737 002300 000716      MOV     #2300,UDES   ;SET PE MODE
1385 005710 012737 000100 000624      MOV     #100,RCNT    ;RESET RECORD COUNT
1386 005716 000662                BR      FT13A        ;GO DO NEXT DENSITY
1387 005720 000137 003074                FT13X: JMP     TSCD2    ;RETURN TO SCHEDULAR
  
```

```

1389                                     ;SPACE TEST*****
1390
1391 005724 000240                      FT14:  NOP
1392 005726 012737 020761 000610      MOV      #MSFT14,EMADDR ;SET TEST HEADER
1393 005734 012737 001700 000716      MOV      #1700,UDES    ;SET NRZ,NORMAL
1394 005742 004737 012442              FT14A1: JSR      PC,RWND  ;GO INITIALIZE
1395 005746 012737 000100 000624      MOV      #100,RCNT    ;SET NUMBER OF RECORDER
1396 005754 012737 177777 021544      MOV      #-1,WDATA   ;SET DATA PATTERN
1397 005762 012737 177700 000620      MOV      #-100,FCNT  ;PRESET FRAME CNT
1398 005770 012737 177740 000622      MOV      #-40,WCNT   ;PRESET WORD CNT
1399 005776 004737 013540              FT14A:  JSR      PC,INIT1 ;GO REWIND
1400 006002 012737 001000 000640      MOV      #1000,OPDYX
1401 006010 012737 040000 000636      MOV      #40000,RDYDX
1402 006016 012737 000061 000710      MOV      #61,FUN      ;SET WRITE OP-CODE
1403 006024 012737 102300 000660      MOV      #102300,STMSK ;MASK DATA RELATED ERRORS
1404 006032 052777 000010 172460      BIS      #10,BCS     ;INHIBIT BUS ADDRESS INCREMENT
1405 006040 004737 012310              JSR      PC,EXEC      ;GO EXECUTE COMMAND
1406 006044 012737 017350 000626      MOV      #MSG46,ERRP  ;SET ERROR CODE
1407 006052 004737 012510              JSR      PC,ERCHK    ;GO CHECK ERRORS
1408 006056 005737 000712              TST      SERFL       ;SEE IF ERROR
1409 006062 001402                      BEQ      FT14A2      ;IF NOT: BR
1410 006064 000137 006550              JMP      FT14X       ;ELSE EXIT
1411 006070 005337 000620              FT14A2: DEC      FCNT  ;BUMP FC
1412 006074 032737 000001 000620      BIT      #1,FCNT    ;SEE IF SHOULD BUMP WC
1413 006102 001403                      BEQ      FT14A3      ;IF NOT: BR
1414 006104 162737 000001 000622      SUB      #1,WCNT    ;BUMP WC
1415 006112 005337 000624              FT14A3: DEC      RCNT  ;SEE IF DONE ALL
1416 006116 001327                      BNE      FT14A       ;WRITE ALL RECORDS
1417 006120 012737 000100 000632      MOV      #100,RRD    ;PRESET RECORD POSITION
1418 006126 012737 000176 000634      MOV      #176,RFD
1419 006134 012737 177701 000642      MOV      #-77,SCNT  ;SET SPACE AMOUNT
1420 006142 012737 000033 000710      FT14B:  MOV      #33,FUN ;SET OP-CODE SPACE REVERSE
1421 006150 004737 012310              JSR      PC,EXEC    ;GO EXECUTE COMMAND
1422 006154 012737 017421 000626      MOV      #MSG48,ERRP ;SET ERROR CODE
1423 006162 004737 012510              JSR      PC,ERCHK   ;GO CHECK ERRORS
1424 006166 005737 000712              TST      SERFL     ;SEE IF ERROR
1425 006172 001166                      BNE      FT14X      ;IF SO: BR
1426 006174 004737 006270              JSR      PC,FT14RR  ;GO READ REVERSE * CHECK DATA
1427 006200 000240                      NOP
1428 006202 012737 000031 000710      MOV      #31,FUN    ;SET SPACE FORWARD OP-CODE
1429 006210 005237 000642              INC      SCNT       ;SET SPACE AMOUNT
1430 006214 001555                      BEQ      FT14X      ;IF DONE: BR
1431 006216 004737 012310              JSR      PC,EXEC    ;GO EXECUTE COMMAND
1432 006222 012737 017374 000626      MOV      #MSG47,ERRP ;SET ERROR CODE
1433 006230 004737 012510              JSR      PC,ERCHK   ;GO CHECK ERROR
1434 006234 005737 000712              TST      SERFL     ;SEE IF ERROR FLAG
1435 006240 001143                      BNE      FT14X      ;IF NO: BR
1436 006242 004737 006332              JSR      PC,FT14RF  ;GO READ FORWARD FOR POSITION CHECK
1437 006246 000240                      NOP
1438 006250 005237 000642              INC      SCNT       ;DECREMENT SPACE AMOUNT
1439 006254 001535                      BEQ      FT14X      ;IF DONE: BR
1440 006256 005237 000632              INC      RRD        ;BUMP DATA EXPT
1441 006262 005337 000634              DEC      RFD        ;BUMP DATA EXPT
1442 006266 000725                      BR       FT14B
1443 006270 000240              FT14RR: NOP
1444 006272 012737 023256 000616      MOV      #RDATA,BADDR ;SET BA

```

1445	006300	012737	000077	000710		MOV	#77,FUN	;SET READ REVERSE OP-CODE
1446	006306	004737	012310			JSR	PC,EXEC	;GO EXECUTE COMMAND
1447	006312	000240				NOP		
1448	006314	013705	000632			MOV	RRD,R5	
1449	006320	020577	172172			CMP	R5,@FC	;SEE IF CORRECT RECORD
1450	006324	001020				BNE	FT14RER	;IF NOT: BR
1451	006326	000137	006360			JMP	FT14EC	;GO CLEAR RM11 ERROR BIT
1452	006332	000240			FT14RF:	NOP		
1453	006334	012737	000071	000710		MOV	#71,FUN	;SET READ FORWARD OP-CODE
1454	006342	004737	012310			JSR	PC,EXEC	;GO EXECUTE COMMAND
1455	006346	013705	000634			MOV	RFD,R5	
1456	006352	020577	172140			CMP	R5,@FC	;SEE IF CORRECT RECORD
1457	006356	001003				BNE	FT14RER	;IF NOT: BR
1458	006360	004737	013540		FT14EC:	JSR	PC,INIT1	;CLEAR RM
1459	006364	000207				RTS	PC	;RETURN
1460	006366	000240			FT14RER:	NOP		
1461	006370	032777	020000	172152		BIT	#20000,@SWR	;SEE IF PRINT INHIBITED
1462	006376	001060				BNE	FT14R3	;IF SO: BR
1463	006400	012704	020761			MOV	#MSFT14,R4	
1464	006404	004737	014414			JSR	PC,TTOUT	;PRINT HEADER
1465	006410	012704	016003			MOV	#MSG9,R4	
1466	006414	004737	014414			JSR	PC,TTOUT	;PRINT ERROR TYPE
1467	006420	012704	016443			MOV	#MSG20,R4	;SET NRZ TAG POINTER
1468	006424	032737	002000	000716		BIT	#2000,UDES	;SEE IF PE
1469	006432	001402				BEQ	FT14R0	;IF NOT: BR
1470	006434	012704	016451			MOV	#MSG21,R4	;ELSE SET PE TAG POINTER
1471	006440	004737	014414		FT14R0:	JSR	PC,TTOUT	;PRINT TAG
1472	006444	032737	000002	000710		BIT	#2,FUN	;SEE IF READ REVERSE
1473	006452	001003				BNE	FT14R1	;IF SO: BR
1474	006454	012704	016423			MOV	#MSG17,R4	
1475	006460	000402				BR	FT14R2	;GO PRINT
1476	006462	012704	016403		FT14R1:	MOV	#MSG16,R4	
1477	006466	004737	014414		FT14R2:	JSR	PC,TTOUT	;PRINT FRWD/REV
1478	006472	012704	016456			MOV	#MSG22,R4	
1479	006476	004737	014414			JSR	PC,TTOUT	;PRINT EXPT TAG
1480	006502	010503				MOV	R5,R3	
1481	006504	042703	177700			BIC	#177700,R3	;MASK RECORD NUMBER
1482	006510	004737	014630			JSR	PC,OCTP	;PRINT EXPT RECORD NUMBER
1483	006514	012704	016466			MOV	#MSG23,R4	
1484	006520	004737	014414			JSR	PC,TTOUT	;PRINT RCVD TAG
1485	006524	017703	171766			MOV	@FC,R3	
1486	006530	042703	177700			BIC	#177700,R3	;MASK RECORD NUMBER
1487	006534	004737	014630			JSR	PC,OCTP	;PRINT ACTUAL RECORD NUMBER
1488	006540	005777	172004		FT14R3:	TST	@SWR	;SEE IF HALT ON ERROR
1489	006544	100001				BPL	FT14X	;IF NOT: BR
1490	006546	000000				HALT		
1491	006550	032737	002000	000716	FT14X:	BIT	#2000,UDES	;SEE IF DONE PE
1492	006556	001005				BNE	FT14XX	;IF SO: BR
1493	006560	012737	002300	000716		MOV	#2300,UDES	;SET TO PE
1494	006566	000137	005742			JMP	FT14A1	;DO IN PE
1495	006572	000137	003074		FT14XX:	JMP	TSCD2	;RETURN TO SCHEDULAR

```
1497                                     ;ERASE TEST*****
1498
1499 006576 000240                      FT15:  NOP
1500 006600 005037 000660              CLR      STMSK
1501 006604 012737 000100 000636      MOV      #100,RDYDX
1502 006612 012737 000010 000640      MOV      #10,OPDYX
1503 006620 012737 021003 000610      MOV      #MSFT15,EMADDR ;SET TEST HEADER
1504 006626 004737 012442              JSR      PC,RWND ;REWIND
1505 006632 012737 023256 000616      MOV      #RDATA,BADDR ;SET BA
1506 006640 012737 001700 000716      MOV      #1700,UDES ;SET NRZ, NORMAL
1507 006646 012737 000025 000710 FT15A: MOV      #25,FUN ;SET ERASE OP-CODE
1508 006654 012737 000454 000624      MOV      #300.,RCNT ;++B SET TO ERASE 300 TIMES
1509 006662 004737 012310              FT15B: JSR      PC,EXEC ;GO EXECUTE COMMAND
1510 006666 012737 017350 000626      MOV      #MSG46,ERRP ;SET ERROR CODE
1511 006674 004737 012510              JSR      PC,ERCHK ;GO CHECK ERRORS
1512 006700 005737 000712              TST      SERFL ;SEE IF ANY ERRORS
1513 006704 001032                      BNE      FT15X ;IF SO EXIT
1514 006706 005337 000624              DEC      RCNT ;SEE IF DONE ERASING
1515 006712 001363                      BNE      FT15B ;IF NOT: BR
1516 006714 000240                      NOP
1517 006716 004737 012442              JSR      PC,RWND ;REWIND
1518 006722 012737 177600 000622      MOV      #-200,WCNT ;SET WC
1519 006730 012737 000071 000710      MOV      #71,FUN ;SET READ FORWARD OP-CODE
1520 006736 012737 000040 000636      MOV      #40,RDYDX ;SET DELAY
1521 006744 004737 012310              JSR      PC,EXEC ;GO EXECUTE COMMAND
1522 006750 000240                      NOP
1523 006752 012737 020022 000626      MOV      #MSG60,ERRP ;SET ERROR CODE
1524 006760 012737 020000 000660      MOV      #20000,STMSK
1525 006766 004737 012510              JSR      PC,ERCHK ;GO CHECK ERRORS
1526                                     ;*****
1527
1528                                     ;THIS CODE ADDED TO FORM REV C
1529
1530                                     ;THE SSC BIT AND THE PIP BIT IN THE DRIVE STATUS REG
1531                                     ;SHOULD NOT BE SET CONCURRENTLY
1532
1533                                     ;*****
1534
1535 006772 012737 000100 000636 FT15X: MOV      #100,RDYDX ;SET DELAY
1536 007000 012737 000010 000640      MOV      #10,OPDYX
1537 007006 012737 000020 000624      MOV      #20,RCNT ;SET UP FOR 20 ERASES
1538 007014 012737 023256 000616 1$:  MOV      #RDATA,BADDR ;SET UP BUSS ADDRS
1539 007022 012737 001700 000716      MOV      #1700,UDES ;SET UP TAPE CONTROL
1540 007030 012737 000025 000710      MOV      #25,FUN ;SET FUN FOR ERASE
1541 007036 004737 012310              JSR      PC,EXEC ;GO EXECUTE CMD
1542 007042 005337 000624              DEC      RCNT ;DECREMENT THE NUMBER OF EXECUTES
1543 007046 001362                      BNE      1$ ;BRANCH IF MORE LEFT
1544 007050 052777 001700 171464      BIS      #1700,@TC
1545 007056 012777 177760 171432      MOV      #-20,@FC
1546 007064 012777 177770 171420      MOV      #-10,@WC
1547 007072 012777 021544 171414      MOV      #WDATA,@BA
1548 007100 012777 000007 171402      MOV      #7,@C1 ;DO REWIND
1549 007106 000240                      NOP
1550 007110 032777 000100 171404 2$:  BIT      #100,@DS ;WAIT FOR SSC
1551 007116 001774                      BEQ      2$
1552 007120 017737 171376 000652      MOV      @DS,TEMP1 ;READ DRIVE STATUS REG IMMEDIATELY
```

1553	007126	032737	020000	000652		BIT	#20000,TEMP1	;CHECK FOR PIP
1554	007134	001420				BEQ	FT15XX	;BRANCH IF NOT SET
1555	007136	052737	000001	007206		BIS	#1,TAG	;SET FLAG FOR ERROR
1556	007144	012704	016375			MOV	#MSG158,R4	
1557	007150	004737	014414			JSR	PC,TTOUT	
1558	007154	010703				MOV	PC,R3	
1559	007156	062703	000010			ADD	#10,R3	
1560	007162	004737	014616			JSR	PC,OCTPE	
1561								
1562	007166	004737	012772			JSR	PC,ERPTB1	;GO PRINT ERROR
1563								;DS REG REPORTED IS ITS CONTENTS
1564								;AT THE TIME OF THE ERROR (5 LINES OF CODE BACK)
1565	007172	005037	007206			CLR	TAG	;CLEAR FLAG
1566	007176	004737	013370		FT15XX:	JSR	PC,ITER	;CHECK FOR ITERATIONS
1567	007202	000137	003074			JMP	TSCD2	;GO TO SCHEDULAR
1568								
1569	007206	000000			TAG:	.WORD	0	;C FLAG FOR ERROR ROUTINE

