

TM03/TU45

BASIC FUNCTION TEST
CZTUQA0

AH-E494A-MC

COPYRIGHT © 75-78
FICHE 1 OF 1

JUL 1978

digital
MADE IN USA

This microfiche card contains a grid of frames, each containing a small table of test data. The data is organized into columns and rows, with some frames containing headers and footers. The text is too small to read clearly but appears to be a structured list of test results or parameters.



.REM 2

IDENTIFICATION

PRODUCT CODE: AC-E493A-MC
PRODUCT TITLE: CZTUQA0 TM03/TU45 BASIC FUNCTION TEST
DATE CREATED: 25 MAY 1978
MAINTAINER: CSS - NASHUA
AUTHOR: J. G. ADAMS/R. J. COLLINS

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) 1975, 1978 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 TM03/TU45 MAG TAPE SYSTEM. ALL FUNCTIONS; WRITE, READ, SPACE, ERASE, REWIND, ETC; WILL BE TESTED. IN ADDITION TO THE TM03/TU45 TESTS, THE RH WILL BE TESTED SEPARATELY IN SO FAR AS IT IS POSSIBLE TO SEPARATE THE RH FROM THE TM03/TU45 ITSELF.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP11 PROCESSOR
- B. 8K OF CORE
- C. CONSOLE TTY
- D. TM03 MAGTAPE CONTROLLER
- E. MASS BUS CONTROLLER
- F. TU45 MAG TAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD BINARY LOADING PROCEDURE

4. STARTING PROCEDURE

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

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

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

4.1 AUTOMATIC MODE OPERATION

IF THIS PROGRAM IS LOADED AND RUN IN AUTOMATIC (CHAIN) MODES
DEFAULT RESPONSES TO OPERATOR REQUESTS ARE USED, AND ALL AVAIL-
ABLE TM03/TU45 COMBINATIONS ARE TESTED. ADDITIONALLY THE SOFTWARE
SWR IS INVOKED WITH A SWITCH SETTING OF 100000 (HALT ON ERROR)
IF LOADED VIA ACT11 CHAIN MODE.

**EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMDP CHAIN MODE THE
PROGRAM WILL NOT TEST TM03 DRIVE #0, TU45 SLAVE #0.

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

4.2 SAMPLE START AT 200

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

PARAMETER REQUEST: <DEFAULT> (RESPONSE) LOCATION:

TM03-TU45 BASIC FUNCTIONS TEST (CZTUQA0)
TYPE C TO RESTART

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

5. CONSOLE SWITCH SETTING

CONTROL:

1) CONTROL G < G>;
SELECTS THE SOFTWARE SWR AND ALLOWS THE USER TO SELECT NEW SWITCH SETTINGS.

THE MACHINE WILL THEN TYPE: SWR=XXXXXXNEW=
WHERE: XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWR.
AFTER THE ''NEW='' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
OF THE FOLLOWING AT THE TTY:

- A) TYPE A NEW SWITCH SETTING
- B) IF A <CR> IS THE FIRST KEY DEPRESED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

2) CONTROL A < A>;
ALTERNATES USAGE OF SWR FROM HARDWARE TO SOFTWARE & VICE VERSA,

3) CONTROL C < C>;
RESTARTS PROGRAM AT 200

4) CONTROL U < U>;
DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST.

ALL SWITCHES EXCEPT 5-9 ARE USED AND THE NORMAL, OR DEFAULT,
RUN IS DONE WITH ALL SWITCHES SET TO ZERO (0).
ALL HARDWARE SWITCHES ARE DYNAMIC, AND MAY BE CHANGED AT ANY TIME.

SW15(100000): 1=HALT ON ERROR
0=CONTINUE
SW14(040000): 1=LOOP ON ERROR (SCOPE: RH TESTS ONLY)
0=CONTINUE
SW13(020000): 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
SW12(010000): 1=CONTINUOUS CYCLE
0=HALT AT END OF PASS
SW11(004000): 1=INHIBIT ITERATION
0=DO ALL ITERATIONS PER TEST
SW10(002000): 1=HALT AT END OF CURRENT TEST
0=CONTINUE
SW9-5: N/A
SW4-0: SELECT TEST NUMBER::00=ALL TESTS

THE USE OF SWC-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/TU45 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/TU45 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 OPERATI^{ON} 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.

FT26: REWIND: OFF LINE THIS TEST WILL ASSURE
THAT THE UNIT WILL REWIND AND
GO OFF LINE. (NOT IF IN CONTINUOUS CYCLE)

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

(THIS TEST IS NOT PERFORMED WHEN CONTINUOUS CYCLE OPERATION IS SELECTED: SW 12 = 1)

520
521
522

.LIST BIN,LOC,SEQ
.TITLE TM03/TU45 BASIC FUNCTION TEST
;CZTUQA0

523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546

```
;25 MAY 78  
;R. BARNES/R. J. COLLINS  
.MCALL .SACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE  
.NLIST MC  
.LIST ME  
.ENABLE ABS,AMA  
  
;CONSOLE SWITCHES*****  
;  
;SW15(100000) 1=HALT ON ERROR  
; 0=CONTINUE  
;SW14(040000) 1=LOOP ON ERROR (SCOPE(040000) RH TESTS ONLY)  
; 0=CONTINUE  
;SW13(02000): 1=DO NOT PRINT ERRORS  
; 0=PRINT ERRORS  
;SW12(010000): 1=CONTINUOUS CYCLE  
; 0=HALT AT END OF PASS  
;SW11(40000): 1=INHIBIT ITERATIONS  
; 0=DO ITERATIONS  
;SW10(002000): 1=HALT AT END OF EACH TEST  
; 0=CONTINUE  
;SW0-4: SELECT TEST NUMBER :: 00=ALL TESTS  
;USE SOFTWARE SWR IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
```



```
593                                     ;REGISTER EQUIVS*****
594
595          000000          R0=%0
596          000001          R1=%1
597          000002          R2=%2
598          000003          R3=%3
599          000004          R4=%4
600          000005          R5=%5
601          000006          SP=%6
602          000007          PC=%7
603
604
605
606          ;ACT11 HOOK *****
607          000764          $SVPC=.          ;SAVE CURRENT LOCATION CTR
608          000046          .=46
609 000046 003154          .WORD $ENDAD          ;SET LOCATION 46
610          000052          .=52
611 000052 000000          .WORD 0          ;SET LOCATION 52 = 0
612          000764          .=$SVPC          ;RESTORE LOCATION CTR
613
614          ;TTY INTERRUPT VECTOR*****
615
616          000060          .=60
617 000060 012656          .WORD TTINT          ;TTY INTERRUPT HEADER ADDRESS
618 000062 000340          .WORD 340          ;PRIORITY LEVEL 7
619
620          ;SOFTWARE SWITCH REGISTER*****
621          ;USED IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
622
623          000176          .=176
624 000176 000000          SWREG: 0          ;SOFTWARE SWITCH REGISTER
625
626
627          ;START ADDRESS*****
628
629          000200          .=200
630 000200 000137 001600          JMP START ;PROGRAM START
631
632          ;RESTART ADDRESS*****
633          000210          .=210
634 000210 000137 002542          JMP ST4
635
636          ;TM03 INTERRUPT VECTOR*****
637
638          000224          .=224
639 000224 012646          MTINT          ;TAPE INTERRUPT HANDLER ADDRESS
640 000226 000340          340
641
```



```
642
643          000510          .=510
644          :MASS BUS REGISTER EQUIVS*****
645
646 000510 172440          C1: 172440
647 000512 172442          WC: 172442
648 000514 172444          BA: 172444
649 000516 172446          FC: 172446
650 000520 172450          CS: 172450
651 000522 172452          DS: 172452
652 000524 172454          ER: 172454
653 000526 172456          AS: 172456
654 000530 172460          CC: 172460
655 000532 172462          DB: 172462
656 000534 172464          MR: 172464
657 000536 172466          DT: 172466
658 000540 172470          SN: 172470
659 000542 172472          TC: 172472
660 000544 172474          BAE: 172474
661
662          :CONSTANTS*****
663
664 000546 177776          PSW: 177776          :PROCESSOR STATUS
665 000550 177570          SWR: 177570          :SWITCH REGISTER
666 000552 177560          TKS: 177560          :TTY READER STATUS
667 000554 177562          TKB: 177562          :TTY READ BUFFER
668 000556 177564          TPS: 177564          :TTY PUNCH STATUS
669 000560 177566          TPB: 177566          :TTY PUNCH BUFFER
670 000562 177777          SERNUM: 177777          :SERIAL NUMBER
671 000564 000011          DRVTP: 011          :DRIVE TYPE
672 000566 000010          ITAMT: 10          :ITERATION AMOUNT
673 000570 000224          VECT: 224          :INTERRUPT VECTOR(RH)
674 000572 172440          REGS: 172440          :STARTING REGISTER ADDRESS
675 000574 000004          BTRP: 4          :BUS TRAP ADDRESS
676 000576 000006          BTRP2: 6          :BUS TRAP PRIORITY LEVEL 7
```

```
677 ;FLAGS AND COUNTERS*****
678
679 000600 000000 TOB: 0
680 000602 000000 TIB: 0
681 000604 000000 RH17F: 0
682 000606 000000 HDRFL: 0
683 000610 000000 EMADDR: 0
684 000612 000000 DRVN: 0
685 000614 000000 SLVN: 0
686 000616 000000 BADDR: 0
687 000620 000000 FCNT: 0
688 000622 000000 WCNT: 0
689 000624 000000 RCNT: 0
690 000626 000000 ERRP: 0
691 000630 000000 ERRP1: 0
692 000632 000000 RRD: 0
693 000634 000000 RFD: 0
694 000636 000000 RDYDX: 0
695 000640 000000 OPDYX: 0
696 000642 000000 SCNT: 0
697 000644 000000 PFLG: 0
698 000646 000000 RTRN: 0
699 000650 000000 ERADD: 0
700 000652 000000 TEMP1: 0
701 000654 000000 TEMP2: 0
702 000656 000000 TEMP3: 0
703 000660 000000 STMSK: 0
704 000662 000000 ITCNT: 0
705 000664 000000 DSAV: 0
706 000666 000000 SAV1: 0
707 000670 000000 SAV2: 0
708 000672 000000 SAV3: 0
709 000674 000000 SCOLP: 0
710 000676 000000 ITRLP: 0
711 000700 000000 EXFL: 0
712 000702 000000 PEXFL: 0
713 000704 000000 STFLG: 0
714 000706 000000 LTADD: 0
715 000710 000000 FUN: 0
716 000712 000000 SERFL: 0
717 000714 000000 CRCNT: 0
718 000716 000000 UDES: 0
719 000720 000000 PATRN: 0
720 000722 000000 RHTF: 0
721 000724 000000 NRZOF: 0
722 000726 000000 RHOF: 0
723 000730 000000 PCNTR: 0
724 000732 000000 TEMPST: 0
725 000734 000000 COUNT: 0
726 000736 000000 RDSW: 0
727
```

```
728  
729  
730 ;DATA PATTERN GENERATORS*****  
731 000740 000000 DATBL: 0  
732 000742 012440 DATA0: DAT1 ;ALL ONE BITS  
733 000744 012460 DATA1: DAT2 ;ALL ZERO BITS  
734 000746 012464 DATA2: DAT3 ;ALTERNATING ONE/ZERO BITS  
735 000750 012472 DATA3: DAT4 ;ALL BITS 0-377
```

736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786

000752 000000
000754 000000
000756 003216
000760 003216
000762 003316
000764 003316
000766 003640
000770 003640
000772 004060
000774 004060
000776 004206
001000 004206
001002 004400
001004 004400
001006 004652
001010 004652
001012 004746
001014 004746
001016 005102
001020 005102
001022 005220
001024 005220
001026 005332
001030 005332
001032 005644
001034 005644
001036 006516
001040 006516
001042 006716
001044 006716
001046 007144
001050 007144
001052 007546
001054 007546
001056 007772
001060 007772
001062 010324
001064 010324
001066 010530
001070 010530
001072 010750
001074 010750
001076 011142
001100 011142
001102 011334
001104 011334
001106 003110
001110 000026

: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

TLAST: .WORD TEND
.WORD 26

:CONTAINS # OF TESTS

```

787          001600          . =1600
788          ;PROGRAM START AND HOUSEKEEPING*****
789
790 001600 012706 000500      START: MOV #500,SP          ;SET STACK POINTER
791 001604 013746 000006      MOV @#6,-(SP)          ;SAVE VECTORS
792 001610 013746 000004      MOV @#4,-(SP)
793 001614 012737 001640 000004  MOV #1$,@#4          ;SET UP FOR TIMEOUT
794 001622 005037 000006      CLR @#6
795 001626 022777 177777 176714  CMP #-1,@SWR          ;REFERENCE HARDWARE SWITCH REGISTER
796 001634 001402          BEQ 2$
797 001636 000404          BR 3$
798 001640 022626          1$: CMP (SP)+,(SP)+          ;ADJUST STACK
799 001642 012737 000176 000550 2$: MOV #SWREG,SWR          ;POINT TO SOFTWARE SWITCH REG
800 001650 012637 000004          3$: MOV (SP)+,@#4          ;RESTORE VECTORS
801 001654 012637 000006      MOV (SP)+,@#6
802 001660 005027          CLR (PC)+
803 001662 000000          CHNFLG: .WORD 0          ;;CLEAR CHAIN INDICATOR
804          ;;CHAIN MODE INDICATOR
805 001664 022737 003154 000042  CMP #SENDAD,@#42      ;;1/0 = CHAIN/NOT CHAIN MODE
806 001672 001404          BEQ 50$              ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
807 001674 005737 000042      TST @#42              ;;BRANCH IF IN DUMP MODE
808 001700 001413          BEQ 52$
809 001702 000406          BR 51$
810 001704 012737 000176 000550 50$: MOV #SWREG,SWR          ;;INVOKE SOFTWARE SWR
811 001712 012777 100000 176630  MOV #100000,@SWR      ;;WITH HALT ON ERROR SET
812 001720 005237 001662          51$: INC CHNFLG          ;;SET CHNFLG = CHAIN MODE
813 001724 000137 002556          JMP TSCD              ;;GO TO CHAIN ADDRESS
814 001730          52$:
815 001730 122737 000006 000041 4$: CMPB #6,@#41          ;BRANCH IF LOADED VIA TMDP (DUMP MODE)
816 001736 001004          BNE 5$
817 001740 012704 016675      MOV #MSG69,R4          ;ADVISE USER TO REMOVE TMDP FROM UUT
818 001744 004737 013442      JSR PC,TTOUT
819 001750 012704 014470          5$: MOV #MSG3,R4
820 001754 004737 013442      JSR PC,TTOUT          ;PRINT TITLE
821 001760 112737 000043 014470  MOVB #'#,MSG3          ;DO NOT PRINT TITLE ON RESTART
822 001766 012704 014623      STOB: MOV #MSG4,R4
823 001772 004737 013442      JSR PC,TTOUT          ;REQUEST REGISTER ADDRESS
824 001776 013703 000572      MOV REGS,R3
825 002002 004737 013572      JSR PC,OCTP          ;PRINT CURRENT ADDRESS
826 002006 012705 000572      MOV #REGS,R5          ;SET ADDRESS SAVE LOC
827 002012 012701 000007      MOV #7,R1            ;SET SIZE OF RESPONSE
828 002016 012702 176400      MOV #176400,R2       ;SET UPPER LIMIT
829 002022 012703 172300      MOV #172300,R3       ;SET LOWER LIMIT
830 002026 004737 013120      JSR PC,TTR           ;GO GET RESPONSE
831 002032 012704 014646      MOV #MSG5,R4
832 002036 004737 013442      JSR PC,TTOUT          ;REQUEST VECTOR
833 002042 013703 000570      MOV VECT,R3
834 002046 004737 013572      JSR PC,OCTP          ;PRINT CURRENT VECTOR
835 002052 012705 000570      MOV #VECT,R5          ;SET ADDRESS SAVE LOC
836 002056 012701 000004      MOV #4,R1            ;SET SIZE OF RESPONSE
837 002062 012702 000224      MOV #224,R2          ;SET UPPER LIMIT
838 002066 012703 000150      MOV #150,R3          ;SET LOWER LIMIT
839 002072 004737 013120      JSR PC,TTR           ;GO GET RESPONSE
840 002076 013700 000570      MOV VECT,R0          ;GET VECTOR
841 002102 012720 012646      MOV #MTINT,(R0)+     ;LOAD INTERRUPT ADDRESS IN VECTOR
842 002106 012710 000340      MOV #340,(R0)        ;LOAD PRIORITY

