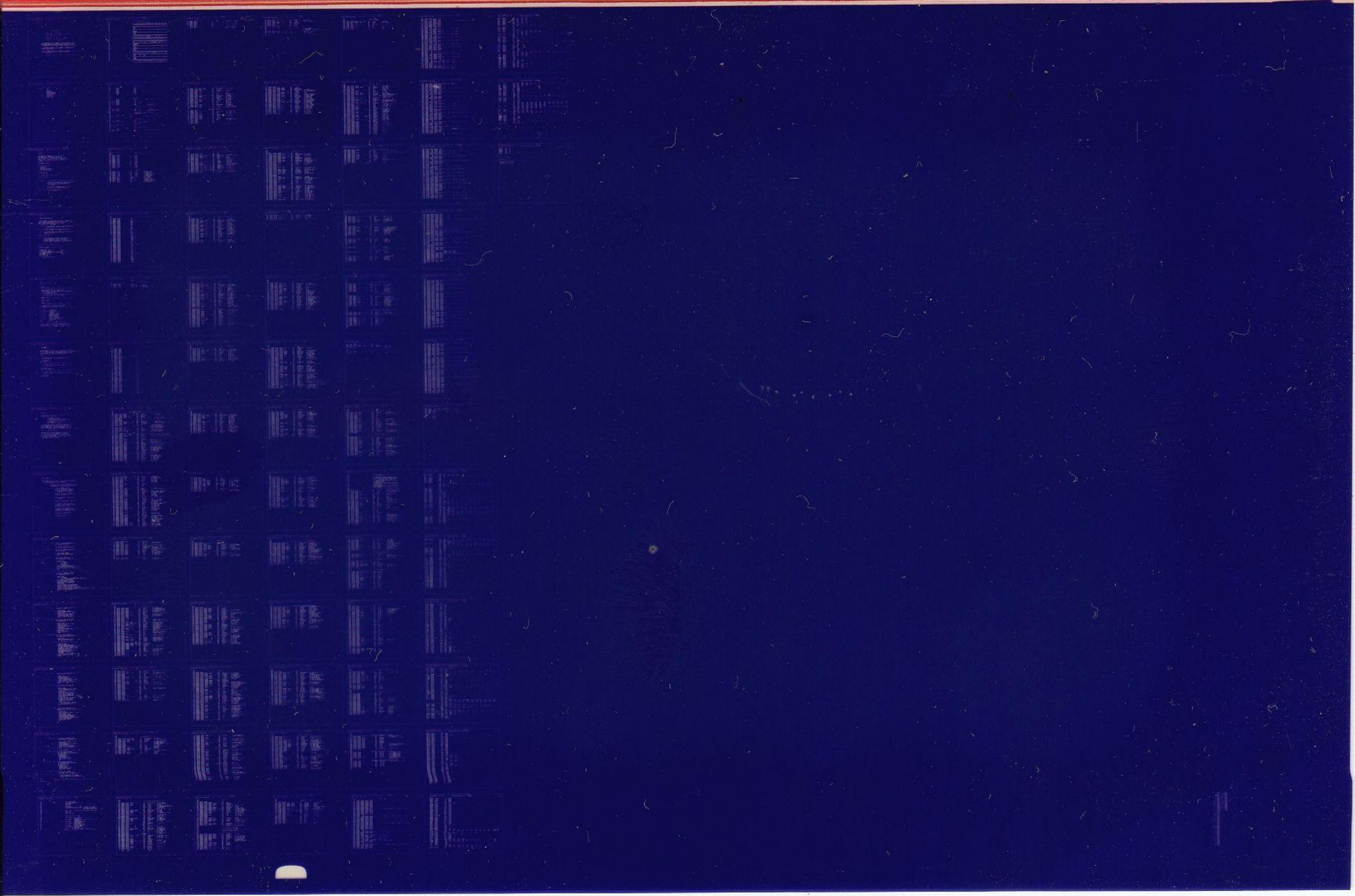


# TM03,TE16

TM03/TE16.TU77 BFT  
CZTECC0

AH-A798C-MC  
COPYRIGHT 77-80  
FICHE 1 OF 1

JAN 1980  
**digital**  
MADE IN USA



.REM %

IDENTIFICATION

PRODUCT CODE: AC-A797C-MC  
PRODUCT NAME: CZTECCO TM03-TE16/TU77 BASIC FUNCTION TEST  
DATE CREATED: 24 JUL 79  
MAINTAINER: DIAGNOSTIC ENGINEERING  
AUTHOR: J. G. ADAMS

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

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

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

COPYRIGHT (C) 1977, 1979 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

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

1. ABSTRACT

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

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP11 PROCESSOR
- B. 8K OF CORE
- C. CONSOLE TTY
- D. TMO3 MAGTAPE CONTROLLER
- E. MASS BUS CONTROLLER
- F. TE16 MAG TAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD BINARY LOADING PROCEDURE

4. STARTING PROCEDURE

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

- A. 200(8): STARTING AT THIS ADDRESS WILL CAUSE THE PROGRAM IDENTIFICATION TO BE PRINTED FOLLOWED BY REQUESTS FOR THE VARIOUS PARAMETERS NEEDED BY THE PROGRAM.
- B. 210(8): THIS ADDRESS IS INTENDED FOR USE AS A RESTART ONLY AND WILL USE THE 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 TMO3/TE16 COMBINATIONS ARE TESTED. ADDITIONALLY THE SOFTWARE  
SWR IS INVOKED WITH A SWITCH SETTING OF 000000  
IF LOADED VIA ACT11 CHAIN MODE.

\*\*EXCEPTION: IF THIS PROGRAM IS LOADED VIA TMDP CHAIN MODE THE  
PROGRAM WILL NOT TEST TMO3 DRIVE #0, TE16 SLAVE #0.

\*\*NOTE: IN ORDER TO CHANGE THE SETTING OF THE SOFTWARE SWR,  
SET LOC: 176(SWREG:) TO THE DESIRED SETTING.

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

4.2 SAMPLE START AT 200

NOTE: DEFAULT RESPONSES ARE SHOWN IN ANGLE BRACKETS <>,  
OPERATOR RESPONSES ARE SHOWN IN PARENTHESES (), AND  
LOCATIONS CONTAINING THE DEFAULT ARE SHOWN IN [].  
TO INVOKE THE DEFAULT RESPONSE TYPE (CR). NON STANDARD  
MODE FOR JUMPERS IS M8931 (W2-IN) ,M8937(W2-IN,W1-OUT).

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

TMO3-TE16/TU77 BASIC FUNCTIONS TEST (DZTEC-B)  
TYPE ^C TO RESTART

REGISTER START: <172440> (CR) [REGS:]  
VECTOR ADDRESS: <224> (CR) [VECT:]  
IS CONTROLLER JUMPERED IN NON-STANDARD MODE  
TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD: <3> [JUMPER:]  
DRIVE NUMBER: <0> (CR) [DRVN:]  
SLAVE NUMBER: <0> (CR) [SLVN:]  
SERIAL NO: 12345  
RH ONLY (NO=0,YES=1): <0> (0) [RHOF:]  
IF THE SOFTWARE SWR IS INVOKED:  
SWR = <000000> NEW = (CR)

5. CONSOLE SWITCH SETTING

CONTROL:

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

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

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

2) CONTROL A <^A>:  
ALTERNATES USAGE OF SWR FROM HARDWARE TO SOFTWARE & VICE VERSA.

3) CONTROL C <^C>:  
RESTARTS PROGRAM AT 200

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

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

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

THE USE OF SW0-4 IS TO ALLOW SELECTION AND CONTINUOUS  
EXECUTION OF ANY TEST. THE TEST SELECTION MAY BE CHANGED AT  
ANY TIME, HOWEVER IT IS ADVISABLE TO USE SW10 TO STOP THE  
PROGRAM AT THE END OF THE CURRENT TEST BEFORE SELECTING A TEST.

6. ERROR PRINTOUTS

THE ERROR PRINTOUTS FOR EACH TEST WILL APPEAR IN THE SAME GENERAL FORMAT. THE FIRST LINE WILL ALWAYS SHOW THE TEST NUMBER AND ITS TITLE. THE SECOND LINE WILL BE AN EXPLANATION OF THE ERROR. THE FOLLOWING LINES WILL SHOW THE APPROPRIATE REGISTER OR ADDRESS VALUES THAT ARE APPLICABLE TO THE INDIVIDUAL TEST

EXAMPLES:

1. THIS EXAMPLE SHOWS A TYPICAL ERROR PRINTOUT FOR THE WRITE READ TEST: A WRITE CRC ERROR OCCURRED ON SLAVE 6.

FT13: WRITE-READ TEST

WRITE ERROR NRZ

CS1	WC	BA	FC	CS2	DS	ER	TC
144260	000000	015650	000000	000103	150600	100000	101306

2. THIS EXAMPLE SHOWS A TYPICAL SPACE ERROR:  
THE FC IS NOT ZERO AT THE END OF THE OPERATION.

FT14: SPACE TEST

SPACE REVERSE ERROR NRZ

CS1	WC	BA	FC	CS2	DS	ER	TC
144230	177700	017162	177740	000114	150600	001000	161700

3. THIS EXAMPLE SHOWS A SPACE OPERATION WHICH RESULTED IN INCORRECT POSITIONING. SHOULD BE AT RECORD 20, IS AT RECORD 22.

FT14: SPACE TEST

POSITION ERROR:

REVERSE ERROR EXPT:20 RCVD:22

7. OPERATION

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

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE  
\*\*\*\*REFER TO SECTION 5 FOR DYNAMIC LOADING  
OF SOFTWARE SWITCH REGISTER.\*\*\*
3. PRESS START
4. ENTER APPROPRIATE 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 TMO3-TE16/TU77 ITSELF. (SEE RH ONLY OPTION; PAR 7)

FT1: RH ADDRESSING: THIS TEST WILL ASSURE THAT THE  
RH WILL RESPOND WITHOUT CAUSING A BUS  
TRAP TO ALL TMO2 REGISTER ADDRESS  
IN SEQUENCE STARTING AT THE ADDRESS  
OF CS1 ENTERED BY THE OPERATOR.

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

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

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

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

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

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

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

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

B. TM03-TE16/TU77 BASIC FUNCTIONS: THE FOLLOWING FOURTEEN (14) TESTS WILL ASSURE OPERATION OF THE MAG TAPE BASIC FUNCTIONS.

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

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

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

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

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

DATA READ IS NOT CHECKED; ONLY THE FUNCTION IS TESTED, NOT THE MEDIUM.

FT14: SPACE TEST: THIS TEST WILL ASSURE THAT PROPER POSITIONING IS MAINTAINED BY BOTH SPACE FORWARD AND REVERSE.

1. REWIND TO BOT
2. WRITE 100 RECORDS
  - A. EACH RECORD IS ONE FRAME LARGER THAN THE LAST. THIS WILL ALLOW FOR POSITION CHECKING BY RECORD SIZE.
3. EACH RECORD IS ERROR CHECKED.
4. DATA RELATED ERRORS ARE IGNORED.
5. NOW SPACE REVERSE 77 RECORDS AND READ REVERSE 1, THE FRAME COUNT SHOULD BE 100. THIS IS THE SIZE OF THE FIRST RECORD.
6. NOW SPACE FORWARD 76 RECORDS AND READ FORWARD 1, THE FRAME COUNT SHOULD BE 177. THIS IS THE SIZE OF THE NEXT TO LAST RECORD.
7. CONTINUE THE SPACE AND READ (DECREMENTING THE RECORD COUNT EACH TIME) UNTIL ALL POSITIONS HAVE BEEN CHECKED. IF POSITION IS LOST; TEST ENDS.
8. REPEAT STEPS 1 THRU 7 FOR PE.
9. END

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

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

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

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

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

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

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

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

FT20: WRITE CHECK TEST: BOTH WRITE CHECK FORWARD AND REVERSE ARE TESTED IN BOTH PE AND NRZ.

1. REWIND TO BOT.
2. WRITE A 400 FRAME RECORD USING DATA PATTERN 3 (125125).
3. ASSURE NO ERRORS OCCURED.
4. ISSUE A REVERSE WRITE CHECK COMMAND.
5. ASSURE NO ERRORS OCCURED.
6. REPEAT STEP 5 FOR A FORWARD WRITE CHECK.
7. REPEAT STEPS 1 THRU 6 FOR PE.
8. END

FT21: ERASE HEAD TEST: THIS TEST WILL ASSURE THAT THE ERASE HEAD ITSELF IS OPERATING.

1. REWIND TO BOT.
2. WRITE 2 RECORDS OF 800(10) FRAMES EACH. EACH RECORD WILL BE 1 INCH OF TAPE. DATA IS NOT ALL ONES.
3. REWIND TO BOT.
4. NOW WRITE A 400(10) FRAME RECORD. THIS RECORD WILL BE ONE HALF INCH OF TAPE. THE ERASE HEAD SHOULD CLEAR THE REMAINDER OF THE FIRST RECORD (ONE HALF INCH).
5. REWIND TO BOT.
6. NOW READ THE SHORT FIRST RECORD. IT SHOULD BE 400(10) FRAMES.
7. NOW READ THE SECOND RECORD. IT SHOULD BE STILL 800(10) FRAMES.
8. IF THE SECOND RECORD IS TOO LONG, THE ERASE HEAD DID NOT FUNCTION OR IT IS IN THE WRONG POLARITY.
10. END

FT22: BUFFERED COMMAND: THIS TEST WILL ASSURE THAT THE TMO2 WILL ACCEPT AND EXECUTE ANOTHER COMMAND WHILE ITS SELECTED SLAVE IS REWINDING.