```

1571                                     ;TAPE MARK WRITE/READ TEST*****
1572
1573 007210 000240                      FT16:  NOP
1574 007212 012737 000001 000636      MOV     #1,RDYDX
1575 007220 012737 001000 000640      MOV     #1000,OPDYX
1576 007226 012737 021025 000610      MOV     #MSFT16,EMADDR ;SET HEADER
1577 007234 012737 001700 000716      MOV     #1700,UDES ;SET TO NRZ,NORMAL,ODD
1578 007242 004737 012442              FT16A: JSR     PC,RWND ;INIT AND REWIND SLAVE
1579 007246 012737 177760 000620      FT16B: MOV     #-20,FCNT ;FC=20
1580 007254 012737 177770 000622      MOV     #-10,WCNT ;WC=10
1581 007262 012737 000027 000710      MOV     #27,FUN ;SET WRITE TAPE MARK OP-CODE
1582 007270 004737 012310              JSR     PC,EXEC ;GO EXECUTE COMMAND
1583 007274 012737 001000 000660      MOV     #1000,STMSK ;SET FOR FCE MASK
1584 007302 012737 016315 000626      MOV     #MSG15,ERRP ;SET ERROR CODE
1585 007310 004737 012510              JSR     PC,ERCHK ;GO CHECK ERROR
1586 007314 004737 013164              JSR     PC,TMCHK ;GO SEE IF TM SET
1587 007320 012737 000077 000710      MOV     #77,FUN ;SET READ REVERSE OP-CODE
1588 007326 004737 012310              JSR     PC,EXEC ;GO EXECUTE COMMAND
1589 007332 012737 001000 000660      MOV     #1000,STMSK ;SET FCE ERROR MASK
1590 007340 012737 016243 000626      MOV     #MSG13,ERRP ;SET ERROR CODE
1591 007346 004737 012510              JSR     PC,ERCHK ;GO CHECK ERRORS
1592 007352 004737 013164              JSR     PC,TMCHK ;GO SEE IF TM SET
1593 007356 012737 000071 000710      MOV     #71,FUN ;SET READ FORWARD OP-CODE
1594 007364 004737 012310              JSR     PC,EXEC ;GO EXECUTE COMMAND
1595 007370 012737 016270 000626      MOV     #MSG14,ERRP ;SET ERROR CODE
1596 007376 004737 012510              JSR     PC,ERCHK ;TO CHECK ERRORS
1597 007402 004737 013164              JSR     PC,TMCHK ;GO SEE IF TM SET
1598 007406 032737 002000 000716      BIT     #2000,UDES ;SEE IF DONE PE
1599 007414 001004                      BNE     FT16X ;IF SO: BR
1600 007416 012737 002300 000716      MOV     #2300,UDES ;SET PE, NORMAL
1601 007424 000706                      BR      FT16A ;DO IN PE
1602 007426 004737 013370              FT16X: JSR     PC,ITER ;DO ITERATIONS
1603 007432 000137 003074              JMP     TSCD2 ;RETURN TO SCHEDULAR
1604

```

```

1606
1607
1608 ;TAPE MARK SPACE TEST*****
1609 007436 005037 000624 FT17: CLR RCNT
1610 007442 012737 021066 000610 MOV #MSFT17,EMADDR ;SET HEADER
1611 007450 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ
1612 007456 004737 012442 FT17A: JSR PC,RWIND ;REWIND TAPE
1613 007462 012737 000027 000710 FT17B: MOV #27,FUN
1614 007470 012737 040000 000636 MOV #40000,RDYDX ;SET DRY DELAY
1615 007476 012737 040000 000640 MOV #40000,OPDYX ;SET OP DELAY
1616 007504 004737 012310 JSR PC,EXEC ;GO WRITE TM
1617 007510 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1618 007516 012737 016315 000626 MOV #MSG15,ERRP ;SET ERROR TYPE
1619 007524 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1620 007530 005737 000712 TST SERFL ;SEE IF ERROR
1621 007534 001137 BNE FT17X ;IF SO: BR
1622 007536 004737 013164 JSR PC,TMCHK ;GO SEE IF TM SET
1623 007542 000240 NOP
1624 007544 000240 NOP
1625 007546 032737 000100 000624 BIT #100,RCNT ;SEE IF DONE PATTERN
1626 007554 001045 BNE FT17D ;IF SO: BR
1627 007556 062737 000020 000624 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1628 007564 013737 000624 000652 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1629 007572 012737 177600 000622 MOV #-200,WCNT ;WC=128
1630 007600 012737 177400 000620 MOV #-400,FCNT ;FC=256
1631 007606 012737 021544 000616 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1632 007614 012737 000061 000710 MOV #61,FUN ;SET WRITE OP CODE
1633 007622 000240 FT17C: NOP
1634 007624 000240 NOP
1635 007626 004737 012310 JSR PC,EXEC ;GO WRITE
1636 007632 012737 016225 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1637 007640 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1638 007646 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1639 007652 005737 000712 TST SERFL ;SEE IF ERROR
1640 007656 001066 BNE FT17X ;IF SO: BR
1641 007660 005337 000652 DEC TEMP1 ;SEE IF DONE ALL
1642 007664 001356 BNE FT17C ;IF NOT: BR
1643 007666 000675 BR FT17B ;ELSE GO DO TM
1644 007670 000240 FT17D: NOP
1645 007672 012737 000033 000710 MOV #33,FUN ;SET SPACE REVERSE
1646 007700 012737 016403 000626 MOV #MSG16,ERRP ;SET ERROR CODE
1647 007706 012737 177600 000642 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1648 007714 012737 000005 000624 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1649 007722 004737 013540 FT17E: JSR PC,INIT1 ;GO INIT
1650 007726 004737 012310 JSR PC,EXEC ;GO SPACE
1651 007732 012737 001000 000660 MOV #1000,STMSK ;SET ERROR MASK
1652 007740 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1653 007744 005737 000712 TST SERFL ;SEE IF ERROR
1654 007750 001031 BNE FT17X ;IF SO: BR
1655 007752 004737 013164 JSR PC,TMCHK ;GO SEE IF TM SET
1656 007756 005337 000624 DEC RCNT ;SEE IF DONE SPACES
1657 007762 001357 BNE FT17E ;IF NOT: BR
1658 007764 022737 000031 000710 CMP #31,FUN ;SEE IF DONE FORWARD
1659 007772 001407 BEQ FT17F ;IF SO: BR
1660 007774 012737 016423 000626 MOV #MSG17,ERRP ;SET ERROR CODE
1661 010002 012737 000031 000710 MOV #31,FUN ;SET TO SPACE FORWARD
  
```

1662	010010	000736				BR	FT17D1		;DO FORWARD
1663	010012	032737	002000	000716	FT17F:	BIT	#2000, UDES		;SEE IF DONE PE
1664	010020	001005				BNE	FT17X		;IF SO; BR
1665	010022	012737	002300	000716		MOV	#2300, UDES		;SET TO PE
1666	010030	000137	007456			JMP	FT17A		;GO PE
1667	010034	000137	003074		FT17X:	JMP	TSCD2		;RETURN TO SCHEDULAR

```

1669
1670
1671
1672 010040 000240
1673 010042 012737 021114 000610
1674 010050 012737 001700 000716
1675 010056 004737 012442
1676 010062 012737 000003 000720
1677 010070 004737 013222
1678 010074 012737 021544 000616
1679 010102 012737 177400 000620
1680 010110 012737 177600 000622
1681 010116 012737 000061 000710
1682 010124 004737 012310
1683 010130 012737 017350 000626
1684 010136 004737 012510
1685 010142 005737 000712
1686 010146 001042
1687 010150 012737 016403 000626
1688 010156 012737 000057 000710
1689 010164 062737 000376 000616
1690 010172 004737 012310
1691 010176 004737 012510
1692 010202 012737 016423 000626
1693 010210 012737 000051 000710
1694 010216 162737 000376 000616
1695 010224 004737 012310
1696 010230 004737 012510
1697 010234 032737 002000 000716
1698 010242 001004
1699 010244 012737 002300 000716
1700 010252 000701
1701 010254 004737 013370
1702 010260 000137 003074

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

```

1704
1705
1706
1707 010264 012737 021145 000610 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1708 010272 004737 012442 FT21A: JSR PC,RWND ;GO REWIND
1709 010276 012737 000003 000720 MOV #3,PATRN
1710 010304 004737 013222 JSR PC,DSUP ;GO SET PATTERN 3
1711 010310 012737 021544 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1712 010316 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1713 010324 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1714 010332 012737 001700 000716 MOV #1700,UDES ;SET NRZ, NORMAL
1715 010340 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1716 010346 004737 012310 JSR PC,EXEC ;GO DO WRITE 1
1717 010352 012737 016225 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1718 010360 004737 012510 JSR PC,ERCHK ;GO CHECK FOR ERROR
1719 010364 004737 012310 JSR PC,EXEC ;YES DO WRITE 2
1720 010370 004737 012510 JSR PC,ERCHK ;YES CHECK FOR ERROR
1721 010374 000240 NOP
1722 010376 004737 012442 JSR PC,RWND ;GO REWIND
1723 010402 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400(10)
1724 010410 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200(10)
1725 010416 004737 012310 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1726 010422 000240 FT21SCP: NOP
1727 010424 004737 012442 JSR PC,RWND ;REWIND
1728 010430 012737 023256 000616 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1729 010436 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400
1730 010444 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200
1731 010452 012737 000071 000710 MOV #71,FUN ;SET READ OP-CODE
1732 010460 004737 012310 JSR PC,EXEC ;GO READ RECORD 1
1733 010464 012737 016270 000626 MOV #MSG14,ERRP ;SET ERROR CODE
1734 010472 004737 012510 JSR PC,ERCHK ;GO CHECK FOR ERROR
1735 010476 000240 NOP
1736 010500 052777 000010 170012 BIS #10,SCS ;INHIBIT BA INCREMENT
1737 010506 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1738 010514 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1739 010522 004737 012310 JSR PC,EXEC ;GO READ RECORD 2
1740 010526 022777 001440 167762 CMP #800.,@FC ;SEE IF READ RECORD 2 OK
1741 010534 001424 BEQ FT21X ;IF SO: BR
1742 010536 022777 001441 167752 CMP #801.,@FC ;BRANCH IF IN GREY AREA
1743 010544 001420 BEQ FT21X
1744 010546 022777 001440 167742 1$: CMP #800.,@FC ;BRANCH IF ERASE HEAD REVERSED
1745 010554 101404 BLOS FT21B ;IF SO: BR
1746 010556 012737 017243 000650 MOV #MSG44,ERADD ;SET ERASE HEAD INOPERATIVE ERROR CODE
1747 010564 000403 BR FT21C
1748 010566 012737 017273 000650 FT21B: MOV #MSG45,ERADD ;SET ERASE HEAD REVERSED ERROR CODE
1749 010574 012737 010422 000674 FT21C: MOV #FT21SCP,SCOLP ;SET SCOPE ADDRESS
1750 010602 004737 004054 JSR PC,FT3ER ;GO PRINT ERROR
1751 010606 004737 013370 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1752 010612 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
1753
1754

```

```

1756                                     ;BUFFERED COMMAND TEST*****
1757
1758 010616 012737 021174 000610 FT22: MOV #MSFT22,EMADDR ;SET TEST HEADER
1759 010624 004737 012442 JSR PC,RWND ;GO REWIND
1760 010630 012700 000003 MOV #3,R0 ;SET NUMBER OF WRITES
1761 010634 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ NORMAL
1762 010642 012737 021544 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1763 010650 012737 177000 000620 MOV #-1000,FCNT ;SET FC=1000
1764 010656 012737 177400 000622 MOV #-400,WCNT ;SET WC=400
1765 010664 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1766 010672 004737 012310 FT22A: JSR PC,EXEC ;GO DO WRITE
1767 010676 005300 DEC R0 ;SEE IF DONE ALL
1768 010700 001374 BNE FT22A ;IF NOT: BR
1769 010702 000240 NOP
1770 010704 012777 000007 167576 MOV #7,BC1 ;START REWIND
1771 010712 032777 000200 167602 FT22B: BIT #200,BDS
1772 010720 001774 BEQ FT22B
1773 010722 004737 013540 JSR PC,INIT1 ;INITIALIZE
1774 010726 012737 000010 000635 MOV #10,RDYDX ;SET LONG READY DELAY
1775 010734 004737 012310 JSR PC,EXEC ;ISSUE BUFFERED WRITE
1776 010740 000240 NOP
1777 010742 012737 017446 000626 MOV #MSG49,ERRP ;SET ERROR CODE
1778 010750 012737 102300 000660 MOV #102300,STMSK ;MARK DATA ERROR
1779 010756 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1780 010762 032777 000002 167532 BIT #2,BDS ;SEE IF BOT IS SET
1781 010770 001410 BEQ FT22X ;IF NOT: BR
1782 010772 012737 017474 000650 MOV #MSG50,ERADD ;SET ERROR CODE
1783 011000 012737 010616 000674 MOV #FT22,SCOLP
1784 011006 004737 004054 JSR PC,FT3ER
1785 011012 004737 013370 FT22X: JSR PC,ITER ;GO DO ERROR
1786 011016 000137 003074 JMP TSCD2 ;GO SEE IF ITERATION
1787 ;RETURN TO SCHEDULAR
1788