```

843	002112	013700	000572		MOV	REGS,R0		;GET START OF REGS
844	002116	012701	000017		MOV	#17,R1		;SET NUMBER OF REGS
845	002122	012702	000510		MOV	#C1,R2		;GET START OF TABLE
846	002126	010022		ST0:	MOV	R0,(R2)+		;BUILD TABLE
847	002130	062700	000002		ADD	#2,R0		;BUMP ADDRESS
848	002134	005301			DEC	R1		;SEE IF DONE
849	002136	001373			BNE	ST0		;IF NOT: BR
850	002140	012702	000600		MOV	#TOB,R2		
851	002144	012700	000054		MOV	#54,R0		
852	002150	005022		ST1:	CLR	(R2)+		;CLEAR FLAGS + COUNTERS
853	002152	005300			DEC	R0		
854	002154	001375			BNE	ST1		
855	002156	012737	000001	000722	MOV	#1,RHTF		;SET ADDRESS TEST FLAG
856	002164	000137	002750		JMP	TSRH		;GO DO INITIAL ADDRESS TEST PASS
857	002170	012704	014725	ST1A:	MOV	#MSG10,R4		
858	002174	004737	013442		JSR	PC,TTOUT		;REQUEST DRIVE NUMBER
859	002200	013703	000612		MOV	DRVN,R3		;GET CURRENT DRIVE #
860	002204	004737	013572		JSR	PC,OCTP		;AND TYPE IT
861	002210	012705	000612		MOV	#DRVN,R5		;SET ADDRESS OF DRIVE NUMBER SAVE
862	002214	012701	000002		MOV	#2,R1		;SET SIZE OF RESPONSE
863	002220	012702	000007		MOV	#7,R2		;SET UPPER LIMIT
864	002224	012703	000000		MOV	#0,R3		;SET LOWER LIMIT
865	002230	004737	013120		JSR	PC,TTR		;GO GET RESPONSE
866	002234	012777	000040	176256	MOV	#40,acs		;SET INIT
867	002242	053777	000612	176250	BIS	DRVN,acs		;SET DRIVE NUMBER
868	002250	005777	176234		TST	ac1		;ACCESS DRIVE
869	002254	032777	010000	176236	BIT	#10000,acs		;SEE IF NED
870	002262	001405			BEQ	ST2		;IF NOT: BR
871	002264	012704	015657		MOV	#MSG41,R4		
872	002270	004737	013442		JSR	PC,TTOUT		;PRINT NOT AVAIL
873	002274	000735			BR	ST1A		;REDO DRIVE REQUEST
874	002276	012704	014745	ST2:	MOV	#MSG11,R4		
875	002302	004737	013442		JSR	PC,TTOUT		;REQUEST SLAVE NUMBER
876	002306	013703	000614		MOV	SLVN,R3		;GET CURRENT SLAVE #
877	002312	004737	013572		JSR	PC,OCTP		;AND TYPE IT
878	002316	012705	000614		MOV	#SLVN,R5		;SET ADDRESS OF SLAVE SAVE
879	002322	012701	000002		MOV	#2,R1		;SET SIZE OF RESPONSE
880	002326	012702	000007		MOV	#7,R2		;SET UPPER LIMIT
881	002332	012703	000000		MOV	#0,R3		;SET LOWER LIMIT
882	002336	004737	013120		JSR	PC,TTR		;GO GET RESPONSE
883	002342	012777	000040	176150	MOV	#40,acs		;INIT
884	002350	053777	000612	176142	BIS	DRVN,acs		;SET DRIVE NUMBER
885	002356	013777	000614	176156	MOV	SLVN,atc		;LOAD SLAVE NUMBER
886	002364	032777	002000	176144	BIT	#2000,adt		;SEE IF SLAVE PRESENT
887	002372	001005			BNE	ST3		;IF SO: BR
888	002374	012704	015700		MOV	#MSG42,R4		
889	002400	004737	013442		JSR	PC,TTOUT		;PRINT NON-EXIST SLAVE
890	002404	000734			BR	ST2		;REDO SLAVE REQUEST
891	002406	012704	015721	ST3:	MOV	#MSG43,R4		
892	002412	004737	013442		JSR	PC,TTOUT		;PRINT SERIAL NUMBER TAG
893	002416	017703	176116		MOV	@SN,R3		
894	002422	004737	014120		JSR	PC,SNPT		;PRINT SERIAL NUMBER
895	002426	005037	000604		CLR	RH17F		;SET RH INDICATOR = RH11
896	002432	013746	000004		MOV	@#4,-(SP)		;SAVE ERROR TRAP VECTORS
897	002436	013746	000006		MOV	@#6,-(SP)		;AND PRIORITY
898	002442	012737	002466	000004	MOV	#1\$,@#4		;SET TIME OUT TRAP TO 1\$ BELOW

```
899 002450 005037 000006          CLR    @#6
900 002454 005777 176064          TST    @BAE          ;REFERENCE BAE REGISTER
901 002460 012737 000001    000604  MOV    #1,RH17F      ;SET FLAG = RH70
902 002466 012637 000006    1$:   MOV    (SP)+,@#6     ;RESTORE ERROR TRAP
903 002472 012637 000004          MOV    (SP)+,@#4
904 002476 012704 016552          MOV    #MSG62,R4     ;GET REQUEST
905 002502 004737 013442          JSR    PC,TTOUT      ;REQUEST RH11 ONLY RESPONSE
906 002506 013703 000726          MOV    RHOF,R3       ;GET CURRENT FLAG SETTING
907 002512 004737 013572          JSR    PC,OCTP       ;AND TYPE IT
908 002516 012705 000726          MOV    #RHOF,R5      ;SET FLAG ADDRESS
909 002522 012701 000002          MOV    #2,R1         ;SET SIZE OF RESPONSE
910 002526 012702 000001          MOV    #1,R2         ;SET UPPER LIMIT
911 002532 012703 000000          MOV    #0,R3         ;SET LOWER LIMIT
912 002536 004737 013120          JSR    PC,TTR        ;GO GET RESPONSE
913
914                                ;START 210
915 002542 012706 000500    ST4:  MOV    #500,SP  ;SET STACK PTR
916 002546 005037 000730          CLR    PCNTR        ;CLEAR PASS COUNTER
917 002552 004737 014222          JSR    PC,GTSWR     ;GET SWITCHES
```

```

918                                     ;TEST SCHEDULAR*****
919
920 002556 052777 000100 175766 TSCD: BIS #100,@TKS ;SET KEYBOARD IE BIT
921 002564 005037 000704 CLR STFLG ;CLEAR SINGLE TEST FLAG
922 002570 017700 175754 MOV @SWR,RO
923 002574 042700 177740 BIC #177740,RO
924 002600 001125 BNE STSCD ;GO SELECT SINGLE TEST
925 002602 005737 001662 TST CHNFLG ;:BRANCH IF NOT IN CHAIN MODE
926 002606 001457 BEQ TSCDA
927 002610 012737 177777 000612 MOV #-1,DRVN ;;INITIALIZE DRIVE #
928 002616 012737 177777 000614 NXTDRV: MOV #-1,SLVN ;;INITIALIZE SLAVE #
929 002624 012777 000040 175666 1$: MOV #40,@CS ;;INIT CONTROLLER
930 002632 005237 000612 INC DRVN ;;STEP DRIVE #
931 002636 022737 000010 000612 CMP #10,DRVN ;;EXIT IF ALL DRIVES TESTED
932 002644 001524 BEQ $DONE ;;FOR AVAILABILITY
933 002646 013777 000612 175644 MOV DRVN,@CS ;;LOAD DRIVE #
934 002654 005777 175630 TST @C1 ;;ACCESS DRIVE
935 002660 032777 010000 175632 BIT #10000,@CS ;;BRANCH IF DRIVE NON EXISTANT
936 002666 001356 BNE 1$ ;;(NED = 1)
937 002670 005237 000614 NXTSLV: INC SLVN ;;STEP SLAVE # AND BRANCH
938 002674 001011 BNE 1$ ;;IF NOT SLAVE 0
939 002676 005737 000612 TST DRVN ;;BRANCH IF NOT DRIVE # 0
940 002702 001006 BNE 1$
941 002704 122737 000006 000041 CMPB #6,@#41 ;;BRANCH IF NOT TMDP
942 002712 001002 BNE 1$
943 002714 005237 000614 INC SLVN ;;STEP TO SLAVE # 1
944 002720 022737 000010 000614 1$: CMP #10,SLVN ;;BRANCH IF ALL SLAVES TESTED
945 002726 001733 BEQ NXTDRV ;;FOR AVAILABILITY
946 002730 013777 000614 175604 MOV SLVN,@TC ;;LOAD SLAVE UNIT #
947 002736 032777 002000 175572 BIT #2000,@DT ;;BRANCH IF SLAVE NOT
948 002744 001751 BEQ NXTSLV ;;PRESENT (SPR = 0)
949 002746 000240 TSCDA: NOP
950 002750 012737 000752 000706 TSRH: MOV #TSTTBL,LTADD
951 002756 062737 000004 000706 TSCD0: ADD #4,LTADD
952 002764 013737 000706 000676 TSCD1: MOV LTADD,ITRLP
953 002772 062737 000002 000676 ADD #2,ITRLP ;SET ITERATION ADDRESS
954 003000 005037 000660 CLR STMSK
955 003004 005037 000626 CLR ERRP
956 003010 005037 000606 CLR HDRFL ;CLEAR PRINT HEADER FLAG
957 003014 017700 175666 MOV @LTADD,RO ;SET POINTER TO TEST
958 003020 000110 JMP (RO) ;GO TO TEST
959 003022 032777 002000 175520 TSCD2: BIT #2000,@SWR ;SEE IF HALT ON TEST
960 003030 001401 BEQ TSCD3 ;IF NOT: BR
961 003032 000000 HALT
962 003034 005737 000704 TSCD3: TST STFLG ;SE IF SINGLE TEST
963 003040 001746 BEQ TSCD0 ;IF NOT: BR
964 003042 017700 175502 MOV @SWR,RO
965 003046 042700 177740 BIC #177740,RO ;BRANCH IF ALL TESTS SELECTED
966 003052 001641 BEQ TSCD
967 003054 012737 000001 000704 STSCD: MOV #1,STFLG ;SET SINGLE TEST FLAG
968 003062 023700 001110 CMP TLAST,RO ;SEE IF EXCEEDED TESTS
969 003066 002410 BLT TEND ;IF SO: BR
970 003070 006300 ASL RO
971 003072 006100 ROL RO ;SET TABLE MODIFIER
972 003074 012737 000752 000706 MOV #TSTTBL,LTADD
973 003102 060037 000706 ADD RO,LTADD ;SET TEST POINTER

```


974	003106	000726			BR	TSCD1		
975	003110	005737	001662		TEND:	TST	CHNFLG	;BRANCH IF IN CHAIN MODE
976	003114	001265				BNE	NXTSLV	
977	003116	012704	014661		\$DONE:	MOV	#MSG6,R4	
978	003122	004737	013442			JSR	PC,TTOUT	;PRINT END OF PASS
979	003126	013703	000730			MOV	PCNTR,R3	
980	003132	004737	013572			JSR	PC,OCTP	;PRINT PASS NUMBER
981	003136	005000				CLR	RO	
982	003140	005300			1\$:	DEC	RO	
983	003142	001376				BNE	1\$	
984	003144	013700	000042			MOV	@#42,RO	;GET ACT11 RETURN ADDRESS
985	003150	001405				BEQ	HERE	;BRANCH IF NOT ACT11
986	003152	000005				RESET		
987	003154	004710			\$ENDAD:	JSR	PC,(RO)	
988	003156	000240				NOP		
989	003160	000240				NOP		
990	003162	000240				NOP		
991	003164	000240			HERE:	NOP		
992	003166	005737	001662			TST	CHNFLG	;BRANCH IF IN CHAIN MODE
993	003172	001005				BNE	TENDX	
994	003174	032777	010000	175346		BIT	#10000,@SWR	;SEE IF HALT ON PASS
995	003202	001001				BNE	TENDX	;IF NOT: BR
996	003204	000000				HALT		
997	003206	005237	000730		TENDX:	INC	PCNTR	;BUMP PASS COUNTER
998	003212	000137	002556			JMP	TSCD	;RESTART

```
999
1000 ;RH ADDRESSING TEST*****
1001
1002 003216 012737 016771 000610 FT1: MOV #MSFT1,EMADDR ;SET HEADER
1003 003224 012777 013004 175342 MOV #TRAP,@BTRP ;SET TRAP HANDLER ADDRESS
1004 003232 012777 000340 175336 MOV #340,@BTRP2
1005 003240 012700 000016 MOV #16,R0 ;SET NUMBER OF REGISTERS
1006 003244 013701 000510 MOV C1,R1 ;GET FIRST ADDRESS (CS1)
1007 003250 005711 FT1A: TST (R1) ;REFERENCE REGISTER
1008 003252 000240 NOP ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
1009 003254 005300 FT1B: DEC R0 ;SEE IF DONE ALL
1010 003256 001403 BEQ FT1X ;IF SO: BR
1011 003260 062701 000002 ADD #2,R1 ;BUMP ADDRESS POINTER
1012 003264 000771 BR FT1A ;CONTINUE
1013 003266 012777 000006 175300 FT1X: MOV #6,@BTRP ;RESET TRAP CATCHER
1014 003274 005737 000722 TST RHTF ;SEE IF INITIAL ADDRESS TEST PASS
1015 003300 001404 BEQ FT1XX ;IF NOT: BR
1016 003302 005037 000722 CLR RHTF ;CLEAR FLAG
1017 003306 000137 002170 JMP ST1A ;RETURN
1018 003312 000137 003022 FT1XX: JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

1019
1020                                     ;RH REGISTER BITS READ/WRITE*****
1021
1022 003316 012737 017016 000610 FT2:  MOV    #MSFT2,EMADDR  ;SET TEST HEADER
1023 003324 012701 177777          MOV    #-1,R1        ;SET ALL ONES PATTERN
1024 003330 004737 012620          FT2A: JSR    PC,INIT1   ;GO INIT
1025 003334 013700 000512          MOV    WC,R0        ;GET ADDRESS OF WORD COUNT
1026 003340 010102          MOV    R1,R2        ;SET EXPT REGISTER BIT PATTERN
1027 003342 010110          MOV    R1,(R0)      ;LOAD PATTERN
1028 003344 021002          CMP    (R0),R2      ;SEE IF EXPT=RCVD
1029 003346 001410          BEQ    FT2B         ;IF SO: BR
1030 003350 012737 015205 000650  MOV    #MSG25,ERADD ;SET CODE
1031 003356 012737 003330 000674  MOV    #FT2A,SCOLP  ;SET SCOPE
1032 003364 004737 003504          JSR    PC,FT2ER     ;GO DO ERROR
1033 003370 013700 000514          FT2B: MOV    BA,R0        ;GET ADDRESS OF BUS ADDRESS
1034 003374 010102          MOV    R1,R2
1035 003376 042702 000001          BIC    #1,R2        ;SET EXPT PATTERN
1036 003402 010110          MOV    R1,(R0)      ;LOAD PATTERN
1037 003404 020210          CMP    R2,(R0)      ;SEE IF EXPT=RCVD
1038 003406 001410          BEQ    FT2C         ;IF SO:BR
1039 003410 012737 015213 000650  MOV    #MSG26,ERADD ;SET ERROR CODE
1040 003416 012737 003370 000674  MOV    #FT2B,SCOLP  ;SET SCOPE ADDRESS
1041 003424 004737 003504          JSR    PC,FT2ER     ;GO DO ERROR
1042 003430 013700 000532          FT2C: MOV    DB,R0        ;GET ADDRESS OF DATA BUFFER
1043 003434 010102          MOV    R1,R2
1044 003436 010110          MOV    R1,(R0)      ;LOAD PATTERN
1045 003440 012703 004000          MOV    #4000,R3
1046 003444 005303          FT2D: DEC    R3        ;DELAY
1047 003446 001376          BNE    FT2D
1048 003450 020210          CMP    R2,(R0)      ;SEE IF EXPT=RCVD
1049 003452 001410          BEQ    FT2E         ;IF SO: BR
1050 003454 012737 015221 000650  MOV    #MSG27,ERADD ;SET ERROR CODE
1051 003462 012737 003430 000674  MOV    #FT2C,SCOLP  ;SET SCOPE ADDRESS
1052 003470 004737 003504          JSR    PC,FT2ER     ;GO DO ERROR
1053 003474 005701          FT2E: TST    R1        ;SEE IF DONE RESET
1054 003476 001453          BEQ    FT2X         ;IF SO: BR
1055 003500 005001          CLR    R1          ;SET ZERO PATTERN
1056 003502 000712          BR    FT2A         ;DO ZERO BITS
1057 003504 000240          FT2ER: NOP
1058 003506 032777 020000 175034  BIT    #20000,@SWR  ;SEE IF PRINT ERROR
1059 003514 001034          BNE    FT2ERB       ;IF NOT: BR
1060 003516 005737 000606          TST    HDRFL        ;SEE ID DONE HEADER
1061 003522 001004          BNE    FT2ERA       ;IF SO: BR
1062 003524 013704 000610          MOV    EMADDR,R4
1063 003530 004737 013442          JSR    PC,TTOUT     ;DO HEADER
1064 003534 012737 000001 000606  FT2ERA: MOV    #1,HDRFL   ;SET FLAG
1065 003542 013704 000650          MOV    ERADD,R4
1066 003546 004737 013442          JSR    PC,TTOUT     ;PRINT ERROR CODE
1067 003552 012704 015151          MOV    #MSG22,R4
1068 003556 004737 013442          JSR    PC,TTOUT     ;PRINT EXPT TAG
1069 003562 010103          MOV    R1,R3
1070 003564 004737 013560          JSR    PC,OCTPE     ;PRINT EXPT
1071 003570 012704 015161          MOV    #MSG23,R4
1072 003574 004737 013442          JSR    PC,TTOUT     ;PRINT RCVD TAG
1073 003600 011003          MOV    (R0),R3
1074 003602 004737 013560          JSR    PC,OCTPE     ;PRINT RCVD

```

1075	003606	005777	174736	FT2ERB:	TST	@SWR		;SEE IF HALT ON ERROR
1076	003612	100001			BPL	FT2ERC		;IF NOT: BR
1077	003614	000000			HALT			
1078	003616	004737	012512	FT2ERC:	JSR	PC,SCOPE		;GO SEE IF SCOPE ON ERROR
1079	003622	000240			NOP			
1080	003624	000207			RTS	PC		;IF NO SCOPE: CONTINUE TEST
1081	003626	000240		FT2X:	NOP			
1082	003630	004737	012546		JSR	PC,ITER		;GO SEE IF ITERATIONS
1083	003634	000137	003022		JMP	TSCD2		;RETURN TO SCHEDULAR

```

1084
1085
1086
1087 003640 012737 017053 000610 FT3:  MOV    #MSFT3,EMADDR  ;SET TEST HEADER
1088 003646 012737 003640 000674      MOV    #FT3,SCOLP
1089 003654 004737 012620      JSR    PC,INIT1      ;GO INIT
1090 003660 052777 020000 174632      BIS    #20000,@CS    ;FORCE UPE =1
1091 003666 000240      NOP
1092 003670 004737 012620      JSR    PC,INIT1      ;GO INIT
1093 003674 005777 174610      TST    @C1           ;SEE IF SC IS RESET
1094 003700 100005      BPL    FT3A          ;IF SO: BR
1095 003702 012737 015257 000650      MOV    #MSG29,ERADD  ;SET ERROR CODE
1096 003710 004737 003774      JSR    PC,FT3ER      ;GO DO ERROR
1097 003714 032777 040000 174566 FT3A:  BIT    #40000,@C1    ;SEE IF TRE IS RESET
1098 003722 001405      BEQ    FT3B          ;IF SO: BR
1099 003724 012737 015306 000650      MOV    #MSG30,ERADD  ;SET ERROR CODE.
1100 003732 004737 003774      JSR    PC,FT3ER      ;GO DO ERROR
1101 003736 017701 174556      FT3B:  MOV    @CS,R1       ;GET CS2
1102 003742 042701 000307      BIC    #307,R1      ;MARK IR/OR
1103 003746 005701      TST    R1           ;SEE IF RESET
1104 003750 001405      BEQ    FT3X          ;IF SO: BR
1105 003752 012737 015336 000650      MOV    #MSG31,ERADD  ;SET ERROR CODE
1106 003760 004737 003774      JSR    PC,FT3ER      ;GO DO ERROR
1107 003764 004737 012546      FT3X:  JSR    PC,ITER      ;GO SEE IF ITERATION
1108 003770 000137 003022      JMP    TSCD2         ;RETURN TO SCHEDULAR
1109
1110      ;ERROR REPORT SUBROUTINE
1111 003774 000240      FT3ER:  NOP
1112 003776 032777 020000 174544      BIT    #20000,@SWR   ;SEE IF PRINT ERROR
1113 004004 001015      BNE    2$           ;IF NOT: BR
1114 004006 005737 000606      TST    HDRFL        ;SEE IF DONE HEADER
1115 004012 001006      BNE    1$           ;IF SO: BR
1116 004014 013704 000610      MOV    EMADDR,R4
1117 004020 004737 013442      JSR    PC,TTOUT      ;PRINT HEADER
1118 004024 005237 000606      INC    HDRFL
1119 004030 013704 000650      1$:   MOV    ERADD,R4
1120 004034 004737 013442      JSR    PC,TTOUT      ;PRINT ERROR CODE
1121 004040 005777 174504      2$:   TST    @SWR        ;SEE IF HALT ON ERROR
1122 004044 100001      BPL    3$           ;IF NOT: BR
1123 004046 000000      HALT
1124 004050 000240      3$:   NOP
1125 004052 004737 012512      JSR    PC,SCOPE      ;GO SEE IF SCOPE
1126 004056 000207      RTS    PC            ;IF NOT: BR