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

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

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

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

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

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

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

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

FT27: REWIND: OFF LINE THIS TEST WILL ASSURE  
THAT THE UNIT WILL REWIND AND  
GO OFF LINE. (NOT IF IN CONTINUOUS CYCLE)

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

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

527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537  
538  
539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558

```
.LIST BIN,LOC,SEQ  
.TITLE CZTECCO TM03-TE16/TU77 BFT  
:BASIC FUNCTION TEST  
:AC-A797B-MC  
:FEB 77  
:J.G. ADAMS  
:REVISED JUN 1977 BY J.G. ADAMS ;++B ADDED TU77 CAPABILITY  
:REVISED NOV 1978 BY MIKE PAGE ;+ DESIGNATES CODE ADDED FOR  
:NON-STANDARD JUMPER CONFIG.  
  
.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.
```



```
605 ;REGISTER EQUIVS*****
606
607 000000 R0=%0
608 000001 R1=%1
609 000002 R2=%2
610 000003 R3=%3
611 000004 R4=%4
612 000005 R5=%5
613 000006 SP=%6
614 000007 PC=%7
615
616
617
618 ;ACT11 HOOK *****
619 000764 $SVPC= ;SAVE CURRENT LOCATION CTR
620 000042 .=42
621 000042 000000 .WORD 0
622 000046 000046 .=46
623 000046 003222 .WORD $ENDAD ;SET LOCATION 46
624 000052 000052 .=52
625 000052 000000 .WORD 0 ;SET LOCATION 52 = 0
626 000764 000764 .= $SVPC ;RESTORE LOCATION CTR
627
628 ;TTY INTERRUPT VECTOR*****
629
630 000060 .=60
631 000060 013532 .WORD TTINT ;TTY INTERRUPT HEADER ADDRESS
632 000062 000340 .WORD 340 ;PRIORITY LEVEL 7
633
634 ;SOFTWARE SWITCH REGISTER*****
635 ;USED IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
636
637 000176 .=176
638 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
639
640
641 ;START ADDRESS*****
642
643 000200 .=200
644 000200 000137 001600 JMP START ;PROGRAM START
645
646 ;RESTART ADDRESS*****
647 000210 .=210
648 000210 000137 002540 JMP ST4
649
650 ;TM03 INTERRUPT VECTOR*****
651
652 000224 .=224
653 000224 013522 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
654 000226 000340 340
655
```

```
656
657      000510
658      .=510
659      ;MASS BUS REGISTER EQUIVS*****
660 000510 172440 C1: 172440
661 000512 172442 WC: 172442
662 000514 172444 BA: 172444
663 000516 172446 FC: 172446
664 000520 172450 CS: 172450
665 000522 172452 DS: 172452
666 000524 172454 ER: 172454
667 000526 172456 AS: 172456
668 000530 172460 CC: 172460
669 000532 172462 DB: 172462
670 000534 172464 MR: 172464
671 000536 172466 DT: 172466
672 000540 172470 SN: 172470
673 000542 172472 TC: 172472
674 000544 172474 BAE: 172474
675
676      ;CONSTANTS*****
677
678 000546 177776 PSW: 177776 ;PROCESSOR STATUS
679 000550 177570 SWR: 177570 ;SWITCH REGISTER
680 000552 177560 TKS: 177560 ;TTY READER STATUS
681 000554 177562 TKB: 177562 ;TTY READ BUFFER
682 000556 177564 TPS: 177564 ;TTY PUNCH STATUS
683 000560 177566 TPB: 177566 ;TTY PUNCH BUFFER
684 000562 177777 SERNUM: 177777 ;SERIAL NUMBER
685 000564 000011 DRVTP: 011 ;DRIVE TYPE
686 000566 000010 ITAMT: 10 ;ITERATION AMOUNT
687 000570 000224 VECT: 224 ;INTERRUPT VECTOR(RH)
688 000572 172440 REGS: 172440 ;STARTING REGISTER ADDRESS
689 000574 000004 BTRP: 4 ;BUS TRAP ADDRESS
690 000576 000006 BTRP2: 6 ;BUS TRAP PRIORITY LEVEL 7
```

```
691 ;FLAGS AND COUNTERS*****
692
693 000600 000000 TOB: 0
694 000602 000000 TIB: 0
695 000604 000000 RH17F: 0
696 000606 000000 HDRFL: 0
697 000610 000000 EMADDR: 0
698 000612 000000 DRVN: 0
699 000614 000000 SLVN: 0
700 000616 000000 BADDR: 0
701 000620 000000 FCNT: 0
702 000622 000000 WCNT: 0
703 000624 000000 RCNT: 0
704 000626 000000 ERRP: 0
705 000630 000000 ERRP1: 0
706 000632 000000 RRD: 0
707 000634 000000 RFD: 0
708 000636 000000 RDYDX: 0
709 000640 000000 OPDYX: 0
710 000642 000000 SCNT: 0
711 000644 000000 PFLG: 0
712 000646 000000 RTRN: 0
713 000650 000000 ERADD: 0
714 000652 000000 TEMP1: 0
715 000654 000000 TEMP2: 0
716 000656 000000 TEMP3: 0
717 000660 000000 STMSK: 0
718 000662 000000 ITCNT: 0
719 000664 000000 DSAV: 0
720 000666 000000 SAV1: 0
721 000670 000000 SAV2: 0
722 000672 000000 SAV3: 0
723 000674 000000 SCOLP: 0
724 000676 000000 ITRLP: 0
725 000700 000000 EXFL: 0
726 000702 000000 PEXFL: 0
727 000704 000000 STFLG: 0
728 000706 000000 LTADD: 0
729 000710 000000 FUN: 0
730 000712 000000 SERFL: 0
731 000714 000000 CRCNT: 0
732 000716 000000 UDES: 0
733 000720 000000 PATRN: 0
734 000722 000000 RHTF: 0
735 000724 000000 NRZOF: 0
736 000726 000000 RHOF: 0
737 000730 000000 PCNTR: 0
738 000732 000000 TEMPST: 0
739 000734 000000 COUNT: 0
740 000736 000000 RDSW: 0
741 000740 000000 NONSTD: 0
742 000742 000000 JUMPER: 0
743
```

744  
745  
746  
747 000744 000000  
748 000746 013216  
749 000750 013236  
750 000752 013242  
751 000754 013250

;DATA PATTERN GENERATORS\*\*\*\*\*

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

```
752  
753 ;LOGIC TEST ENTRY TABLE*****  
754  
755 TSTTBL: 0  
756 000756 000000 0  
757 000760 000000 FT1  
758 000762 003264 FT1  
759 000764 003264 FT1  
760 000766 003372 FT2  
761 000770 003372 FT2  
762 000772 003714 FT3  
763 000774 003714 FT3  
764 000776 004134 FT4  
765 001000 004134 FT4  
766 001002 004262 FT5  
767 001004 004262 FT5  
768 001006 004454 FT6  
769 001010 004454 FT6  
770 001012 004726 FT7  
771 001014 004726 FT7  
772 001016 005022 FT10  
773 001020 005022 FT10  
774 001022 005156 FT11  
775 001024 005156 FT11  
776 001026 005274 FT12  
777 001030 005274 FT12  
778 001032 005406 FT13  
779 001034 005406 FT13  
780 001036 005720 FT14  
781 001040 005720 FT14  
782 001042 006572 FT15  
783 001044 006572 FT15  
784 001046 007204 FT16  
785 001050 007204 FT16  
786 001052 007432 FT17  
787 001054 007432 FT17  
788 001056 010034 FT20  
789 001060 010034 FT20  
790 001062 010260 FT21  
791 001064 010260 FT21  
792 001066 010612 FT22  
793 001070 010612 FT22  
794 001072 011016 FT23  
795 001074 011016 FT23  
796 001076 011236 FT24  
797 001100 011236 FT24  
798 001102 011430 FT25  
799 001104 011430 FT25  
800 001106 011622 FT26  
801 001110 011622 FT26  
802 001112 012060 FT27  
803 001114 012060 FT27  
804 001116 003156  
805 001120 000027
```

TLAST: .WORD TEND  
.WORD 27

;CONTAINS # OF TESTS

```

805          001600          . =1600
806          ;PROGRAM START AND HOUSEKEEPING*****
807
808 001600 012706 000500      START: MOV #500,SP          ;SET STACK POINTER
809 001604 013746 000006      MOV @#6,-(SP)        ;SAVE VECTORS
810 001610 013746 000004      MOV @#4,-(SP)
811 001614 012737 001640 000004  MOV #1$,@#4          ;SET UP FOR TIMEOUT
812 001622 005037 000006      CLR @#6
813 001626 022777 177777 176714  CMP #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
814 001634 001402
815 001636 000404
816 001640 022626      1$: CMP (SP)+,(SP)+    ;ADJUST STACK
817 001642 012737 000176 000550  2$: MOV #SWREG,SWR    ;POINT TO SOFTWARE SWITCH REG
818 001650 012637 000004      3$: MOV (SP)+,@#4      ;RESTORE VECTORS
819 001654 012637 000006      MOV (SP)+,@#6
820 001660 005027      CLR (PC)+          ;;CLEAR CHAIN INDICATOR
821 001662 000000      CHNFLG: .WORD 0    ;;CHAIN MODE INDICATOR
822                                     ;;1/0 = CHAIN/NOT CHAIN MODE
823 001664 005737 000042      TST @#42           ;;BRANCH IF IN DUMP MODE
824 001670 001407
825 001672 012737 000176 000550  BEQ 50$
826 001700 005237 001662      MOV #SWREG,SWR    ;;INVOKE SOFTWARE SWR
827 001704 000137 001710      INC CHNFLG        ;;SET CHNFLG = CHAIN MODE
828 001710      JMP SCHN         ;;GO TO CHAIN ADDRESS
829 001710 000240      50$: NOP
830 001712 122737 000006 000041  SCHN: CMPB #6,@#41    ;BRANCH IF LOADED VIA TMDP (DUMP MODE)
831 001720 001005      BNE 5$
832 001722 012704 020032      MOV #MSG69,R4     ;ADVISE USER TO REMOVE TMDP FROM UUT
833 001726 004737 014316      JSR PC,TTOUT
834 001732 000000      HALT
835 001734 012704 015401      5$: MOV #MSG3,R4
836 001740 004737 014316      JSR PC,TTOUT      ;PRINT TITLE
837 001744 005737 001662      TST CHNFLG        ;SEE IF IN CHAIN MODE
838 001750 001402      BEQ 6$           ;IF NOT: BR
839 001752 000137 002554      JMP TSCD         ;ELSE GO START TEST
840 001756 112737 000043 015401  6$: MOVB #'#,MSG3    ;DO NOT PRINT TITLE ON RESTART
841 001764 012704 015541      STOB: MOV #MSG4,R4
842 001770 004737 014316      JSR PC,TTOUT      ;REQUEST REGISTER ADDRESS
843 001774 013703 000572      MOV REGS,R3
844 002000 004737 014446      JSR PC,OCTP       ;PRINT CURRENT ADDRESS
845 002004 012705 000572      MOV #REGS,R5      ;SET ADDRESS SAVE LOC
846 002010 012701 000007      MOV #7,R1         ;SET SIZE OF RESPONSE
847 002014 012702 176400      MOV #176400,R2    ;SET UPPER LIMIT
848 002020 012703 172300      MOV #172300,R3    ;SET LOWER LIMIT
849 002024 004737 013774      JSR PC,TTR        ;GO GET RESPONSE
850 002030 012704 015564      MOV #MSG5,R4
851 002034 004737 014316      JSR PC,TTOUT      ;REQUEST VECTOR
852 002040 013703 000570      MOV VECT,R3
853 002044 004737 014446      JSR PC,OCTP       ;PRINT CURRENT VECTOR
854 002050 012705 000570      MOV #VECT,R5      ;SET ADDRESS SAVE LOC
855 002054 012701 000004      MOV #4,R1         ;SET SIZE OF RESPONSE
856 002060 012702 000224      MOV #224,R2       ;SET UPPER LIMIT
857 002064 012703 000150      MOV #150,R3       ;SET LOWER LIMIT
858 002070 004737 013774      JSR PC,TTR        ;GO GET RESPONSE
859 002074 013700 000570      MOV VECT,R0       ;GET VECTOR
860 002100 012720 013522      MOV #MTINT,(R0)+  ;LOAD INTERRUPT ADDRESS IN VECTOR
  
```

861	002104	012710	000340		MOV	#340,(R0)	:LOAD PRIORITY
862	002110	013700	000572		MOV	REGS,R0	:GET START OF REGS
863	002114	012701	000017		MOV	#17,R1	:SET NUMBER OF REGS
864	002120	012702	000510		MOV	#C1,R2	:GET START OF TABLE
865	002124	010022		ST0:	MOV	R0,(R2)+	:BUILD TABLE
866	002126	062700	000002		ADD	#2,R0	:BUMP ADDRESS
867	002132	005301			DEC	R1	:SEE IF DONE
868	002134	001373			BNE	ST0	:IF NOT: BR
869	002136	012702	000600		MOV	#TOB,R2	
870	002142	012700	000054		MOV	#54,R0	
871	002146	005022		ST1:	CLR	(R2)+	:CLEAR FLAGS + COUNTERS
872	002150	005300			DEC	R0	
873	002152	001375			BNE	ST1	
874	002154	012737	000001	000722	MOV	#1,RHTF	:SET ADDRESS TEST FLAG
875	002162	000137	003016		JMP	TSRH	:GO DO INITIAL ADDRESS TEST PASS
876	002166	012704	015643	ST1A:	MOV	#MSG10A,R4	
877	002172	004737	014316		JSR	PC,TTOUT	:REQUEST JUMPER CONFIGURATION
878	002176	012705	000742		MOV	#JUMPER,R5	:GET ADDRESS OF RESPONSE
879	002202	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
880	002206	012702	000004		MOV	#4,R2	:SET RANGE
881	002212	012703	000000		MOV	#0,R3	:LOWER LIMIT
882	002216	004737	013774		JSR	PC,TTR	:GET RESPONSE
883	002222	022737	000002	000742	CMP	#2,JUMPER	:TEST FOR NON-STANDARD MODE
884	002230	001002			BNE	1\$	
885	002232	004737	013376		JSR	PC,NOST	:MODIFY TEST SCHEDULE
886	002236	012704	016003	1\$:	MOV	#MSG10,R4	
887	002242	004737	014316		JSR	PC,TTOUT	:REQUEST DRIVE NUMBER
888	002246	013703	000612		MOV	DRVN,R3	:GET CURRENT DRIVE #
889	002252	004737	014446		JSR	PC,OCTP	:AND TYPE IT
890	002256	012705	000612		MOV	#DRVN,R5	:SET ADDRESS OF DRIVE NUMBER SAVE
891	002262	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
892	002266	012702	000007		MOV	#7,R2	:SET UPPER LIMIT
893	002272	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
894	002276	004737	013774		JSR	PC,TTR	:GO GET RESPONSE
895	002302	012777	000040	176210	MOV	#40,@CS	:SET INIT
896	002310	053777	000612	176202	BIS	DRVN,@CS	:SET DRIVE NUMBER
897	002316	005777	176166		TST	@C1	:ACCESS DRIVE
898	002322	032777	010000	176170	BIT	#10000,@CS	:SEE IF NED
899	002330	001405			BEQ	ST2	:IF NOT: BR
900	002332	012704	017002		MOV	#MSG41,R4	
901	002336	004737	014316		JSR	PC,TTOUT	:PRINT NOT AVAIL
902	002342	000711			BR	ST1A	:REDO DRIVE REQUEST
903	002344	012704	016023	ST2:	MOV	#MSG11,R4	
904	002350	004737	014316		JSR	PC,TTOUT	:REQUEST SLAVE NUMBER
905	002354	013703	000614		MOV	SLVN,R3	:GET CURRENT SLAVE #
906	002360	004737	014446		JSR	PC,OCTP	:AND TYPE IT
907	002364	012705	000614		MOV	#SLVN,R5	:SET ADDRESS OF SLAVE SAVE
908	002370	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
909	002374	012702	000007		MOV	#7,R2	:SET UPPER LIMIT
910	002400	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
911	002404	004737	013774		JSR	PC,TTR	:GO GET RESPONSE
912	002410	012777	000040	176102	MOV	#40,@CS	:INIT
913	002416	053777	000612	176074	BIS	DRVN,@CS	:SET DRIVE NUMBER
914	002424	013777	000614	176110	MOV	SLVN,@TC	:LOAD SLAVE NUMBER
915	002432	032777	002000	176076	BIT	#2000,@DT	:SEE IF SLAVE PRESENT
916	002440	001005			BNE	ST3	:IF SO: BR