```

```

1790                                     ;READ-IN PRESET TEST*****
1791
1792 011022 005737 000614           FT23: TST      SLVN      ;SEE IF SLAVE SELECT=0
1793 011026 001103                   BNE      FT23X    ;IF NOT:BR
1794 011030 012737 021231 000610    MOV      #MSFT23,EMADDR ;SET TEST HEADER
1795 011036 004737 013540           JSR      PC,INIT1  ;GO INIT
1796 011042 012737 001700 000716    MOV      #1700,UDES   ;SET TO NRZ NORMAL
1797 011050 012737 021544 000616    MOV      #WDATA,BADDR ;SET BA=WRITE BUFFER
1798 011056 012737 177400 000620    MOV      #-400,FCNT  ;SET FC=400
1799 011064 012737 177600 000622    MOV      #-200,WCNT  ;SET WC=200
1800 011072 012737 000061 000710    MOV      #61,FUN     ;SET WRITE OP-CODE
1801 011100 004737 012310           JSR      PC,EXEC    ;GO DO WRITE
1802 011104 000240                   NOP
1803 011106 004737 013540           JSR      PC,INIT1  ;INITIALIZE
1804 011112 012737 000021 000710    MOV      #21,FUN     ;SET READ-IN PRESET OP CODE
1805 011120 004737 012310           JSR      PC,EXEC    ;GO DO COMMAND
1806 011124 005000                   CLR      R0
1807 011126 012703 000004           MOV      #4,R3      ;SET MULT
1808 011132 032777 020000 167362    FT23A: BIT      #20000,#DS ;SEE IF PIP RESET
1809 011140 001404                   BEQ      FT23B     ;IF SO: BR
1810 011142 005300                   DEC      R0
1811 011144 001372                   BNE      FT23A     ;AWAIT PIP RESET
1812 011146 005303                   DEC      R3
1813 011150 001370                   BNE      FT23A     ;DELAY
1814 011152 032777 000002 167342    FT23B: BIT      #2,#DS ;SEE IF BOT
1815 011160 001010                   BNE      FT23C     ;IF SO: BR
1816 011162 012737 017532 000650    MOV      #MSG51,ERADD ;SET ERROR CODE
1817 011170 012737 011022 000674    MOV      #FT23,SCOLP
1818 011176 004737 004054           JSR      PC,FT3ER   ;GO DO ERROR
1819 011202 012701 141000           FT23C: MOV      #141000,R1 ;SET EXPT TC
1820 011206 013700 000542           MOV      TC,R0      ;SET TC ADDRESS
1821 011212 020110                   CMP      R1,(R0)    ;SEE IF EXPT=RCVD
1822 011214 001410                   BEQ      FT23X     ;IF SO: BR
1823 011216 012737 017566 000650    MOV      #MSG52,ERADD ;SET ERROR CODE
1824 011224 012737 011022 000674    MOV      #FT23,SCOLP ;CLEAR SCOPE ADDRESS
1825 011232 004737 003564           JSR      PC,FT2ER   ;GO DO ERROR
1826 011236 000137 003074           FT23X: JMP      TSCD2 ;RETURN TO SCHEDULAR
1827
1828

```

```
1830
1831
1832
1833 011242 012737 021264 000610 FT24: MOV #MSFT24,EMADDR ;SET ERROR MSG HEADER
1834 011250 004737 012442 JSR PC,RWIND ;REWIND SLAVE
1835 011254 012737 000001 070720 MOV #1,PATRN ;SELECT PATTERN
1836 011262 004737 013222 JSR PC,DSUP ;GO DO DATA SETUP
1837 011266 012737 021544 000616 MOV #WDATA,BADDR ;SET BUS ADDRESS,
1838 011274 012737 177400 000620 MOV #-400,FCNT ;FRAME COUNT,
1839 011302 012737 177600 000622 MOV #-200,WCNT ;WORD COUNT,
1840 011310 012737 001700 000716 MOV #1700,UDES ;& SLAVE DESC = NRZ NORMAL
1841 011316 012737 000061 000710 MOV #61,FUN ;LOAD OP CODE WRITE FWD
1842 011324 004737 012310 JSR PC,EXEC ;GO EXECUTE COMMAND
1843 011330 012737 017350 000626 MOV #MSG46,ERRP ;SET ERROR MSG ADDRESS
1844 011336 004737 012510 JSR PC,ERCHK ;GO CHECK ERRORS
1845 011342 005737 000712 TST SERFL ;BRANCH IF AN ERROR OCCURRED
1846 011346 001026 BNE FT24X
1847 011350 004737 012442 JSR PC,RWIND ;REWIND SLAVE
1848 011354 012737 023256 000616 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1849 011362 012737 002300 000716 MOV #2300,UDES ;SET SLAVE DESC = PE,NORMAL
1850 011370 012737 000071 000710 MOV #71,FUN ;SET OP CODE = READ FWD
1851 011376 004737 012310 JSR PC,EXEC ;GO READ RECORD
1852 011402 032777 000040 167112 BIT #40,SDS ;BRANCH ID PES BIT CLEARED
1853 011410 001405 BEQ FT24X
1854 011412 012737 020121 000650 MOV #MSG63,ERADD
1855 011420 004737 004054 JSR PC,FT3ER ;GO PROCESS ERROR
1856 011424 004737 013370 FT24X: JSR PC,ITER
1857 011430 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULER
1858
```

```

1860
1861 ;AUTO-DENSITY SELECT TEST: WRITE-PE,READ-NRZ
1862 011434 012737 021342 000610 FT25: MOV #MSFT25,EMADDR ;SET ERROR MESSAGE ADDRESS
1863 011442 004737 012442 JSR PC,RWIND ;REWIND SLAVE
1864 011446 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN
1865 011454 004737 013222 JSR PC,DSUP ;GO DO DATA SETUP
1866 011460 012737 021544 000616 MOV #WDATA,BADDR ;SET BUS ADDRESS
1867 011466 012737 177400 000620 MOV #-400,FCNT ;FRAME COUNT,
1868 011474 012737 177600 000622 MOV #-200,WCNT ;WORD COUNT,
1869 011502 012737 002300 000716 MOV #2300,UDES ;& SLAVE DESC = PE,NORMAL
1870 011510 012737 000061 000710 MOV #61,FUN ;LOAD WRITE OP CODE
1871 011516 004737 012310 JSR PC,EXEC ;GO EXECUTE WRITE
1872 011522 012737 017350 000626 MOV #MSG46,ERRP ;SET ERROR MSG HDR
1873 011530 004737 012510 JSR PC,ERCHK ;GO CHECK FOR ERRORS
1874 011534 005737 000712 TST SERFL ;BRANCH IF ERROR OCURRED
1875 011540 001026 BNE FT25X
1876 011542 004737 012442 JSR PC,RWIND ;REWIND SLAVE
1877 011546 012737 023256 000616 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1878 011554 012737 001700 000716 MOV #1700,UDES ;SET SLAVE DESC = NRZ,NORMAL
1879 011562 012737 000071 000710 MOV #71,FUN ;SET READ FWD OP CODE
1880 011570 004737 012310 JSR PC,EXEC ;GO EXECUTE
1881 011574 032777 000040 166720 BIT #40,BDS ;BRANCH ID PES BIT GOT SET
1882 011602 001005 BNE FT25X
1883 011604 012737 020152 000650 MOV #MSG64,ERADD
1884 011612 004737 004054 JSR PC,FT3ER ;GO PROCESS ERROR
1885 011616 004737 013370 FT25X: JSR PC,ITER ;ITERATION LOOP
1886 011622 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULER
1887

```

```

1890                                     ;**B SEQUENTIAL TAPE MARK TEST
1891
1892 011626 000240                       FT26:  NOP
1893 011630 012737 021420 000610        MOV    #MSFT26,EMADDR ;SET TEST ERROR MSG HEADER
1894 011636 012737 001700 000716        MOV    #1700,UDES    ;SET NRZ
1895 011644 004737 012442                1$:   JSR    PC,RWND      ;REWIND SLAVE
1896 011650 012737 000027 000710        MOV    #27,FUN      ;SET WRITE TAPE MARK FUNCTION CODE
1897 011656 004737 012310                JSR    PC,EXEC      ;GO DO TAPE MARK
1898 011662 005037 000660                CLR    STMSK        ;CLEAR EXPECTED ERROR MASK
1899 011666 012737 016315 000626        MOV    #MSG15,ERRP  ;SET ERROR MESSAGE
1900 011674 004737 012510                JSR    PC,ERCHK     ;GO CHECK FOR ERRORS
1901 011700 004737 013164                JSR    PC,TMCHK     ;GO CHECK FOR TAPE MARK
1902 011704 005737 000712                TST    SERFL        ;EXIT TEST IF ERROR DETECTED
1903 011710 001061                       BNE    FT26X
1904 011712 004737 012310                JSR    PC,EXEC      ;WRITE SECOND TAPE MARK
1905 011716 012737 016336 000626        MOV    #MSG15A,ERRP ;SET ERROR MESSAGE
1906 011724 004737 012510                JSR    PC,ERCHK     ;GO CHECK ERROR
1907 011730 004737 013164                JSR    PC,TMCHK
1908 011734 005737 000712                TST    SERFL        ;EXIT TEST IF ERROR DETECTED
1909 011740 001045                       BNE    FT26X
1910 011742 004737 012442                JSR    PC,RWND      ;REWIND
1911 011746 012737 000031 000710        MOV    #31,FUN      ;SET SPACE FORWARD OP CODE
1912 011754 012737 177777 000642        MOV    #-1,SCNT     ;SET # OF RECORDS TO SPACE
1913 011762 004737 012310                JSR    PC,EXEC      ;GO SPACE FORWARD
1914 011766 012737 017374 000626        MOV    #MSG47,ERRP  ;SET SPACE FORWARD ERROR
1915 011774 004737 012510                JSR    PC,ERCHK     ;GO CHECK ERROR BITS
1916 012000 004737 013164                JSR    PC,TMCHK     ;GO CHECK IF TAPE MARK DETECTED
1917 012004 005737 000712                TST    SERFL        ;EXIT TEST IF ERROR DETECTED
1918 012010 001021                       BNE    FT26X
1919 012012 004737 012310                JSR    PC,EXEC      ;SPACE TO SECOND TAPE MARK
1920 012016 004737 012510                JSR    PC,ERCHK     ;GO CHECK ERROR BITS
1921 012022 004737 013164                JSR    PC,TMCHK     ;CHECK IF TAPE MARK DETECTED
1922 012026 005737 000712                TST    SERFL        ;EXIT TEST IF ERROR DETECTED
1923 012032 001010                       BNE    FT26X
1924 012034 032737 002000 000716        BIT    #2000,UDES   ;EXIT TEST IF PE COMPLETED
1925 012042 001004                       BNE    FT26X
1926 012044 012737 002300 000716        MOV    #2300,UDES   ;SET PE MODE
1927 012052 000674                       BR     1$
1928 012054 004737 013370                FT26X: JSR    PC,ITER
1929 012060 000137 003074                JMP    TSCD2
  
```

```

1931                                     ;REWIND: OFF LINE TEST*****
1932
1933 012064 032777 010000 166456 FT27: BIT      #10000,@SWR      ;SEE IF IN CONTINUOUS MODE
1934 012072 001104                                     BNE      FT27XX      ;IF SO: BR
1935 012074 005737 001662                                     TST      CHNFLG      ;BRANCH IF CHAIN MODE
1936 012100 001101                                     BNE      FT27XX
1937 012102 012737 021461 000610 MOV      #MSFT27,EMADDR ;SET TEST HEADER
1938 012110 004737 012442                                     JSR      PC,RWIND    ;REWIND & SELECT SLAVE
1939 012114 012737 000001 000720 MOV      #1,PATRN    ;SELECT PATTERN (ALL 1'S)
1940 012122 004737 013222                                     JSR      PC,DSUP     ;FILL WRITE BUFFER
1941 012126 012737 021544 000616 MOV      #WDATA,BADDR ;SET WRITE BUFFER BUS ADDRESS
1942 012134 012737 177400 000620 MOV      #-400,FCNT  ;SET FRAME COUNT
1943 012142 012737 177600 000622 MOV      #-200,WCNT  ;SET WORD COUNT
1944 012150 012737 001700 000716 MOV      #1700,UDES  ;SET UNIT DESCRIPTION = NRZ
1945 012156 012737 000061 000710 MOV      #61,FUN     ;SET WRITE COMMAND
1946 012164 004737 012310                                     JSR      PC,EXEC     ;GO WRITE A RECORD
1947 012170 004737 013540                                     JSR      PC,INIT1    ;**B CLEAR ANY ERROR BITS
1948 012174 012777 000003 166306 MOV      #3,@C1      ;ISSUE REWIND: OFF LINE COMMAND
1949 012202 005037 000674                                     CLR      SCOLP      ;CLEAR SCOPE LOOP
1950 012206 012700 000042                                     MOV      #42,R0
1951 012212 005001                                     1$: CLR      R1      ;CLEAR TIMER
1952 012214 005301                                     2$: DEC      R1
1953 012216 001376                                     BNE      2$         ;IF NOT TIMED OUT: BR
1954 012220 005300                                     DEC      R0
1955 012222 001373                                     BNE      1$         ;IF NOT ALL TIMED OUT: BR
1956 012224 032777 010000 166270 BIT      #10000,@DS  ;SEE IF MOL IS RESET
1957 012232 001406                                     BEQ      3$         ;IF SO: BR
1958 012234 012737 017605 000650 MOV      #MSG53,ERADD ;SET ERROR CODE
1959 012242 004737 004054                                     JSR      PC,FT3ER    ;GO DO ERROR
1960 012246 000412                                     BR      FT27X
1961 012250 013700 000524                                     3$: MOV      ER,R0   ;GET ADDRESS OF ERROR REG
1962 012254 005001                                     CLR      R1          ;RESULT SHOULD BE 0
1963 012256 020110                                     CMP      R1,(R0)     ;BRANCH IF ERROR REG = 0
1964 012260 001405                                     BEQ      FT27X
1965 012262 012737 020206 000650 MOV      #MSG67,ERADD ;SET ERROR MSG HEADER
1966 012270 004737 003564                                     JSR      PC,FT2ER    ;GO TYPE ERROR
1967 012274 012704 017632 FT27X: MOV      #MSG54,R4
1968 012300 004737 014414 FT27X: JSR      PC,TTOUT ;PRINT ON LINE REQUEST
1969 012304 000137 003074 FT27XX: JMP      TSCD2 ;RETURN TO SCHEDULER