```

```
1127
1128
1129 ;RH11 SILO TEST 1: EPMTY SILO READ*****
1130 004060 005737 000604 FT4: TST RH17F
1131 004064 001141 BNE FT5X ;IF RH70: BR
1132 004066 012737 017105 000610 MOV #MSFT4,EMADDR ;SET TEST TEST HEADER
1133 004074 012777 000040 174416 MOV #40,@CS ;INIT
1134 004102 017700 174424 MOV @DB,R0 ;READ DB
1135 004106 005777 174406 TST @CS ;SEE IF DLT IS SET
1136 004112 100013 BPL FT4ER ;IF NOT: BR
1137 004114 005777 174370 TST @C1 ;SEE IF SC IS SET
1138 004120 100014 BPL FT4ERA ;IF NOT: BR
1139 004122 032777 040000 174360 BIT #40000,@C1 ;SEE IF TRE IS SET
1140 004130 001414 BEQ FT4ERB ;IF NOT: BR
1141 004132 004737 012546 FT4X: JSR PC,ITER ;GO SEE IF ITERATION
1142 004136 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULAR
1143 004142 012737 015366 000650 FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE
1144 004150 000407 BR FT4ERC
1145 004152 012737 015404 000650 FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE
1146 004160 000403 BR FT4ERC
1147 004162 012737 015421 000650 FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.
1148 004170 000240 FT4ERC: NOP
1149 004172 012737 004060 000674 MOV #FT4,SCOLP ;SET SCOPE ADDRESS
1150 004200 004737 003774 JSR PC,FT3ER ;GO PRINT ERROR
1151 004204 000752 BR FT4X
```

```
1152
1153
1154 ;RH11 SILO TEST 2: IR/OR CHECK*****
1155 004206 005737 000604 FT5: TST RH17F ;SEE IF RH70
1156 004212 001066 BNE FT5X ;IF SO: BR
1157 004214 012737 017135 000610 MOV #MSFT5,EMADDR ;SET TEST HEADER
1158 004222 012737 004230 000674 MOV #FT5A,SCOLP ;SET SCOPE ADDRESS
1159 004230 004737 012620 FT5A: JSR PC,INIT1 ;GO INIT
1160 004234 032777 000100 174256 BIT #100,@CS ;SEE IF IR IS SET
1161 004242 001005 BNE FT5B ;IF SO: BR
1162 004244 012737 015437 000650 MOV #MSG35,ERADD ;SET ERROR CODE
1163 004252 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1164 004256 032777 000200 174234 FT5B: BIT #200,@CS ;SEE IF OR IS RESET
1165 004264 001405 BEQ FT5C ;IF SO: BR
1166 004266 012737 015464 000650 MOV #MSG36,ERADD ;SET ERROR CODE
1167 004274 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1168 004300 012777 000000 174224 FT5C: MOV #0,@DB ;LOAD ZERO INTO SILO
1169 004306 032777 000200 174204 BIT #200,@CS ;SEE THAT OR RESET
1170 004314 001405 BEQ FT5D ;IF IT DOES: BR
1171 004316 012737 015513 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1172 004324 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1173 004330 012777 177777 174174 FT5D: MOV #-1,@DB ;LOAD SILO WITH -1
1174 004336 012700 004000 MOV #4000,RO
1175 004342 032777 000200 174150 FT5E: BIT #200,@CS ;SEE IF OR IS SET
1176 004350 001007 BNE FT5X ;IF SO: BR
1177 004352 005300 DEC RO
1178 004354 001372 BNE FT5E ;AWAIT OR
1179 004356 012737 015513 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1180 004364 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1181 004370 004737 012546 FT5X: JSR PC,ITER ;GO SEE IF ITERATION
1182 004374 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

1183
1184
1185
1186 004400 005737 000604 FT6: TST RH17F
1187 004404 001052 BNE FT6X ;IF RH70: BR
1188 004406 012737 017165 000610 MOV #MSFT6,EMADDR ;SET TEST HEADER
1189 004414 012737 004422 000674 MOV #FT6A,SCOLP ;SET SCOPE ADDRESS
1190 004422 004737 012620 FT6A: JSR PC,INIT1 ;GO INIT
1191 004426 005000 CLR R0 ;PRESET DATA
1192 004430 010077 174076 FT6B: MOV R0,@DB ;LOAD SILO
1193 004434 005200 INC R0 ;BUMP DATA
1194 004436 022700 000102 CMP #102,R0 ;SEE IF FILLED ALL
1195 004442 001372 BNE FT6B ;IF NOT: BR
1196 004444 032777 000100 174046 BIT #100,@CS ;SEE IF IR IS RESET.
1197 004452 001405 BEQ FT6C ;IF SO: BR
1198 004454 012737 015624 000650 MOV #MSG40,ERADD ;SET ERROR CODE
1199 004462 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1200 004466 032777 000200 174024 FT6C: BIT #200,@CS ;SEE IF OR IS SET
1201 004474 001005 BNE FT6D ;IF SO: BR
1202 004476 012737 015552 000650 MOV #MSG38,ERADD ;SET ERROR CODE
1203 004504 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1204 004510 005000 FT6D: CLR R0 ;PRESET DATA
1205 004512 017701 174014 FT6E: MOV @DB,R1 ;READ SILO
1206 004516 020001 CMP R0,R1 ;SEE IF EXPT=RCVD
1207 004520 001010 BNE FT6DE ;IF NOT: BR
1208 004522 005200 INC R0 ;BUMP DATA
1209 004524 022700 000102 CMP #102,R0 ;SEE IF DONE ALL
1210 004530 001370 BNE FT6E ;IF NOT: BR
1211 004532 004737 012546 FT6X: JSR PC,ITER ;GO SEE IF ITERATION
1212 004536 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULAR
1213
1214 004542 000240 FT6DE: NOP
1215 004544 032777 020000 173776 BIT #20000,@SWR ;SEE IF PRINT ERROR
1216 004552 001032 BNE FT6DEB ;IF NOT: BR
1217 004554 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1218 004560 013701 000610 MOV EMADDR,R1
1219 004564 004737 013442 JSR PC,TTOUT ;PRINT HEADER
1220 004570 005237 000606 INC HDRFL ;SET FLAG
1221 004574 012704 015604 FT6DEA: MOV #MSG39,R4
1222 004600 004737 013442 JSR PC,TTOUT ;PRINT SILO READ ERROR
1223 004604 012704 015151 MOV #MSG22,R4
1224 004610 004737 013442 JSR PC,TTOUT ;PRINT EXPT TAG
1225 004614 010003 MOV R0,R3
1226 004616 004737 013572 JSR PC,OCTP ;PRINT EXPT
1227 004622 012704 015161 MOV #MSG23,R4
1228 004626 004737 013442 JSR PC,TTOUT ;PRINT RCVD TAG
1229 004632 010103 MOV R1,R3
1230 004634 004737 013572 JSR PC,OCTP ;PRINT RCVD
1231 004640 005777 173704 FT6DEB: TST @SWR ;SEE IF HALT ON ERROR
1232 004644 100001 BPL FT6DEX ;IF NOT: BR
1233 004646 000000 HALT
1234 004650 000207 FT6DEX: RTS PC ;RETURN TO TEST

```



```
1235
1236 ;RH11 SILO TEST 4: SILO OVERFLOW*****
1237
1238 004652 005737 000604 FT7: TST RH17F
1239 004656 001021 BNE FT7X ;IF RH70: BR
1240 004660 012737 017215 000610 MOV #MSFT7,EMADDR ;SET TEST HEADER
1241 004666 012737 004652 000674 MOV #FT7,SCOLP ;SET SCOPE ADDRESS
1242 004674 004737 012620 JSR PC,INIT1 ;GO INIT
1243 004700 012700 000103 MOV #103,R0 ;SET SIZE OF SILO +1
1244 004704 010077 173622 FT7A: MOV R0,@DB ;LOAD SILO
1245 004710 005300 DEC R0 ;SEE IF DONE
1246 004712 001374 BNE FT7A ;IF NOT: BR
1247 004714 005777 173600 TST @CS ;SEE IF DLT IS SET
1248 004720 100004 BPL FT7ER ;IF NOT: BR
1249 004722 004737 012546 FT7X: JSR PC,ITER ;GO SEE IF ITERATION
1250 004726 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULAR
1251 004732 012737 015366 000650 FT7ER: MOV #MSG32,ERADD ;SET ERROR CODE
1252 004740 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1253 004744 000766 BR FT7X
```

```
1254
1255 ;RH11 SILO TEST 5: SILO RESET*****
1256
1257 004746 005737 000604 FT10: TST RH17F
1258 004752 001034 BNE FT10X ;IF RH70: BR
1259 004754 012737 017245 000610 MOV #MSFT10,EMADDR ;SET TEST HEADER
1260 004762 012737 004746 000674 MOV #FT10,SCOLP ;SET SCOPE ADDRESS
1261 004770 012777 000040 173522 MOV #4,@CS ;INITIALIZE
1262 004776 012700 000004 MOV #4,R0 ;SET NUMBER OF SILO WRITER
1263 005002 010077 173524 FT10A: MOV R0,@DB ;WRITE SILO
1264 005006 005300 DEC R0 ;SEE IF DONE
1265 005010 001374 BNE FT10A ;IF NOT: BR
1266 005012 052777 000040 173500 BIS #4,@CS ;INITIALIZE
1267 005020 012777 177777 173504 MOV #-1,@DB ;WRITE SILO
1268 005026 017701 173500 MOV @DB,R1 ;READ SILO 1
1269 005032 017701 173474 MOV @DB,R1 ;READ SILO 2
1270 005036 005777 173456 TST @CS ;SEE IF DLT IS SET
1271 005042 100011 BPL FT10ER ;IF NOT: BR
1272 005044 004737 012546 FT10X: JSR PC,ITER ;GO SEE IF ITERATION
1273 005050 005737 000726 TST RHOF ;SEE IF RH11 ONLY
1274 005054 001402 BEQ FT10XX ;IF NOT: BR
1275 005056 000137 003110 JMP TEND ;ELSE GO TO END
1276 005062 000137 003022 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR
1277 005066 012737 015366 000650 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE
1278 005074 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1279 005100 000761 BR FT10X
```

```
1280 ;NOP TEST*****
1281
1282 005102 000240 FT11: NOP
1283 005104 012737 005102 000674 MOV #FT11,SCOLP ;SET SCOPE ADDRESS
1284 005112 004737 012620 JSR PC,INIT1
1285 005116 012737 000300 000716 MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
1286 005124 012737 177777 000620 MOV #-1,FCNT ;SET FC= ALL OVER
1287 005132 012737 177777 000622 MOV #-1,WCNT ;SET WC= ALL OVER
1288 005140 012737 177777 000616 MOV #-1,BADDR ;SET BA= ALL OVER
1289 005146 012737 000001 000636 MOV #1,RDYDX ;SET DELAY
1290 005154 012737 000001 000640 MOV #1,OPDYX ;SET OP DELAY
1291 005162 012737 000001 000710 MOV #1,FUN ;SET NOP FUNCTIONS CODE
1292 005170 004737 011546 JSR PC,EXEC ;GO EXECUTE COMMAND
1293 005174 000240 NOP
1294 005176 012737 017276 000610 MOV #MSFT11,EMADDR
1295 005204 004737 011776 JSR PC,ERCHK ;GO CHECK REGISTER
1296 005210 004737 012546 JSR PC,ITER ;GO SEE IF ITERATIONS
1297 005214 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

1298                                     ;REWIND TEST*****
1299
1300 005220 000240                      FT12: NOP
1301 005222 012737 005220 000674      MOV #FT12,SCOLP
1302 005230 004737 012620              JSR PC,INIT1 ;GO INITIALIZE
1303 005234 052777 001700 173300      BIS #1700,@TC ;SET TO NRZ,NORMAL
1304 005242 012737 177760 000620      MOV #-20,FCNT ;SET FC=20
1305 005250 012737 177770 000622      MOV #-10,WCNT ;SET WC=10
1306 005256 012737 020112 000616      MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1307 005264 012737 000007 000710      MOV #7,FUN ;SET REWIND OP CODE
1308 005272 004737 011546              JSR PC,EXEC ;GO EXECUTE COMMAND
1309 005276 000240                      NOP
1310 005300 032777 020000 173214      FT12A: BIT #20000,@DS
1311 005306 001374                      BNE FT12A ;AWAIT PIP
1312 005310 012737 017316 000610      MOV #MSFT12,EMADDR
1313 005316 004737 011776              JSR PC,ERCHK ;GO CHECK FOR ERROR
1314 005322 004737 012546              JSR PC,ITER ;GO SEE IF ITERATION
1315 005326 000137 003022              JMP TSCD2 ;RETURN TO SCHEDULAR
1316
  
```

```

1317                                     ;WRITE/READ TEST*****
1318
1319 005332 000240          FT13:  NOP
1320 005334 012737 000001 000636  MOV    #1,RDYDX
1321 005342 012737 000001 000640  MOV    #1,OPDYX
1322 005350 012737 000100 000624  MOV    #100,RCNT      ;SET RECORD COUNT
1323 005356 012737 017341 000610  MOV    #MSFT13,EMADDR ;SET TEST HEADER
1324 005364 012737 000001 000720  MOV    #1,PATRN
1325 005372 004737 012400          JSR    PC,DSUP        ;SET UP ALL ONES DATA PATTERN
1326 005376 012737 001700 000716  MOV    #1700,UDES     ;SET TO 800 BPI NORMAL
1327 005404 004737 011700          FT13A: JSR    PC,RWND       ;GO REWIND
1328 005410 012737 177600 000620  MOV    #-200,FCNT     ;SET FC
1329 005416 012737 177700 000622  MOV    #-100,WCNT     ;SET WC
1330 005424 012737 020112 000616  MOV    #WDATA,BADDR   ;SET BA
1331 005432 012737 000061 000710  MOV    #61,FUN        ;SET WRITE OP-CODE
1332 005440 012737 014765 000626  MOV    #MSG12,ERRP
1333 005446 004737 011546          FT13B: JSR    PC,EXEC     ;GO EXECUTE COMMAND
1334 005452 005037 000674          CLR    SCOLP         ;NO SCOPE LOOP
1335 005456 004737 011776          JSR    PC,ERCHK      ;GO CHECK ERROR
1336 005462 005337 000624          DEC    RCNT          ;SEE IF DONE ALL
1337 005466 001367          BNE    FT13B        ;IF NOT: BR
1338 005470 012737 000100 000624  MOV    #100,RCNT     ;SET RECORD COUNT
1339 005476 012737 021624 000616  MOV    #RDATA,BADDR
1340 005504 062737 000200 000616  ADD    #200,BADDR    ;SET BA
1341 005512 012737 000077 000710  MOV    #77,FUN       ;SET READ REVERSE OP-CPDE
1342 005520 012737 015003 000626  MOV    #MSG13,ERRP
1343 005526 004737 011546          FT13C: JSR    PC,EXEC     ;GO EXECUTE COMMAND
1344 005532 004737 011776          JSR    PC,ERCHK      ;GO CHECK ERROR
1345 005536 005337 000624          DEC    RCNT          ;SEE IF READ ALL
1346 005542 001371          BNE    FT13C        ;IF NOT:BR
1347 005544 162737 000200 000616  SUB    #200,BADDR    ;SET BA
1348 005552 012737 000071 000710  MOV    #71,FUN       ;SET READ FORWARD OP-CODE
1349 005560 012737 015030 000626  MOV    #MSG14,ERRP
1350 005566 012737 000100 000624  MOV    #100,RCNT     ;SET RECORD COUNT
1351 005574 004737 011546          FT13D: JSR    PC,EXEC     ;GO EXECUTE COMMAND
1352 005600 004737 011776          JSR    PC,ERCHK      ;GO CHECK ERRORS
1353 005604 005337 000624          DEC    RCNT          ;SEE IF DONE ALL
1354 005610 001371          BNE    FT13D        ;IF NOT:BR
1355 005612 032737 002000 000716  BIT    #2000,UDES     ;SEE IF DONE PE
1356 005620 001007          BNE    FT13X        ;IF SO: BR
1357 005622 012737 002300 000716  MOV    #2300,UDES    ;SET PE MODE
1358 005630 012737 000100 000624  MOV    #100,RCNT     ;RESET RECORD COUNT
1359 005636 000662          BR     FT13A        ;GO DO NEXT DENSITY
1360 005640 000137 003022          FT13X: JMP    TSCD2   ;RETURN TO SCHEDULAR

```

```

1361                                     ;SPACE TEST*****
1362
1363 005644 000240          FT14:  NOP
1364 005646 012737 017370 000610  MOV      #MSFT14,EMADDR ;SET TEST HEADER
1365 005654 012737 001700 000716  MOV      #1700,UDES   ;SET NRZ,NORMAL
1366 005662 004737 011700          FT14A1: JSR      PC,RWND    ;GO INITIALIZE
1367 005666 012737 000100 000624  MOV      #100,RCNT   ;SET NUMBER OF RECORDER
1368 005674 012737 177777 020112  MOV      #-1,WDATA   ;SET DATA PATTERN
1369 005702 012737 177700 000620  MOV      #-100,FCNT  ;PRESET FRAME CNT
1370 005710 012737 177740 000622  MOV      #-40,WCNT   ;PRESET WORD CNT
1371 005716 004737 012620          FT14A:  JSR      PC,INIT1  ;GO REWIND
1372 005722 012737 001000 000640  MOV      #1000,OPDYX
1373 005730 012737 040000 000636  MOV      #40000,RDYDX
1374 005736 012737 000061 000710  MOV      #61,FUN     ;SET WRITE OP-CODE
1375 005744 012737 102300 000660  MOV      #102300,STMSK ;MASK DATA RELATED ERRORS
1376 005752 052777 000010 172540  BIS      #10,ACS     ;INHIBIT BUS ADDRESS INCREMENT
1377 005760 004737 011546          JSR      PC,EXEC     ;GO EXECUTE COMMAND
1378 005764 012737 016043 000626  MOV      #MSG46,ERRP ;SET ERROR CODE
1379 005772 004737 011776          JSR      PC,ERCHK    ;GO CHECK ERRORS
1380 005776 005737 000712          TST      SERFL      ;SEE IF ERROR
1381 006002 001402          BEQ      FT14A2     ;IF NOT: BR
1382 006004 000137 006470          JMP      FT14X      ;ELSE EXIT
1383 006010 005337 000620          FT14A2: DEC      FCNT  ;BUMP FC
1384 006014 032737 000001 000620  BIT      #1,FCNT    ;SEE IF SHOULD BUMP WC
1385 006022 001403          BEQ      FT14A3     ;IF NOT: BR
1386 006024 162737 000001 000622  SUB      #1,WCNT    ;BUMP WC
1387 006032 005337 000624          FT14A3: DEC      RCNT  ;SEE IF DONE ALL
1388 006036 001327          BNE      FT14A     ;WRITE ALL RECORDS
1389 006040 012737 000100 000632  MOV      #100,RRD   ;PRESET RECORD POSITION
1390 006046 012737 000176 000634  MOV      #176,RFD
1391 006054 012737 177701 000642  MOV      #-77,SCNT  ;SET SPACE AMOUNT
1392 006062 012737 000033 000710  FT14B:  MOV      #33,FUN  ;SET OP-CODE SPACE REVERSE
1393 006070 004737 011546          JSR      PC,EXEC     ;GO EXECUTE COMMAND
1394 006074 012737 016114 000626  MOV      #MSG48,ERRP ;SET ERROR CODE
1395 006102 004737 011776          JSR      PC,ERCHK    ;GO CHECK ERRORS
1396 006106 005737 000712          TST      SERFL      ;SEE IF ERROR
1397 006112 001166          BNE      FT14X      ;IF SO: BR
1398 006114 004737 006210          JSR      PC,FT14RR  ;GO READ REVERSE + CHECK DATA
1399 006120 000240          NOP
1400 006122 012737 000031 000710  MOV      #31,FUN    ;SET SPACE FORWARD OP-CODE
1401 006130 005237 000642          INC      SCNT       ;SET SPACE AMOUNT
1402 006134 001555          BEQ      FT14X      ;IF DONE: BR
1403 006136 004737 011546          JSR      PC,EXEC     ;GO EXECUTE COMMAND
1404 006142 012737 016067 000626  MOV      #MSG47,ERRP ;SET ERROR CODE
1405 006150 004737 011776          JSR      PC,ERCHK    ;GO CHECK ERROR
1406 006154 005737 000712          TST      SERFL      ;SEE IF ERROR FLAG
1407 006160 001143          BNE      FT14X      ;IF NO: BR
1408 006162 004737 006252          JSR      PC,FT14RF  ;GO READ FORWARD FOR POSITION CHECK
1409 006166 000240          NOP
1410 006170 005237 000642          INC      SCNT       ;DECREMENT SPACE AMOUNT
1411 006174 001535          BEQ      FT14X      ;IF DONE: BR
1412 006176 005237 000632          INC      RRD        ;BUMP DATA EXPT
1413 006202 005337 000634          DEC      RFD        ;BUMP DATA EXPT
1414 006206 000725          BR       FT14B
1415 006210 000240          FT14RR: NOP
1416 006212 012737 021624 000616  MOV      #RDATA,BADDR ;SET BA