917	002442	012704	017023	MOV	#MSG42,R4	
918	002446	004737	014316	JSR	PC,TTOUT	:PRINT NON-EXIST SLAVE
919	002452	000734		BR	ST2	:REDO SLAVE REQUEST
920	002454	012704	017044	ST3: MOV	#MSG43,R4	
921	002460	004737	014316	JSR	PC,TTOUT	:PRINT SERIAL NUMBER TAG
922	002464	017703	176050	MOV	@SN,R3	
923	002470	004737	014774	JSR	PC,SNPT	:PRINT SERIAL NUMBER
924	002474	012704	017707	MOV	#MSG62,R4	:GET REQUEST
925	002500	004737	014316	JSR	PC,TTOUT	:REQUEST RH11 ONLY RESPONSE
926	002504	013703	000726	MOV	RHOF,R3	:GET CURRENT FLAG SETTING
927	002510	004737	014446	JSR	PC,OCTP	:AND TYPE IT
928	002514	012705	000726	MOV	#RHOF,R5	:SET FLAG ADDRESS
929	002520	012701	000002	MOV	#2,R1	:SET SIZE OF RESPONSE
930	002524	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
931	002530	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
932	002534	004737	013774	JSR	PC,TTR	:GO GET RESPONSE
933						
934				:START	210	
935	002540	012706	000500	ST4: MOV	#500,SP	:SET STACK PTR
936	002544	005037	000730	CLR	PCNTR	:CLEAR PASS COUNTER
937	002550	004737	015076	JSR	PC,GTSWR	:GET SWITCHES

```
938 ;TEST SCHEDULAR*****
939
940 002554 052777 000100 175770 TSCD: BIS #100,@TKS ;SET KEYBOARD IE BIT
941 002562 005037 000704 CLR STFLG ;CLEAR SINGLE TEST FLAG
942 002566 005037 000604 CLR RH17F ;SET RH INDICATOR = RH11
943 002572 013746 000004 MOV @#4,-(SP) ;SAVE ERROR TRAP VECTORS
944 002576 013746 000006 MOV @#6,-(SP) ;AND PRIORITY
945 002602 012737 002626 000004 MOV #1$,@#4 ;SET TIME OUT TRAP TO 1$ BELOW
946 002610 005037 000006 CLR @#6
947 002614 005777 175724 TST @BAE ;REFERENCE BAE REGISTER
948 002620 012737 000001 000604 MOV #1,RH17F ;SET FLAG = RH70
949 002626 012637 000006 1$: MOV (SP)+,@#6 ;RESTORE ERROR TRAP
950 002632 012637 000004 MOV (SP)+,@#4
951 002636 017700 175706 MOV @SWR,R0
952 002642 042700 177740 BIC #177740,R0
953 002646 001125 BNE STSCD ;GO SELECT SINGLE TEST
954 002650 005737 001662 TST CHNFLG ;:BRANCH IF NOT IN CHAIN MODE
955 002654 001457 BEQ TSCDA
956 002656 012737 177777 000612 MOV #-1,DRVN ;:INITIALIZE DRIVE #
957 002664 012737 177777 000614 NXTDRV: MOV #-1,SLVN ;:INITIALIZE SLAVE #
958 002672 012777 000040 175620 1$: MOV #40,@CS ;:INIT CONTROLLER
959 002700 005237 000612 INC DRVN ;:STEP DRIVE #
960 002704 022737 000010 000612 CMP #10,DRVN ;:EXIT IF ALL DRIVES TESTED
961 002712 001524 BEQ $DONE ;:FOR AVAILABILITY
962 002714 013777 000612 175576 MOV DRVN,@CS ;:LOAD DRIVE #
963 002722 005777 175562 TST @C1 ;:ACCESS DRIVE
964 002726 032777 010000 175564 BIT #10000,@CS ;:BRANCH IF DRIVE NON EXISTANT
965 002734 001356 BNE 1$ ;:(NED = 1)
966 002736 005237 000614 NXTSLV: INC SLVN ;:STEP SLAVE # AND BRANCH
967 002742 001011 BNE 1$ ;:IF NOT SLAVE 0
968 002744 005737 000612 TST DRVN ;:BRANCH IF NOT DRIVE # 0
969 002750 001006 BNE 1$
970 002752 122737 000006 000041 CMPB #6,@#41 ;:BRANCH IF NOT TMDP
971 002760 001002 BNE 1$
972 002762 005237 000614 INC SLVN ;:STEP TO SLAVE # 1
973 002766 022737 000010 000614 1$: CMP #10,SLVN ;:BRANCH IF ALL SLAVES TESTED
974 002774 001733 BEQ NXTDRV ;:FOR AVAILABILITY
975 002776 013777 000614 175536 MOV SLVN,@TC ;:LOAD SLAVE UNIT #
976 003004 032777 002000 175524 BIT #2000,@DT ;:BRANCH IF SLAVE NOT
977 003012 001751 BEQ NXTSLV ;:PRESENT (SPR = 0)
978 003014 000240 TSCDA: NOP
979 003016 012737 000756 000706 TSRH: MOV #TSTTBL,LTADD
980 003024 062737 000004 000706 TSCD0: ADD #4,LTADD
981 003032 013737 000706 000676 TSCD1: MOV LTADD,IIRLP
982 003040 062737 000002 000676 ADD #2,IIRLP ;SET ITERATION ADDRESS
983 003046 005037 000660 CLR STMSK
984 003052 005037 000626 CLR ERRP
985 003056 005037 000606 CLR HDRFL ;CLEAR PRINT HEADER FLAG
986 003062 017700 175620 MOV @LTADD,R0 ;SET POINTER TO TEST
987 003066 000110 JMP (R0) ;GO TO TEST
988 003070 032777 002000 175452 TSCD2: BIT #2000,@SWR ;SEE IF HALT ON TEST
989 003076 001401 BEQ TSCD3 ;IF NOT: BR
990 003100 000000 HALT
991 003102 005737 000704 TSCD3: TST STFLG ;SE IF SINGLE TEST
992 003106 001746 BEQ TSCD0 ;IF NOT: BR
993 003110 017700 175434 MOV @SWR,R0
```



```
1028
1029
1030
1031 003264 012737 020126 000610 FT1:  MOV    #MSFT1,EMADDR  ;SET HEADER
1032 003272 012737 013660 000004      MOV    #TRAP,@#4    ;SET TRAP HANDLER ADDRESS
1033 003300 012737 000340 000006      MOV    #340,@#6
1034 003306 012700 000016      MOV    #16,R0      ;SET NUMBER OF REGISTERS
1035 003312 013701 000510      MOV    C1,R1      ;GET FIRST ADDRESS (CS1)
1036 003316 005711      FT1A:  TST    (R1)    ;REFERENCE REGISTER
1037 003320 000240      NOP                ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
1038 003322 005300      FT1B:  DEC    R0      ;SEE IF DONE ALL
1039 003324 001403      BEQ    FT1X        ;IF SO: BR
1040 003326 062701 000002      ADD    #2,R1      ;BUMP ADDRESS POINTER
1041 003332 000771      BR     FT1A        ;CONTINUE
1042 003334 012737 000006 000004 FT1X:  MOV    #6,@#4    ;RESET TRAP CATCHER
1043 003342 012737 000000 000006      MOV    #HALT,@#6
1044 003350 005737 000722      TST    RHTF        ;SEE IF INITIAL ADDRESS TEST PASS
1045 003354 001404      BEQ    FT1XX       ;IF NOT: BR
1046 003356 005037 000722      CLR    RHTF        ;CLEAR FLAG
1047 003362 000137 002166      JMP    ST1A        ;RETURN
1048 003366 000137 003070      FT1XX: JMP    TSCD2   ;RETURN TO SCHEDULAR
```

```

1049
1050 ;RH REGISTER BITS READ/WRITE*****
1051
1052 003372 012737 020153 000610 FT2:  MOV #MSFT2,EMADDR ;SET TEST HEADER
1053 003400 012701 177777          MOV #-1,R1 ;SET ALL ONES PATTERN
1054 003404 004737 013474          FT2A: JSR PC,INIT1 ;GO INIT
1055 003410 013700 000512          MOV WC,R0 ;GET ADDRESS OF WORD COUNT
1056 003414 010102          MOV R1,R2 ;SET EXPT REGISTER BIT PATTERN
1057 003416 010110          MOV R1,(R0) ;LOAD PATTERN
1058 003420 021002          CMP (R0),R2 ;SEE IF EXPT=RCVD
1059 003422 001410          BEQ FT2B ;IF SO: BR
1060 003424 012737 016330 000650          MOV #MSG25,ERADD ;SET CODE
1061 003432 012737 003404 000674          MOV #FT2A,SCOLP ;SET SCOPE
1062 003440 004737 003560          JSR PC,FT2ER ;GO DO ERROR
1063 003444 013700 000514          FT2B: MOV BA,R0 ;GET ADDRESS OF BUS ADDRESS
1064 003450 010102          MOV R1,R2
1065 003452 042702 000001          BIC #1,R2 ;SET EXPT PATTERN
1066 003456 010110          MOV R1,(R0) ;LOAD PATTERN
1067 003460 020210          CMP R2,(R0) ;SEE IF EXPT=RCVD
1068 003462 001410          BEQ FT2C ;IF SO:BR
1069 003464 012737 016336 000650          MOV #MSG26,ERADD ;SET ERROR CODE
1070 003472 012737 003444 000674          MOV #FT2B,SCOLP ;SET SCOPE ADDRESS
1071 003500 004737 003560          JSR PC,FT2ER ;GO DO ERROR
1072 003504 013700 000532          FT2C: MOV DB,R0 ;GET ADDRESS OF DATA BUFFER
1073 003510 010102          MOV R1,R2
1074 003512 010110          MOV R1,(R0) ;LOAD PATTERN
1075 003514 012703 004000          MOV #4000,R3
1076 003520 005303          FT2D: DEC R3 ;DELAY
1077 003522 001376          BNE FT2D
1078 003524 020210          CMP R2,(R0) ;SEE IF EXPT=RCVD
1079 003526 001410          BEQ FT2E ;IF SO: BR
1080 003530 012737 016344 000650          MOV #MSG27,ERADD ;SET ERROR CODE
1081 003536 012737 003504 000674          MOV #FT2C,SCOLP ;SET SCOPE ADDRESS
1082 003544 004737 003560          JSR PC,FT2ER ;GO DO ERROR
1083 003550 005701          FT2E: TST R1 ;SEE IF DONE RESET
1084 003552 001453          BEQ FT2X ;IF SO: BR
1085 003554 005001          CLR R1 ;SET ZERO PATTERN
1086 003556 000712          BR FT2A ;DO ZERO BITS
1087 003560 000240          FT2ER: NOP
1088 003562 032777 020000 174760          BIT #20000,@SWR ;SEE IF PRINT ERROR
1089 003570 001034          BNE FT2ERB ;IF NOT: BR
1090 003572 005737 000606          TST HDRFL ;SEE IF DONE HEADER
1091 003576 001004          BNE FT2ERA ;IF SO: BR
1092 003600 013704 000610          MOV EMADDR,R4
1093 003604 004737 014316          JSR PC,TTOUT ;DO HEADER
1094 003610 012737 000001 000606          FT2ERA: MOV #1,HDRFL ;SET FLAG
1095 003616 013704 000650          MOV ERADD,R4
1096 003622 004737 014316          JSR PC,TTOUT ;PRINT ERROR CODE
1097 003626 012704 016274          MOV #MSG22,R4
1098 003632 004737 014316          JSR PC,TTOUT ;PRINT EXPT TAG
1099 003636 010103          MOV R1,R3
1100 003640 004737 014434          JSR PC,OCTPE ;PRINT EXPT
1101 003644 012704 016304          MOV #MSG23,R4
1102 003650 004737 014316          JSR PC,TTOUT ;PRINT RCVD TAG
1103 003654 011003          MOV (R0),R3
1104 003656 004737 014434          JSR PC,OCTPE ;PRINT RCVD