```

```
1971
1972
1973
1974 012310 000240
1975 012312 053777 000716 166222
1976 012320 013777 000622 166164
1977 012326 013777 000620 166162
1978 012334 013777 000616 166152
1979 012342 022737 000031 000710
1980 012350 001404
1981 012352 022737 000033 000710
1982 012360 001003
1983 012362 013777 000642 166126
1984 012370 000240
1985 012372 013777 000710 166110
1986 012400 000240
1987 012402 013703 000636
1988 012406 005004
1989 012410 032777 000200 166104
1990 012416 001004
1991 012420 005304
1992 012422 001372
1993 012424 005303
1994 012426 001370
1995 012430 013703 000640
1996 012434 005303
1997 012436 001376
1998 012440 000207
1999

;COMMAND EXECUTE SUBROUTINE*****
EXEC: NOP
      BIS      UDES,@TC      ;LOAD TAPE CONT
      MOV      WCNT,@WC      ;LOAD WC
      MOV      FCNT,@FC      ;LOAD FC
      MOV      BADDR,@BA     ;LOAD BA
      CMP      #31,FUN       ;SEE IF SPACE FORWARD
      BEQ      EXECA         ;IF SO: BR
      CMP      #33,FUN       ;SEE IF SPACE REVERSE
      BNE      EXECB         ;IF NOT: BR
EXECA: MOV      SCNT,@FC      ;SET SPACE COUNT
EXECB: NOP
      MOV      FUN,@C1       ;LOAD OP-CODE + GO
      NOP
      MOV      RDYDX,R3      ;SET DELAY
      CLR      R4
EXECX: BIT      #200,@DS     ;SEE IF DRY
      BNE      EXECX         ;IF SO: BR
      DEC      R4
      BNE      EXECX
      DEC      R3            ;DELAY FOR DRY
      BNE      EXECX
EXECXA: MOV      OPDYX,R3
EXECXX: DEC      R3          ;DELAY
      BNE      EXECXA
      RTS      PC           ;RETURN TO CALLER
```

```

2001
2002
2003 012442 004737 013540          RWND: JSR    PC,INIT1      ;INIT SLAVE
2004 012446 012777 000007 166034  MOV    #7,#C1      ;START REWIND
2005 012454 032777 000002 166040  1$:   BIT    #2,#DS      ;WAIT FOR BOT TO SET
2006 012462 001774                      BEQ    1$
2007 012464 032777 020000 166030  2$:   BIT    #20000,#DS   ;WAIT FOR PIP TO CLEAR
2008 012472 001374                      BNE    2$
2009 012474 105777 166022          3$:   TSTB   #DS          ;...B WAIT FOR DRY
2010 012500 100375                      BPL    3$
2011 012502 004737 013540          JSR    PC,INIT1      ;INIT
2012 012506 000207                      RTS    PC             ;RETURN TO CALLER
2013

```

```

2015                                     ;ERROR CHECK SUBROUTINE*****
2016
2017 012510 005037 000712          ERCHK: CLR      SERFL      ;CLEAR FLAG
2018 012514 017737 166002 000664  MOV      @DS,DSAV   ;SAVE DRIVE STATUS REGISTER
2019 012522 032777 040000 165772  BIT      @40000,@DS ;SEE IF ERROR
2020 012530 001021                    BNE      ERPT      ;IF SO: BR
2021 012532 022777 006576 166146  CMP      @FT15,@LTADD ;CHECK FOR TEST 15
2022 012540 001404                    BEQ      1$        ;BRANCH IF TEST 15
2023 012542 022777 007210 166136  CMP      @FT16,@LTADD ;CHECK FOR TEST 16
2024 012550 001010                    BNE      2$        ;BRANCH IF NOT TEST 16
2025 012552 032777 100000 165742 1$: BIT      @100000,@DS ;TEST FOR ATA
2026 012560 001004                    BNE      2$        ;EXIT IF ATA IS SET
2027 012562 012737 020310 000626  MOV      @MSG71,ERRP ;SET UP ERROR CODE
2028 012570 000434                    BR       ERPTG     ;GO REPORT ERROR
2029 012572 000207                    2$: RTS      PC      ;RETURN
2030 012574 017704 165724          ERPT:  MOV      @ER,R4  ;GET ERROR REGISTER
2031 012600 032737 002000 000716  BIT      @2000,UDES ;SEE IF PE
2032 012606 001403                    BEQ      1$        ;IF SO: BR
2033 012610 042737 000200 000660  BIC      @200,STMSK ;RESET PEF MASK
2034 012616 022737 000003 000742 1$: CMP      @3,JUMPER ;TEST FOR NON-STANDARD JUMPER
2035 012624 001413                    BEQ      ERPTA1    ;BRANCH IF STANDARD
2036 012626 022777 011242 166052  CMP      @FT24,@LTADD ;CHECK FOR TEST 24
2037 012634 001404                    BEQ      2$        ;BRANCH IF TST24
2038 012636 022777 011434 166042  CMP      @FT25,@LTADD ;CHECK FOR TEST 25
2039 012644 001003                    BNE      ERPTA1
2040 012646 052737 020000 000660 2$: BIS      @20000,STMSK ;SET OPI BIT IN ERROR MASK
2041 012654 043704 000660          ERPTA1: BIC      STMSK,R4 ;MASK DONT CARE BITS
2042 012660 001536                    BEQ      ERPTX     ;IF NO UNEXPECTED ERRORS: BR
2043 012662 012737 000001 000712  ERPTG: MOV      @1,SERFL ;SET FLAG
2044 012670 032777 020000 165652  BIT      @20000,@SWR ;SEE IF SHOULD PRINT ERRORS
2045 012676 001123                    BNE      ERPTD     ;IF NOT: BR
2046 012700 005737 000606          TST      HDRFL    ;SEE IF DONE HEADER
2047 012704 001006                    BNE      ERPTA     ;IF SO: BR
2048 012706 005237 000606          INC      HDRFL    ;SET HEADER FLAG
2049 012712 013704 000610          MOV      @MADDR,R4
2050 012716 004737 014414          JSR      PC,TTOUT  ;PRINT HEADER
2051 012722 013704 000626          ERPTA: MOV      @ERRP,R4 ;GET ERROR CODE
2052 012726 001414                    BEQ      ERPTB     ;IF NONE: BR
2053 012730 004737 014414          JSR      PC,TTOUT  ;PRINT ERROR CODE
2054 012734 012704 016443          MOV      @MSG20,R4 ;SET NRZ TAG
2055 012740 032777 002000 165574  BIT      @2000,@TC ;SEE IF PE
2056 012746 001402                    BEQ      ERPT1A    ;IF NOT: BR
2057 012750 012704 016451          MOV      @MSG21,R4 ;ELSE SET PE TAG
2058 012754 004737 014414          ERPT1A: JSR      PC,TTOUT ;PRINT TAG
2059 012760 013704 000630          ERPTB: MOV      @ERRP1,R4 ;SEE IF CODE 2
2060 012764 001402                    BEQ      ERPTB1    ;IF NOT: BR
2061 012766 004737 014414          JSR      PC,TTOUT  ;PRINT CODE 2
2062 012772 032777 004000 165550  ERPTB1: BIT      @4000,@SWR ;SEE IF ITERATION
2063 013000 001010                    BNE      ERPTC     ;IF NOT: BR
2064 013002 012704 017776          MOV      @MSG56,R4
2065 013006 004737 014414          JSR      PC,TTOUT  ;PRINT ITER TAG
2066 013012 013703 000662          MOV      @ITCNT,R3
2067 013016 004737 014630          JSR      PC,OCPT   ;PRINT ITERATION
2068 013022 012704 015422          ERPTC: MOV      @MSG1,R4
2069 013026 004737 014414          JSR      PC,TTOUT  ;PRINT REGISTER TAG
2070 013032 017703 165452          MOV      @C1,R3
  
```

2071	013036	004737	014616		JSR	PC, OCTPE	;PRINT CS1
2072	013042	017703	165444		MOV	@WC, R3	
2073	013046	004737	014616		JSR	PC, OCTPE	;PRINT WC
2074	013052	017703	165436		MOV	@BA, R3	
2075	013056	004737	014616		JSR	PC, OCTPE	;PRINT BA
2076	013062	017703	165430		MOV	@FC, R3	
2077	013066	004737	014616		JSR	PC, OCTPE	;PRINT FC
2078	013072	017703	165422		MOV	@CS, R3	
2079	013076	004737	014616		JSR	PC, OCTPE	;PRINT CS2
2080	013102	005737	007206		TST	TAG	;...C CHECK FOR SPECIAL DS
2081	013106	001403			BEQ	1\$	
2082	013110	013703	000652		MOV	TEMP1, R3	;...C PRINT DS READ INTO TEMP1 AT CRITICAL TIME
2083	013114	000402			BR	2\$	
2084	013116	017703	165400	1\$:	MOV	@DS, R3	
2085	013122	004737	014616	2\$:	JSR	PC, OCTPE	;PRINT DS
2086	013126	017703	165372		MOV	@ER, R3	
2087	013132	004737	014616		JSR	PC, OCTPE	;PRINT ER
2088	013136	017703	165400		MOV	@TC, R3	
2089	013142	004737	014616		JSR	PC, OCTPE	;PRINT TC
2090	013146	005777	165376	ERPTD:	TST	@SWR	;SEE IF HALT ON ERROR
2091	013152	100001			BPL	ERPTX	;IF NOT: BR
2092	013154	000000			HALT		
2093	013156	004737	013540	ERPTX:	JSR	PC, INIT1	;INIT
2094	013162	000207		ERPTXX:	RTS	PC	;RETURN
2095							
2096							

```

2098                                     ;TAPE MARK STATUS CHECK*****
2099
2100 013164 032737 000004 000664 TMCHK: BIT      #4, DSAV      ;BRANCH IF TM SET
2101 013172 001012                                BNE      1$
2102 013174 005737 000712                                TST      SERFL      ;SEE IF HAD ERROR
2103 013200 001007                                BNE      1$          ;IF SO: BR
2104 013202 012737 020006 000630 MOV      #MSG57,ERRP1 ;SET ERROR CODE 2
2105 013210 004737 012662                                JSR      PC,ERPTG   ;GO PRINT TM ERROR
2106 013214 005037 000630                                CLR      ERRP1      ;CLEAR CODE 2 FLAG
2107 013220 000207                                1$:      RTS      PC ;RETURN
2108
2109                                     ;DATA SETUP ROUTINE*****
2110
2111 013222 000240                                DSUP:    NOP
2112 013224 012703 021544                                DSO:    MOV      #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
2113 013230 013701 000720                                MOV      PATRN,R1    ;R1 = PATTERN SELECTOR
2114 013234 006301                                ASL      R1          ;MAKE PATTERN SELECTOR EVEN
2115 013236 004771 000744                                JSR      PC,@DATBL(R1) ;GO GENERATE PATTERN
2116 013242 012702 000640                                MOV      #640,R2    ;R2=BUFFER SIZE *2
2117 013246 012701 023256                                MOV      @RDATA,R1  ;R1=READ DATA START
2118 013252 005021                                1$:    CLR      (R1)+ ;CLEAR BUFFER
2119 013254 005302                                DEC      R2          ;SEE IF DONE ALL
2120 013256 001375                                BNE      1$         ;IF NOT: BR
2121 013260 000207                                RTS      PC         ;EXIT
2122
2123                                     ;ALL ONES*****
2124
2125 013262 012701 177777                                DAT1:    MOV      #-1,R1 ;R1=DATA
2126 013266 012702 000640                                DAT1A:  MOV      #640,R2  ;R2=WORD COUNT *2
2127 013272 010123                                1$:    MOV      R1,(R3)+ ;LOAD BUFFER
2128 013274 005302                                DEC      R2          ;SEE IF DONE
2129 013276 001375                                BNE      1$         ;IF NOT: BR
2130 013300 000207                                RTS      PC
2131
2132                                     ;ALL ZEROS*****
2133
2134 013302 005001                                DAT2:    CLR      R1 ;R1=DATA
2135 013304 000770                                BR      DAT1A      ;LOAD BUFFER
2136
2137                                     ;ONE/ZERO IN ALTERNATING CHARACTERS*****
2138
2139 013306 012701 125125                                DAT3:    MOV      #125125,R1 ;R1=DATA
2140 013312 000765                                BR      DAT1A      ;LOAD BUFFER
2141
2142                                     ;ALL BITS 0-377*****
2143
2144 013314 005001                                DAT4:    CLR      R1 ;R1=STARTING DATA
2145 013316 012702 001500                                MOV      #1500,R2   ;R2=CHARACTER COUNT
2146 013322 110123                                1$:    MOV      R1,(R3)+ ;LOAD BUFFER
2147 013324 105201                                INCB    R1          ;BUMP DATA
2148 013326 005302                                DEC      R2          ;SEE IF DONE
2149 013330 001374                                BNE      1$         ;IF NOT: BR
2150 013332 000207                                RTS      PC
2151