```

```

1417 006220 012737 000077 000710      MOV      #77,FUN      ;SET READ REVERSE OP-CODE
1418 006226 004737 011546      JSR      PC,EXEC     ;GO EXECUTE COMMAND
1419 006232 000240                NOP
1420 006234 013705 000632      MOV      RRD,R5
1421 006240 020577 172252      CMP      R5,@FC      ;SEE IF CORRECT RECORD
1422 006244 001020                BNE      FT14RER     ;IF NOT: BR
1423 006246 000137 006300      JMP      FT14EC      ;GO CLEAR RH11 ERROR BIT
1424 006252 000240                FT14RF: NOP
1425 006254 012737 000071 000710      MOV      #71,FUN     ;SET READ FORWARD OP-CODE
1426 006262 004737 011546      JSR      PC,EXEC     ;GO EXECUTE COMMAND
1427 006266 013705 000634      MOV      RFD,R5
1428 006272 020577 172220      CMP      R5,@FC      ;SEE IF CORRECT RECORD
1429 006276 001003                BNE      FT14RER     ;IF NOT: BR
1430 006300 004737 012620      FT14EC: JSR      PC,INIT1 ;CLEAR RH
1431 006304 000207                RTS      PC          ;RETURN
1432 006306 000240                FT14RER: NOP
1433 006310 032777 020000 172232      BIT      #2000,@SWR  ;SEE IF PRINT INHIBITED
1434 006316 001060                BNE      FT14R3      ;IF SO: BR
1435 006320 012704 017370      MOV      #MSFT14,R4
1436 006324 004737 013442      JSR      PC,TTOUT    ;PRINT HEADER
1437 006330 012704 014703      MOV      #MSG9,R4
1438 006334 004737 013442      JSR      PC,TTOUT    ;PRINT ERROR TYPE
1439 006340 012704 015136      MOV      #MSG20,R4  ;SET NRZ TAG POINTER
1440 006344 032737 002000 000716      BIT      #2000,UDES  ;SEE IF PE
1441 006352 001402                BEQ      FT14R0      ;IF NOT: BR
1442 006354 012704 015144      MOV      #MSG21,R4  ;ELSE SET PE TAG POINTER
1443 006360 004737 013442      FT14R0: JSR      PC,TTOUT ;PRINT TAG
1444 006364 032737 000002 000710      BIT      #2,FUN      ;SEE IF READ REVERSE
1445 006372 001003                BNE      FT14R1      ;IF SO: BR
1446 006374 012704 015116      MOV      #MSG17,R4
1447 006400 000402                BR       FT14R2      ;GO PRINT
1448 006402 012704 015076      FT14R1: MOV      #MSG16,R4
1449 006406 004737 013442      FT14R2: JSR      PC,TTOUT ;PRINT FRWD/REV
1450 006412 012704 015151      MOV      #MSG22,R4
1451 006416 004737 013442      JSR      PC,TTOUT    ;PRINT EXPT TAG
1452 006422 010503                MOV      R5,R3
1453 006424 042703 177700      BIC      #177700,R3  ;MASK RECORD NUMBER
1454 006430 004737 013572      JSR      PC,OCTP     ;PRINT EXPT RECORD NUMBER
1455 006434 012704 015161      MOV      #MSG23,R4
1456 006440 004737 013442      JSR      PC,TTOUT    ;PRINT RCVD TAG
1457 006444 017703 172046      MOV      @FC,R3
1458 006450 042703 177700      BIC      #177700,R3  ;MASK RECORD NUMBER
1459 006454 004737 013572      JSR      PC,OCTP     ;PRINT ACTUAL RECORD NUMBER
1460 006460 005777 172064      FT14R3: TST      @SWR  ;SEE IF HALT ON ERROR
1461 006464 100001                BPL      FT14X      ;IF NOT: BR
1462 006466 000000                HALT
1463 006470 032737 002000 000716      FT14X:  BIT      #2000,UDES ;SEE IF DONE PE
1464 006476 001005                BNE      FT14XX      ;IF SO: BR
1465 006500 012737 002300 000716      MOV      #2300,UDES ;SET TO PE
1466 006506 000137 005662      JMP      FT14A1      ;DO IN PE
1467 006512 000137 003022      FT14XX: JMP      TSCD2    ;RETURN TO SCHEDULAR

```

```
1468                                     ;ERASE TEST*****
1469
1470 006516 000240                      FT15:  NOP
1471 006520 005037 000660                CLR      STMSK
1472 006524 012737 000100 000636        MOV      #100,RDYDX
1473 006532 012737 000010 000640        MOV      #10,OPDYX
1474 006540 012737 017412 000610        MOV      #MSFT15,EMADDR ;SET TEST HEADER
1475 006546 004737 011700                JSR      PC,RWND ;REWIND
1476 006552 012737 021624 000616        MOV      #RDATA,BADDR ;SET BA
1477 006560 012737 001700 000716        MOV      #1700,UDES ;SET NRZ, NORMAL
1478 006566 012737 000025 000710 FT15A: MOV      #25,FUN ;SET ERASE OP-CODE
1479 006574 012737 000400 000624        MOV      #400,RCNT ;SET TO ERASE 256 TIMES
1480 006602 004737 011546                FT15B: JSR      PC,EXEC ;GO EXECUTE COMMAND
1481 006606 012737 016043 000626        MOV      #MSG46,ERRP ;SET ERROR CODE
1482 006614 004737 011776                JSR      PC,ERCHK ;GO CHECK ERRORS
1483 006620 005737 000712                TST      SERFL ;SEE IF ANY ERRORS
1484 006624 001032                      BNE      FT15X ;IF SO EXIT
1485 006626 005337 000624                DEC      RCNT ;SEE IF DONE ERASING
1486 006632 001363                      BNE      FT15B ;IF NOT: BR
1487 006634 000240                      NOP
1488 006636 004737 011700                JSR      PC,RWND ;REWIND
1489 006642 012737 177600 000622        MOV      #-200,WCNT ;SET WC
1490 006650 012737 000071 000710        MOV      #71,FUN ;SET READ FORWARD OP-CODE
1491 006656 012737 000040 000636        MOV      #40,RDYDX ;SET DELAY
1492 006664 004737 011546                JSR      PC,EXEC ;GO EXECUTE COMMAND
1493 006670 000240                      NOP
1494 006672 012737 016503 000626        MOV      #MSG60,ERRP ;SET ERROR CODE
1495 006700 012737 020000 000660        MOV      #20000,STMSK
1496 006706 004737 011776                JSR      PC,ERCHK ;GO CHECK ERRORS
1497 006712 000137 003022                FT15X: JMP      TSCD2 ;RETURN TO SCHEDULAR
```



```

1498                                     ;TAPE MARK WRITE/READ TEST*****
1499
1500 006716 000240                      FT16:  NOP
1501 006720 012737 000001 000636      MOV    #1,RDYDX
1502 006726 012737 001000 000640      MOV    #1000,OPDYX
1503 006734 012737 017434 000610      MOV    #MSFT16,EMADDR ;SET HEADER
1504 006742 012737 001700 000716      MOV    #1700,UDES    ;SET TO NRZ,NORMAL,ODD
1505 006750 004737 011700              FT16A: JSR    PC,RWND  ;INIT AND REWIND SLAVE
1506 006754 012737 177760 000620      FT16B: MOV    #-20,FCNT ;FC=20
1507 006762 012737 177770 000622      MOV    #-10,WCNT    ;WC=10
1508 006770 012737 000027 000710      MOV    #27,FUN      ;SET WRITE TAPE MARK OP-CODE
1509 006776 004737 011546              JSR    PC,EXEC      ;GO EXECUTE COMMAND
1510 007002 012737 001000 000660      MOV    #1000,STMSK  ;SET FOR FCE MASK
1511 007010 012737 014765 000626      MOV    #MSG12,ERRP  ;SET ERROR CODE
1512 007016 004737 011776              JSR    PC,ERCHK     ;GO CHECK ERROR
1513 007022 004737 012340              JSR    PC,TMCHK     ;GO SEE IF TM SET
1514 007026 012737 000077 000710      MOV    #77,FUN      ;SET USED REVERSE OP-CODE
1515 007034 004737 011546              JSR    PC,EXEC      ;GO EXECUTE COMMAND
1516 007040 012737 001000 000660      MOV    #1000,STMSK  ;SET FCE ERROR MASK
1517 007046 012737 015003 000626      MOV    #MSG13,ERRP  ;SET ERROR CODE
1518 007054 004737 011776              JSR    PC,ERCHK     ;GO CHECK ERRORS
1519 007060 004737 012340              JSR    PC,TMCHK     ;GO SEE IF TM SET
1520 007064 012737 000071 000710      MOV    #71,FUN      ;SET READ FORWARD OP-CODE
1521 007072 004737 011546              JSR    PC,EXEC      ;GO EXECUTE COMMAND
1522 007076 012737 015030 000626      MOV    #MSG14,ERRP  ;SET ERROR CODE
1523 007104 004737 011776              JSR    PC,ERCHK     ;TO CHECK ERRORS
1524 007110 004737 012340              JSR    PC,TMCHK     ;GO SEE IF TM SET
1525 007114 032737 002000 000716      BIT    #2000,UDES   ;SEE IF DONE PE
1526 007122 001004                      BNE    FT16X        ;IF SO: BR
1527 007124 012737 002300 000716      MOV    #2300,UDES   ;SET PE, NORMAL
1528 007132 000706                      BR     FT16A        ;DO IN PE
1529 007134 004737 012546              FT16X: JSR    PC,ITER ;DO ITERATIONS
1530 007140 000137 003022              JMP    TSCD2        ;RETURN TO SCHEDULAR
1531

```

```

1532
1533                                     ;TAPE MARK SPACE TEST*****
1534
1535 007144 005037 000624 FT17: CLR RCNT
1536 007150 012737 017475 000610 MOV #MSFT17,EMADDR ;SET HEADER
1537 007156 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ
1538 007164 004737 011700 FT17A: JSR PC,RWND ;REWIND TAPE
1539 007170 012737 000027 000710 FT17B: MOV #27,FUN
1540 007176 012737 040000 000636 MOV #40000,RDYDX ;SET DRY DELAY
1541 007204 012737 040000 000640 MOV #40000,OPDYX ;SET OP DELAY
1542 007212 004737 011546 JSR PC,EXEC ;GO WRITE TM
1543 007216 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1544 007224 012737 015055 000626 MOV #MSG15,ERRP ;SET ERROR TYPE
1545 007232 004737 011776 JSR PC,ERCHK ;GO CHECK ERROR
1546 007236 005737 000712 TST SERFL ;SEE IF ERROR
1547 007242 001137 BNE FT17X ;IF SO: BR
1548 007244 004737 012340 JSR PC,TMCHK ;GO SEE IF TM SET
1549 007250 000240 NOP
1550 007252 000240 NOP
1551 007254 032737 000100 000624 BIT #100,RCNT ;SEE IF DONE PATTERN
1552 007262 001045 BNE FT17D ;IF SO: BR
1553 007264 062737 000020 000624 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1554 007272 013737 000624 000652 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1555 007300 012737 177600 000622 MOV #-200,WCNT ;WC=128
1556 007306 012737 177400 000620 MOV #-400,FCNT ;FC=256
1557 007314 012737 020112 000616 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1558 007322 012737 000061 000710 MOV #61,FUN ;SET WRITE OP CODE
1559 007330 000240 FT17C: NOP
1560 007332 000240 NOP
1561 007334 004737 011546 JSR PC,EXEC ;GO WRITE
1562 007340 012737 014765 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1563 007346 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1564 007354 004737 011776 JSR PC,ERCHK ;GO CHECK ERROR
1565 007360 005737 000712 TST SERFL ;SEE IF ERROR
1566 007364 001066 BNE FT17X ;IF SO: BR
1567 007366 005337 000652 DEC TEMP1 ;SEE IF DONE ALL
1568 007372 001356 BNE FT17C ;IF NOT: BR
1569 007374 000675 BR FT17B ;ELSE GO DO TM
1570 007376 000240 FT17D: NOP
1571 007400 012737 000033 000710 MOV #33,FUN ;SET SPACE REVERSE
1572 007406 012737 015076 000626 MOV #MSG16,ERRP ;SET ERROR CODE
1573 007414 012737 177600 000642 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1574 007422 012737 000005 000624 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1575 007430 004737 012620 FT17E: JSR PC,INIT1 ;GO INIT
1576 007434 004737 011546 JSR PC,EXEC ;GO SPACE
1577 007440 012737 001000 000660 MOV #1000,STMSK ;SET ERROR MASK
1578 007446 004737 011776 JSR PC,ERCHK ;GO CHECK ERROR
1579 007452 005737 000712 TST SERFL ;SEE IF ERROR
1580 007456 001031 BNE FT17X ;IF SO: BR
1581 007460 004737 012340 JSR PC,TMCHK ;GO SEE IF TM SET
1582 007464 005337 000624 DEC RCNT ;SEE IF DONE SPACES
1583 007470 001357 BNE FT17E ;IF NOT: BR
1584 007472 022737 000031 000710 CMP #31,FUN ;SEE IF DONE FORWARD
1585 007500 001407 BEQ FT17F ;IF SO: BR
1586 007502 012737 015116 000626 MOV #MSG17,ERRP ;SET ERROR CODE
1587 007510 012737 000031 000710 MOV #31,FUN ;SET TO SPACE FORWARD

```

1588	007516	000736			BR	FT17D1		:DO FORWARD
1589	007520	032737	002000	000716	FT17F:	BIT	#2000,UDES	:SEE IF DONE PE
1590	007526	001005			BNE	FT17X		:IF SO: BR
1591	007530	012737	002300	000716	MOV	#2300,UDES		:SET TO PE
1592	007536	000137	007164		JMP	FT17A		:GO PE
1593	007542	000137	003022		FT17X:	JMP	TSCD2	:RETURN TO SCHEDULAR

```

1594
1595
1596
1597 007546 000240          FT20:  NOP
1598 007550 012737 017523 000610  MOV      #MSFT20,EMADDR ;SET HEADER
1599 007556 012737 001700 000716  MOV      #1700,UDES   ;SET UNIT DESCRIPTION
1600 007564 004737 011700          FT20A: JSR      PC,RWND     ;INIT AND REWIND SLAVE
1601 007570 012737 000003 000720  MOV      #3,PATRN
1602 007576 004737 012400          JSR      PC,DSUP     ;GO SET PATTERN 3
1603 007602 012737 020112 000616  MOV      #WDATA,BADDR ;SET BA
1604 007610 012737 177400 000620  MOV      #-400,FCNT   ;SET FC
1605 007616 012737 177600 000622  MOV      #-200,WCNT   ;SET WC
1606 007624 012737 000061 000710  MOV      #61,FUN     ;SET WRITE OP CODE
1607 007632 004737 011546          JSR      PC,EXEC     ;GO WRITE RECORD
1608 007636 012737 016043 000626  MOV      #MSG46,ERRP  ;SET ERROR CODE
1609 007644 004737 011776          JSR      PC,ERCHK    ;GO CHECK ERROR
1610 007650 005737 000712          TST      SERFL       ;SEE IF ERORR
1611 007654 001042          BNE      FT20X       ;IF SO: BR
1612 007656 012737 015076 000626  MOV      #MSG16,ERRP  ;SET REVERSE ERROR TAG
1613 007664 012737 000057 000710  MOV      #57,FUN     ;SET REVERSE WRITE CHECK OP-CODE
1614 007672 062737 000376 000616  ADD      #376,BADDR  ;SET BA FOR REVERSE CHECK
1615 007700 004737 011546          JSR      PC,EXEC     ;GO DO REVERSE CHECK
1616 007704 004737 011776          JSR      PC,ERCHK    ;GO CHECK ERROR
1617 007710 012737 015116 000626  FT20B: MOV      #MSG17,ERRP ;SET FORWARD TAG
1618 007716 012737 000051 000710  MOV      #51,FUN     ;SET FORWARD CHECK OP CODE
1619 007724 162737 000376 000616  SUB      #376,BADDR  ;SET BA FOR FORWARD CHECK
1620 007732 004737 011546          JSR      PC,EXEC     ;GO DO FORWARD CHECK
1621 007736 004737 011776          JSR      PC,ERCHK    ;GO CHECK ERROR
1622 007742 032737 002000 000716  FT20C: BIT      #2000,UDES ;SEE IF DONE PE
1623 007750 001004          BNE      FT20X       ;IF SO: BR
1624 007752 012737 002300 000716  MOV      #2300,UDES  ;ELSE SET PE
1625 007760 000701          BR       FT20A       ;DO IN PE
1626 007762 004737 012546          FT20X: JSR      PC,ITER  ;DO ITERATIONS
1627 007766 000137 003022          JMP      TSCD2      ;RETURN TO SCHEDULAR