```

1105	003662	005777	174662	FT2ERB:	TST	@SWR		;SEE IF HALT ON ERROR
1106	003666	100001			BPL	FT2ERC		;IF NOT: BR
1107	003670	000000			HALT			
1108	003672	004737	013270	FT2ERC:	JSR	PC,SCOPE		;GO SEE IF SCOPE ON ERROR
1109	003676	000240			NOP			
1110	003700	000207			RTS	PC		;IF NO SCOPE: CONTINUE TEST
1111	003702	000240		FT2X:	NOP			
1112	003704	004737	013324		JSR	PC,ITER		;GO SEE IF ITERATIONS
1113	003710	000137	003070		JMP	TSCD2		;RETURN TO SCHEDULAR

```

1114
1115
1116 ;RH INITIALIZE TEST*****
1117 003714 012737 020210 000610 FT3: MOV #MSFT3,EMADDR ;SET TEST HEADER
1118 003722 012737 003714 000674 MOV #FT3,SCOLP
1119 003730 004737 013474 JSR PC,INIT1 ;GO INIT
1120 003734 052777 020000 174556 BIS #20000,@CS ;FORCE UPE =1
1121 003742 000240 NOP
1122 003744 004737 013474 JSR PC,INIT1 ;GO INIT
1123 003750 005777 174534 TST @C1 ;SEE IF SC IS RESET
1124 003754 100005 BPL FT3A ;IF SO: BR
1125 003756 012737 016402 000650 MOV #MSG29,ERADD ;SET ERROR CODE
1126 003764 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1127 003770 032777 040000 174512 FT3A: BIT #40000,@C1 ;SEE IF TRE IS RESET
1128 003776 001405 BEQ FT3B ;IF SO: BR
1129 004000 012737 016431 000650 MOV #MSG30,ERADD ;SET ERROR CODE.
1130 004006 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1131 004012 017701 174502 FT3B: MOV @CS,R1 ;GET CS2
1132 004016 042701 000307 BIC #307,R1 ;MARK IR/OR
1133 004022 005701 TST R1 ;SEE IF RESET
1134 004024 001405 BEQ FT3X ;IF SO: BR
1135 004026 012737 016461 000650 MOV #MSG31,ERADD ;SET ERROR CODE
1136 004034 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1137 004040 004737 013324 FT3X: JSR PC,ITER ;GO SEE IF ITERATION
1138 004044 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1139
1140 ;ERROR REPORT SUBROUTINE
1141 004050 000240 FT3ER: NOP
1142 004052 032777 020000 174470 BIT #20000,@SWR ;SEE IF PRINT ERROR
1143 004060 001015 BNE 2$ ;IF NOT: BR
1144 004062 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1145 004066 001006 BNE 1$ ;IF SO: BR
1146 004070 013704 000610 MOV EMADDR,R4
1147 004074 004737 014316 JSR PC,TTOUT ;PRINT HEADER
1148 004100 005237 000606 INC HDRFL
1149 004104 013704 000650 1$: MOV ERADD,R4
1150 004110 004737 014316 JSR PC,TTOUT ;PRINT ERROR CODE
1151 004114 005777 174430 2$: TST @SWR ;SEE IF HALT ON ERROR
1152 004120 100001 BPL 3$ ;IF NOT: BR
1153 004122 000000 HALT
1154 004124 000240 3$: NOP
1155 004126 004737 013270 JSR PC,SCOPE ;GO SEE IF SCOPE
1156 004132 000207 RTS PC ;IF NOT: BR
  
```

```
1157
1158 ;RH11 SILO TEST 1: EPMTY SILO READ*****
1159
1160 004134 005737 000604 FT4: TST RH17F
1161 004140 001141 BNE FT5X ;IF RH70: BR
1162 004142 012737 020242 000610 MOV #MSFT4,EMADDR ;SET TEST TEST HEADER
1163 004150 012777 000040 174342 MOV #40,@CS ;INIT
1164 004156 017700 174350 MOV @DB,R0 ;READ DB
1165 004162 005777 174332 TST @CS ;SEE IF DLT IS SET
1166 004166 100013 BPL FT4ER ;IF NOT: BR
1167 004170 005777 174314 TST @C1 ;SEE IF SC IS SET
1168 004174 100014 BPL FT4ERA ;IF NOT: BR
1169 004176 032777 040000 174304 BIT #40000,@C1 ;SEE IF TRE IS SET
1170 004204 001414 BEQ FT4ERB ;IF NOT: BR
1171 004206 004737 013324 FT4X: JSR PC,ITER ;GO SEE IF ITERATION
1172 004212 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1173 004216 012737 016511 000650 FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE
1174 004224 000407 BR FT4ERC
1175 004226 012737 016527 000650 FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE
1176 004234 000403 BR FT4ERC
1177 004236 012737 016544 000650 FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.
1178 004244 000240 FT4ERC: NOP
1179 004246 012737 004134 000674 MOV #FT4,SCOLP ;SET SCOPE ADDRESS
1180 004254 004737 004050 JSR PC,FT3ER ;GO PRINT ERROR
1181 004260 000752 BR FT4X
```

```
1182
1183
1184
1185 004262 005737 000604 FT5: TST RH17F ;SEE IF RH70
1186 004266 001066 BNE FT5X ;IF SO: BR
1187 004270 012737 020272 000610 MOV #MSFT5,EMADDR ;SET TEST HEADER
1188 004276 012737 004304 000674 MOV #FT5A,SCOLP ;SET SCOPE ADDRESS
1189 004304 004737 013474 FT5A: JSR PC,INIT1 ;GO INIT
1190 004310 032777 000100 174202 BIT #100,@CS ;SEE IF IR IS SET
1191 004316 001005 BNE FT5B ;IF SO: BR
1192 004320 012737 016562 000650 MOV #MSG35,ERADD ;SET ERROR CODE
1193 004326 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1194 004332 032777 000200 174160 FT5B: BIT #200,@CS ;SEE IF OR IS RESET
1195 004340 001405 BEQ FT5C ;IF SO: BR
1196 004342 012737 016607 000650 MOV #MSG36,ERADD ;SET ERROR CODE
1197 004350 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1198 004354 012777 000000 174150 FT5C: MOV #0,@DB ;LOAD ZERO INTO SILO
1199 004362 032777 000200 174130 BIT #200,@CS ;SEE THAT OR RESET
1200 004370 001405 BEQ FT5D ;IF IT DOES: BR
1201 004372 012737 016636 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1202 004400 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1203 004404 012777 177777 174120 FT5D: MOV #-1,@DB ;LOAD SILO WITH -1
1204 004412 012700 004000 MOV #4000,R0
1205 004416 032777 000200 174074 FT5E: BIT #200,@CS ;SEE IF OR IS SET
1206 004424 001007 BNE FT5X ;IF SO: BR
1207 004426 005300 DEC R0
1208 004430 001372 BNE FT5E ;AWAIT OR
1209 004432 012737 016636 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1210 004440 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1211 004444 004737 013324 FT5X: JSR PC,ITER ;GO SEE IF ITERATION
1212 004450 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
1213
1214 ;RH11 SILO TEST 3: SILO DATA TEST*****
1215
1216 004454 005737 000604 FT6: TST RH17F
1217 004460 001052 BNE FT6X ;IF RH70: BR
1218 004462 012737 020322 000610 MOV #MSFT6,EMADDR ;SET TEST HEADER
1219 004470 012737 004476 000674 MOV #FT6A,SCOLP ;SET SCOPE ADDRESS
1220 004476 004737 013474 FT6A: JSR PC,INIT1 ;GO INIT
1221 004502 005000 CLR R0 ;PRESET DATA
1222 004504 010077 174022 FT6B: MOV R0,@DB ;LOAD SILO
1223 004510 005200 INC R0 ;BUMP DATA
1224 004512 022700 000102 CMP #102,R0 ;SEE IF FILLED ALL
1225 004516 001372 BNE FT6B ;IF NOT: BR
1226 004520 032777 000100 173772 BIT #100,@CS ;SEE IF IR IS RESET.
1227 004526 001405 BEQ FT6C ;IF SO: BR
1228 004530 012737 016747 000650 MOV #MSG40,ERADD ;SET ERROR CODE
1229 004536 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1230 004542 032777 000200 173750 FT6C: BIT #200,@CS ;SEE IF OR IS SET
1231 004550 001005 BNE FT6D ;IF SO: BR
1232 004552 012737 016675 000650 MOV #MSG38,ERADD ;SET ERROR CODE
1233 004560 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1234 004564 005000 FT6D: CLR R0 ;PRESET DATA
1235 004566 017701 173740 FT6E: MOV @DB,R1 ;READ SILO
1236 004572 020001 CMP R0,R1 ;SEE IF EXPT=RCVD
1237 004574 001010 BNE FT6DE ;IF NOT: BR
1238 004576 005200 INC R0 ;BUMP DATA
1239 004600 022700 000102 CMP #102,R0 ;SEE IF DONE ALL
1240 004604 001370 BNE FT6E ;IF NOT: BR
1241 004606 004737 013324 FT6X: JSR PC,ITER ;GO SEE IF ITERATION
1242 004612 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1243
1244 004616 000240 FT6DE: NOP
1245 004620 032777 020000 173722 BIT #20000,@SWR ;SEE IF PRINT ERROR
1246 004626 001032 BNE FT6DEB ;IF NOT: BR
1247 004630 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1248 004634 013701 000610 MOV EMADDR,R1
1249 004640 004737 014316 JSR PC,TTOUT ;PRINT HEADER
1250 004644 005237 000606 INC HDRFL ;SET FLAG
1251 004650 012704 016727 FT6DEA: MOV #MSG39,R4
1252 004654 004737 014316 JSR PC,TTOUT ;PRINT SILO READ ERROR
1253 004660 012704 016274 MOV #MSG22,R4
1254 004664 004737 014316 JSR PC,TTOUT ;PRINT EXPT TAG
1255 004670 010003 MOV R0,R3
1256 004672 004737 014446 JSR PC,OCTP ;PRINT EXPT
1257 004676 012704 016304 MOV #MSG23,R4
1258 004702 004737 014316 JSR PC,TTOUT ;PRINT RCVD TAG
1259 004706 010103 MOV R1,R3
1260 004710 004737 014446 JSR PC,OCTP ;PRINT RCVD
1261 004714 005777 173630 FT6DEB: TST @SWR ;SEE IF HALT ON ERROR
1262 004720 100001 BPL FT6DEX ;IF NOT: BR
1263 004722 000000 HALT
1264 004724 000207 FT6DEX: RTS PC ;RETURN TO TEST
```

```
1265
1266 ;RH11 SILO TEST 4: SILO OVERFLOW*****
1267
1268 004726 005737 000604 FT7: TST RH17F
1269 004732 001021 BNE FT7X ;IF RH70: BR
1270 004734 012737 020352 000610 MOV #MSFT7,EMADDR ;SET TEST HEADER
1271 004742 012737 004726 000674 MOV #FT7,SCOLP ;SET SCOPE ADDRESS
1272 004750 004737 013474 JSR PC,INIT1 ;GO INIT
1273 004754 012700 000103 MOV #103,R0 ;SET SIZE OF SILO +1
1274 004760 010077 173546 FT7A: MOV R0,@DB ;LOAD SILO
1275 004764 005300 DEC R0 ;SEE IF DONE
1276 004766 001374 BNE FT7A ;IF NOT: BR
1277 004770 005777 173524 TST @CS ;SEE IF DLT IS SET
1278 004774 100004 BPL FT7ER ;IF NOT: BR
1279 004776 004737 013324 FT7X: JSR PC,ITER ;GO SEE IF ITERATION
1280 005002 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1281 005006 012737 016511 000650 FT7ER: MOV #MSG32,ERADD ;SET ERROR CODE
1282 005014 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1283 005020 000766 BR FT7X
```

```
1284
1285
1286
1287 005022 005737 000604          FT10:  TST      RH17F
1288 005026 001034                    BNE      FT10X      ;IF RH70: BR
1289 005030 012737 020402 000610    MOV      #MSFT10,EMADDR ;SET TEST HEADER
1290 005036 012737 005022 000674    MOV      #FT10,SCOLP   ;SET SCOPE ADDRESS
1291 005044 012777 000040 173446    MOV      #4,@CS       ;INITIALIZE
1292 005052 012700 000004          MOV      #4,R0        ;SET NUMBER OF SILO WRITER
1293 005056 010077 173450          FT10A: MOV      R0,@DB    ;WRITE SILO
1294 005062 005300                    DEC      R0           ;SEE IF DONE
1295 005064 001374                    BNE      FT10A       ;IF NOT: BR
1296 005066 052777 000040 173424    BIS      #4,@CS       ;INITIALIZE
1297 005074 012777 177777 173430    MOV      #-1,@DB     ;WRITE SILO
1298 005102 017701 173424          MOV      @DB,R1      ;READ SILO 1
1299 005106 017701 173420          MOV      @DB,R1      ;READ SILO 2
1300 005112 005777 173402          TST      @CS         ;SEE IF DLT IS SET
1301 005116 100011                    BPL      FT10ER      ;IF NOT: BR
1302 005120 004737 013324          FT10X: JSR      PC,ITER ;GO SEE IF ITERATION
1303 005124 005737 000726          TST      RHOF        ;SEE IF RH11 ONLY
1304 005130 001402                    BEQ      FT10XX      ;IF NOT: BR
1305 005132 000137 003156          JMP      TEND        ;ELSE GO TO END
1306 005136 000137 003070          FT10XX: JMP      TSCD2    ;RETURN TO SCHEDULAR
1307 005142 012737 016511 000650    FT10ER: MOV      #MSG32,ERADD ;SET ERROR CODE
1308 005150 004737 004050          JSR      PC,FT3ER   ;GO DO ERROR
1309 005154 000761                    BR       FT10X
```

```
1310 ;NOP TEST*****
1311
1312 005156 000240 FT11: NOP
1313 005160 012737 005156 000674 MOV #FT11,SCOLP ;SET SCOPE ADDRESS
1314 005166 004737 013474 JSR PC,INIT1
1315 005172 012737 000300 000716 MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
1316 005200 012737 177777 000620 MOV #-1,FCNT ;SET FC= ALL OVER
1317 005206 012737 177777 000622 MOV #-1,WCNT ;SET WC= ALL OVER
1318 005214 012737 177777 000616 MOV #-1,BADDR ;SET BA= ALL OVER
1319 005222 012737 000001 000636 MOV #1,RDYDX ;SET DELAY
1320 005230 012737 000001 000640 MOV #1,OPDYX ;SET OP DELAY
1321 005236 012737 000001 000710 MOV #1,FUN ;SET NOP FUNCTIONS CODE
1322 005244 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1323 005250 000240 NOP
1324 005252 012737 020433 000610 MOV #MSFT11,EMADDR
1325 005260 004737 012504 JSR PC,ERCHK ;GO CHECK REGISTER
1326 005264 004737 013324 JSR PC,ITER ;GO SEE IF ITERATIONS
1327 005270 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
1328                                     ;REWIND TEST*****
1329
1330 005274 000240          FT12:  NOP
1331 005276 012737 005274 000674  MOV    #FT12,SCOLP
1332 005304 004737 013474          JSR    PC,INIT1      ;GO INITIALIZE
1333 005310 052777 001700 173224  BIS    #1700,@TC     ;SET TO NRZ,NORMAL
1334 005316 012737 177760 000620  MOV    #-20,FCNT    ;SET FC=20
1335 005324 012737 177770 000622  MOV    #-10,WCNT    ;SET WC=10
1336 005332 012737 021310 000616  MOV    #WDATA,BADDR ;SET BA=WRITE BUFFER
1337 005340 012737 000007 000710  MOV    #7,FUN       ;SET REWIND OP CODE
1338 005346 004737 012304          JSR    PC,EXEC       ;GO EXECUTE COMMAND
1339 005352 000240          NOP
1340 005354 032777 020000 173140  FT12A: BIT    #20000,@DS
1341 005362 001374          BNE    FT12A        ;AWAIT PIP
1342 005364 012737 020453 000610  MOV    #MSFT12,EMADDR
1343 005372 004737 012504          JSR    PC,ERCHK     ;GO CHECK FOR ERROR
1344 005376 004737 013324          JSR    PC,ITER      ;GO SEE IF ITERATION
1345 005402 000137 003070          JMP    TSCD2        ;RETURN TO SCHEDULAR
1346
```

```

1347                                     ;WRITE/READ TEST*****
1348
1349 005406 000240          FT13:  NOP
1350 005410 012737 000001 000636  MOV    #1,RDYDX
1351 005416 012737 000001 000640  MOV    #1,OPDYX
1352 005424 012737 000100 000624  MOV    #100,RCNT      ;SET RECORD COUNT
1353 005432 012737 020476 000610  MOV    #MSFT13,EMADDR ;SET TEST HEADER
1354 005440 012737 000001 000720  MOV    #1,PATRN
1355 005446 004737 013156          JSR    PC,DSUP        ;SET UP ALL ONES DATA PATTERN
1356 005452 012737 001700 000716  MOV    #1700,UDES     ;SET TO 800 BP: NORMAL
1357 005460 004737 012436          FT13A: JSR    PC,RWIND      ;GO REWIND
1358 005464 012737 177600 000620  MOV    #-200,FCNT     ;SET FC
1359 005472 012737 177700 000622  MOV    #-100,WCNT     ;SET WC
1360 005500 012737 021310 000616  MOV    #WDATA,BADDR  ;SET BA
1361 005506 012737 000061 000710  MOV    #61,FUN        ;SET WRITE OP-CODE
1362 005514 012737 016043 000626  MOV    #MSG12,ERRP
1363 005522 004737 012304          FT13B: JSR    PC,EXEC    ;GO EXECUTE COMMAND
1364 005526 005037 000674          CLR    SCOLP         ;NO SCOPE LOOP
1365 005532 004737 012504          JSR    PC,ERCHK      ;GO CHECK ERROR
1366 005536 005337 000624          DEC    RCNT          ;SEE IF DONE ALL
1367 005542 001367          BNE    FT13B        ;IF NOT: BR
1368 005544 012737 000100 000624  MOV    #100,RCNT     ;SET RECORD COUNT
1369 005552 012737 023022 000616  MOV    #RDATA,BADDR
1370 005560 062737 000200 000616  ADD    #200,BADDR    ;SET BA
1371 005566 012737 000077 000710  MOV    #77,FUN       ;SET READ REVERSE OP-CPDE
1372 005574 012737 016061 000626  MOV    #MSG13,ERRP
1373 005602 004737 012304          FT13C: JSR    PC,EXEC    ;GO EXECUTE COMMAND
1374 005606 004737 012504          JSR    PC,ERCHK      ;GO CHECK ERROR
1375 005612 005337 000624          DEC    RCNT          ;SEE IF READ ALL
1376 005616 001371          BNE    FT13C        ;IF NOT:BR
1377 005620 162737 000200 000616  SUB    #200,BADDR    ;SET BA
1378 005626 012737 000071 000710  MOV    #71,FUN       ;SET READ FORWARD OP-CODE
1379 005634 012737 016106 000626  MOV    #MSG14,ERRP
1380 005642 012737 000100 000624  MOV    #100,RCNT     ;SET RECORD COUNT
1381 005650 004737 012304          FT13D: JSR    PC,EXEC    ;GO EXECUTE COMMAND
1382 005654 004737 012504          JSR    PC,ERCHK      ;GO CHECK ERRORS
1383 005660 005337 000624          DEC    RCNT          ;SEE IF DONE ALL
1384 005664 001371          BNE    FT13D        ;IF NOT:BR
1385 005666 032737 002000 000716  BIT    #2000,UDES     ;SEE IF DONE PE
1386 005674 001007          BNE    FT13X        ;IF SO: BR
1387 005676 012737 002300 000716  MOV    #2300,UDES    ;SET PE MODE
1388 005704 012737 000100 000624  MOV    #100,RCNT     ;RESET RECORD COUNT
1389 005712 000662          BR     FT13A        ;GO DO NEXT DENSITY
1390 005714 000137 003070          FT13X: JMP    TSCD2   ;RETURN TO SCHEDULAR