```

```
2153
2154
2155
2156 ;SCOPE LOOP ON ERROR SUBROUTINE*****
2157
2158 013334 000240 SCOPE: NOP
2159 013336 032777 040000 165204 BIT @40000,@SWR ;SEE IF LOOP ON ERROR
2160 013344 001001 BNE 1$ ;IF SO: BR
2161 013346 000207 RTS PC ;ELSE EXIT
2162 013350 000240 1$: NOP
2163 013352 005737 000674 TST SCOLP ;SEE IF SCOPE ADDRESS
2164 013356 001001 BNE 2$ ;IF NOT: BR
2165 013360 000207 RTS PC ;ELSE EXIT
2166 013362 022626 2$: CMP (SP)+,(SP)+ ;RESET STACK
2167 013364 000177 165304 JMP @SCOLP ;LOOP ON ERROR
2168
2169 ;TEST ITERATION SUBROUTINE*****
2170
2171 013370 000240 ITER: NOP
2172 013372 032777 004000 165150 BIT @4000,@SWR ;SEE IF ITERATIONS
2173 013400 001403 BEQ 2$ ;IF SO: BR
2174 013402 005037 000662 1$: CLR ITCNT ;CLEAR ITERATION COUNTER
2175 013406 000207 RTS PC ;ELSE EXIT
2176 013410 005737 000730 2$: TST PCNTR ;DO SINGLE SUBTEST ITERATION
2177 013414 001772 BEQ 1$ ;ON FIRST PASS
2178 013416 005237 000662 INC ITCNT ;BUMP COUNTER
2179 013422 023737 000662 000566 CMP ITCNT,ITAMT ;SEE IF DONE ALL
2180 013430 001764 BEQ 1$ ;IF SO: BR
2181 013432 005726 TST (SP)+ ;RESET STACK
2182 013434 017700 165236 MOV @ITRLP,RO ;SET ITERATION POINTER
2183 013440 000110 JMP (RO) ;GO ITERATE
2184
2185
2186 ;NON-STANDARD JUMPER HANDLER SUBROUTINE*****
2187
2188
2189 013442 010046 NOST: MOV RO,-(SP) ;SAVE RO
2190 013444 012700 000120 MOV @120,RO ;SET UP INDEX
2191 013450 012760 011626 000756 MOV @FT26,TSTTBL(RO);ADJUST SCHEDULAR TEST TABLE
2192 013456 005720 TST (RO)+
2193 013460 012760 011626 000756 MOV @FT26,TSTTBL(RO) ;OVERLAY TEST LIST
2194
2195 013466 005720 TST (RO)+
2196 013470 012760 012064 000756 MOV @FT27,TSTTBL(RO)
2197 013476 005720 TST (RO)+
2198 013500 012760 012064 000756 MOV @FT27,TSTTBL(RO)
2199 013506 005720 TST (RO)+
2200 013510 012760 003162 000756 MOV @TEND,TSTTBL(RO)
2201 013516 005720 TST (RO)+
2202 013520 012760 000027 000756 MOV @27,TSTTBL(RO)
2203 013526 012737 000027 001120 MOV @27,TLAST
2204 013534 012600 MOV (SP)+,RO ;RESTO RO
2205 013536 000207 RTS PC
2206
2207 ;INITIALIZE SUBROUTINE*****
2208
```

```
2209 013540 000240          INIT1:  NOP
2210 013542 012777 000040 164750      MOV     #40,@CS      ;INIT
2211 013550 013777 000612 164742      MOV     DRVN,@CS    ;SELECT DRIVE
2212 013556 013777 000614 164756      MOV     SLVN,@TC    ;SELECT SLAVE
2213 013564 000207          RTS     PC           ;RETURN
2214
2215                          ;MAG TAPE INTERRUPT HANDLER*****
2216
2217 013566 000240          MTINT:  NOP
2218 013570 013716 000646      MOV     RTRN,(SP)   ;RETURN TO (RTRN)
2219 013574 000002          RTI
2220
```

2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244

013576 017746 164752
013602 042716 000200
013606 122716 000003
013612 001010
013614 005737 001662
013620 001005
013622 005077 164720
013626 000005
013630 000137 000200
013634 122716 000001
013640 001017
013642 022737 000176 000550
013650 001016
013652 012737 177570 000550
013660 004737 015356
013664 012704 020262
013670 004737 014414
013674 004737 015400
013700 122716 000007
013704 001006

```
      ;TTY INTERRUPT HANDLER*****  
TTINT: MOV      @TKB,-(SP)      ;GET CHARACTER  
      BIC      @200,(SP)      ;CLEAR PARITY BIT  
      CMPB    @3,(SP)        ;BRANCH IF NOT CONTROL C  
      BNE     1$  
      TST     CHNFLG          ;INHIBIT 'C IF CHAIN MODE  
      BNE     1$  
      CLR     @PSW  
      RESET  
      JMP     @@200           ;RESTART PROGRAM  
1$:    CMPB    @1,(SP)        ;BRANCH IF NOT 'A  
      BNE     2$  
      CMP     @SWREG,SWR      ;BRANCH IF HARDWARE SWR IS INVOKED  
      BNE     3$  
      MOV     @177570,SWR     ;INVOKE HARDWARE SWR  
      JSR    PC,SAVE         ;SAVE REGISTERS ON THE STACK  
      MOV     @MSG70,R4      ;TYPE 'HARDWARE SWR IN USE'  
      JSR    PC,TTOUT  
      JSR    PC,RESTORE  
2$:    CMPB    @7,(SP)        ;BRANCH IF NOT 'G  
      BNE     4$
```

```
2246  
2247 013706 012737 000176 000550 3#: MOV #SWREG,SWR ;INVOKE SOFTWARE SWR  
2248 013714 004737 015260 JSR PC,GTSWR ;GET SWITCHES  
2249 013720 000414 BR 6#  
2250 013722 122716 000023 4#: CMPB #23,(SP) ;SEE IF 'S  
2251 013726 001004 BNE 5# ;BRANCH IF NOT  
2252 013730 112737 000377 001122 MOVB #377,#CNTRLS ;SET XOFF FLAG  
2253 013736 000405 BR 6#  
2254 013740 122716 000021 5#: CMPB #21,(SP) ;SEE IF 'Q  
2255 013744 001002 BNE 6# ;BRANCH IF NOT  
2256 013746 105037 001122 CLRB #CNTRLS  
2257 013752 005726 6#: TST (SP)+ ;POP CHARACTER OFF STACK  
2258 013754 000002 RTI ;RETURN
```

```

2260                                     ;BUS ADDRESS TRAP HANDLER*****
2261
2262 013756 000240                       TRAP:  NOP
2263 013760 032777 020000 164562        BIT    #20000,@SWR      ;SEE IF SHOULD PRINT ERRORS
2264 013766 001020                       BNE    TRAP2          ;IF NOT; BR
2265 013770 005737 000606                TST    HDRFL          ;SEE IF DONE HEADER
2266 013774 001006                       BNE    TRAP1          ;IF SO; BR
2267 013776 005237 000606                INC    HDRFL          ;ELSE SET HEADER FLAG
2268 014002 013704 000610                MOV    EMADDR,R4
2269 014006 004737 014414                JSR    PC,TTOUT       ;PRINT HEADER
2270 014012 012704 016476                TRAP1: MOV    #MSG24,R4
2271 014016 004737 014414                JSR    PC,TTOUT       ;PRINT ERROR
2272 014022 010103                       MOV    R1,R3         ;GET ADDRESS THAT CAUSED THE TRAP
2273 014024 004737 014630                JSR    PC,OCTP        ;PRINT ADDRESS OF TRAP
2274 014030 005777 164514                TRAP2: TST    @SWR    ;SEE IF HALT ON ERROR
2275 014034 100001                       BPL    TRAPX          ;IF NOT; BR
2276 014036 000000                       HALT
2277 014040 022626                       TRAPX: CMP    (SP), (SP) ;RESET STACK
2278 014042 012737 003322 000674        MOV    #FT1A,SCOLP   ;SET SCOPE ADDRESS
2279 014050 004737 013334                JSR    PC,SCOPE      ;GO SEE IF SCOPE LOOP
2280 014054 005737 000722                TST    RHTF          ;SEE IF INITIAL ADDRESS TEST
2281 014060 001402                       BEQ    TRAPXX        ;IF NOT; BR
2282 014062 000137 001764                JMP    STOB          ;ELSE REDO ADDRESS REQUEST
2283 014066 000137 003326                TRAPXX: JMP   FT1B   ;RETURN TO TEST 1
2284

```

```

2286 ;*****
2287 ;TTY ENTRY SUBROUTINE:
2288 ;
2289 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
2290 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
2291 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
2292 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
2293 ; (0-7) AND MUST FALL WITHIN THE LIMITS SET BY
2294 ;THE CALLING ROUTINE.
2295 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
2296 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
2297 ;MAY BE REENTERED.
2298 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
2299 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
2300 ;CARRIAGE RETURN
2301 ;*****
2302
2303 014072 010146 TTR: MOV R1, -(SP) ;SAVE CHAR COUNT ON STACK
2304 014074 011601 10$: MOV (SP), R1 ;RESTORE CHAR COUNT (FOR ↑U)
2305 014076 005037 000652 CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
2306 014102 005000 CLR R0
2307 014104 004737 014352 1$: JSR PC, TTIN ;GO READ CHARACTER
2308 014110 122737 000003 000602 CMPB #3, TIB ;BRANCH IF NOT ↑C
2309 014116 001003 BNE 11$
2310 014120 000005 RESET ;RESET
2311 014122 000137 000200 JMP #200 ;RESTART
2312 014126 122737 000015 000602 11$: CMPB #15, TIB ;SEE IF CR
2313 014134 001004 BNE 2$ ;IF NOT: BR
2314 014136 005737 000652 TST TEMP1 ;SEE IF FIRST CHARACTER
2315 014142 001471 BEQ 9$ ;IF SO: BR
2316 014144 000457 BR 6$ ;ELSE GO LOAD VALUE
2317 014146 122737 000025 000602 2$: CMPB #25, TIB ;BRANCH IF NOT CONTROL U
2318 014154 001005 BNE 21$
2319 014156 012704 020202 MOV #MSG65, R4 ;TYPE <CR><LF>
2320 014162 004737 014414 JSR PC, TTOUT
2321 014166 000742 BR 10$ ;RESTART
2322 014170 122737 000177 000602 21$: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'
2323 014176 001012 BNE 3$
2324 014200 000241 CLC ;REMOVE LAST CHARACTER
2325 014202 006000 ROR R0
2326 014204 006200 ASR R0
2327 014206 006200 ASR R0
2328 014210 012704 020204 MOV #MSG66, R4 ;TYPE '\ '
2329 014214 004737 014414 JSR PC, TTOUT
2330 014220 005201 INC R1 ;DECREMENT CHAR RECEIVED COUNT
2331 014222 000730 BR 1$ ;GET NEXT CHARACTER
2332 014224 122737 000060 000602 3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
2333 014232 101402 RLOS 4$ ;IF NOT: BR
2334 014234 000137 014332 JMP TNER ;ELSE GO TO ERROR
2335 014240 122737 000070 000602 4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
2336 014246 101002 BHI 5$ ;IF NOT: BR
2337 014250 000137 014332 JMP TNER ;ELSE GO TO ERROR
2338 014254 005237 000652 5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
2339 014260 006300 ASL R0
2340 014262 006300 ASL R0 ;SHIFT 3 LEFT
2341 014264 006300 ASL R0
  