```

```

1628
1629
1630
1631 007772 012737 017554 000610 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1632 010000 004737 011700 FT21A: JSR PC,RWND ;GO REWIND
1633 010004 012737 000003 000720 MOV #3,PATRN
1634 010012 004737 012400 JSR PC,DSUP ;GO SET PATTERN 3
1635 010016 012737 020112 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1636 010024 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1637 010032 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1638 010040 012737 001700 000716 MOV #1700,UDES ;SET NRZ, NORMAL
1639 010046 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1640 010054 004737 011546 JSR PC,EXEC ;GO DO WRITE 1
1641 010060 012737 014765 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1642 010066 004737 011776 JSR PC,ERCHK ;GO CHECK FOR ERROR
1643 010072 004737 011546 JSR PC,EXEC ;YES DO WRITE 2
1644 010076 004737 011776 JSR PC,ERCHK ;YES CHECK FOR ERROR
1645 010102 000240 NOP
1646 010104 004737 011700 JSR PC,RWND ;GO REWIND
1647 010110 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400(10)
1648 010116 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200(10)
1649 010124 004737 011546 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1650 010130 000240 FT21SCP:NOP
1651 010132 004737 011700 JSR PC,RWND ;REWIND
1652 010136 012737 021624 000616 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1653 010144 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400
1654 010152 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200
1655 010160 012737 000071 000710 MOV #71,FUN ;SET READ OP-CODE
1656 010166 004737 011546 JSR PC,EXEC ;GO READ RECORD 1
1657 010172 012737 015030 000626 MOV #MSG14,ERRP ;SET ERROR CODE
1658 010200 004737 011776 JSR PC,ERCHK ;GO CHECK FOR ERROR
1659 010204 000240 NOP
1660 010206 052777 000010 170304 BIS #10,@CS ;INHIBIT BA INCREMENT
1661 010214 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1662 010222 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1663 010230 004737 011546 JSR PC,EXEC ;GO READ RECORD 2
1664 010234 022777 001440 170254 CMP #800.,@FC ;SEE IF READ RECORD 2 OK
1665 010242 001424 BEQ FT21X ;IF SO: BR
1666 010244 022777 001441 170244 CMP #801.,@FC ;BRANCH IF IN GREY AREA
1667 010252 001420 BEQ FT21X
1668 010254 022777 001440 170234 1$: CMP #800.,@FC ;BRANCH IF ERASE HEAD REVERSED
1669 010262 101404 BLOS FT21B ;IF SO: BR
1670 010264 012737 015736 000650 MOV #MSG44,ERADD ;SET ERASE HEAD INOPERATIVE ERROR CODE
1671 010272 000403 BR FT21C
1672 010274 012737 015766 000650 FT21B: MOV #MSG45,ERADD ;SET ERASE HEAD REVERSED ERROR CODE
1673 010302 012737 010130 000674 FT21C: MOV #FT21SCP,SCOLP ;SET SCOPE ADDRESS
1674 010310 004737 003774 JSR PC,FT3ER ;GO PRINT ERROR
1675 010314 004737 012546 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1676 010320 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULAR
1677
1678

```

```

1679                                     ;BUFFERED COMMAND TEST*****
1680
1681 010324 012737 017603 000610 FT22:  MOV    #MSFT22,EMADDR ;SET TEST HEADER
1682 010332 004737 011700             JSR    PC,RWND      ;GO REWIND
1683 010336 012700 000003             MOV    #3,R0        ;SET NUMBER OF WRITES
1684 010342 012737 001700 000716     MOV    #1700,UDES   ;SET TO NRZ NORMAL
1685 010350 012737 020112 000616     MOV    #WDATA,BADDR ;SET BA=WRITE BUFFER
1686 010356 012737 177000 000620     MOV    #-1000,FCNT  ;SET FC=1000
1687 010364 012737 177400 000622     MOV    #-400,WCNT   ;SET WC=400
1688 010372 012737 000061 000710     MOV    #61,FUN      ;SET WRITE OP-CODE
1689 010400 004737 011546             FT22A: JSR    PC,EXEC   ;GO DO WRITE
1690 010404 005300                   DEC    R0            ;SEE IF DONE ALL
1691 010406 001374                   BNE    FT22A        ;IF NOT: BR
1692 010410 000240                   NOP
1693 010412 012777 000007 170070     MOV    #7,@C1       ;START REWIND
1694 010420 032777 000200 170074     FT22B: BIT    #200,@DS
1695 010426 001774                   BEQ    FT22B
1696 010430 004737 012620             JSR    PC,INIT1     ;INITIALIZE
1697 010434 012737 000010 000636     MOV    #10,RDYDX    ;SET LONG READY DELAY
1698 010442 004737 011546             JSR    PC,EXEC      ;ISSUE BUFFERED WRITE
1699 010446 000240                   NOP
1700 010450 012737 016141 000626     MOV    #MSG49,ERRP  ;SET ERROR CODE
1701 010456 012737 102300 000660     MOV    #102300,STMSK ;MARK DATA ERROR
1702 010464 004737 011776             JSR    PC,ERCHK     ;GO CHECK ERROR
1703 010470 032777 000002 170024     BIT    #2,@DS       ;SEE IF BOT IS SET
1704 010476 001410                   BEQ    FT22X        ;IF NOT: BR
1705 010500 012737 016167 000650     MOV    #MSG50,ERADD ;SET ERROR CODE
1706 010506 012737 010324 000674     MOV    #FT22,SCOLP
1707 010514 004737 003774             JSR    PC,FT3ER     ;GO DO ERROR
1708 010520 004737 012546             FT22X: JSR    PC,ITER  ;GO SEE IF ITERATION
1709 010524 000137 003022             JMP    TSCD2        ;RETURN TO SCHEDULAR
1710
1711

```

```

1712                                     ;READ-IN PRESET TEST*****
1713
1714 010530 005737 000614          FT23: TST      SLVN          ;SEE IF SLAVE SELECT=0
1715 010534 001103                   BNE      FT23X        ;IF NOT:BR
1716 010536 012737 017640 000610  MOV      #MSFT23,EMADDR ;SET TEST HEADER
1717 010544 004737 012620                   JSR      PC,INIT1     ;GO INIT
1718 010550 012737 001700 000716  MOV      #1700,UDES    ;SET TO NRZ NORMAL
1719 010556 012737 020112 000616  MOV      #WDATA,BADDR  ;SET BA=WRITE BUFFER
1720 010564 012737 177400 000620  MOV      #-400,FCNT    ;SET FC=400
1721 010572 012737 177600 000622  MOV      #-200,WCNT    ;SET WC=200
1722 010600 012737 000061 000710  MOV      #61,FUN      ;SET WRITE OP-CODE
1723 010606 004737 011546                   JSR      PC,EXEC      ;GO DO WRITE
1724 010612 000240                   NOP
1725 010614 004737 012620                   JSR      PC,INIT1     ;INITIALIZE
1726 010620 012737 000021 000710  MOV      #21,FUN      ;SET READ-IN PRESET OP CODE
1727 010626 004737 011546                   JSR      PC,EXEC      ;GO DO COMMAND
1728 010632 005000                   CLR      R0
1729 010634 012703 000004                   MOV      #4,R3        ;SET MULT
1730 010640 032777 020000 167654  FT23A: BIT      #20000,@DS  ;SEE IF PIP RESET
1731 010646 001404                   BEQ      FT23B        ;IF SO: BR
1732 010650 005300                   DEC      R0
1733 010652 001372                   BNE      FT23A        ;AWAIT PIP RESET
1734 010654 005303                   DEC      R3
1735 010656 001370                   BNE      FT23A        ;DELAY
1736 010660 032777 000002 167634  FT23B: BIT      #2,@DS  ;SEE IF BOT
1737 010666 001010                   BNE      FT23C        ;IF SO: BR
1738 010670 012737 016225 000650  MOV      #MSG51,ERADD  ;SET ERROR CODE
1739 010676 012737 010530 000674  MOV      #FT23,SCOLP
1740 010704 004737 003774                   JSR      PC,FT3ER     ;GO DO ERROR
1741 010710 012701 141000          FT23C: MOV      #141000,R1 ;SET EXPT TC
1742 010714 013700 000542                   MOV      TC,R0        ;SET TC ADDRESS
1743 010720 020110                   CMP      R1,(R0)      ;SEE IF EXPT=RCVD
1744 010722 001410                   BEQ      FT23X        ;IF SO: BR
1745 010724 012737 016261 000650  MOV      #MSG52,ERADD  ;SET ERROR CODE
1746 010732 012737 010530 000674  MOV      #FT23,SCOLP  ;CLEAR SCOPE ADDRESS
1747 010740 004737 003504                   JSR      PC,FT2ER     ;GO DO ERROR
1748 010744 000137 003022          FT23X: JMP      TSCD2   ;RETURN TO SCHEDULAR
1749
1750

```

```
1751
1752
1753
1754 010750 012737 017727 000610 FT24: MOV #MSFT24,EMADDR ;SET ERROR MSG HEADER
1755 010756 004737 011700 JSR PC,RWND ;REWIND SLAVE
1756 010762 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN
1757 010770 004737 012400 JSR PC,DSUP ;GO DO DATA SETUP
1758 010774 012737 020112 000616 MOV #WDATA,BADDR ;SET BUS ADDRESS,
1759 011002 012737 177400 000620 MOV #-400,FCNT ;FRAME COUNT,
1760 011010 012737 177600 000622 MOV #-200,WCNT ;WORD COUNT,
1761 011016 012737 001700 000716 MOV #1700,UDES ;& SLAVE DESC = NRZ NORMAL
1762 011024 012737 000061 000710 MOV #61,FUN ;LOAD OP CODE WRITE FWD
1763 011032 004737 011546 JSR PC,EXEC ;GO EXECUTE COMMAND
1764 011036 012737 016043 000626 MOV #MSG46,ERRP ;SET ERROR MSG ADDRESS
1765 011044 004737 011776 JSR PC,ERCHK ;GO CHECK ERRORS
1766 011050 005737 000712 TST SERFL ;BRANCH IF AN ERROR OCCURRED
1767 011054 001026 BNE FT24X
1768 011056 004737 011700 JSR PC,RWND ;REWIND SLAVE
1769 011062 012737 021624 000616 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1770 011070 012737 002300 000716 MOV #2300,UDES ;SET SLAVE DESC = PE,NORMAL
1771 011076 012737 000071 000710 MOV #71,FUN ;SET OP CODE = READ FWD
1772 011104 004737 011546 JSR PC,EXEC ;GO READ RECORD
1773 011110 032777 000040 167404 BIT #40,ADS ;BRANCH ID PES BIT CLEARED
1774 011116 001405 BEQ FT24X
1775 011120 012737 016602 000650 MOV #MSG63,ERADD
1776 011126 004737 003774 JSR PC,FT3ER ;GO PROCESS ERROR
1777 011132 004737 012546 FT24X: JSR PC,ITER
1778 011136 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULER
1779
```



```

1780
1781 ;AUTO-DENSITY SELECT TEST: WRITE-PE,READ-NRZ
1782 011142 012737 020005 000610 FT25: MOV #MSFT25,EMADDR ;SET ERROR MESSAGE ADDRESS
1783 011150 004737 011700 JSR PC,RWND ;REWIND SLAVE
1784 011154 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN
1785 011162 004737 012400 JSR PC,DSUP ;GO DO DATA SETUP
1786 011166 012737 020112 000616 MOV #WDATA,BADDR ;SET BUS ADDRESS
1787 011174 012737 177400 000620 MOV #-400,FCNT ;FRAME COUNT,
1788 011202 012737 177600 000622 MOV #-200,WCNT ;WORD COUNT,
1789 011210 012737 002300 000716 MOV #2300,UDES ;& SLAVE DESC = PE,NORMAL
1790 011216 012737 000061 000710 MOV #61,FUN ;LOAD WRITE OP CODE
1791 011224 004737 011546 JSR PC,EXEC ;GO EXECUTE WRITE
1792 011230 012737 016043 000626 MOV #MSG46,ERRP ;SET ERROR MSG HDR
1793 011236 004737 011776 JSR PC,ERCHK ;GO CHECK FOR ERRORS
1794 011242 005737 000712 TST SERFL ;BRANCH IF ERROR OCURRED
1795 011246 001026 BNE FT25X
1796 011250 004737 011700 JSR PC,RWND ;REWIND SLAVE
1797 011254 012737 021624 000616 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1798 011262 012737 001700 000716 MOV #1700,UDES ;SET SLAVE DESC = NRZ,NORMAL
1799 011270 012737 000071 000710 MOV #71,FUN ;SET READ FWD OP CODE
1800 011276 004737 011546 JSR PC,EXEC ;GO EXECUTE
1801 011302 032777 000040 167212 BIT #40,ADS ;BRANCH ID PES BIT GOT SET
1802 011310 001005 BNE FT25X
1803 011312 012737 016633 000650 MOV #MSG64,ERADD
1804 011320 004737 003774 JSR PC,FT3ER ;GO PROCESS ERROR
1805 011324 004737 012546 FT25X: JSR PC,ITER ;ITERATION LOOP
1806 011330 000137 003022 JMP TSCD2 ;RETURN TO SCHEDULER
1807
1808 ;REWIND: OFF LINE TEST*****
1809
1810 011334 032777 010000 167206 FT26: BIT #10000,ASWR ;SEE IF IN CONTINUOUS MODE
1811 011342 001077 BNE FT26XX ;IF SO: BR
1812 011344 005737 001662 TST CHNFLG ;BRANCH IF CHAIN MODE
1813 011350 001074 BNE FT26XX
1814 011352 012737 017673 000610 MOV #MSFT26,EMADDR ;SET TEST HEADER
1815 011360 004737 011700 JSR PC,RWND ;REWIND & SELECT SLAVE
1816 011364 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN (ALL 1'S)
1817 011372 004737 012400 JSR PC,DSUP ;FILL WRITE BUFFER
1818 011376 012737 020112 000616 MOV #WDATA,BADDR ;SET WRITE BUFFER BUS ADDRESS
1819 011404 012737 177400 000620 MOV #-400,FCNT ;SET FRAME COUNT
1820 011412 012737 177600 000622 MOV #-200,WCNT ;SET WORD COUNT
1821 011420 012737 001700 000716 MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
1822 011426 012737 000061 000710 MOV #61,FUN ;SET WRITE COMMAND
1823 011434 004737 011546 JSR PC,EXEC ;GO WRITE A RECORD
1824 011440 012777 000003 167042 MOV #3,AC1 ;ISSUE REWIND: OFF LINE COMMAND
1825 011446 005037 000674 CLR SCOLP ;CLEAR SCOPE LOOP
1826 011452 012700 004000 MOV #4000,R0
1827 011456 005300 1$: DEC R0 ;DELAY
1828 011460 001376 BNE 1$
1829 011462 032777 010000 167032 BIT #10000,ADS ;SEE IF MOL IS RESET
1830 011470 001406 BEQ 2$ ;IF SO: BR
1831 011472 012737 016300 000650 MOV #MSG53,ERADD ;SET ERROR CODE
1832 011500 004737 003774 JSR PC,FT3ER ;GO DO ERROR
1833 011504 000412 BR FT26X
1834 011506 013700 000524 2$: MOV ER,R0 ;GET ADDRESS OF ERROR REG
1835 011512 005001 CLR R1 ;RESULT SHOULD BE 0