```

```

1391                                     ;SPACE TEST*****
1392
1393 005720 000240 FT14:  NOP
1394 005722 012737 020525 000610  MOV #MSFT14,EMADDR ;SET TEST HEADER
1395 005730 012737 001700 000716  MOV #1700,UDES ;SET NRZ,NORMAL
1396 005736 004737 012436 FT14A1: JSR PC,RWND ;GO INITIALIZE
1397 005742 012737 000100 000624  MOV #100,RCNT ;SET NUMBER OF RECORDER
1398 005750 012737 177777 021310  MOV #-1,WDATA ;SET DATA PATTERN
1399 005756 012737 177700 000620  MOV #-100,FCNT ;PRESET FRAME CNT
1400 005764 012737 177740 000622  MOV #-40,WCNT ;PRESET WORD CNT
1401 005772 004737 013474 FT14A:  JSR PC,INIT1 ;GO REWIND
1402 005776 012737 001000 000640  MOV #1000,OPDYX
1403 006004 012737 040000 000636  MOV #40000,RDYDX
1404 006012 012737 000061 000710  MOV #61,FUN ;SET WRITE OP-CODE
1405 006020 012737 102300 000660  MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1406 006026 052777 000010 172464  BIS #10,@CS ;INHIBIT BUS ADDRESS INCREMENT
1407 006034 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1408 006040 012737 017166 000626  MOV #MSG46,ERRP ;SET ERROR CODE
1409 006046 004737 012504 JSR PC,ERCHK ;GO CHECK ERRORS
1410 006052 005737 000712 TST SERFL ;SEE IF ERROR
1411 006056 001402 BEQ FT14A2 ;IF NOT: BR
1412 006060 000137 006544 JMP FT14X ;ELSE EXIT
1413 006064 005337 000620 FT14A2: DEC FCNT ;BUMP FC
1414 006070 032737 000001 000620  BIT #1,FCNT ;SEE IF SHOULD BUMP WC
1415 006076 001403 BEQ FT14A3 ;IF NOT: BR
1416 006100 162737 000001 000622  SUB #1,WCNT ;BUMP WC
1417 006106 005337 000624 FT14A3: DEC RCNT ;SEE IF DONE ALL
1418 006112 001327 BNE FT14A ;WRITE ALL RECORDS
1419 006114 012737 000100 000632  MOV #100,RRD ;PRESET RECORD POSITION
1420 006122 012737 000176 000634  MOV #176,RFD
1421 006130 012737 177701 000642  MOV #-77,SCNT ;SET SPACE AMOUNT
1422 006136 012737 000033 000710  FT14B: MOV #33,FUN ;SET OP-CODE SPACE REVERSE
1423 006144 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1424 006150 012737 017237 000626  MOV #MSG48,ERRP ;SET ERROR CODE
1425 006156 004737 012504 JSR PC,ERCHK ;GO CHECK ERRORS
1426 006162 005737 000712 TST SERFL ;SEE IF ERROR
1427 006166 001166 BNE FT14X ;IF SO: BR
1428 006170 004737 006264 JSR PC,FT14RR ;GO READ REVERSE + CHECK DATA
1429 006174 000240 NOP
1430 006176 012737 000031 000710  MOV #31,FUN ;SET SPACE FORWARD OP-CODE
1431 006204 005237 000642 INC SCNT ;SET SPACE AMOUNT
1432 006210 001555 BEQ FT14X ;IF DONE: BR
1433 006212 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1434 006216 012737 017212 000626  MOV #MSG47,ERRP ;SET ERROR CODE
1435 006224 004737 012504 JSR PC,ERCHK ;GO CHECK ERROR
1436 006230 005737 000712 TST SERFL ;SEE IF ERROR FLAG
1437 006234 001143 BNE FT14X ;IF NO: BR
1438 006236 004737 006326 JSR PC,FT14RF ;GO READ FORWARD FOR POSITION CHECK
1439 006242 000240 NOP
1440 006244 005237 000642 INC SCNT ;DECREMENT SPACE AMOUNT
1441 006250 001535 BEQ FT14X ;IF DONE: BR
1442 006252 005237 000632 INC RRD ;BUMP DATA EXPT
1443 006256 005337 000634 DEC RFD ;BUMP DATA EXPT
1444 006262 000725 BR FT14B
1445 006264 000240 FT14RR: NOP
1446 006266 012737 023022 000616  MOV #RDATA,BADDR ;SET RA
  
```

1447	006274	012737	000077	000710	MOV	#77,FUN	:SET READ REVERSE OP-CODE
1448	006302	004737	012304		JSR	PC,EXEC	:GO EXECUTE COMMAND
1449	006306	000240			NOP		
1450	006310	013705	000632		MOV	RRD,R5	
1451	006314	020577	172176		CMP	R5,@FC	:SEE IF CORRECT RECORD
1452	006320	001020			BNE	FT14RER	:IF NOT: BR
1453	006322	000137	006354		JMP	FT14EC	:GO CLEAR RH11 ERROR BIT
1454	006326	000240			FT14RF: NOP		
1455	006330	012737	000071	000710	MOV	#71,FUN	:SET READ FORWARD OP-CODE
1456	006336	004737	012304		JSR	PC,EXEC	:GO EXECUTE COMMAND
1457	006342	013705	000634		MOV	RFD,R5	
1458	006346	020577	172144		CMP	R5,@FC	:SEE IF CORRECT RECORD
1459	006352	001003			BNE	FT14RER	:IF NOT: BR
1460	006354	004737	013474		FT14EC: JSR	PC,INIT1	:CLEAR RH
1461	006360	000207			RTS	PC	:RETURN
1462	006362	000240			FT14RER: NOP		
1463	006364	032777	020000	172156	BIT	#2000,@SWR	:SEE IF PRINT INHIBITED
1464	006372	001060			BNE	FT14R3	:IF SO: BR
1465	006374	012704	020525		MOV	#MSFT14,R4	
1466	006400	004737	014316		JSR	PC,TTOUT	:PRINT HEADER
1467	006404	012704	015621		MOV	#MSG9,R4	
1468	006410	004737	014316		JSR	PC,TTOUT	:PRINT ERROR TYPE
1469	006414	012704	016261		MOV	#MSG20,R4	:SET NRZ TAG POINTER
1470	006420	032737	002000	000716	BIT	#2000,UDES	:SEE IF PE
1471	006426	001402			BEQ	FT14R0	:IF NOT: BR
1472	006430	012704	016267		MOV	#MSG21,R4	:ELSE SET PE TAG POINTER
1473	006434	004737	014316		FT14R0: JSR	PC,TTOUT	:PRINT TAG
1474	006440	032737	000002	000710	BIT	#2,FUN	:SEE IF READ REVERSE
1475	006446	001003			BNE	FT14R1	:IF SO: BR
1476	006450	012704	016241		MOV	#MSG17,R4	
1477	006454	000402			BR	FT14R2	:GO PRINT
1478	006456	012704	016221		FT14R1: MOV	#MSG16,R4	
1479	006462	004737	014316		FT14R2: JSR	PC,TTOUT	:PRINT FRWD/REV
1480	006466	012704	016274		MOV	#MSG22,R4	
1481	006472	004737	014316		JSR	PC,TTOUT	:PRINT EXPT TAG
1482	006476	010503			MOV	R5,R3	
1483	006500	042703	177700		BIC	#177700,R3	:MASK RECORD NUMBER
1484	006504	004737	014446		JSR	PC,OCTP	:PRINT EXPT RECORD NUMBER
1485	006510	012704	016304		MOV	#MSG23,R4	
1486	006514	004737	014316		JSR	PC,TTOUT	:PRINT RCVD TAG
1487	006520	017703	171772		MOV	@FC,R3	
1488	006524	042703	177700		BIC	#177700,R3	:MASK RECORD NUMBER
1489	006530	004737	014446		JSR	PC,OCTP	:PRINT ACTUAL RECORD NUMBER
1490	006534	005777	172010		FT14R3: TST	@SWR	:SEE IF HALT ON ERROR
1491	006540	100001			BPL	FT14X	:IF NOT: BR
1492	006542	000000			HALT		
1493	006544	032737	002000	000716	FT14X: BIT	#2000,UDES	:SEE IF DONE PE
1494	006552	001005			BNE	FT14XX	:IF SO: BR
1495	006554	012737	002300	000716	MOV	#2300,UDES	:SET TO PE
1496	006562	000137	005736		JMP	FT14A1	:DO IN PE
1497	006566	000137	003070		FT14XX: JMP	TSCD2	:RETURN TO SCHEDULAR

```
1498                                     ;ERASE TEST*****
1499
1500 006572 000240                      FT15:  NOP
1501 006574 005037 000660              CLR      STMSK
1502 006600 012737 000100 000636      MOV      #100,RDYDX
1503 006606 012737 000010 000640      MOV      #10,OPDYX
1504 006614 012737 020547 000610      MOV      #MSFT15,EMADDR ;SET TEST HEADER
1505 006622 004737 012436              JSR      PC,RWND ;REWIND
1506 006626 012737 023022 000616      MOV      #RDATA,BADDR ;SET BA
1507 006634 012737 001700 000716      MOV      #1700,UDES ;SET NRZ, NORMAL
1508 006642 012737 000025 000710 FT15A: MOV      #25,FUN ;SET ERASE OP-CODE
1509 006650 012737 000454 000624      MOV      #300,RCNT ;++B SET TO ERASE 300 TIMES
1510 006656 004737 012304              FT15B: JSR      PC,EXEC ;GO EXECUTE COMMAND
1511 006662 012737 017166 000626      MOV      #MSG46,ERRP ;SET ERROR CODE
1512 006670 004737 012504              JSR      PC,ERCHK ;GO CHECK ERRORS
1513 006674 005737 000712              TST      SERFL ;SEE IF ANY ERRORS
1514 006700 001032                      BNE      FT15X ;IF SO EXIT
1515 006702 005337 000624              DEC      RCNT ;SEE IF DONE ERASING
1516 006706 001363                      BNE      FT15B ;IF NOT: BR
1517 006710 000240                      NOP
1518 006712 004737 012436              JSR      PC,RWND ;REWIND
1519 006716 012737 177600 000622      MOV      #-200,WCNT ;SET WC
1520 006724 012737 000071 000710      MOV      #71,FUN ;SET READ FORWARD OP-CODE
1521 006732 012737 000040 000636      MOV      #40,RDYDX ;SET DELAY
1522 006740 004737 012304              JSR      PC,EXEC ;GO EXECUTE COMMAND
1523 006744 000240                      NOP
1524 006746 012737 017640 000626      MOV      #MSG60,ERRP ;SET ERROR CODE
1525 006754 012737 020000 000660      MOV      #20000,STMSK
1526 006762 004737 012504              JSR      PC,ERCHK ;GO CHECK ERRORS
1527                                     ;*****
1528
1529                                     ;THIS CODE ADDED TO FORM REV C
1530
1531                                     ;THE SSC BIT AND THE PIP BIT IN THE DRIVE STATUS REG
1532                                     ;SHOULD NOT BE SET CONCURRENTLY
1533
1534                                     ;*****
1535
1536 006766 012737 000100 000636 FT15X: MOV      #100,RDYDX ;SET DELAY
1537 006774 012737 000010 000640      MOV      #10,OPDYX
1538 007002 012737 000020 000624      MOV      #20,RCNT ;SET UP FOR 20 ERASES
1539 007010 012737 023022 000616 1$:  MOV      #RDATA,BADDR ;SET UP BUSS ADDRS
1540 007016 012737 001700 000716      MOV      #1700,UDES ;SET UP TAPE CONTROL
1541 007024 012737 000025 000710      MOV      #25,FUN ;SET FUN FOR ERASE
1542 007032 004737 012304              JSR      PC,EXEC ;GO EXECUTE CMD
1543 007036 005337 000624              DEC      RCNT ;DECREMENT THE NUMBER OF EXECUTES
1544 007042 001362                      BNE      1$ ;BRANCH IF MORE LEFT
1545 007044 012777 001700 171470      MOV      #1700,@TC
1546 007052 012777 177760 171436      MOV      #-20,@FC
1547 007060 012777 177770 171424      MOV      #-10,@WC
1548 007066 012777 021310 171420      MOV      #WDATA,@BA
1549 007074 012777 000007 171406      MOV      #7,@C1 ;DO REWIND
1550 007102 000240                      NOP
1551 007104 032777 000100 171410 2$:  BIT      #100,@DS ;WAIT FOR SSC
1552 007112 001774                      BEQ      2$
1553 007114 017737 171402 000652      MOV      @DS,TEMP1 ;READ DRIVE STATUS REG IMMEDIATELY
```