```

```

2342 014266 042737 177770 000602      BIC      #177770,TIB      ;STRIP ASCII
2343 014274 053700 000602      BIS      TIB,R0          ;LOAD CHARACTER
2344 014300 005301                DEC      R1              ;SEE IF DONE
2345 014302 001300                SNE      1#              ;IF NOT: BR
2346 014304 020002                6#:     CMP      RO,R2    ;SEE IF EXCEEDED MAXIMUM LIMIT
2347 014306 101402                BLOS     7#              ;IF NOT: BR
2348 014310 000137 014332      JMP      TINNER          ;ELSE GO TO ERROR
2349 014314 020300                7#:     CMP      R3,RO    ;SEE IF BELOW MINIMUM LIMIT
2350 014316 101402                BLOS     8#              ;IF NOT: BR
2351 014320 000137 014332      JMP      TINNER          ;ELSE GO TO ERROR
2352 014324 010015                8#:     MOV      RO,(R5)  ;LOAD VALUE
2353 014326 005726                9#:     TST      (SP)+    ;POP CHAR COUNT OFF STACK
2354 014330 000207                RTS      PC              ;EXIT
2355
2356                                ;TTY ENTRY ERROR SUBROUTINE*****
2357
2358 014332 012704 015777      TINNER:  MOV      #MSG7,R4
2359 014336 004737 014414      JSR      PC,TTOUT        ;PRINT?
2360 014342 005726                TST      (SP)+          ;POP CHAR COUNT OFF STACK
2361 014344 162716 000020      SUB      #20,(SP)       ;RESET SP TO START OF VALUE ROUTINE
2362 014350 000207                RTS      PC              ;REDO VALUE ENTRY
2363
2364                                ;TTY READ SUBROUTINE*****
2365
2366 014352 005277 164174      TTIN:   INC      @TKS
2367 014356 105777 164170      1#:    TSTB     @TKS
2368 014362 100375                BPL      1#
2369 014364 117737 164164 000602  MOVB     @TKB,TIB
2370 014372 042737 000200 000602  BIC      #200,TIB      ;STRIP PARITY BIT
2371 014400 013737 000602 000600  MOV      TIB,TOB       ;MOVE CHAR TO OUTPUT BFR
2372 014406 004737 014514      JSR      PC,TOG         ;AND TYPE IT
2373 014412 000207                RTS      PC
2374
2375                                ;TTY OUTPUT SUBROUTINE*****
2376
2377 014414 112437 000600      TTOUT:  MOVB     (R4)+,TOB
2378 014420 122737 000043 000600  CMPB     #43,TOB
2379 014426 001472                BEQ      TEX
2380 014430 122737 000045 000600  CMPB     #45,TOB
2381 014436 001403                BEQ      1#
2382 014440 004737 014514      JSR      PC,TOG
2383 014444 000763                BR       TTOUT
2384 014446 112737 000015 000600  1#:    MOVB     #15,TOB
2385 014454 004737 014514      JSR      PC,TOG
2386 014460 012703 000004                MOV      #4,R3
2387 014464 005037 000600      2#:    CLR      TOB
2388 014470 004737 014514      JSR      PC,TOG
2389 014474 005303                DEC      R3
2390 014476 001372                BNE     2#              ;DO FILLERS
2391 014500 112737 000012 000600  MOVB     #12,TOB
2392 014506 004737 014514      JSR      PC,TOG
2393 014512 000740                BR       TTOUT

```

2395	014514	105777	164032		TOG:	TSTB	@TKS		;SEE IF INPUT AT KEYBOARD
2396	014520	100024				BPL	3:		;IF SO, THEN
2397	014522	117737	164026	001123		MOVB	@TKB,%CNTRLS+1		;MOVE CHARACTER AND
2398	014530	142737	000200	001123		BICB	@200,%CNTRLS+1		;MASK OFF PARITY BIT.
2399	014536	122737	000023	001123		CMPB	@23,%CNTRLS+1		;SEE IF CHARACTER IS XOFF
2400	014544	001004				BNE	2:		;IF XOFF, THEN
2401	014546	112737	000377	001122		MOVB	@377,%CNTRLS		;SET XOFF FLAG
2402	014554	000757				BR	TOG		
2403	014556	122737	000021	001123	2:	CMPB	@21,%CNTRLS+1		;SEE IF CHARACTER IS XON
2404	014564	001002				BNE	3:		;IF SO THEN
2405	014566	105037	001122			CLRB	%CNTRLS		;CLEAR XOFF FLAG
2406	014572	105737	001122		3:	TSTB	%CNTRLS		;SEE IF IN XOFF MODE
2407	014576	100746				BMI	TOG		;IF NOT THEN
2408	014600	105777	163752			TSTB	@TPS		;CHECK IF PRINTER READY
2409	014604	100343				BPL	TOG		;
2410	014606	113777	000600	163744		MOVB	TOB,@TPB		
2411	014614	000207			TEX:	RTS	PC		;RETURN

```

2413                                ;OCTAL OUTPUT SUBROUTINE*****
2414
2415 014616 012737 000001 015046 OCTPE: MOV    #1,OFL
2416 014624 010304          MOV    R3,R4
2417 014626 000410          BR     OCTP0
2418 014630 005037 015046          OCTP:  CLR    OFL                ;CLEAR FLAG FOR LEADING ZERO
2419 014634 010304          OCTPE1: MOV   R3,R4                ;SEE IF NUMBER IS ZERO
2420 014636 001004          BNE    OCTP0                ;IF NOT ZERO: BR
2421 014640 004737 015026          JSR    PC,OCTPG1            ;ELSE PRINT ZERO
2422 014644 000137 014770          JMP    OCTP3                ;SPACE AND EXIT
2423 014650 032704 100000          OCTP0: BIT   #100000,R4       ;SEE IF MSD = 1
2424 014654 001406          BEQ    OCTP1                ;IF NOT: BR
2425 014656 012704 000001          MOV    #1,R4
2426 014662 004737 015004          JSR    PC,OCTPG            ;PRINT 1
2427 014666 000137 014700          JMP    OCTP2
2428 014672 005004          OCTP1: CLR    R4
2429 014674 004737 015004          JSR    PC,OCTPG            ;PRINT 0
2430 014700 010304          OCTP2: MOV   R3,R4
2431 014702 006004          ROR    R4
2432 014704 006004          ROR    R4
2433 014706 006004          ROR    R4                ;POSITION DIGIT
2434 014710 006004          ROR    R4
2435 014712 000304          SWAB   R4
2436 014714 004737 015004          JSR    PC,OCTPG            ;PRINT DIGIT 2
2437 014720 010304          MOV    R3,R4
2438 014722 006004          ROR    R4
2439 014724 000304          SWAB   R4
2440 014726 004737 015004          JSR    PC,OCTPG            ;PRINT DIGIT 3
2441 014732 010304          MOV    R3,R4
2442 014734 006104          ROL    R4
2443 014736 006104          ROL    R4
2444 014740 000304          SWAB   R4
2445 014742 004737 015004          JSR    PC,OCTPG            ;PRINT DIGIT 4
2446 014746 010304          MOV    R3,R4
2447 014750 006004          ROR    R4
2448 014752 006004          ROR    R4
2449 014754 006004          ROR    R4
2450 014756 004737 015004          JSR    PC,OCTPG
2451 014762 010304          MOV    R3,R4
2452 014764 004737 015004          JSR    PC,OCTPG            ;PRINT DIGIT 5
2453 014770 012737 000240 000600 OCTP3: MOV   #240,TOB
2454 014776 004737 014514          JSR    PC,TOG                ;PRINT SPACE
2455 015002 000207          RTS    PC                    ;EXIT
2456 015004 042704 177770          OCTPG: BIC   #177770,R4
2457 015010 001004          BNE    OCTPG0
2458 015012 005737 015046          TST   OFL
2459 015016 001001          BNE    OCTPG0
2460 015020 000207          RTS    PC
2461
2462 015022 005237 015046          OCTPG0: INC   OFL
2463 015026 052704 000260          OCTPG1: BIS   #260,R4
2464 015032 010437 000600          MOV    R4,TOB
2465 015036 004737 014514          JSR    PC,TOG
2466 015042 010304          MOV    R3,R4
2467 015044 000207          RTS    PC
2468 015046 000000          OFL:   0                    ;FIRST CHAR FLAG
  
```

```

2469
2470
2471
2472 015050 005037 000600          DOUT:  CLR      TOB
2473 015054 012704 000010          MOV      #10,R4          ;SET NUMBER TO PRINT
2474 015060 110337 000600          MOV      R3,TOB
2475 015064 105777 163466          1$:     TSTB     #TPS
2476 015070 100375                    BPL      1$
2477 015072 132737 000200 000600    BITB     #200,TOB
2478 015100 001404                    BEQ      2$
2479 015102 012777 000061 163450    MOV      #061,#TPB
2480 015110 000403                    BR       3$
2481 015112 012777 000060 163440    2$:     MOV      #060,#TPB
2482 015120 006137 000600          3$:     ROL      TOB
2483 015124 005304                    DEC      R4
2484 015126 001356                    BNE     1$
2485 015130 000207                    RTS      PC
2486
2487 015132 013703 000656          DOUTD:  MOV      TEMP3,R3
2488 015136 000303                    SWAB    R3
2489 015140 004737 015050          JSR     PC,DOUT
2490 015144 013703 000656          MOV      TEMP3,R3
2491 015150 004737 015050          JSR     PC,DOUT
2492 015154 000207                    RTS      PC
2493
2494
2495
2496 015156 010304          SNPT:  MOV      R3,R4
2497 015160 000304          SWAB    R4
2498 015162 006004          ROR     R4
2499 015164 006004          ROR     R4
2500 015166 006004          ROR     R4
2501 015170 006004          ROR     R4          ;GET FIRST DIGIT
2502 015172 004737 015234    JSR     PC,SNPG      ;GO PRINT
2503 015176 010304          MOV      R3,R4
2504 015200 000304          SWAB    R4          ;GET SECOND DIGIT
2505 015202 004737 015234    JSR     PC,SNPG      ;GO PRINT
2506 015206 010304          MOV      R3,R4
2507 015210 006004          ROR     R4
2508 015212 006004          ROR     R4
2509 015214 006004          ROR     R4
2510 015216 006004          ROR     R4          ;GET THIRD DIGIT
2511 015220 004737 015234    JSR     PC,SNPG      ;GO PRINT
2512 015224 010304          MOV      R3,R4          ;GET FOURTH DIGIT
2513 015226 004737 015234    JSR     PC,SNPG      ;GO PRINT
2514 015232 000207          RTS      PC          ;EXIT
2515 015234 012737 000260 000600  SNPG:  MOV      #260,TOB      ;SET BASE = 0
2516 015242 042704 177760          BIC     #177760,R4    ;MASK DIGIT
2517 015246 050437 000600          BIS     R4,TOB        ;SET ASCII
2518 015252 004737 014514          JSR     PC,TOG        ;TYPE DIGIT
2519 015256 000207          RTS      PC          ;RETURN
2520

```

```

2522
2523 ;ROUTINE TO LOAD NEW VALUE INTO SWITCHES
2524 015260 022737 000176 000550 GTSWR: CMP @SWREG,SWR ;BRANCH IF SOFTWARE SWR
2525 015266 001032 BNE 1$ ;NOT INVOKED
2526 015270 004737 015356 JSR PC,.SAVE ;SAVE REGISTERS ON THE STACK
2527 015274 012704 021521 MOV @MSWR,R4
2528 015300 004737 014414 JSR PC,TTOUT
2529 015304 017703 163240 MOV @SWR,R3
2530 015310 004737 014616 JSR PC,OCTPE
2531 015314 012704 021530 MOV @MNEW,R4
2532 015320 004737 014414 JSR PC,TTOUT
2533 015324 013705 000550 MOV SWR,R5 ;TTR ROUTINE RETURNS NEW VALUE TO (R5)
2534 015330 012701 000007 MOV #7,R1 ;LIMIT RESPONSE TO 7 CHARS
2535 015334 012702 177777 MOV #177777,R2 ;BETWEEN 0 AND 177777
2536 015340 012703 000000 MOV #0,R3
2537 015344 004737 014072 JSR PC,TTR
2538 015350 004737 015400 JSR PC,.RESTORE ;RESTORE REGISTERS
2539 015354 000207 1$: RTS PC
2540
2541 ;;ROUTINE TO SAVE REGISTERS ON THE STACK
(1) 015356 010546 .SAVE: MOV #5,-(SP) ;;R5 IS SAVED AT 12(SP)
(1) 015360 010446 MOV #4,-(SP) ;;R4 IS SAVED AT 10(SP)
(1) 015362 010346 MOV #3,-(SP) ;;R3 IS SAVED AT 6(SP)
(1) 015364 010246 MOV #2,-(SP) ;;R2 IS SAVED AT 4(SP)
(1) 015366 010146 MOV #1,-(SP) ;;R1 IS SAVED AT 2(SP)
(1) 015370 010046 MOV #0,-(SP) ;;R0 IS SAVED AT (SP)
(1) 015372 016646 000014 MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
(1) 015376 000207 RTS PC ;;RETURN TO CALLER
2542 ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
(1) 015400 012666 000014 .RESTORE:MOV (SP),.14(SP) ;;STORE RETURN PC ON STACK
(1) 015404 012600 MOV (SP),.R0
(1) 015406 012601 MOV (SP),.R1
(1) 015410 012602 MOV (SP),.R2
(1) 015412 012603 MOV (SP),.R3
(1) 015414 012604 MOV (SP),.R4
(1) 015416 012605 MOV (SP),.R5
(1) 015420 000207 RTS PC ;;RETURN
2543
  