```

1836	011514	020110			CMP	R1,(R0)	;BRANCH IF ERROR REG = 0
1837	011516	001405			BEQ	FT26X	
1838	011520	012737	016667	000650	MOV	#MSG67,ERADD	;SET ERROR MSG HEADER
1839	011526	004737	003504		JSR	PC,FT2ER	;GO TYPE ERROR
1840	011532	012704	016325		FT26X: MOV	#MSG54,R4	
1841	011536	004737	013442		JSR	PC,TTOUT	;PRINT ON LINE REQUEST
1842	011542	000137	003022		FT26XX: JMP	TSCD2	;RETURN TO SCHEDULER
1843							

```
1844                                     ;COMMAND EXECUTE SUBROUTINE*****
1845
1846 011546 000240                               EXEC:  NOP
1847 011550 053777 000716 166764                BIS    UDES,@TC           ;LOAD TAPE CONT
1848 011556 013777 000622 166726                MOV    WCNT,@WC          ;LOAD WC
1849 011564 013777 000620 166724                MOV    FCNT,@FC          ;LOAD FC
1850 011572 013777 000616 166714                MOV    BADDR,@BA        ;LOAD BA
1851 011600 022737 000031 000710                CMP    #31,FUN           ;SEE IF SPACE FORWARD
1852 011606 001404                                BEQ    EXECA              ;IF SO: BR
1853 011610 022737 000033 000710                CMP    #33,FUN           ;SEE IF SPACE REVERSE
1854 011616 001003                                BNE    EXECB              ;IF NOT: BR
1855 011620 013777 000642 166670                EXECA: MOV    SCNT,@FC    ;SET SPACE COUNT
1856 011626 000240                                EXECB: NOP
1857 011630 013777 000710 166652                MOV    FUN,@C1           ;LOAD OP-CODE + GO
1858 011636 000240                                NOP
1859 011640 013703 000636                        MOV    RDYDX,R3          ;SET DELAY
1860 011644 005004                                CLR    R4
1861 011646 032777 000200 166646                EXECC: BIT    #200,@DS    ;SEE IF DRY
1862 011654 001004                                BNE    EXECX              ;IF SO: BR
1863 011656 005304                                DEC    R4
1864 011660 001372                                BNE    EXECC
1865 011662 005303                                DEC    R3                 ;DELAY FOR DRY
1866 011664 001370                                BNE    EXECC
1867 011666 013703 000640                                EXECX: MOV    OPDYX,R3
1868 011672 005303                                EXECXA: DEC    R3         ;DELAY
1869 011674 001376                                BNE    EXECXA
1870 011676 000207                                EXECXX: RTS    PC        ;RETURN TO CALLER
1871
```

```
1872                                     ;REWIND SUBROUTINE*****
1873
1874 011700 000240          RWND:  NOP
1875 011702 004737 012620   JSR    PC,INIT1          ;INIT
1876 011706 012777 000007 166574  MOV    #7,@C1          ;START REWIND
1877 011714 012700 040000   MOV    #40000,R0
1878 011720 005300          RWNDA: DEC    R0
1879 011722 001376          BNE    RWNDA          ;DELAY
1880 011724 032777 020000 166570  RWNDB: BIT    #20000,@DS
1881 011732 001374          BNE    RWNDB          ;AWAIT PIP
1882 011734 032777 000002 166560  BIT    #2,@DS        ;SEE IF BOT
1883 011742 001012          BNE    RWNDX          ;IF SO: BR
1884 011744 013704 000610   MOV    EMADDR,R4
1885 011750 004737 013442   JSR    PC,TTOUT        ;PRINT HEADER
1886 011754 012704 014452   MOV    #MSG2,R4
1887 011760 004737 013442   JSR    PC,TTOUT        ;PRINT REWIND ERROR
1888 011764 000137 003022   JMP    TSCD2           ;RETURN TO SECDULAR
1889 011770 004737 012620   RWNDX: JSR    PC,INIT1 ;INIT
1890 011774 000207          RTS    PC              ;RETURN TO CALLER
1891
```

```

1892                                     ;ERROR CHECK SUBROUTINE*****
1893
1894 011776 005037 000712          ERCHK: CLR      SERFL      ;CLEAR FLAG
1895 012002 017737 166514 000664  MOV      @DS,DSAV    ;SAVE DRIVE STATUS REGISTER
1896 012010 032777 040000 166504  BIT      #40000,@DS  ;SEE IF ERROR
1897 012016 001001                    BNE      ERPT        ;IF SO: BR
1898 012020 000207                    RTS      PC          ;RETURN
1899 012022 017704 166476          ERPT:  MOV      @ER,R4    ;GET ERROR REGISTER
1900 012026 032737 002000 000716  BIT      #2000,UDES  ;SEE IF PE
1901 012034 001403                    BEQ      ERPTA1      ;IF SO: BR
1902 012036 042737 000200 000660  BIC      #200,STMSK  ;RESET PEF MASK
1903 012044 043704 000660          ERPTA1: BIC      STMSK,R4 ;MASK DONT CARE BITS
1904 012050 001530                    BEQ      ERPTX       ;IF NO UNEXPECTED ERRORS: BR
1905 012052 012737 000001 000712  ERPTG: MOV      #1,SERFL ;SET FLAG
1906 012060 032777 020000 166462  BIT      #20000,@SWR ;SEE IF SHOULD PRINT ERRORS
1907 012066 001115                    BNE      ERPTD       ;IF NOT: BR
1908 012070 005737 000606          TST      HDRFL      ;SEE IF DONE HEADER
1909 012074 001006                    BNE      ERPTA       ;IF SO: BR
1910 012076 005237 000606          INC      HDRFL      ;SET HEADER FLAG
1911 012102 013704 000610          MOV      EMADDR,R4
1912 012106 004737 013442          JSR      PC,TTOUT    ;PRINT HEADER
1913 012112 013704 000626          ERPTA: MOV      ERRP,R4 ;GET ERROR CODE
1914 012116 001414                    BEQ      ERPTB       ;IF NONE: BR
1915 012120 004737 013442          JSR      PC,TTOUT    ;PRINT ERROR CODE
1916 012124 012704 015136          MOV      #MSG20,R4  ;SET NRZ TAG
1917 012130 032777 002000 166404  BIT      #2000,@TC  ;SEE IF PE
1918 012136 001402                    BEQ      ERPT1A      ;IF NOT: BR
1919 012140 012704 015144          MOV      #MSG21,R4  ;ELSE SET PE TAG
1920 012144 004737 013442          ERPT1A: JSR      PC,TTOUT ;PRINT TAG
1921 012150 013704 000630          ERPTB: MOV      ERRP1,R4 ;SEE IF CODE 2
1922 012154 001402                    BEQ      ERPTB1      ;IF NOT: BR
1923 012156 004737 013442          JSR      PC,TTOUT    ;PRINT CODE 2
1924 012162 032777 004000 166360  ERPTB1: BIT      #4000,@SWR ;SEE IF ITERATION
1925 012170 001010                    BNE      ERPTC       ;IF NOT: BR
1926 012172 012704 016457          MOV      #MSG56,R4
1927 012176 004737 013442          JSR      PC,TTOUT    ;PRINT ITER TAG
1928 012202 013703 000662          MOV      ITCNT,R3
1929 012206 004737 013572          JSR      PC,OCTP     ;PRINT ITERATION
1930 012212 012704 014364          ERPTC: MOV      #MSG1,R4
1931 012216 004737 013442          JSR      PC,TTOUT    ;PRINT REGISTER TAG
1932 012222 017703 166262          MOV      @C1,R3
1933 012226 004737 013560          JSR      PC,OCTPE    ;PRINT CS1
1934 012232 017703 166254          MOV      @WC,R3
1935 012236 004737 013560          JSR      PC,OCTPE    ;PRINT WC
1936 012242 017703 166246          MOV      @BA,R3
1937 012246 004737 013560          JSR      PC,OCTPE    ;PRINT BA
1938 012252 017703 166240          MOV      @FC,R3
1939 012256 004737 013560          JSR      PC,OCTPE    ;PRINT FC
1940 012262 017703 166232          MOV      @CS,R3
1941 012266 004737 013560          JSR      PC,OCTPE    ;PRINT CS2
1942 012272 017703 166224          MOV      @DS,R3
1943 012276 004737 013560          JSR      PC,OCTPE    ;PRINT DS
1944 012302 017703 166216          MOV      @ER,R3
1945 012306 004737 013560          JSR      PC,OCTPE    ;PRINT ER
1946 012312 017703 166224          MOV      @TC,R3
1947 012316 004737 013560          JSR      PC,OCTPE    ;PRINT TC

```

1948	012322	005777	166222	ERPTD:	TST	@SWR	:SEE IF HALT ON ERROR
1949	012326	100001			BPL	ERPTX	:IF NOT: BR
1950	012330	000000			HALT		
1951	012332	004737	012620	ERPTX:	JSR	PC,INIT1	:INIT
1952	012336	000207		ERPTXX:	RTS	PC	:RETURN
1953							
1954							

```

1955                                     ;TAPE MARK STATUS CHECK*****
1956
1957 012340 032737 000004 000664 TMCHK: BIT #4,DSAV ;SEE IF TM SET
1958 012346 001401 BEQ TMCHK1 ;IF NOT: BR
1959 012350 000207 TMCHK0: RTS PC ;ELSE RETURN
1960 012352 005737 000712 TMCHK1: TST SERFL ;SEE IF HAD ERROR
1961 012356 001374 BNE TMCHK0 ;IF SO: BR
1962 012360 012737 016467 000630 MOV #MSG57,ERRP1 ;SET ERROR CODE 2
1963 012366 004737 012052 JSR PC,ERPTG ;GO PRINT TM ERROR
1964 012372 005037 000630 CLR ERRP1 ;CLEAR CODE 2 FLAG
1965 012376 000207 RTS PC ;RETURN
1966
1967                                     ;DATA SETUP ROUTINE*****
1968
1969 012400 000240 DSUP: NOP
1970 012402 012703 020112 DSO: MOV #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
1971 012406 013701 000720 MOV PATRN,R1 ;R1 = PATTERN SELECTOR
1972 012412 006301 ASL R1 ;MAKE PATTERN SELECTOR EVEN
1973 012414 004771 000740 JSR PC,@DATBL(R1) ;GO GENERATE PATTERN
1974 012420 012702 000640 MOV #640,R2 ;R2=BUFFER SIZE +2
1975 012424 012701 021624 MOV #RDATA,R1 ;R1=READ DATA START
1976 012430 005021 1$: CLR (R1)+ ;CLEAR BUFFER
1977 012432 005302 DEC R2 ;SEE IF DONE ALL
1978 012434 001375 BNE 1$ ;IF NOT: BR
1979 012436 000207 RTS PC ;EXIT
1980
1981                                     ;ALL ONES*****
1982
1983 012440 012701 177777 DAT1: MOV #-1,R1 ;R1=DATA
1984 012444 012702 000640 DAT1A: MOV #640,R2 ;R2=WORD COUNT +2
1985 012450 010123 1$: MOV R1,(R3)+ ;LOAD BUFFER
1986 012452 005302 DEC R2 ;SEE IF DONE
1987 012454 001375 BNE 1$ ;IF NOT: BR
1988 012456 000207 RTS PC
1989
1990                                     ;ALL ZEROS*****
1991
1992 012460 005001 DAT2: CLR R1 ;R1=DATA
1993 012462 000770 BR DAT1A ;LOAD BUFFER
1994
1995                                     ;ONE/ZERO IN ALTERNATING CHARACTERS*****
1996
1997 012464 012701 125125 DAT3: MOV #125125,R1 ;R1=DATA
1998 012470 000765 BR DAT1A ;LOAD BUFFER
1999
2000                                     ;ALL BITS 0-377*****
2001
2002 012472 005001 DAT4: CLR R1 ;R1=STARTING DATA
2003 012474 012702 001500 MOV #1500,R2 ;R2=CHARACTER COUNT
2004 012500 110123 1$: MOV R1,(R3)+ ;LOAD BUFFER
2005 012502 105201 INCB R1 ;BUMP DATA
2006 012504 005302 DEC R2 ;SEE IF DONE
2007 012506 001374 BNE 1$ ;IF NOT: BR
2008 012510 000207 RTS PC
2009

```

```

2010
2011
2012
2013
2014
2015 012512 000240
2016 012514 032777 040000 166026
2017 012522 001001
2018 012524 000207
2019 012526 000240
2020 012530 005737 000674
2021 012534 001001
2022 012536 000207
2023 012540 022626
2024 012542 000177 166126
2025
2026
2027
2028 012546 000240
2029 012550 032777 004000 165772
2030 012556 001403
2031 012560 005037 000662
2032 012564 000207
2033 012566 005737 000730
2034 012572 001772
2035 012574 005237 000662
2036 012600 023737 000662 000566
2037 012606 001764
2038 012610 005726
2039 012612 017700 166060
2040 012616 000110
2041
2042
2043
2044 012620 000240
2045 012622 012777 000040 165670
2046 012630 013777 000612 165662
2047 012636 013777 000614 165676
2048 012644 000207
2049

;SCOPE LOOP ON ERROR SUBROUTINE*****
SCOPE: NOP
BIT #40000,@SWR ;SEE IF LOOP ON ERROR
BNE 1$ ;IF SO: BR
RTS PC ;ELSE EXIT
1$: NOP
TST SCOLP ;SEE IF SCOPE ADDRESS
BNE 2$ ;IF NOT: BR
RTS PC ;ELSE EXIT
2$: CMP (SP)+,(SP)+ ;RESET STACK
JMP @SCOLP ;LOOP ON ERROR

;TEST ITERATION SUBROUTINE*****
ITER: NOP
BIT #4000,@SWR ;SEE IF ITERATIONS
BEQ 2$ ;IF SO: BR
1$: CLR ITCNT ;CLEAR ITERATION COUNTER
RTS PC ;ELSE EXIT
2$: TST PCNTR ;DO SINGLE SUBTEST ITERATION
BEQ 1$ ;ON FIRST PASS
INC ITCNT ;BUMP COUNTER
CMP ITCNT,ITAMT ;SEE IF DONE ALL
BEQ 1$ ;IF SO: BR
TST (SP)+ ;RESET STACK
MOV @ITRLP,R0 ;SET ITERATION POINTER
JMP (R0) ;GO ITERATE

;INITIALIZE SUBROUTINE*****
INIT1: NOP
MOV #40,@CS ;INIT
INIT2: MOV DRVN,@CS ;SELECT DRIVE
MOV SLVN,@TC ;SELECT SLAVE
RTS PC ;RETURN

```



```

2050                                     ;MAG TAPE INTERRUPT HANDLER*****
2051
2052 012646 000240          MTINT:  NOP
2053 012650 013716 000646      MOV     RTRN,(SP)      ;RETURN TO (RTRN)
2054 012654 000002          RTI          ;RETURN
2055
2056                                     ;TTY INTERRUPT HANDLER*****
2057
2058 012656 017746 165672      TTINT:  MOV     @TKB,-(SP)    ;GET CHARACTER
2059 012662 042716 000200      BIC     #200,(SP)     ;CLEAR PARITY BIT
2060 012666 122716 000003      CMPB   #3,(SP)       ;BRANCH IF NOT CONTROL C
2061 012672 001010          BNE     1$
2062 012674 005737 001662      TST    CHNFLG        ;INHIBIT C IF CHAIN MODE
2063 012700 001005          BNE     1$
2064 012702 005077 165640      CLR    @PSW
2065 012706 000005          RESET
2066 012710 000137 000200      JMP    @#200         ;RESTART PROGRAM
2067 012714 122716 000001      1$:    CMPB   #1,(SP)     ;BRANCH IF NOT A
2068 012720 001017          BNE     2$
2069 012722 022737 000176 000550  CMP    #SWREG,SWR    ;BRANCH IF HARDWARE SWR IS INVOKED
2070 012730 001016          BNE     3$
2071 012732 012737 177570 000550  MOV    #177570,SWR   ;INVOKE HARDWARE SWR
2072 012740 004737 014320      JSR    PC,.SAVE      ;SAVE REGISTERS ON THE STACK
2073 012744 012704 016743      MOV    #MSG70,R4    ;TYPE 'HARDWARE SWR IN USE'
2074 012750 004737 013442      JSR    PC,TTOUT
2075 012754 004737 014342      JSR    PC,.RESTORE
2076 012760 122716 000007      2$:    CMPB   #7,(SP)     ;BRANCH IF NOT G
2077 012764 001005          BNE     4$
2078 012766 012737 000176 000550  3$:    MOV    #SWREG,SWR   ;INVOKE SOFTWARE SWR
2079 012774 004737 014222      JSR    PC,GTSWR     ;GET SOFTWARE SWITCHES
2080 013000 005726          4$:    TST    (SP)+       ;POP CHARACTER OFF THE STACK
2081 013002 000002          RTI
2082
2083                                     ;BUS ADDRESS TRAP HANDLER*****
2084
2085 013004 000240          TRAP:  NOP
2086 013006 032777 020000 165534  BIT    #20000,@SWR   ;SEE IF SHOULD PRINT ERRORS
2087 013014 001020          BNE     TRAP2        ;IF NOT: BR
2088 013016 005737 000606      TST    HDRFL        ;SEE IF DONE HEADER
2089 013022 001006          BNE     TRAP1        ;IF SO: BR
2090 013024 005237 000606      INC    HDRFL        ;ELSE SET HEADER FLAG
2091 013030 013704 000610      MOV    EMADDR,R4
2092 013034 004737 013442      JSR    PC,TTOUT     ;PRINT HEADER
2093 013040 012704 015171      TRAP1: MOV    #MSG24,R4
2094 013044 004737 013442      JSR    PC,TTOUT     ;PRINT ERROR
2095 013050 010103          MOV    R1,R3        ;GET ADDRESS THAT CAUSED THE TRAP
2096 013052 004737 013572      JSR    PC,OCTP      ;PRINT ADDRESS OF TRAP
2097 013056 005777 165466      TRAP2: TST    @SWR   ;SEE IF HALT ON ERROR
2098 013062 100001          BPL    TRAPX        ;IF NOT: BR
2099 013064 000000          HALT
2100 013066 022626          TRAPX: CMP    (SP)+,(SP)+ ;RESET STACK
2101 013070 012737 003250 000674  MOV    #FT1A,SCOLP  ;SET SCOPE ADDRESS
2102 013076 004737 012512      JSR    PC,SCOPE     ;GO SEE IF SCOPE LOOP
2103 013102 005737 000722      TST    RHTF        ;SEE IF INITIAL ADDRESS TEST
2104 013106 001402          BEQ    TRAPXX       ;IF NOT: BR
2105 013110 000137 001766      JMP    STOB         ;ELSE REDO ADDRESS REQUEST

```

2106 013114 000137 003254 TRAPXX: JMP FT1B ;RETURN TO TEST 1
2107

```
2108 ;*****
2109 ;TTY ENTRY SUBROUTINE:
2110 ;
2111 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
2112 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
2113 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
2114 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
2115 ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
2116 ;THE CALLING ROUTINE.
2117 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
2118 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
2119 ;MAY BE REENTERED.
2120 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
2121 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
2122 ;CARRIAGE RETURN
2123 ;*****
2124
2125 013120 010146 TTR: MOV R1,-(SP) ;SAVE CHAR COUNT ON STACK
2126 013122 011601 10$: MOV (SP),R1 ;RESTORE CHAR COUNT (FOR U)
2127 013124 005037 000652 CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
2128 013130 005000 CLR R0
2129 013132 004737 013400 1$: JSR PC,TTIN ;GO READ CHARACTER
2130 013136 122737 000003 000602 CMPB #3,TIB ;BRANCH IF NOT C
2131 013144 001003 BNE 11$
2132 013146 000005 RESET ;RESET
2133 013150 000137 000200 JMP @#200 ;RESTART
2134 013154 122737 000015 000602 11$: CMPB #15,TIB ;SEE IF CR
2135 013162 001004 BNE 2$ ;IF NOT: BR
2136 013164 005737 000652 TST TEMP1 ;SEE IF FIRST CHARACTER
2137 013170 001471 BEQ 9$ ;IF SO: BR
2138 013172 000457 BR 6$ ;ELSE GO LOAD VALUE
2139 013174 122737 000025 000602 2$: CMPB #25,TIB ;BRANCH IF NOT CONTROL U
2140 013202 001005 BNE 21$
2141 013204 012704 016663 MOV #MSG65,R4 ;TYPE <CR><LF>
2142 013210 004737 013442 JSR PC,TTOUT
2143 013214 000742 BR 10$ ;RESTART
2144 013216 122737 000177 000602 21$: CMPB #177,TIB ;BRANCH IF NOT 'RUBOUT'
2145 013224 001012 BNE 3$
2146 013226 000241 CLC ;REMOVE LAST CHARACTER
2147 013230 006000 ROR R0
2148 013232 006200 ASR R0
2149 013234 006200 ASR R0
2150 013236 012704 016665 MOV #MSG66,R4 ;TYPE ' '
2151 013242 004737 013442 JSR PC,TTOUT
2152 013246 005201 INC R1 ;DECREMENT CHAR RECEIVED COUNT
2153 013250 000730 BR 1$ ;GET NEXT CHARACTER
2154 013252 122737 000060 000602 3$: CMPB #60,TIB ;SEE IF CHAR IS LESS THAN 0
2155 013260 101402 BLOS 4$ ;IF NOT: BR
2156 013262 000137 013360 JMP T1NER ;ELSE GO TO ERROR
2157 013266 122737 000070 000602 4$: CMPB #70,TIB ;SEE IF CHAR IS GREATER THAN 7
2158 013274 101002 BHI 5$ ;IF NOT: BR
2159 013276 000137 013360 JMP T1NER ;ELSE GO TO ERROR
2160 013302 005237 000652 5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
2161 013306 006300 ASL R0
2162 013310 006300 ASL R0 ;SHIFT 3 LEFT
2163 013312 006300 ASL R0
```

```

2164 013314 042737 177770 000602      BIC      #177770,TIB      ;STRIP ASCII
2165 013322 053700 000602      BIS      TIB,R0          ;LOAD CHARACTER
2166 013326 005301                DEC      R1              ;SEE IF DONE
2167 013330 001300                BNE      1$              ;IF NOT: BR
2168 013332 020002      6$:    CMP      RO,R2        ;SEE IF EXCEEDED MAXIMUM LIMIT
2169 013334 101402                BLOS    7$              ;IF NOT: BR
2170 013336 000137 013360      JMP      TINER           ;ELSE GO TO ERROR
2171 013342 020300      7$:    CMP      R3,R0        ;SEE IF BELOW MINIMUM LIMIT
2172 013344 101402                BLOS    8$              ;IF NOT: BR
2173 013346 000137 013360      JMP      TINER           ;ELSE GO TO ERROR
2174 013352 010015      8$:    MOV      RO,(R5)    ;LOAD VALUE
2175 013354 005726      9$:    TST      (SP)+      ;POP CHAR COUNT OFF STACK
2176 013356 000207                RTS      PC              ;EXIT
2177
2178                                ;TTY ENTRY ERROR SUBROUTINE*****
2179
2180 013360 012704 014677      TINER:  MOV      #MSG7,R4
2181 013364 004737 013442      JSR      PC,TTOUT        ;PRINT?
2182 013370 005726                TST      (SP)+          ;POP CHAR COUNT OFF STACK
2183 013372 162716 000020      SUB      #20,(SP)       ;RESET SP TO START OF VALUE ROUTINE
2184 013376 000207                RTS      PC              ;REDO VALUE ENTRY
2185
2186                                ;TTY READ SUBROUTINE*****
2187
2188 013400 005277 165146      TTIN:   INC      @TKS
2189 013404 105777 165142      1$:    TSTB     @TKS
2190 013410 100375                BPL     1$
2191 013412 117737 165136 000602      MOVB    @TKB,TIB
2192 013420 042737 000200 000602      BIC     #200,TIB        ;STRIP PARITY BIT
2193 013426 013737 000602 000600      MOV     TIB,TOB        ;MOVE CHAR TO OUTPUT BFR
2194 013434 004737 013542      JSR     PC,TOG         ;AND TYPE IT
2195 013440 000207                RTS     PC
2196
2197                                ;TTY OUTPUT SUBROUTINE*****
2198
2199 013442 112437 000600      TTOUT:  MOVB    (R4)+,TOB
2200 013446 122737 000043 000600      CMPB   #43,TOB
2201 013454 001440                BEQ     TEX
2202 013456 122737 000045 000600      CMPB   #45,TOB
2203 013464 001403                BEQ     1$
2204 013466 004737 013542      JSR     PC,TOG
2205 013472 000763                BR      TTOUT
2206 013474 112737 000015 000600  1$:    MOVB    #15,TOB
2207 013502 004737 013542      JSR     PC,TOG
2208 013506 012703 000004                MOV     #4,R3
2209 013512 005037 000600      2$:    CLR     TOB
2210 013516 004737 013542      JSR     PC,TOG
2211 013522 005303                DEC     R3
2212 013524 001372                BNE     2$              ;DO FILLERS
2213 013526 112737 000012 000600      MOVB    #12,TOB
2214 013534 004737 013542      JSR     PC,TOG
2215 013540 000740                BR      TTOUT
2216 013542 105777 165010      TOG:   TSTB     @TPS
2217 013546 100375                BPL     TOG
2218 013550 113777 000600 165002      MOVB    TOB,@TPB
2219 013556 000207                RTS     PC