1554	007122	032737	020000	000652	BIT	#20000,TEMP1	:CHECK FOR PIP
1555	007130	001420			BEQ	FT15XX	:BRANCH IF NOT SET
1556	007132	052737	000001	007202	BIS	#1,TAG	:SET FLAG FOR ERROR
1557	007140	012704	016213		MOV	#MSG15B,R4	
1558	007144	004737	014316		JSR	PC,TTOUT	
1559	007150	010703			MOV	PC,R3	
1560	007152	062703	000010		ADD	#10,R3	
1561	007156	004737	014434		JSR	PC,OCTPE	
1562							
1563	007162	004737	012726		JSR	PC,ERPTB1	:GO PRINT ERROR
1564							:DS REG REPORTED IS ITS CONTENTS
1565							:AT THE TIME OF THE ERROR (5 LINES OF CODE BACK)
1566	007166	005037	007202		CLR	TAG	:CLEAR FLAG
1567	007172	004737	013324	FT15XX:	JSR	PC,ITER	:CHECK FOR ITERATIONS
1568	007176	000137	003070		JMP	TSCD2	:GO TO SCHEDULAR
1569							
1570	007202	000000		TAG:	.WORD	0	:++C FLAG FOR ERROR ROUTINE

```
1571                                     ;TAPE MARK WRITE/READ TEST*****
1572
1573 007204 000240 FT16: NOP
1574 007206 012737 000001 000636 MOV #1,RDYDX
1575 007214 012737 001000 000640 MOV #1000,OPDYX
1576 007222 012737 020571 000610 MOV #MSFT16,EMADDR ;SET HEADER
1577 007230 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ,NORMAL,ODD
1578 007236 004737 012436 FT16A: JSR PC,RWND ;INIT AND REWIND SLAVE
1579 007242 012737 177760 000620 FT16B: MOV #-20,FCNT ;FC=20
1580 007250 012737 177770 000622 MOV #-10,WCNT ;WC=10
1581 007256 012737 000027 000710 MOV #27,FUN ;SET WRITE TAPE MARK OP-CODE
1582 007264 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1583 007270 012737 001000 000660 MOV #1000,STMSK ;SET FOR FCE MASK
1584 007276 012737 016133 000626 MOV #MSG15,ERRP ;SET ERROR CODE
1585 007304 004737 012504 JSR PC,ERCHK ;GO CHECK ERROR
1586 007310 004737 013120 JSR PC,TMCHK ;GO SEE IF TM SET
1587 007314 012737 000077 000710 MOV #77,FUN ;SET READ REVERSE OP-CODE
1588 007322 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1589 007326 012737 001000 000660 MOV #1000,STMSK ;SET FCE ERROR MASK
1590 007334 012737 016061 000626 MOV #MSG13,ERRP ;SET ERROR CODE
1591 007342 004737 012504 JSR PC,ERCHK ;GO CHECK ERRORS
1592 007346 004737 013120 JSR PC,TMCHK ;GO SEE IF TM SET
1593 007352 012737 000071 000710 MOV #71,FUN ;SET READ FORWARD OP-CODE
1594 007360 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1595 007364 012737 016106 000626 MOV #MSG14,ERRP ;SET ERROR CODE
1596 007372 004737 012504 JSR PC,ERCHK ;TO CHECK ERRORS
1597 007376 004737 013120 JSR PC,TMCHK ;GO SEE IF TM SET
1598 007402 032737 002000 000716 BIT #2000,UDES ;SEE IF DONE PE
1599 007410 001004 BNE FT16X ;IF SO: BR
1600 007412 012737 002300 000716 MOV #2300,UDES ;SET PE, NORMAL
1601 007420 000706 BR FT16A ;DO IN PE
1602 007422 004737 013324 FT16X: JSR PC,ITER ;DO ITERATIONS
1603 007426 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1604
```

```
1605
1606 ;TAPE MARK SPACE TEST*****
1607
1608 007432 005037 000624 FT17: CLR RCNT
1609 007436 012737 020632 000610 MOV #MSFT17,EMADDR ;SET HEADER
1610 007444 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ
1611 007452 004737 012436 FT17A: JSR PC,RWIND ;REWIND TAPE
1612 007456 012737 000027 000710 FT17B: MOV #27,FUN
1613 007464 012737 040000 000636 MOV #40000,RDYDX ;SET DRY DELAY
1614 007472 012737 040000 000640 MOV #40000,OPDYX ;SET OP DELAY
1615 007500 004737 012304 JSR PC,EXEC ;GO WRITE TM
1616 007504 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1617 007512 012737 016133 000626 MOV #MSG15,ERRP ;SET ERROR TYPE
1618 007520 004737 012504 JSR PC,ERCHK ;GO CHECK ERROR
1619 007524 005737 000712 TST SERFL ;SEE IF ERROR
1620 007530 001137 BNE FT17X ;IF SO: BR
1621 007532 004737 013120 JSR PC,TMCHK ;GO SEE IF TM SET
1622 007536 000240 NOP
1623 007540 000240 NOP
1624 007542 032737 000100 000624 BIT #100,RCNT ;SEE IF DONE PATTERN
1625 007550 001045 BNE FT17D ;IF SO: BR
1626 007552 062737 000020 000624 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1627 007560 013737 000624 000652 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1628 007566 012737 177600 000622 MOV #-200,WCNT ;WC=128
1629 007574 012737 177400 000620 MOV #-400,FCNT ;FC=256
1630 007602 012737 021310 000616 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1631 007610 012737 000061 000710 MOV #61,FUN ;SET WRITE OP CODE
1632 007616 000240 FT17C: NOP
1633 007620 000240 NOP
1634 007622 004737 012304 JSR PC,EXEC ;GO WRITE
1635 007626 012737 016043 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1636 007634 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1637 007642 004737 012504 JSR PC,ERCHK ;GO CHECK ERROR
1638 007646 005737 000712 TST SERFL ;SEE IF ERROR
1639 007652 001066 BNE FT17X ;IF SO: BR
1640 007654 005337 000652 DEC TEMP1 ;SEE IF DONE ALL
1641 007660 001356 BNE FT17C ;IF NOT: BR
1642 007662 000675 BR FT17B ;ELSE GO DO TM
1643 007664 000240 FT17D: NOP
1644 007666 012737 000033 000710 MOV #33,FUN ;SET SPACE REVERSE
1645 007674 012737 016221 000626 MOV #MSG16,ERRP ;SET ERROR CODE
1646 007702 012737 177600 000642 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1647 007710 012737 000005 000624 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1648 007716 004737 013474 FT17E: JSR PC,INIT1 ;GO INIT
1649 007722 004737 012304 JSR PC,EXEC ;GO SPACE
1650 007726 012737 001000 000660 MOV #1000,STMSK ;SET ERROR MASK
1651 007734 004737 012504 JSR PC,ERCHK ;GO CHECK ERROR
1652 007740 005737 000712 TST SERFL ;SEE IF ERROR
1653 007744 001031 BNE FT17X ;IF SO: BR
1654 007746 004737 013120 JSR PC,TMCHK ;GO SEE IF TM SET
1655 007752 005337 000624 DEC RCNT ;SEE IF DONE SPACES
1656 007756 001357 BNE FT17E ;IF NOT: BR
1657 007760 022737 000031 000710 CMP #31,FUN ;SEE IF DONE FORWARD
1658 007766 001407 BEQ FT17F ;IF SO: BR
1659 007770 012737 016241 000626 MOV #MSG17,ERRP ;SET ERROR CODE
1660 007776 012737 000031 000710 MOV #31,FUN ;SET TO SPACE FORWARD
```

1661	010004	000736			BR	FT17D1	:DO FORWARD
1662	010006	032737	002000	000716	FT17F: BIT	#2000, UDES	:SEE IF DONE PE
1663	010014	001005			BNE	FT17X	:IF SO: BR
1664	010016	012737	002300	000716	MOV	#2300, UDES	:SET TO PE
1665	010024	000137	007452		JMP	FT17A	:GO PE
1666	010030	000137	003070		FT17X: JMP	TSCD2	:RETURN TO SCHEDULAR

```

1667
1668
1669
1670 010034 000240          FT20:  NOP
1671 010036 012737 020660 000610  MOV    #MSFT20,EMADDR  ;SET HEADER
1672 010044 012737 001700 000716  MOV    #1700,UDES      ;SET UNIT DESCRIPTION
1673 010052 004737 012436          FT20A: JSR    PC,RWIND       ;INIT AND REWIND SLAVE
1674 010056 012737 000003 000720  MOV    #3,PATRN
1675 010064 004737 013156          JSR    PC,DSUP         ;GO SET PATTERN 3
1676 010070 012737 021310 000616  MOV    #WDATA,BADDR   ;SET BA
1677 010076 012737 177400 000620  MOV    #-400,FCNT     ;SET FC
1678 010104 012737 177600 000622  MOV    #-200,WCNT     ;SET WC
1679 010112 012737 000061 000710  MOV    #61,FUN        ;SET WRITE OP CODE
1680 010120 004737 012304          JSR    PC,EXEC        ;GO WRITE RECORD
1681 010124 012737 017166 000626  MOV    #MSG46,ERRP    ;SET ERROR CODE
1682 010132 004737 012504          JSR    PC,ERCHK       ;GO CHECK ERROR
1683 010136 005737 000712          TST    SERFL          ;SEE IF ERORR
1684 010142 001042          BNE    FT20X          ;IF SO: BR
1685 010144 012737 016221 000626  MOV    #MSG16,ERRP    ;SET REVERSE ERROR TAG
1686 010152 012737 000057 000710  MOV    #57,FUN        ;SET REVERSE WRITE CHECK OP-CODE
1687 010160 062737 000376 000616  ADD    #376,BADDR     ;SET BA FOR REVERSE CHECK
1688 010166 004737 012304          JSR    PC,EXEC        ;GO DO REVERSE CHECK
1689 010172 004737 012504          JSR    PC,ERCHK       ;GO CHECK ERROR
1690 010176 012737 016241 000626  FT20B: MOV    #MSG17,ERRP    ;SET FORWARD TAG
1691 010204 012737 000051 000710  MOV    #51,FUN        ;SET FORWARD CHECK OP CODE
1692 010212 162737 000376 000616  SUB    #376,BADDR     ;SET BA FOR FORWARD CHECK
1693 010220 004737 012304          JSR    PC,EXEC        ;GO DO FORWARD CHECK
1694 010224 004737 012504          JSR    PC,ERCHK       ;GO CHECK ERROR
1695 010230 032737 002000 000716  FT20C: BIT    #2000,UDES  ;SEE IF DONE PE
1696 010236 001004          BNE    FT20X          ;IF SO: BR
1697 010240 012737 002300 000716  MOV    #2300,UDES    ;ELSE SET PE
1698 010246 000701          BR     FT20A          ;DO IN PE
1699 010250 004737 013324          FT20X: JSR    PC,ITER      ;DO ITERATIONS
1700 010254 000137 003070          JMP    TSCD2         ;RETURN TO SCHEDULAR
  