```

```
2545                                     ;MESSAGE TABLE*****
2546
2547 015422 041445 030523 020040 MSG1: .ASCII /#CS1 WC BA FC CS2 /
      015430 020040 041527 020040
      015436 020040 041040 020101
      015444 020040 020040 041506
      015452 020040 020040 041440
      015460 031123 020040 020040
2548 015466 051504 020040 020040 .ASCII /DS ER TC#0/
      015474 042440 020122 020040
      015502 020040 041524 021445
2549 015510 051045 053505 047111 MSG2: .ASCII /#REWIND ERROR-BOT NOT SET WHEN PIP CLEARED#/
      015516 020104 051105 047522
      015524 026522 047502 020124
      015532 047516 020124 042523
      015540 020124 044127 047105
      015546 050040 050111 041440
      015554 042514 051101 042105
      015562 043
2550 015563 045 052045 030115 MSG3: .ASCII '#TM03-TE16/TU77 BASIC FUNCTION TEST (CZTECF0)#';..B
      015570 026463 042524 033061
      015576 052057 033525 020067
      015604 040502 044523 020103
      015612 052506 041516 044524
      015620 047117 052040 051505
      015626 020124 041450 052132
      015634 041505 030106 022451
2551 015642 054524 042520 036040 .ASCII /TYPE <CR> TO TERMINATE RESPONSE & 'C TO RESTART#/
      015650 051103 020076 047524
      015656 052040 051105 044515
      015664 040516 042524 051040
      015672 051505 047520 051516
      015700 020105 020046 041536
      015706 052040 020117 042522
      015714 052123 051101 022524
      015722 043
2552 015723 045 042522 044507 MSG4: .ASCII /#REGISTER START = #/
      015730 052123 051105 051440
      015736 040524 052122 036440
      015744 021440
2553 015746 053045 041505 047524 MSG5: .ASCII /#VECTOR = #/
      015754 020122 020075 043
2554 015761 045 047105 020104 MSG6: .ASCII /#END OF PASS #/
      015766 043117 050040 051501
      015774 020123 043
2555 015777 040 020077 043 MSG7: .ASCII / ? #/
2556 016003 045 047520 044523 MSG9: .ASCII /#POSITION ERROR: #/
      016010 044524 047117 042440
      016016 051122 051117 020072
      016024 043
2557 016025 045 051511 041440 MSG10A: .ASCII /#IS CONTROLLER JUMPERED IN NON-STANDARD MODE/<15><12>
      016032 047117 051124 046117
      016040 042514 020122 052512
      016046 050115 051105 042105
      016054 044440 020116 047516
      016062 026516 052123 047101
```

	016070	040504	042122	046440			
	016076	042117	006505	012			
2558	016103	124	050131	020105	.ASCII	/TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD;	#/
	016110	020062	047506	020122			
	016116	047516	026516	052123			
	016124	047101	040504	042122			
	016132	047440	020122	051103			
	016140	043040	051117	051440			
	016146	040524	042116	051101			
	016154	035104	020040	020040			
	016162	020040	043				
2559	016165	045	051104	053111	MSG10:	.ASCII /#DRIVE NUMBER: #/	
	016172	020105	052516	041115			
	016200	051105	020072	043			
2560	016205	045	046123	053101	MSG11:	.ASCII /#SLAVE NUMBER: #/	
	016212	020105	052516	041115			
	016220	051105	020072	043			
2561	016225	045	051127	052111	MSG12:	.ASCII /#WRITE ERROR #/	
	016232	020105	051105	047522			
	016240	020122	043				
2562	016243	045	042522	042101	MSG13:	.ASCII /#READ REVERSE ERROR #/	
	016250	051040	053105	051105			
	016256	042523	042440	051122			
	016264	051117	021440				
2563	016270	051045	040505	020104	MSG14:	.ASCII /#READ FORWARD ERROR #/	
	016276	047506	053522	051101			
	016304	020104	051105	047522			
	016312	020122	043				
2564	016315	045	051127	052111	MSG15:	.ASCII /#WRITE TM ERROR #/	
	016322	020105	046524	042440			
	016330	051122	051117	021440			
2565	016336	020045	051127	052111	MSG15A:	.ASCII /#WRITE TM ERROR ON SECOND TM #/	
	016344	020105	046524	042440			
	016352	051122	051117	047440			
	016360	020116	042523	047503			
	016366	042116	052040	020115			
	016374	043					
2566	016375	045	041520	020040	MSG15B:	.ASCII /#PC #/	
	016402	043					
2567	016403	045	042522	042526	MSG16:	.ASCII /#REVERSE ERROR #/	
	016410	051522	020105	051105			
	016416	047522	020122	043			
2568	016423	045	047506	053522	MSG17:	.ASCII /#FORWARD ERROR #/	
	016430	051101	020104	051105			
	016436	047522	020122	043			
2569	016443	040	051116	020132	MSG20:	.ASCII / NRZ #/	
	016450	043					
2570	016451	040	042520	021440	MSG21:	.ASCII / PE #/	
2571	016456	042440	050130	035124	MSG22:	.ASCII / EXPT: #/	
	016464	021440					
2572	016466	051040	053103	035104	MSG23:	.ASCII / RCVD: #/	
	016474	021440					
2573	016476	041045	051525	052040	MSG24:	.ASCII /#BUS TRAP: #/	
	016504	040522	035120	021440			
2574	016512	053445	035103	021440	MSG25:	.ASCII /#WC: #/	
2575	016520	041045	035101	021440	MSG26:	.ASCII /#BA: #/	

2576	016526	042045	035102	021440	MSG27:	.ASCII	/#DB: #/
2577	016534	044445	044516	020124	MSG28:	.ASCII	/#INIT DID NOT CLEAR RH #/
	016542	044504	020104	047516			
	016550	020124	046103	040505			
	016556	020122	044122	021440			
2578	016564	051445	020103	047516	MSG29:	.ASCII	/#SC NOT RESET BY INIT #/
	016572	020124	042522	042523			
	016600	020124	054502	044440			
	016606	044516	020124	043			
2579	016613	045	051124	020105	MSG30:	.ASCII	/#TRE NOT RESET BY INIT #/
	016620	047516	020124	042522			
	016626	042523	020124	054502			
	016634	044440	044516	020124			
	016642	043					
2580	016643	045	051503	020062	MSG31:	.ASCII	/#CS2 NOT RESET BY INIT #/
	016650	047516	020124	042522			
	016656	042523	020124	054502			
	016664	044440	044516	020124			
	016672	043					
2581	016673	045	046104	020124	MSG32:	.ASCII	/#DLT NOT SET #/
	016700	047516	020124	042523			
	016706	020124	043				
2582	016711	045	041523	047040	MSG33:	.ASCII	/#SC NOT SET #/
	016716	052117	051440	052105			
	016724	021440					
2583	016726	052045	042522	047040	MSG34:	.ASCII	/#TRE NOT SET #/
	016734	052117	051440	052105			
	016742	021440					
2584	016744	044445	020122	047516	MSG35:	.ASCII	/#IR NOT SET BY INIT #/
	016752	020124	042523	020124			
	016760	054502	044440	044516			
	016766	020124	043				
2585	016771	045	051117	047040	MSG36:	.ASCII	/#OR NOT RESET BY INIT #/
	016776	052117	051040	051505			
	017004	052105	041040	020131			
	017012	047111	052111	021440			
2586	017020	047445	020122	047516	MSG37:	.ASCII	/#OR NOT RESET BY 1 SILO ENTRY #/
	017026	020124	042522	042523			
	017034	020124	054502	030440			
	017042	051440	046111	020117			
	017050	047105	051124	020131			
	017056	043					
2587	017057	045	051117	047040	MSG38:	.ASCII	/#OR NOT SET BY SILO FULL #/
	017064	052117	051440	052105			
	017072	041040	020131	044523			
	017100	047514	043040	046125			
	017106	020114	043				
2588	017111	045	040502	020104	MSG39:	.ASCII	/#BAD SILO READ #/
	017116	044523	047514	051040			
	017124	040505	020104	043			
2589	017131	045	051111	047040	MSG40:	.ASCII	/#IR NOT RESET BY SILO FULL #/
	017136	052117	051040	051505			
	017144	052105	041040	020131			
	017152	044523	047514	043040			
	017160	046125	021514				
2590	017164	047045	047117	042455	MSG41:	.ASCII	/#NON-EXIST DRIVE #/

	017172	044530	052123	042040		
	017200	044522	042526	043		
2591	017205	045	047516	026516	MSG42:	.ASCII /NON-EXIST SLAVE#/
	017212	054105	051511	020124		
	017220	046123	053101	021505		
2592	017226	051445	051105	040511	MSG43:	.ASCII /SERIAL NO: #/
	017234	020114	047516	020072		
	017242	043				
2593	017243	045	051105	051501	MSG44:	.ASCII /ERASE HEAD INOPERATIVE#/
	017250	020105	042510	042101		
	017256	044440	047516	042520		
	017264	040522	044524	042526		
	017272	043				
2594	017273	045	047520	051523	MSG45:	.ASCII /POSSIBLE ERASE HEAD PROBLEM: /
	017300	041111	042514	042440		
	017306	040522	042523	044040		
	017314	040505	020104	051120		
	017322	041117	042514	035115		
	017330	040				
2595	017331	103	042510	045503		.ASCII /CHECK POLARITY#/
	017336	050040	046117	051101		
	017344	052111	021531			
2596	017350	051445	052105	052455	MSG46:	.ASCII /SET-UP WRITE ERROR#/
	017356	020120	051127	052111		
	017364	020105	051105	047522		
	017372	021522				
2597	017374	051445	040520	042503	MSG47:	.ASCII /SPACE FORWARD ERROR#/
	017402	043040	051117	040527		
	017410	042122	042440	051122		
	017416	051117	043			
2598	017421	045	050123	041501	MSG48:	.ASCII /SPACE REVERSE ERROR#/
	017426	020105	042522	042526		
	017434	051522	020105	051105		
	017442	047522	021522			
2599	017446	041045	043125	042506	MSG49:	.ASCII /BUFFERED WRITE ERROR#/
	017454	042522	020104	051127		
	017462	052111	020105	051105		
	017470	047522	021522			
2600	017474	041045	052117	051440	MSG50:	.ASCII /BOT SET AFTER BUFFERED WRITE#/
	017502	052105	040440	052106		
	017510	051105	041040	043125		
	017516	042506	042522	020104		
	017524	051127	052111	021505		
2601	017532	047045	020117	047502	MSG51:	.ASCII /NO BOT FROM READ IN PRESET#/
	017540	020124	051106	046517		
	017546	051040	040505	020104		
	017554	047111	050040	042522		
	017562	042523	021524			
2602	017566	052045	020103	047111	MSG52:	.ASCII /BTC INCORRECT #/
	017574	047503	051122	041505		
	017602	020124	043			
2603	017605	045	047515	020114	MSG53:	.ASCII /MOL FAILED TO CLEAR#/
	017612	040506	046111	042105		
	017620	052040	020117	046103		
	017626	040505	021522			
2604	017632	022445	042522	042523	MSG54:	.ASCII /RESET SLAVE TO ON LINE BEFORE CONTINUING/

	017640	020124	046123	053101	
	017646	020105	047524	047440	
	017654	020116	044514	042516	
	017662	041040	043105	051117	
	017670	020105	047503	052116	
	017676	047111	044525	043516	
2605	017704	051445	052105	051440	.ASCII /#SET SW12=1 IF YOU DOT WISH TO REPEAT REWIND OFFLINE TEST#/
	017712	030527	036462	020061	
	017720	043111	054440	052517	
	017726	042040	052117	053440	
	017734	051511	020110	047524	
	017742	051040	050105	040505	
	017750	020124	042522	044527	
	017756	042116	047440	043106	
	017764	044514	042516	052040	
	017772	051505	021524		
2606	017776	044440	042524	035122	MSG56: .ASCII / ITER: #/
	020004	021440			
2607	020006	052045	020115	047516	MSG57: .ASCII /#TM NOT SET#/
	020014	020124	042523	021524	
2608	020022	042445	052111	042510	MSG60: .ASCII /#EITHER TAPE NOT ERASED OR OPI PROBLEM#/
	020030	020122	040524	042520	
	020036	047040	052117	042440	
	020044	040522	042523	020104	
	020052	051117	047440	044520	
	020060	050040	047522	046102	
	020066	046505	043		
2609	020071	045	044122	047440	MSG62: .ASCII /#RH ONLY (NO=0,YES=1): #/
	020076	046116	020131	047050	
	020104	036517	026060	042531	
	020112	036523	024461	020072	
	020120	043			
2610	020121	045	044504	020104	MSG63: .ASCII /#DID NOT AUTO SELECT NRZ#/
	020126	047516	020124	052501	
	020134	047524	051440	046105	
	020142	041505	020124	051116	
	020150	021532			
2611	020152	042045	042111	047040	MSG64: .ASCII /#DID NOT AUTO SELECT PE#/
	020160	052117	040440	052125	
	020166	020117	042523	042514	
	020174	052103	050040	021505	
2612	020202	021445			MSG65: .ASCII /##/
2613	020204	021534			MSG66: .ASCII /##/
2614	020206	042445	035122	021440	MSG67: .ASCII /#ER: #/
2615	020214	051045	046505	053117	MSG69: .ASCII /#REMOVE TMDP FROM SLAVE TO BE TESTED#/
	020222	020105	046524	050104	
	020230	043040	047522	020115	
	020236	046123	053101	020105	
	020244	047524	041040	020105	
	020252	042524	052123	042105	
	020260	021445			
2616	020262	044045	051101	053504	MSG70: .ASCII /#HARDWARE SWR IN USE#/
	020270	051101	020105	053523	
	020276	020122	047111	052440	
	020304	042523	021445		
2617	020310	040445	040524	047040	MSG71: .ASCII /#ATA NOT SET AFTER NON-TRANSFER OPERATION#/

020316	052117	051440	052105
020324	040440	052106	051105
020332	047040	047117	052055
020340	040522	051516	042506
020346	020122	050117	051105
020354	052101	047511	021516