```

2220
2221

```
2222 ;OCTAL OUTPUT SUBROUTINE*****
2223
2224 013560 012737 000001 014010 OCTPE: MOV #1,OFL
2225 013566 010304 MOV R3,R4
2226 013570 000410 BR OCTP0
2227 013572 005037 014010 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
2228 013576 010304 OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
2229 013600 001004 BNE OCTP0 ;IF NOT ZERO: BR
2230 013602 004737 013770 JSR PC,OCTPG1 ;ELSE PRINT ZERO
2231 013606 000137 013732 JMP OCTP3 ;SPACE AND EXIT
2232 013612 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
2233 013616 001406 BEQ OCTP1 ;IF NOT: BR
2234 013620 012704 000001 MOV #1,R4
2235 013624 004737 013746 JSR PC,OCTPG ;PRINT 1
2236 013630 000137 013642 JMP OCTP2
2237 013634 005004 OCTP1: CLR R4
2238 013636 004737 013746 JSR PC,OCTPG ;PRINT 0
2239 013642 010304 OCTP2: MOV R3,R4
2240 013644 006004 ROR R4
2241 013646 006004 ROR R4
2242 013650 006004 ROR R4 ;POSITION DIGIT
2243 013652 006004 ROR R4
2244 013654 000304 SWAB R4
2245 013656 004737 013746 JSR PC,OCTPG ;PRINT DIGIT 2
2246 013662 010304 MOV R3,R4
2247 013664 006004 ROR R4
2248 013666 000304 SWAB R4
2249 013670 004737 013746 JSR PC,OCTPG ;PRINT DIGIT 3
2250 013674 010304 MOV R3,R4
2251 013676 006104 ROL R4
2252 013700 006104 ROL R4
2253 013702 000304 SWAB R4
2254 013704 004737 013746 JSR PC,OCTPG ;PRINT DIGIT 4
2255 013710 010304 MOV R3,R4
2256 013712 006004 ROR R4
2257 013714 006004 ROR R4
2258 013716 006004 ROR R4
2259 013720 004737 013746 JSR PC,OCTPG
2260 013724 010304 MOV R3,R4
2261 013726 004737 013746 JSR PC,OCTPG ;PRINT DIGIT 5
2262 013732 012737 000240 000600 OCTP3: MOV #240,TOB
2263 013740 004737 013542 JSR PC,TOG ;PRINT SPACE
2264 013744 000207 RTS PC ;EXIT
2265 013746 042704 177770 OCTPG: BIC #177770,R4
2266 013752 001004 BNE OCTPG0
2267 013754 005737 014010 TST OFL
2268 013760 001001 BNE OCTPG0
2269 013762 000207 RTS PC
2270
2271 013764 005237 014010 OCTPG0: INC OFL
2272 013770 052704 000260 OCTPG1: BIS #260,R4
2273 013774 010437 000600 MOV R4,TOB
2274 014000 004737 013542 JSR PC,TOG
2275 014004 010304 MOV R3,R4
2276 014006 000207 RTS PC
2277 014010 000000 OFL: 0 ;FIRST CHAR FLAG
```

```
2278
2279 ;DATA CHARACTER OUTPUT SUBROUTINE*****
2280
2281 014012 005037 000600 DOUT: CLR TOB
2282 014016 012704 000010 MOV #10,R4 ;SET NUMBER TO PRINT
2283 014022 110337 000600 MOVB R3,TOB
2284 014026 105777 164524 1$: TSTB @TPS
2285 014032 100375 BPL 1$
2286 014034 132737 000200 000600 BITB #200,TOB
2287 014042 001404 BEQ 2$
2288 014044 012777 000061 164506 MOV #061,@TPB
2289 014052 000403 BR 3$
2290 014054 012777 000060 164476 2$: MOV #060,@TPB
2291 014062 006137 000600 3$: ROL TOB
2292 014066 005304 DEC R4
2293 014070 001356 BNE 1$
2294 014072 000207 RTS PC
2295
2296 014074 013703 000656 DOUTD: MOV TEMP3,R3
2297 014100 000303 SWAB R3
2298 014102 004737 014012 JSR PC,DOUT
2299 014106 013703 000656 MOV TEMP3,R3
2300 014112 004737 014012 JSR PC,DOUT
2301 014116 000207 RTS PC
2302
2303 ;TU45 SERIAL NUMBER PRINT SUBROUTINE*****
2304
2305 014120 010304 SNPT: MOV R3,R4
2306 014122 000304 SWAB R4
2307 014124 006004 ROR R4
2308 014126 006004 ROR R4
2309 014130 006004 ROR R4
2310 014132 006004 ROR R4 ;GET FIRST DIGIT
2311 014134 004737 014176 JSR PC,SNPG ;GO PRINT
2312 014140 010304 MOV R3,R4
2313 014142 000304 SWAB R4 ;GET SECOND DIGIT
2314 014144 004737 014176 JSR PC,SNPG ;GO PRINT
2315 014150 010304 MOV R3,R4
2316 014152 006004 ROR R4
2317 014154 006004 ROR R4
2318 014156 006004 ROR R4
2319 014160 006004 ROR R4 ;GET THIRD DIGIT
2320 014162 004737 014176 JSR PC,SNPG ;GO PRINT
2321 014166 010304 MOV R3,R4 ;GET FOURTH DIGIT
2322 014170 004737 014176 JSR PC,SNPG ;GO PRINT
2323 014174 000207 RTS PC ;EXIT
2324 014176 012737 000260 000600 SNPG: MOV #260,TOB ;SET BASE = 0
2325 014204 042704 177760 BIC #177760,R4 ;MASK DIGIT
2326 014210 050437 000600 BIS R4,TOB ;SET ASCII
2327 014214 004737 013542 JSR PC,TOG ;TYPE DIGIT
2328 014220 000207 RTS PC ;RETURN
2329
```

```

2330
2331      ;ROUTINE TO LOAD NEW VALUE INTO SWITCHES
2332 014222 022737 000176 000550 GTSWR:  CMP    #SWREG,SWR    ;BRANCH IF SOFTWARE SWR
2333 014230 001032                BNE    1$          ;NOT INVOKED
2334 014232 004737 014320                JSR    PC,.SAVE   ;SAVE REGISTERS ON THE STACK
2335 014236 012704 020067                MOV    #SMSWR,R4
2336 014242 004737 013442                JSR    PC,TTOUT
2337 014246 017703 164276                MOV    @SWR,R3
2338 014252 004737 013560                JSR    PC,OCTPE
2339 014256 012704 020076                MOV    #SMNEW,R4
2340 014262 004737 013442                JSR    PC,TTOUT
2341 014266 013705 000550                MOV    SWR,R5    ;TTR ROUTINE RETURNS NEW VALUE TO (R5)
2342 014272 012701 000007                MOV    #7,R1    ;LIMIT RESPONSE TO 7 CHARS
2343 014276 012702 177777                MOV    #177777,R2 ;BETWEEN 0 AND 177777
2344 014302 012703 000000                MOV    #0,R3
2345 014306 004737 013120                JSR    PC,TTR
2346 014312 004737 014342                JSR    PC,.RESTORE ;RESTORE REGISTERS
2347 014316 000207      1$:          RTS    PC
2348
2349      ;;ROUTINE TO SAVE REGISTERS ON THE STACK
2350 014320 010546      .SAVE:  MOV    %5,-(SP)    ;;R5 IS SAVED AT 12(SP)
2351 014322 010446                MOV    %4,-(SP)    ;;R4 IS SAVED AT 10(SP)
2352 014324 010346                MOV    %3,-(SP)    ;;R3 IS SAVED AT 6(SP)
2353 014326 010246                MOV    %2,-(SP)    ;;R2 IS SAVED AT 4(SP)
2354 014330 010146                MOV    %1,-(SP)    ;;R1 IS SAVED AT 2(SP)
2355 014332 010046                MOV    %0,-(SP)    ;;R0 IS SAVED AT (SP)
2356 014334 016646 000014                MOV    14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
2357 014340 000207                RTS    PC          ;;RETURN TO CALLER
2358
2359      ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
2360 014342 012666 000014      .RESTORE:MOV    (SP)+,14(SP) ;;STORE RETURN PC ON STACK
2361 014346 012600                MOV    (SP)+,%0
2362 014350 012601                MOV    (SP)+,%1
2363 014352 012602                MOV    (SP)+,%2
2364 014354 012603                MOV    (SP)+,%3
2365 014356 012604                MOV    (SP)+,%4
2366 014360 012605                MOV    (SP)+,%5
2367 014362 000207                RTS    PC          ;;RETURN
2368
2369

```



```
2370 ;MESSAGE TABLE*****
2371
2372 014364 041445 030523 020040 MSG1: .ASCII /%CS1 WC BA FC CS2 /
2373 014372 020040 041527 020040
2374 014400 020040 041040 020101
2375 014406 020040 020040 041506
2376 014414 020040 020040 041440
2377 014422 031123 020040 020040
2378 014430 051504 020040 020040 .ASCII /DS ER TC%/
2379 014436 042440 020122 020040
2380 014444 020040 041524 021445
2381 014452 051045 053505 047111 MSG2: .ASCII /%REWIND ERROR#/
2382 014460 020104 051105 047522
2383 014466 021522
2384 014470 022445 046524 031460 MSG3: .ASCII /%TM03-TU45 BASIC FUNCTION TEST (CZTUQA0)%/
2385 014476 052055 032125 020065
2386 014504 040502 044523 020103
2387 014512 052506 041516 044524
2388 014520 047117 052040 051505
2389 014526 020124 041450 052132
2390 014534 050525 030101 022451
2391 014542 054524 042520 036040 .ASCII /TYPE <CR> TO TERMINATE RESPONSE & C TO RESTART%/
2392 014550 051103 020076 047524
2393 014556 052040 051105 044515
2394 014564 040516 042524 051040
2395 014572 051505 047520 051516
2396 014600 020105 020046 041536
2397 014606 052040 020117 042522
2398 014614 052123 051101 022524
2399 014622 043
2400 014623 045 042522 044507 MSG4: .ASCII /%REGISTER START = #/
2401 014630 052123 051105 051440
2402 014636 040524 052122 036440
2403 014644 021440
2404 014646 053045 041505 047524 MSG5: .ASCII /%VECTOR = #/
2405 014654 020122 020075 043
2406 014661 045 047105 020104 MSG6: .ASCII /%END OF PASS #/
2407 014666 043117 050040 051501
2408 014674 020123 043
2409 014677 040 020077 043 MSG7: .ASCII / ? #/
2410 014703 045 047520 044523 MSG9: .ASCII /%POSITION ERROR: #/
2411 014710 044524 047117 042440
2412 014716 051122 051117 020072
2413 014724 043
2414 014725 045 051104 053111 MSG10: .ASCII /%DRIVE NUMBER: #/
2415 014732 020105 052516 041115
2416 014740 051105 020072 043
2417 014745 045 046123 053101 MSG11: .ASCII /%SLAVE NUMBER: #/
2418 014752 020105 052516 041115
2419 014760 051105 020072 043
2420 014765 045 051127 052111 MSG12: .ASCII /%WRITE ERROR #/
2421 014772 020105 051105 047522
2422 015000 020122 043
2423 015003 045 042522 042101 MSG13: .ASCII /%READ REVERSE ERROR #/
2424 015010 051040 053105 051105
2425 015016 042523 042440 051122
```

2426	015024	051117	021440				
2427	015030	051045	040505	020104	MSG14:	.ASCII	/%READ FORWARD ERROR #/
2428	015036	047506	053522	051101			
2429	015044	020104	051105	047522			
2430	015052	020122	043				
2431	015055	045	051127	052111	MSG15:	.ASCII	/%WRITE TM ERROR #/
2432	015062	020105	046524	042440			
2433	015070	051122	051117	021440			
2434	015076	051045	053105	051105	MSG16:	.ASCII	/%REVERSE ERROR #/
2435	015104	042523	042440	051122			
2436	015112	051117	021440				
2437	015116	043045	051117	040527	MSG17:	.ASCII	/%FORWARD ERROR #/
2438	015124	042122	042440	051122			
2439	015132	051117	021440				
2440	015136	047040	055122	021440	MSG20:	.ASCII	/ NRZ #/
2441	015144	050040	020105	043	MSG21:	.ASCII	/ PE #/
2442	015151	040	054105	052120	MSG22:	.ASCII	/ EXPT: #/
2443	015156	020072	043				
2444	015161	040	041522	042126	MSG23:	.ASCII	/ RCVD: #/
2445	015166	020072	043				
2446	015171	045	052502	020123	MSG24:	.ASCII	/%BUS TRAP: #/
2447	015176	051124	050101	020072			
2448	015204	043					
2449	015205	045	041527	020072	MSG25:	.ASCII	/%WC: #/
2450	015212	043					
2451	015213	045	040502	020072	MSG26:	.ASCII	/%BA: #/
2452	015220	043					
2453	015221	045	041104	020072	MSG27:	.ASCII	/%DB: #/
2454	015226	043					
2455	015227	045	047111	052111	MSG28:	.ASCII	/%INIT DID NOT CLEAR RH #/
2456	015234	042040	042111	047040			
2457	015242	052117	041440	042514			
2458	015250	051101	051040	020110			
2459	015256	043					
2460	015257	045	041523	047040	MSG29:	.ASCII	/%SC NOT RESET BY INIT #/
2461	015264	052117	051040	051505			
2462	015272	052105	041040	020131			
2463	015300	047111	052111	021440			
2464	015306	052045	042522	047040	MSG30:	.ASCII	/%TRE NOT RESET BY INIT #/
2465	015314	052117	051040	051505			
2466	015322	052105	041040	020131			
2467	015330	047111	052111	021440			
2468	015336	041445	031123	047040	MSG31:	.ASCII	/%CS2 NOT RESET BY INIT #/
2469	015344	052117	051040	051505			
2470	015352	052105	041040	020131			
2471	015360	047111	052111	021440			
2472	015366	042045	052114	047040	MSG32:	.ASCII	/%DLT NOT SET #/
2473	015374	052117	051440	052105			
2474	015402	021440					
2475	015404	051445	020103	047516	MSG33:	.ASCII	/%SC NOT SET #/
2476	015412	020124	042523	020124			
2477	015420	043					
2478	015421	045	051124	020105	MSG34:	.ASCII	/%TRE NOT SET #/
2479	015426	047516	020124	042523			
2480	015434	020124	043				
2481	015437	045	051111	047040	MSG35:	.ASCII	/%IR NOT SET BY INIT #/

2482	015444	052117	051440	052105	
2483	015452	041040	020131	047111	
2484	015460	052111	021440		
2485	015464	047445	020122	047516	MSG36: .ASCII /%OR NOT RESET BY INIT #/
2486	015472	020124	042522	042523	
2487	015500	020124	054502	044440	
2488	015506	044516	020124	043	
2489	015513	045	051117	047040	MSG37: .ASCII /%OR NOT RESET BY 1 SILO ENTRY #/
2490	015520	052117	051040	051505	
2491	015526	052105	041040	020131	
2492	015534	020061	044523	047514	
2493	015542	042440	052116	054522	
2494	015550	021440			
2495	015552	047445	020122	047516	MSG38: .ASCII /%OR NOT SET BY SILO FULL #/
2496	015560	020124	042523	020124	
2497	015566	054502	051440	046111	
2498	015574	020117	052506	046114	
2499	015602	021440			
2500	015604	041045	042101	051440	MSG39: .ASCII /%BAD SILO READ #/
2501	015612	046111	020117	042522	
2502	015620	042101	021440		
2503	015624	044445	020122	047516	MSG40: .ASCII /%IR NOT RESET BY SILO FULL#/
2504	015632	020124	042522	042523	
2505	015640	020124	054502	051440	
2506	015646	046111	020117	052506	
2507	015654	046114	043		
2508	015657	045	047516	026516	MSG41: .ASCII /%NON-EXIST DRIVE#/
2509	015664	054105	051511	020124	
2510	015672	051104	053111	021505	
2511	015700	047045	047117	042455	MSG42: .ASCII /%NON-EXIST SLAVE#/
2512	015706	044530	052123	051440	
2513	015714	040514	042526	043	
2514	015721	045	042523	044522	MSG43: .ASCII /%SERIAL NO: #/
2515	015726	046101	047040	035117	
2516	015734	021440			
2517	015736	042445	040522	042523	MSG44: .ASCII /%ERASE HEAD INOPERATIVE#/
2518	015744	044040	040505	020104	
2519	015752	047111	050117	051105	
2520	015760	052101	053111	021505	
2521	015766	050045	051517	044523	MSG45: .ASCII /%POSSIBLE ERASE HEAD PROBLEM: /
2522	015774	046102	020105	051105	
2523	016002	051501	020105	042510	
2524	016010	042101	050040	047522	
2525	016016	046102	046505	020072	
2526	016024	044103	041505	020113	.ASCII /CHECK POLARITY#/
2527	016032	047520	040514	044522	
2528	016040	054524	043		
2529	016043	045	042523	026524	MSG46: .ASCII /%SET-UP WRITE ERROR#/
2530	016050	050125	053440	044522	
2531	016056	042524	042440	051122	
2532	016064	051117	043		
2533	016067	045	050123	041501	MSG47: .ASCII /%SPACE FORWARD ERROR#/
2534	016074	020105	047506	053522	
2535	016102	051101	020104	051105	
2536	016110	047522	021522		
2537	016114	051445	040520	042503	MSG48: .ASCII /%SPACE REVERSE ERROR#/