```

```

1701
1702
1703
1704 010260 012737 020711 000610 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1705 010266 004737 012436 FT21A: JSR PC,RWIND ;GO REWIND
1706 010272 012737 000003 000720 MOV #3,PATRN
1707 010300 004737 013156 JSR PC,DSUP ;GO SET PATTERN 3
1708 010304 012737 021310 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1709 010312 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1710 010320 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1711 010326 012737 001700 000716 MOV #1700,UDES ;SET NRZ, NORMAL
1712 010334 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1713 010342 004737 012304 JSR PC,EXEC ;GO DO WRITE 1
1714 010346 012737 016043 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1715 010354 004737 012504 JSR PC,ERCHK ;GO CHECK FOR ERROR
1716 010360 004737 012304 JSR PC,EXEC ;YES DO WRITE 2
1717 010364 004737 012504 JSR PC,ERCHK ;YES CHECK FOR ERROR
1718 010370 000240 NOP
1719 010372 004737 012436 JSR PC,RWIND ;GO REWIND
1720 010376 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400(10)
1721 010404 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200(10)
1722 010412 004737 012304 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1723 010416 000240 FT21SCP:NOP
1724 010420 004737 012436 JSR PC,RWIND ;REWIND
1725 010424 012737 023022 000616 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1726 010432 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400
1727 010440 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200
1728 010446 012737 000071 000710 MOV #71,FUN ;SET READ OP-CODE
1729 010454 004737 012304 JSR PC,EXEC ;GO READ RECORD 1
1730 010460 012737 016106 000626 MOV #MSG14,ERRP ;SET ERROR CODE
1731 010466 004737 012504 JSR PC,ERCHK ;GO CHECK FOR ERROR
1732 010472 000240 NOP
1733 010474 052777 000010 170016 BIS #10,@CS ;INHIBIT BA INCREMENT
1734 010502 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1735 010510 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1736 010516 004737 012304 JSR PC,EXEC ;GO READ RECORD 2
1737 010522 022777 001440 167766 CMP #800.,@FC ;SEE IF READ RECORD 2 OK
1738 010530 001424 BEQ FT21X ;IF SO: BR
1739 010532 022777 001441 167756 CMP #801.,@FC ;BRANCH IF IN GREY AREA
1740 010540 001420 BEQ FT21X
1741 010542 022777 001440 167746 1$: CMP #800.,@FC ;BRANCH IF ERASE HEAD REVERSED
1742 010550 101404 BLOS FT21B ;IF SO: BR
1743 010552 012737 017061 000650 MOV #MSG44,ERADD ;SET ERASE HEAD INOPERATIVE ERROR CODE
1744 010560 000403 BR FT21C
1745 010562 012737 017111 000650 FT21B: MOV #MSG45,ERADD ;SET ERASE HEAD REVERSED ERROR CODE
1746 010570 012737 010416 000674 FT21C: MOV #FT21SCP,SCOLP ;SET SCOPE ADDRESS
1747 010576 004737 004050 JSR PC,FT3ER ;GO PRINT ERROR
1748 010602 004737 013324 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1749 010606 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1750
1751

```

```
1752                                     ;BUFFERED COMMAND TEST*****
1753
1754 010612 012737 020740 000610 FT22: MOV #MSFT22,EMADDR ;SET TEST HEADER
1755 010620 004737 012436 JSR PC,RWIND ;GO REWIND
1756 010624 012700 000003 MOV #3,R0 ;SET NUMBER OF WRITES
1757 010630 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ NORMAL
1758 010636 012737 021310 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1759 010644 012737 177000 000620 MOV #-1000,FCNT ;SET FC=1000
1760 010652 012737 177400 000622 MOV #-400,WCNT ;SET WC=400
1761 010660 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1762 010666 004737 012304 FT22A: JSR PC,EXEC ;GO DO WRITE
1763 010672 005300 DEC R0 ;SEE IF DONE ALL
1764 010674 001374 BNE FT22A ;IF NOT: BR
1765 010676 000240 NOP
1766 010700 012777 000007 167602 MOV #7,@C1 ;START REWIND
1767 010706 032777 000200 167606 FT22B: BIT #200,@DS
1768 010714 001774 BEQ FT22B
1769 010716 004737 013474 JSR PC,INIT1 ;INITIALIZE
1770 010722 012737 000910 000636 MOV #10,RDYDX ;SET LONG READY DELAY
1771 010730 004737 012304 JSR PC,EXEC ;ISSUE BUFFERED WRITE
1772 010734 000240 NOP
1773 010736 012737 017264 000626 MOV #MSG49,ERRP ;SET ERROR CODE
1774 010744 012737 102300 000660 MOV #102300,STMSK ;MARK DATA ERROR
1775 010752 004737 012504 JSR PC,ERCHK ;GO CHECK ERROR
1776 010756 032777 000002 167536 BIT #2,@DS ;SEE IF BOT IS SET
1777 010764 001410 BEQ FT22X ;IF NOT: BR
1778 010766 012737 017312 000650 MOV #MSG50,ERADD ;SET ERROR CODE
1779 010774 012737 010612 000674 MOV #FT22,SCOLP
1780 011002 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1781 011006 004737 013324 FT22X: JSR PC,ITER ;GO SEE IF ITERATION
1782 011012 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1783
1784
```

```

1785                                     ;READ-IN PRESET TEST*****
1786
1787 011016 005737 000614          FT23: TST      SLVN          ;SEE IF SLAVE SELECT=0
1788 011022 001103                    BNE      FT23X        ;IF NOT:BR
1789 011024 012737 020775 000610    MOV      #MSFT23,EMADDR ;SET TEST HEADER
1790 011032 004737 013474          JSR      PC,INIT1     ;GO INIT
1791 011036 012737 001700 000716    MOV      #1700,UDES    ;SET TO NRZ NORMAL
1792 011044 012737 021310 000616    MOV      #WDATA,BADDR ;SET BA=WRITE BUFFER
1793 011052 012737 177400 000620    MOV      #-400,FCNT   ;SET FC=400
1794 011060 012737 177600 000622    MOV      #-200,WCNT   ;SET WC=200
1795 011066 012737 000061 000710    MOV      #61,FUN      ;SET WRITE OP-CODE
1796 011074 004737 012304          JSR      PC,EXEC      ;GO DO WRITE
1797 011100 000240                    NOP
1798 011102 004737 013474          JSR      PC,INIT1     ;INITIALIZE
1799 011106 012737 000021 000710    MCV      #21,FUN      ;SET READ-IN PRESET OP CODE
1800 011114 004737 012304          JSR      PC,EXEC      ;GO DO COMMAND
1801 011120 005000                    CLR      R0
1802 011122 012703 000004          MOV      #4,R3        ;SET MULT
1803 011126 032777 020000 167366    FT23A: BIT      #20000,@DS ;SEE IF PIP RESET
1804 011134 001404                    BEQ      FT23B        ;IF SO: BR
1805 011136 005300                    DEC      R0
1806 011140 001372                    BNE      FT23A        ;AWAIT PIP RESET
1807 011142 005303                    DEC      R3
1808 011144 001370                    BNE      FT23A        ;DELAY
1809 011146 032777 000002 167346    FT23B: BIT      #2,@DS  ;SEE IF BOT
1810 011154 001010                    BNE      FT23C        ;IF SO: BR
1811 011156 012737 017350 000650    MOV      #MSG51,ERADD ;SET ERROR CODE
1812 011164 012737 011016 000674    MOV      #FT23,SCOLP
1813 011172 004737 004050          JSR      PC,FT3ER     ;GO DO ERROR
1814 011176 012701 141000          FT23C: MOV      #141000,R1 ;SET EXPT TC
1815 011202 013700 000542          MOV      TC,R0        ;SET TC ADDRESS
1816 011206 020110                    CMP      R1,(R0)      ;SEE IF EXPT=RCVD
1817 011210 001410                    BEQ      FT23X        ;IF SO: BR
1818 011212 012737 017404 000650    MOV      #MSG52,ERADD ;SET ERROR CODE
1819 011220 012737 011016 000674    MOV      #FT23,SCOLP ;CLEAR SCOPE ADDRESS
1820 011226 004737 003560          JSR      PC,FT2ER     ;GO DO ERROR
1821 011232 000137 003070          FT23X: JMP      TSCD2  ;RETURN TO SCHEDULAR
1822
1823

```

```
1824
1825           ;AUTO-DENSITY SELECT TEST: WRITE-NRZ,READ-PE
1826
1827 011236 012737 021030 000610 FT24: MOV #MSFT24,EMADDR ;SET ERROR MSG HEADER
1828 011244 004737 012436 JSR PC,RWND ;REWIND SLAVE
1829 011250 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN
1830 011256 004737 013156 JSR PC,DSUP ;GO DO DATA SETUP
1831 011262 012737 021310 000616 MOV #WDATA,BADDR ;SET BUS ADDRESS,
1832 011270 012737 177400 000620 MOV #-400,FCNT ;FRAME COUNT,
1833 011276 012737 177600 000622 MOV #-200,WCNT ;WORD COUNT,
1834 011304 012737 001700 000716 MOV #1700,UDES ;& SLAVE DESC = NRZ NORMAL
1835 011312 012737 000061 00071C MOV #61,FUN ;LOAD OP CODE WRITE FWD
1836 011320 004737 012304 JSR PC,EXEC ;GO EXECUTE COMMAND
1837 011324 012737 017166 000626 MOV #MSG46,ERRP ;SET ERROR MSG ADDRESS
1838 011332 004737 012504 JSR PC,ERCHK ;GO CHECK ERRORS
1839 011336 005737 000712 TST SERFL ;BRANCH IF AN ERROR OCCURRED
1840 011342 001026 BNE FT24X
1841 011344 004737 012436 JSR PC,RWND ;REWIND SLAVE
1842 011350 012737 023022 000616 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1843 011356 012737 002300 000716 MOV #2300,UDES ;SET SLAVE DESC = PE,NORMAL
1844 011364 012737 000071 000710 MOV #71,FUN ;SET OP CODE = READ FWD
1845 011372 004737 012304 JSR PC,EXEC ;GO READ RECORD
1846 011376 032777 000040 167116 BIT #40,@DS ;BRANCH ID PES BIT CLEARED
1847 011404 001405 BEQ FT24X
1848 011406 012737 017737 000650 MOV #MSG63,ERADD
1849 011414 004737 004050 JSR PC,FT3ER ;GO PROCESS ERROR
1850 011420 004737 013324 FT24X: JSR PC,ITER
1851 011424 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULER
1852
```

```
1853
1854 ;AUTO-DENSITY SELECT TEST: WRITE-PE,READ-NRZ
1855 011430 012737 021106 000610 FT25: MOV #MSFT25,EMADDR ;SET ERROR MESSAGE ADDRESS
1856 011436 004737 012436 JSR PC,RWIND ;REWIND SLAVE
1857 011442 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN
1858 011450 004737 013156 JSR PC,DSUP ;GO DO DATA SETUP
1859 011454 012737 021310 000616 MOV #WDATA,BADDR ;SET BUS ADDRESS
1860 011462 012737 177400 000620 MOV #-400,FCNT ;FRAME COUNT,
1861 011470 012737 177600 000622 MOV #-200,WCNT ;WORD COUNT,
1862 011476 012737 002300 000716 MOV #2300,UDES ;& SLAVE DESC = PE,NORMAL
1863 011504 012737 000061 000710 MOV #61,FUN ;LOAD WRITE OP CODE
1864 011512 004737 012304 JSR PC,EXEC ;GO EXECUTE WRITE
1865 011516 012737 017166 000626 MOV #MSG46,ERRP ;SET ERROR MSG HDR
1866 011524 004737 012504 JSR PC,ERCHK ;GO CHECK FOR ERRORS
1867 011530 005737 000712 TST SERFL ;BRANCH IF ERROR OCURRED
1868 011534 001026 BNE FT25X
1869 011536 004737 012436 JSR PC,RWIND ;REWIND SLAVE
1870 011542 012737 023022 000616 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1871 011550 012737 001700 000716 MOV #1700,UDES ;SET SLAVE DESC = NRZ,NORMAL
1872 011556 012737 000071 000710 MOV #71,FUN ;SET READ FWD OP CODE
1873 011564 004737 012304 JSR PC,EXEC ;GO EXECUTE
1874 011570 032777 000040 166724 BIT #40,ADS ;BRANCH ID PES BIT GOT SET
1875 011576 001005 BNE FT25X
1876 011600 012737 017770 000650 MOV #MSG64,ERADD
1877 011606 004737 004050 JSR PC,FT3ER ;GO PROCESS ERROR
1878 011612 004737 013324 FT25X: JSR PC,ITER ;ITERATION LOOP
1879 011616 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULER
1880
```

```

1881                                     ;++B SEQUENTIAL TAPE MARK TEST
1882
1883 011622 000240                       FT26:  NOP
1884 011624 012737 021164 000610        MOV    #MSFT26,EMADDR ;SET TEST ERROR MSG HEADER
1885 011632 012737 001700 000716        MOV    #1700,UDES     ;SET NRZ
1886 011640 004737 012436                 1$:   JSR    PC,RWIND      ;REWIND SLAVE
1887 011644 012737 000027 000710        MOV    #27,FUN       ;SET WRITE TAPE MARK FUNCTION CODE
1888 011652 004737 012304                 JSR    PC,EXEC       ;GO DO TAPE MARK
1889 011656 005037 000660                 CLR    STMSK         ;CLEAR EXPECTED ERROR MASK
1890 011662 012737 016133 000626        MOV    #MSG15,ERRP   ;SET ERROR MESSAGE
1891 011670 004737 012504                 JSR    PC,ERCHK      ;GO CHECK FOR ERRORS
1892 011674 004737 013120                 JSR    PC,TMCHK      ;GO CHECK FOR TAPE MARK
1893 011700 005737 000712                 TST    SERFL         ;EXIT TEST IF ERROR DETECTED
1894 011704 001061                       BNE    FT26X
1895 011706 004737 012304                 JSR    PC,EXEC       ;WRITE SECOND TAPE MARK
1896 011712 012737 016154 000626        MOV    #MSG15A,ERRP  ;SET ERROR MESSAGE
1897 011720 004737 012504                 JSR    PC,ERCHK      ;GO CHECK ERROR
1898 011724 004737 013120                 JSR    PC,TMCHK
1899 011730 005737 000712                 TST    SERFL         ;EXIT TEST IF ERROR DETECTED
1900 011734 001045                       BNE    FT26X
1901 011736 004737 012436                 JSR    PC,RWIND      ;REWIND
1902 011742 012737 000031 000710        MOV    #31,FUN       ;SET SPACE FORWARD OP CODE
1903 011750 012737 177777 000642        MOV    #-1,SCNT      ;SET # OF RECORDS TO SPACE
1904 011756 004737 012304                 JSR    PC,EXEC       ;GO SPACE FORWARD
1905 011762 012737 017212 000626        MOV    #MSG47,ERRP   ;SET SPACE FORWARD ERROR
1906 011770 004737 012504                 JSR    PC,ERCHK      ;GO CHECK ERROR BITS
1907 011774 004737 013120                 JSR    PC,TMCHK      ;GO CHECK IF TAPE MARK DETECTED
1908 012000 005737 000712                 TST    SERFL         ;EXIT TEST IF ERROR DETECTED
1909 012004 001021                       BNE    FT26X
1910 012006 004737 012304                 JSR    PC,EXEC       ;SPACE TO SECOND TAPE MARK
1911 012012 004737 012504                 JSR    PC,ERCHK      ;GO CHECK ERROR BITS
1912 012016 004737 013120                 JSR    PC,TMCHK      ;CHECK IF TAPE MARK DETECTED
1913 012022 005737 000712                 TST    SERFL         ;EXIT TEST IF ERROR DETECTED
1914 012026 001010                       BNE    FT26X
1915 012030 032737 002000 000716        BIT    #2000,UDES    ;EXIT TEST IF PE COMPLETED
1916 012036 001004                       BNE    FT26X
1917 012040 012737 002300 000716        MOV    #2300,UDES    ;SET PE MODE
1918 012046 000674                       BR     1$
1919 012050 004737 013324                 FT26X: JSR   PC,ITER
1920 012054 000137 003070                 JMP    TSCD2
  