2618

```

2620                                     ;TEST HEADERS*****
2621
2622 020362 022445 052106 035061 MSFT1: .ASCII /MSFT1:RH ADDRESSING #/
      020370 044122 040440 042104
      020376 042522 051523 047111
      020404 020107      043
2623 020407      045 043045 031124 MSFT2: .ASCII /MSFT2:RH REGISTER BITS TEST #/
      020414 051072 020110 042522
      020422 044507 052123 051105
      020430 041040 052111 020123
      020436 042524 052123 021440
2624 020444 022445 052106 035063 MSFT3: .ASCII /MSFT3:RH INITIALIZE TEST #/
      020452 044122 044440 044516
      020460 044524 046101 055111
      020466 020105 042524 052123
      020474 02144C
2625 020476 022445 052106 035064 MSFT4: .ASCII /MSFT4:RH11 SILO TEST 1 #/
      020504 044122 030461 051440
      020512 046111 020117 042524
      020520 052123 030440 021440
2626 020526 022445 052106 035065 MSFT5: .ASCII /MSFT5:RH11 SILO TEST 2 #/
      020534 044122 030461 051440
      020542 046111 020117 042524
      020550 052123 031040 021440
2627 020556 022445 052106 035066 MSFT6: .ASCII /MSFT6:RH11 SILO TEST 3 #/
      020564 044122 030461 051440
      020572 046111 020117 042524
      020600 052123 031440 021440
2628 020606 022445 052106 035067 MSFT7: .ASCII /MSFT7:RH11 SILO TEST 4 #/
      020614 044122 030461 051440
      020622 046111 020117 042524
      020630 052123 032040 021440
2629 020636 022445 052106 030061 MSFT10: .ASCII /MSFT10:RH11 SILO TEST 5 #/
      020644 051072 030510 020061
      020652 044523 047514 052040
      020660 051505 020124 020065
      020666      043
2630 020667      045 043045 030524 MSFT11: .ASCII /MSFT11:NOP TEST#/
      020674 035061 047516 020120
      020702 042524 052123      043
2631 020707      045 043045 030524 MSFT12: .ASCII /MSFT12:REWIND TEST#/
      020714 035062 042522 044527
      020722 042116 052040 051505
      020730 021524
2632 020732 022445 052106 031461 MSFT13: .ASCII /MSFT13:WRITE-READ TEST#/
      020740 053472 044522 042524
      020746 051055 040505 020104
      020754 042524 052123      043
2633 020761      045 043045 030524 MSFT14: .ASCII /MSFT14:SPACE TEST#/
      020766 035064 050123 041501
      020774 020105 042524 052123
      021002      043
2634 021003      045 043045 030524 MSFT15: .ASCII /MSFT15:ERASE TEST#/
      021010 035065 051105 051501
      021016 020105 042524 052123
      021024      043
    
```

2635	021025	045	043045	030524	MSFT16: .ASCII	/MSFT16:TAPE MARK WRITE-READ TEST#/
	021032	035066	040524	042520		
	021040	046440	051101	020113		
	021046	051127	052111	026505		
	021054	042522	042101	052040		
	021062	051505	021524			
2636	021066	022445	052106	033461	MSFT17: .ASCII	/MSFT17:TM SPACE TEST #/
	021074	052072	020115	050123		
	021102	041501	020105	042524		
	021110	052123	021440			
2637	021114	022445	052106	030062	MSFT20: .ASCII	/MSFT20:WRITE CHECK TEST #/
	021122	053472	044522	042524		
	021130	041440	042510	045503		
	021136	052040	051505	020124		
	021144	043				
2638	021145	045	043045	031124	MSFT21: .ASCII	/MSFT21:ERASE HEAD TEST#/
	021152	035061	051105	051501		
	021160	020105	042510	042101		
	021166	052040	051505	021524		
2639	021174	022445	052106	031062	MSFT22: .ASCII	/MSFT22:BUFFERED COMMAND TEST#/
	021202	041072	043125	042506		
	021210	042522	020104	047503		
	021216	046515	047101	020104		
	021224	042524	052123	043		
2640	021231	045	043045	031124	MSFT23: .ASCII	/MSFT23:READ IN PRESET TEST#/
	021236	035063	042522	042101		
	021244	044440	020116	051120		
	021252	051505	052105	052040		
	021260	051505	021524			
2641	021264	022445	052106	032062	MSFT24: .ASCII	/MSFT24:AUTO DENSITY SELECT: WRITE-NRZ,READ-PE#/
	021272	040472	052125	020117		
	021300	042504	051516	052111		
	021306	020131	042523	042514		
	021314	052103	020072	051127		
	021322	052111	026505	051116		
	021330	026132	042522	042101		
	021336	050055	021505			
2642	021342	022445	052106	032462	MSFT25: .ASCII	/MSFT25:AUTO DENSITY SELECT: WRITE-PE,READ-NRZ#/
	021350	040472	052125	020117		
	021356	042504	051516	052111		
	021364	020131	042523	042514		
	021372	052103	020072	051127		
	021400	052111	026505	042520		
	021406	051054	040505	026504		
	021414	051116	021532			
2643	021420	022445	052106	033062	MSFT26: .ASCII	/MSFT26:SEQUENTIAL TAPE MARK TEST#/
	021426	051472	050505	042525		
	021434	052116	040511	020114		
	021442	040524	042520	046440		
	021450	051101	020113	042524		
	021456	052123	043			
2644	021461	045	043045	031124	MSFT27: .ASCII	/MSFT27:REWIND-OFF LINE TEST#/
	021466	035067	042522	044527		
	021474	042116	047455	043106		
	021502	046040	047111	020105		
	021510	042524	052123	043		

2645	021515	045	043536	043	\$CNTG:	.ASCII	/#†G#/
2646	021521	045	053523	036522	\$MSWR:	.ASCII	/#SWR= #/
	021526	021440					
2647	021530	020040	042516	036527	\$MNEW:	.ASCII	/ NEW= #/
	021536	021440					
2648	021540	022477	043		\$QUEST:	.ASCII	/?##/
2649							
2650							
2651		021544				.EVEN	
2652	021544	000000			WDATA:	0	
2653		023256				.+.1510	
2654	023256	000000			RDATA:	0	
2655							
2656		000001				.END	

FT15B	006662	15090	1515				
FT15X	006772	1513	15350				
FT15XX	007176	1554	15660				
FT16	007210	802	803	15730	2023		
FT16A	007242	15780	1601				
FT16B	007246	15790					
FT16X	007426	1599	16020				
FT17	007436	804	805	16090			
FT17A	007456	16120	1666				
FT17B	007462	16130	1643				
FT17C	007622	16330	1642				
FT17D	007670	1626	16440				
FT17D1	007706	16470	1662				
FT17E	007722	16490	1657				
FT17F	010012	1659	16630				
FT17X	010034	1621	1640	1654	1664	16670	
FT2	003376	778	779	10400			
FT2A	003410	10420	1049	1074			
FT2B	003450	1047	10510	1058			
FT2C	003510	1056	10600	1069			
FT2D	003524	10640	1065				
FT2E	003554	1067	10710				
FT2ER	003564	1050	1059	1070	10750	1825	1966
FT2ERA	003614	1079	10820				
FT2ERB	003666	1077	10930				
FT2ERC	003676	1094	10960				
FT2X	003706	1072	10990				
FT20	010040	806	807	16720			
FT20A	010056	16750	1700				
FT20B	010202	16920					
FT20C	010234	16970					
FT20X	010254	1686	1698	17010			
FT21	010264	808	809	17070			
FT21A	010272	17080					
FT21B	010566	1745	17480				
FT21C	010574	1747	17490				
FT21SC	010422	17260	1749				
FT21X	010606	1741	1743	17510			
FT22	010616	810	811	17580	1783		
FT22A	010672	17660	1768				
FT22B	010712	17710	1772				
FT22X	011012	1781	17850				
FT23	011022	812	813	17920	1817	1824	
FT23A	011132	18080	1811	1813			
FT23B	011152	1809	18140				
FT23C	011202	1815	18190				
FT23X	011236	1793	1822	18260			
FT24	011242	814	815	18330	2036		
FT24X	011424	1846	1853	18560			
FT25	011434	816	817	18620	2038		
FT25X	011616	1875	1882	18850			
FT26	011626	818	819	18920	2191	2193	
FT26X	012054	1903	1909	1918	1923	1925	19280
FT27	012064	820	821	19330	2196	2198	
FT27X	012274	1960	1964	19670			
FT27XX	012304	1934	1936	19690			

MSFT13	020732	1350	26320		
MSFT14	020761	1392	1463	26330	
MSFT15	021003	1503	26340		
MSFT16	021025	1576	26350		
MSFT17	021066	1610	26360		
MSFT2	020407	1040	26230		
MSFT20	021114	1673	26370		
MSFT21	021145	1707	26380		
MSFT22	021174	1758	26390		
MSFT23	021231	1794	26400		
MSFT24	021264	1833	26410		
MSFT25	021342	1862	26420		
MSFT26	021420	1893	26430		
MSFT27	021461	1937	26440		
MSFT3	020444	1106	26240		
MSFT4	020476	1152	26250		
MSFT5	020526	1178	26260		
MSFT6	020556	1210	26270		
MSFT7	020606	1263	26280		
MSG1	015422	2068	25470		
MSG10	016165	899	25590		
MSG10A	016025	889	25570		
MSG11	016205	916	25600		
MSG12	016225	1359	1636	1717	25610
MSG13	016243	1369	1590	25620	
MSG14	016270	1376	1595	1733	25630
MSG15	016315	1584	1618	1899	25640
MSG15A	016336	1905	25650		
MSG15B	016375	1556	25660		
MSG16	016403	1476	1646	1687	25670
MSG17	016423	1474	1660	1692	25680
MSG2	015510	25490			
MSG20	016443	1467	2054	25690	
MSG21	016451	1470	2057	25700	
MSG22	016456	1085	1245	1478	25710
MSG23	016466	1089	1249	1483	25720
MSG24	016476	2270	25730		
MSG25	016512	1048	25740		
MSG26	016520	1057	25750		
MSG27	016526	1068	25760		
MSG28	016534	25770			
MSG29	016564	1114	25780		
MSG3	015563	848	853	25500	
MSG30	016613	1118	25790		
MSG31	016643	1124	25800		
MSG32	016673	1163	1274	1301	25810
MSG33	016711	1165	25820		
MSG34	016726	1167	25830		
MSG35	016744	1183	25840		
MSG36	016771	1187	25850		
MSG37	017020	1192	1200	25860	
MSG38	017057	1224	25870		
MSG39	017111	1243	25880		
MSG4	015723	854	25520		
MSG40	017131	1220	25890		
MSG41	017164	913	25900		

RDYDX	000636	725*	1314*	1347*	1401*	1501*	1520*	1535*	1574*	1614*	1774*	1987		
REGS	000572	704*	856	858	875									
RFD	000634	724*	1418*	1441*	1455									
RHOF	000726	753*	939	941	1297									
RHTF	000722	751*	887*	1031	1033*	2280								
RH17F	000604	712*	956*	962*	1150	1176	1208	1261	1281					
RRD	000632	723*	1417*	1440*	1448									
RTRN	000646	729*	2218											
RWND	012442	1354	1394	1504	1517	1578	1612	1675	1708	1722	1727	1759	1834	1847
		1863	1876	1895	1910	1938	2003*							
SAV1	000666	737*												
SAV2	000670	738*												
SAV3	000672	739*												
SCHN	001710	841	842*											
SCNT	000642	727*	1419*	1429*	1438*	1647*	1912*	1983						
SCOLP	000674	740*	1049*	1058*	1069*	1107*	1169*	1179*	1211*	1264*	1284*	1308*	1327*	1361*
		1749*	1783*	1817*	1824*	1949*	2163	2167	2278*					
SCOPE	013334	1096	1144	2158*	2279									
SERFL	000712	747*	1408	1424	1434	1512	1620	1639	1653	1685	1845	1874	1902	1908
		1917	1922	2017*	2043*	2102								
SERNUM	000562	700*												
SLVN	000614	716*	918	920	927	970*	1792	2212						
SN	000540	688*	935											
SNPG	015234	2502	2505	2511	2513	2515*								
SNPT	015156	936	2496*											
START	001600	659	829*											
STFLG	000704	744*	955*	984	989*									
STMSK	000660	734*	976*	1403*	1500*	1524*	1583*	1589*	1617*	1637*	1651*	1778*	1898*	2033*
		2040*	2041											
STSCD	003126	969	989*											
STO	002124	878*	881											
STOB	001764	854*	2282											
ST1	002146	884*	886											
ST1A	002166	889*	915	1034										
ST2	002344	912	916*	932										
ST3	002454	929	933*											
ST4	002540	663	948*											
SWR	000550	695*	834	838*	841*	967	981	986	1009	1076	1093	1131	1140	1237
		1253	1461	1488	1933	2044	2062	2090	2159	2172	2236	2238*	2247*	2263
		2274	2524	2529	2533									
SWREG	000176	653*	838	841	2236	2247	2524							
TAG	007206	1555*	1565*	1569*	2080									
TC	000542	689*	927*	970*	1329*	1544*	1820	1975*	2055	2088	2212*			
TEMPST	000732	755*												
TEMP1	000652	731*	1552*	1553	1628*	1641*	2082	2305*	2314	2338*				
TEMP2	000654	732*												
TEMP3	000656	733*	2487	2490										
TEND	003162	822	991	997*	1299	2200								
TENDX	003260	1008	1010	1012*										
TEX	014614	2379	2411*											
TIB	000602	711*	2308	2312	2317	2322	2332	2335	2342*	2343	2369*	2370*	2371	
TINER	014332	2334	2337	2348	2351	2358*								
TKB	000554	697*	2225	2369	2397									
TKS	000552	696*	954*	2366*	2367	2395								
TLAST	001120	823*	990	2203*										
TMCHK	013164	1586	1592	1597	1622	1655	1901	1907	1916	1921	2100*			

\$CATCH	5590	640
\$CHAIN	5590	841
\$CHNMO	5590	970
\$RESTO	5590	2542
\$SAVE	5590	2541
.\$ACT1	5590	642
.\$EOP	5590	1006

. ABS. 023260 000

ERRORS DETECTED: 0

CZTECF.CZTECF/CRF=CZTEAE.SML/ML.CZTECF.P11
RUN-TIME: 3 5 .9 SECONDS
RUN-TIME RATIO: 19/10=1.8
CORE USED: 11K (22 PAGES)