2538	016122	051040	053105	051105	
2539	016130	042523	042440	051122	
2540	016136	051117	043		
2541	016141	045	052502	043106	MSG49: .ASCII /%BUFFERED WRITE ERROR#/
2542	016146	051105	042105	053440	
2543	016154	044522	042524	042440	
2544	016162	051122	051117	043	
2545	016167	045	047502	020124	MSG50: .ASCII /%BOT SET AFTER BUFFERED WRITE#/
2546	016174	042523	020124	043101	
2547	016202	042524	020122	052502	
2548	016210	043106	051105	042105	
2549	016216	053440	044522	042524	
2550	016224	043			
2551	016225	045	047516	041040	MSG51: .ASCII /%NO BOT FROM READ IN PRESET#/
2552	016232	052117	043040	047522	
2553	016240	020115	042522	042101	
2554	016246	044440	020116	051120	
2555	016254	051505	052105	043	
2556	016261	045	041524	044440	MSG52: .ASCII /%TC INCORRECT #/
2557	016266	041516	051117	042522	
2558	016274	052103	021440		
2559	016300	046445	046117	043040	MSG53: .ASCII /%MOL FAILED TO CLEAR#/
2560	016306	044501	042514	020104	
2561	016314	047524	041440	042514	
2562	016322	051101	043		
2563	016325	045	051045	051505	MSG54: .ASCII /%%RESET SLAVE TO ON LINE BEFORE CONTINUING/
2564	016332	052105	051440	040514	
2565	016340	042526	052040	020117	
2566	016346	047117	046040	047111	
2567	016354	020105	042502	047506	
2568	016362	042522	041440	047117	
2569	016370	044524	052516	047111	
2570	016376	107			
2571	016377	045	042523	020124	.ASCII /%SET SW12=1 IF YOU DOT WISH TO REPEAT THIS TEST#/
2572	016404	053523	031061	030475	
2573	016412	044440	020106	047531	
2574	016420	020125	047504	020124	
2575	016426	044527	044123	052040	
2576	016434	020117	042522	042520	
2577	016442	052101	052040	044510	
2578	016450	020123	042524	052123	
2579	016456	043			
2580	016457	040	052111	051105	MSG56: .ASCII / ITER: #/
2581	016464	020072	043		
2582	016467	045	046524	047040	MSG57: .ASCII /%TM NOT SET#/
2583	016474	052117	051440	052105	
2584	016502	043			
2585	016503	045	044505	044124	MSG60: .ASCII /%EITHER TAPE NOT ERASED OR OPI PROBLEM#/
2586	016510	051105	052040	050101	
2587	016516	020105	047516	020124	
2588	016524	051105	051501	042105	
2589	016532	047440	020122	050117	
2590	016540	020111	051120	041117	
2591	016546	042514	021515		
2592	016552	051045	020110	047117	MSG62: .ASCII /%RH ONLY (NO=0,YES=1): #/
2593	016560	054514	024040	047516	

2594	016566	030075	054454	051505	
2595	016574	030475	035051	021440	
2596	016602	042045	042111	047040	MSG63: .ASCII /%DID NOT AUTO SELECT NRZ#/
2597	016610	052117	040440	052125	
2598	016616	020117	042523	042514	
2599	016624	052103	047040	055122	
2600	016632	043			
2601	016633	045	044504	020104	MSG64: .ASCII /%DID NOT AUTO SELECT PE#/
2602	016640	047516	020124	052501	
2603	016646	047524	051440	046105	
2604	016654	041505	020124	042520	
2605	016662	043			
2606	016663	045	043		MSG65: .ASCII /%#/
2607	016665	134	043		MSG66: .ASCII / #/
2608	016667	045	051105	020072	MSG67: .ASCII /%ER: #/
2609	016674	043			
2610	016675	045	042522	047515	MSG69: .ASCII /%REMOVE TMDP FROM SLAVE TO BE TESTED%#/
2611	016702	042526	052040	042115	
2612	016710	020120	051106	046517	
2613	016716	051440	040514	042526	
2614	016724	052040	020117	042502	
2615	016732	052040	051505	042524	
2616	016740	022504	043		
2617	016743	045	040510	042122	MSG70: .ASCII /%HARDWARE SWR IN USE%#/
2618	016750	040527	042522	051440	
2619	016756	051127	044440	020116	
2620	016764	051525	022505	043	
2621					

```

2622                                     ;TEST HEADERS*****
2623
2624 016771      045  043045  030524 MSFT1: .ASCII  /%%FT1:RH ADDRESSING #/
2625 016776 051072  020110  042101
2626 017004 051104  051505  044523
2627 017012 043516  021440
2628 017016 022445  052106  035062 MSFT2: .ASCII  /%%FT2:RH REGISTER BITS TEST #/
2629 017024 044122  051040  043505
2630 017032 051511  042524  020122
2631 017040 044502  051524  052040
2632 017046 051505  020124    043
2633 017053    045  043045  031524 MSFT3: .ASCII  /%%FT3:RH INITIALIZE TEST #/
2634 017060 051072  020110  047111
2635 017066 052111  040511  044514
2636 017074 042532  052040  051505
2637 017102 020124    043
2638 017105    045  043045  032124 MSFT4: .ASCII  /%%FT4:RH11 SILO TEST 1 #/
2639 017112 051072  030510  020061
2640 017120 044523  047514  052040
2641 017126 051505  020124  020061
2642 017134    043
2643 017135    045  043045  032524 MSFT5: .ASCII  /%%FT5:RH11 SILO TEST 2 #/
2644 017142 051072  030510  020061
2645 017150 044523  047514  052040
2646 017156 051505  020124  020062
2647 017164    043
2648 017165    045  043045  033124 MSFT6: .ASCII  /%%FT6:RH11 SILO TEST 3 #/
2649 017172 051072  030510  020061
2650 017200 044523  047514  052040
2651 017206 051505  020124  020063
2652 017214    043
2653 017215    045  043045  033524 MSFT7: .ASCII  /%%FT7:RH11 SILO TEST 4 #/
2654 017222 051072  030510  020061
2655 017230 044523  047514  052040
2656 017236 051505  020124  020064
2657 017244    043
2658 017245    045  043045  030524 MSFT10: .ASCII  /%%FT10:RH11 SILO TEST 5 #/
2659 017252 035060  044122  030461
2660 017260 051440  046111  020117
2661 017266 042524  052123  032440
2662 017274 021440
2663 017276 022445  052106  030461 MSFT11: .ASCII  /%%FT11:NOP TEST#/
2664 017304 047072  050117  052040
2665 017312 051505  021524
2666 017316 022445  052106  031061 MSFT12: .ASCII  /%%FT12:REWIND TEST#/
2667 017324 051072  053505  047111
2668 017332 020104  042524  052123
2669 017340    043
2670 017341    045  043045  030524 MSFT13: .ASCII  /%%FT13:WRITE-READ TEST#/
2671 017346 035063  051127  052111
2672 017354 026505  042522  042101
2673 017362 052040  051505  021524
2674 017370 022445  052106  032061 MSFT14: .ASCII  /%%FT14:SPACE TEST#/
2675 017376 051472  040520  042503
2676 017404 052040  051505  021524
2677 017412 022445  052106  032461 MSFT15: .ASCII  /%%FT15:ERASE TEST#/

```

2678	017420	042472	040522	042523	
2679	017426	052040	051505	021524	
2680	017434	022445	052106	033061	MSFT16: .ASCII /%%FT16:TAPE MARK WRITE-READ TEST#/
2681	017442	052072	050101	020105	
2682	017450	040515	045522	053440	
2683	017456	044522	042524	051055	
2684	017464	040505	020104	042524	
2685	017472	052123	043		
2686	017475	045	043045	030524	MSFT17: .ASCII /%%FT17:TM SPACE TEST #/
2687	017502	035067	046524	051440	
2688	017510	040520	042503	052040	
2689	017516	051505	020124	043	
2690	017523	045	043045	031124	MSFT20: .ASCII /%%FT20:WRITE CHECK TEST #/
2691	017530	035060	051127	052111	
2692	017536	020105	044103	041505	
2693	017544	020113	042524	052123	
2694	017552	021440			
2695	017554	022445	052106	030462	MSFT21: .ASCII /%%FT21:ERASE HEAD TEST#/
2696	017562	042472	040522	042523	
2697	017570	044040	040505	020104	
2698	017576	042524	052123	043	
2699	017603	045	043045	031124	MSFT22: .ASCII /%%FT22:BUFFERED COMMAND TEST#/
2700	017610	035062	052502	043106	
2701	017616	051105	042105	041440	
2702	017624	046517	040515	042116	
2703	017632	052040	051505	021524	
2704	017640	022445	052106	031462	MSFT23: .ASCII /%%FT23:READ IN PRESET TEST#/
2705	017646	051072	040505	020104	
2706	017654	047111	050040	042522	
2707	017662	042523	020124	042524	
2708	017670	052123	043		
2709	017673	045	043045	031124	MSFT26: .ASCII /%%FT26:REWIND-OFF LINE TEST#/
2710	017700	035066	042522	044527	
2711	017706	042116	047455	043106	
2712	017714	046040	047111	020105	
2713	017722	042524	052123	043	
2714	017727	045	043045	031124	MSFT24: .ASCII /%%FT24:AUTO DENSITY SELECT: WRITE-NRZ,READ-PE#/
2715	017734	035064	052501	047524	
2716	017742	042040	047105	044523	
2717	017750	054524	051440	046105	
2718	017756	041505	035124	053440	
2719	017764	044522	042524	047055	
2720	017772	055122	051054	040505	
2721	020000	026504	042520	043	
2722	020005	045	043045	031124	MSFT25: .ASCII /%%FT25:AUTO DENSITY SELECT: WRITE-PE,READ-NRZ#/
2723	020012	035065	052501	047524	
2724	020020	042040	047105	044523	
2725	020026	054524	051440	046105	
2726	020034	041505	035124	053440	
2727	020042	044522	042524	050055	
2728	020050	026105	042522	042101	
2729	020056	047055	055122	043	
2730	020063	045	043536	043	\$CNTG: .ASCII /% G#/
2731	020067	045	053523	036522	\$MSWR: .ASCII /%SWR= #/
2732	020074	021440			
2733	020076	020040	042516	036527	\$MNEW: .ASCII / NEW= #/

2734	020104	021440		
2735	020106	022477	043	\$QUEST: .ASCII /?%#/
2736				
2737				
2738		020112		WDATA: .EVEN
2739	020112	000000		0
2740		021624		.=.+1510
2741	021624	000000		RDATA: 0
2742				
2743		000001		.END

AS	000526	EXECX	011666	FT17D	007376	FT4X	004132	MSFT4	017105
BA	000514	EXECXA	011672	FT17D1	007414	FT5	004206	MSFT5	017135
BADDR	000616	EXECXX	011676	FT17E	007430	FT5A	004230	MSFT6	017165
BAE	000544	EXFL	000700	FT17F	007520	FT5B	004256	MSFT7	017215
BTRP	000574	FC	000516	FT17X	007542	FT5C	004300	MSG1	014364
BTRP2	000576	FCNT	000620	FT2	003316	FT5D	004330	MSG10	014725
CC	000530	FT1	003216	FT2A	003330	FT5E	004342	MSG11	014745
CHNFLG	001662	FT1A	003250	FT2B	003370	FT5X	004370	MSG12	014765
COUNT	000734	FT1B	003254	FT2C	003430	FT6	004400	MSG13	015003
CRCNT	000714	FT1X	003266	FT2D	003444	FT6A	004422	MSG14	015030
CS	000520	FT1XX	003312	FT2E	003474	FT6B	004430	MSG15	015055
C1	000510	FT10	004746	FT2ER	003504	FT6C	004466	MSG16	015076
DATA0	000742	FT10A	005002	FT2ERA	003534	FT6D	004510	MSG17	015116
DATA1	000744	FT10ER	005066	FT2ERB	003606	FT6DE	004542	MSG2	014452
DATA2	000746	FT10X	005044	FT2ERC	003616	FT6DEA	004574	MSG20	015136
DATA3	000750	FT10XX	005062	FT2X	003626	FT6DEB	004640	MSG21	015144
DATBL	000740	FT11	005102	FT20	007546	FT6DEX	004650	MSG22	015151
DAT1	012440	FT12	005220	FT20A	007564	FT6E	004512	MSG23	015161
DAT1A	012444	FT12A	005300	FT20B	007710	FT6X	004532	MSG24	015171
DAT2	012460	FT13	005332	FT20C	007742	FT7	004652	MSG25	015205
DAT3	012464	FT13A	005404	FT20X	007762	FT7A	004704	MSG26	015213
DAT4	012472	FT13B	005446	FT21	007772	FT7ER	004732	MSG27	015221
DB	000532	FT13C	005526	FT21A	010000	FT7X	004722	MSG28	015227
DOUT	014012	FT13D	005574	FT21B	010274	FUN	000710	MSG29	015257
DOUTD	014074	FT13X	005640	FT21C	010302	GTSWR	014222	MSG3	014470
DRVN	000612	FT14	005644	FT21SC	010130	HDRFL	000606	MSG30	015306
DRVTP	000564	FT14A	005716	FT21X	010314	HERE	003164	MSG31	015336
DS	000522	FT14A1	005662	FT22	010324	INIT1	012620	MSG32	015366
DSAV	000664	FT14A2	006010	FT22A	010400	INIT2	012630	MSG33	015404
DSUP	012400	FT14A3	006032	FT22B	010420	ITAMT	000566	MSG34	015421
DSO	012402	FT14B	006062	FT22X	010520	ITCNT	000662	MSG35	015437
DT	000536	FT14EC	006300	FT23	010530	ITER	012546	MSG36	015464
EMADDR	000610	FT14RE	006306	FT23A	010640	ITRLP	000676	MSG37	015513
ER	000524	FT14RF	006252	FT23B	010660	LTADD	000706	MSG38	015552
ERADD	000650	FT14RR	006210	FT23C	010710	MR	000534	MSG39	015604
ERCHK	011776	FT14R0	006360	FT23X	010744	MSFT1	016771	MSG4	014623
ERPT	012022	FT14R1	006402	FT24	010750	MSFT10	017245	MSG40	015624
ERPTA	012112	FT14R2	006406	FT24X	011132	MSFT11	017276	MSG41	015657
ERPTA1	012044	FT14R3	006460	FT25	011142	MSFT12	017316	MSG42	015700
ERPTB	012150	FT14X	006470	FT25X	011324	MSFT13	017341	MSG43	015721
ERPTB1	012162	FT14XX	006512	FT26	011334	MSFT14	017370	MSG44	015736
ERPTC	012212	FT15	006516	FT26X	011532	MSFT15	017412	MSG45	015766
ERPTD	012322	FT15A	006566	FT26XX	011542	MSFT16	017434	MSG46	016043
ERPTG	012052	FT15B	006602	FT3	003640	MSFT17	017475	MSG47	016067
ERPTX	012332	FT15X	006712	FT3A	003714	MSFT2	017016	MSG48	016114
ERPTXX	012336	FT16	006716	FT3B	003736	MSFT20	017523	MSG49	016141
ERPT1A	012144	FT16A	006750	FT3ER	003774	MSFT21	017554	MSG5	014646
ERRP	000626	FT16B	006754	FT3X	003764	MSFT22	017603	MSG50	016167
ERRP1	000630	FT16X	007134	FT4	004060	MSFT23	017640	MSG51	016225
EXEC	011546	FT17	007144	FT4ER	004142	MSFT24	017727	MSG52	016261
EXECA	011620	FT17A	007164	FT4ERA	004152	MSFT25	020005	MSG53	016300
EXECB	011626	FT17B	007170	FT4ERB	004162	MSFT26	017673	MSG54	016325
EXECC	011646	FT17C	007330	FT4ERC	004170	MSFT3	017053	MSG56	016457

MSG57	016467	OCTP3	013732	SCNT	000642	TEMP3	000656	TSCD3	003034
MSG6	014661	OFL	014010	SCOLP	000674	TEND	003110	TSRH	002750
MSG60	016503	OPDYX	000640	SCOPE	012512	TENDX	003206	TSTTBL	000752
MSG62	016552	PATRN	000720	SERFL	000712	TEX	013556	TTIN	013400
MSG63	016602	PCNTR	000730	SERNUM	000562	TIB	000602	TTINT	012656
MSG64	016633	PEXFL	000702	SLVN	000614	TINER	013360	TTOUT	013442
MSG65	016663	PFLG	000644	SN	000540	TKB	000554	TTR	013120
MSG66	016665	PSW	000546	SNPG	014176	TKS	000552	UDES	000716
MSG67	016667	RCNT	000624	SNPT	014120	TLAST	001110	VECT	000570
MSG69	016675	RDATA	021624	START	001600	TMCHK	012340	WC	000512
MSG7	014677	RDSW	000736	STFLG	000704	TMCHK0	012350	WCNT	000622
MSG70	016743	RDYDX	000636	STMSK	000660	TMCHK1	012352	WDATA	020112
MSG9	014703	REGS	000572	STSCD	003054	TOB	000600	\$CNTG	020063
MTINT	012646	RFD	000634	STO	002126	TOG	013542	\$DONE	003116
NRZOF	000724	RHOF	000726	STOB	001766	TPB	000560	\$ENDAD	003154
NXTDRV	002616	RHTF	000722	ST1	002150	TPS	000556	\$MNEW	020076
NXTSLV	002670	RH17F	000604	ST1A	002170	TRAP	013004	\$MSWR	020067
OCTP	013572	RRD	000632	ST2	002276	TRAPX	013066	\$QUEST	020106
OCTPE	013560	RTRN	000646	ST3	002406	TRAPXX	013114	\$SVPC =	000764
OCTPE1	013576	RWND	011700	ST4	002542	TRAP1	013040	. =	021626
OCTPG	013746	RWNDA	011720	SWR	000550	TRAP2	013056	.RESTO	014342
OCTPG0	013764	RWNDB	011724	SWREG	000176	TSCD	002556	.SAVE	014320
OCTPG1	013770	RWDX	011770	TC	000542	TSCDA	002746		
OCTP0	013612	SAV1	000666	TEMPST	000732	TSCD0	002756		
OCTP1	013634	SAV2	000670	TEMP1	000652	TSCD1	002764		
OCTP2	013642	SAV3	000672	TEMP2	000654	TSCD2	003022		

. ABS. 021626 000

ERRORS DETECTED: 0

,CZTUQA.SEQ/SOL_CZTUQA.P11
RUN-TIME: 24 40 2 SECONDS
RUN-TIME RATIO: 112/67=1.6
CORE USED: 14K (28 PAGES)