```

```

1921                                     ;REWIND: OFF LINE TEST*****
1922
1923 012060 032777 010000 166462 FT27: BIT #10000,@SWR ;SEE IF IN CONTINUOUS MODE
1924 012066 001104 BNE FT27XX ;IF SO: BR
1925 012070 005737 001662 TST CHNFLG ;BRANCH IF CHAIN MODE
1926 012074 001101 BNE FT27XX
1927 012076 012737 021225 000610 MOV #MSFT27,EMADDR ;SET TEST HEADER
1928 012104 004737 012436 JSR PC,RWIND ;REWIND & SELECT SLAVE
1929 012110 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN (ALL 1'S)
1930 012116 004737 013156 JSR PC,DSUP ;FILL WRITE BUFFER
1931 012122 012737 021310 000616 MOV #WDATA,BADDR ;SET WRITE BUFFER BUS ADDRESS
1932 012130 012737 177400 000620 MOV #-400,FCNT ;SET FRAME COUNT
1933 012136 012737 177600 000622 MOV #-200,WCNT ;SET WORD COUNT
1934 012144 012737 001700 000716 MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
1935 012152 012737 000061 000710 MOV #61,FUN ;SET WRITE COMMAND
1936 012160 004737 012304 JSR PC,EXEC ;GO WRITE A RECORD
1937 012164 004737 013474 JSR PC,INIT1 ;++B CLEAR ANY ERROR BITS
1938 012170 012777 000003 166312 MOV #3,@C1 ;ISSUE REWIND: OFF LINE COMMAND
1939 012176 005037 000674 CLR SCOLP ;CLEAR SCOPE LOOP
1940 012202 012700 000042 MOV #42,R0
1941 012206 005001 1$: CLR R1 ;CLEAR TIMER
1942 012210 005301 2$: DEC R1
1943 012212 001376 BNE 2$ ;IF NOT TIMED OUT: BR
1944 012214 005300 DEC R0
1945 012216 001373 BNE 1$ ;IF NOT ALL TIMED OUT: BR
1946 012220 032777 010000 166274 BIT #10000,@DS ;SEE IF MOL IS RESET
1947 012226 001406 BEQ 3$ ;IF SO: BR
1948 012230 012737 017423 000650 MOV #MSG53,ERADD ;SET ERROR CODE
1949 012236 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1950 012242 000412 BR FT27X
1951 012244 013700 000524 3$: MOV ER,R0 ;GET ADDRESS OF ERROR REG
1952 012250 005001 CLR R1 ;RESULT SHOULD BE 0
1953 012252 020110 CMP R1,(R0) ;BRANCH IF ERROR REG = 0
1954 012254 001405 BEQ FT27X
1955 012256 012737 020024 000650 MOV #MSG67,ERADD ;SET ERROR MSG HEADER
1956 012264 004737 003560 JSR PC,FT2ER ;GO TYPE ERROR
1957 012270 012704 017450 FT27X: MOV #MSG54,R4
1958 012274 004737 014316 JSR PC,TTOUT ;PRINT ON LINE REQUEST
1959 012300 000137 003070 FT27XX: JMP TSCD2 ;RETURN TO SCHEDULER
  
```

```

1960
1961
1962
1963 012304 000240
1964 012306 053777 000716 166226
1965 012314 013777 000622 166170
1966 012322 013777 000620 166166
1967 012330 013777 000616 166156
1968 012336 022737 000031 000710
1969 012344 001404
1970 012346 022737 000033 000710
1971 012354 001003
1972 012356 013777 000642 166132
1973 012364 000240
1974 012366 013777 000710 166114
1975 012374 000240
1976 012376 013703 000636
1977 012402 005004
1978 012404 032777 000200 166110
1979 012412 001004
1980 012414 005304
1981 012416 001372
1982 012420 005303
1983 012422 001370
1984 012424 013703 000640
1985 012430 005303
1986 012432 001376
1987 012434 000207
1988

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



```

2002                                     :ERROR CHECK SUBROUTINE*****
2003
2004 012504 005037 000712 ERCHK: CLR SERFL ;CLEAR FLAG
2005 012510 017737 166006 000664 MOV @DS,DSAV ;SAVE DRIVE STATUS REGISTER
2006 012516 032777 040000 165776 BIT #40000,@DS ;SEE IF ERROR
2007 012524 001001 BNE ERPT ;IF SO: BR
2008 012526 000207 RTS PC ;RETURN
2009 012530 017704 165770 ERPT: MOV @ER,R4 ;GET ERROR REGISTER
2010 012534 032737 002000 000716 BIT #2000,UDES ;SEE IF PE
2011 012542 001403 BEQ 2$ ;IF SO: BR
2012 012544 042737 000200 000660 BIC #200,STMSK ;RESET PEF MASK
2013 012552 022737 000003 000742 2$: CMP #3,JUMPER ;+TEST FOR NON-STANDARD JUMPER
2014 012560 001413 BEQ ERPTA1 ;+BRANCH IF STANDARD
2015 012562 022777 011236 166116 CMP #FT24,@LTADD ;+CHECK FOR TEST 24
2016 012570 001404 BEQ 1$ ;+BRANCH IF TST24
2017 012572 022777 011430 166106 CMP #FT25,@LTADD ;+CHECK FOR TEST 25
2018 012600 001003 BNE ERPTA1
2019 012602 052737 020000 000660 1$: BIS #20000,STMSK ;+SET OPI BIT IN ERROR MASK
2020 012610 043704 000660 ERPTA1: BIC STMSK,R4 ;MASK DONT CARE BITS
2021 012614 001536 BEQ ERPTX ;IF NO UNEXPECTED ERRORS: BR
2022 012616 012737 000001 000712 ERPTG: MOV #1,SERFL ;SET FLAG
2023 012624 032777 020000 165716 BIT #20000,@SWR ;SEE IF SHOULD PRINT ERRORS
2024 012632 001123 BNE ERPTD ;IF NOT: BR
2025 012634 005737 000606 TST HDRFL ;SEE IF DONE HEADER
2026 012640 001006 BNE ERPTA ;IF SO: BR
2027 012642 005237 000606 INC HDRFL ;SET HEADER FLAG
2028 012646 013704 000610 MOV EMADDR,R4
2029 012652 004737 014316 JSR PC,TTOUT ;PRINT HEADER
2030 012656 013704 000626 ERPTA: MOV ERRP,R4 ;GET ERROR CODE
2031 012662 001414 BEQ ERPTB ;IF NONE: BR
2032 012664 004737 014316 JSR PC,TTOUT ;PRINT ERROR CODE
2033 012670 012704 016261 MOV #MSG20,R4 ;SET NRZ TAG
2034 012674 032777 002000 165640 BIT #2000,@TC ;SEE IF PE
2035 012702 001402 BEQ ERPT1A ;IF NOT: BR
2036 012704 012704 016267 MOV #MSG21,R4 ;ELSE SET PE TAG
2037 012710 004737 014316 ERPT1A: JSR PC,TTOUT ;PRINT TAG
2038 012714 013704 000630 ERPTB: MOV ERRP1,R4 ;SEE IF CODE 2
2039 012720 001402 BEQ ERPTB1 ;IF NOT: BR
2040 012722 004737 014316 JSR PC,TTOUT ;PRINT CODE 2
2041 012726 032777 004000 165614 ERPTB1: BIT #4000,@SWR ;SEE IF ITERATION
2042 012734 001010 BNE ERPTC ;IF NOT: BR
2043 012736 012704 017614 MOV #MSG56,R4
2044 012742 004737 014316 JSR PC,TTOUT ;PRINT ITER TAG
2045 012746 013703 000662 MOV ITCNT,R3
2046 012752 004737 014446 JSR PC,OCTP ;PRINT ITERATION
2047 012756 012704 015240 ERPTC: MOV #MSG1,R4
2048 012762 004737 014316 JSR PC,TTOUT ;PRINT REGISTER TAG
2049 012766 017703 165516 MOV @C1,R3
2050 012772 004737 014434 JSR PC,OCTPE ;PRINT CS1
2051 012776 017703 165510 MOV @WC,R3
2052 013002 004737 014434 JSR PC,OCTPE ;PRINT WC
2053 013006 017703 165502 MOV @BA,R3
2054 013012 004737 014434 JSR PC,OCTPE ;PRINT BA
2055 013016 017703 165474 MOV @FC,R3
2056 013022 004737 014434 JSR PC,OCTPE ;PRINT FC
2057 013026 017703 165466 MOV @CS,R3
  
```

```
2058 013032 004737 014434      JSR    PC,OCTPE      ;PRINT CS2
2059 013036 005737 007202      TST    TAG           ;++C CHECK FOR SPECIAL DS
2060 013042 001403                BEQ    1$            ;
2061 013044 013703 000652      MOV    TEMP1,R3     ;++C PRINT DS READ INTO TEMP1 AT CRITICAL TIME
2062 013050 000402                BR     2$            ;
2063 013052 017703 165444      1$:  MOV    @DS,R3
2064 013056 004737 014434      2$:  JSR    PC,OCTPE      ;PRINT DS
2065 013062 017703 165436      MOV    @ER,R3
2066 013066 004737 014434      JSR    PC,OCTPE      ;PRINT ER
2067 013072 017703 165444      MOV    @TC,R3
2068 013076 004737 014434      JSR    PC,OCTPE      ;PRINT TC
2069 013102 005777 165442      ERPTD: TST    @SWR     ;SEE IF HALT ON ERROR
2070 013106 100001                BPL    ERPTX        ;IF NOT: BR
2071 013110 000000                HALT
2072 013112 004737 013474      ERPTX: JSR    PC,INIT1 ;INIT
2073 013116 000207                ERPTXX: RTS    PC     ;RETURN
2074
2075
```

```

2076                                     ;TAPE MARK STATUS CHECK*****
2077
2078 013120 032737 000004 000664 TMCHK: BIT #4,DSAV ;BRANCH IF TM SET
2079 013126 001012 BNE 1$
2080 013130 005737 000712 TST SERFL ;SEE IF HAD ERROR
2081 013134 001007 BNE 1$ ;IF SO: BR
2082 013136 012737 017624 000630 MOV #MSG57,ERRP1 ;SET ERROR CODE 2
2083 013144 004737 012616 JSR PC,ERPTG ;GO PRINT TM ERROR
2084 013150 005037 000630 CLR ERRP1 ;CLEAR CODE 2 FLAG
2085 013154 000207 1$: RTS PC ;RETURN
2086
2087                                     ;DATA SETUP ROUTINE*****
2088
2089 013156 000240 DSUP: NOP
2090 013160 012703 021310 DSO: MOV #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
2091 013164 013701 000720 MOV PATRN,R1 ;R1 = PATTERN SELECTOR
2092 013170 006301 ASL R1 ;MAKE PATTERN SELECTOR EVEN
2093 013172 004771 000744 JSR PC,@DATBL(R1) ;GO GENERATE PATTERN
2094 013176 012702 000640 MOV #640,R2 ;R2=BUFFER SIZE +2
2095 013202 012701 023022 MOV #RDATA,R1 ;R1=READ DATA START
2096 013206 005021 1$: CLR (R1)+ ;CLEAR BUFFER
2097 013210 005302 DEC R2 ;SEE IF DONE ALL
2098 013212 001375 BNE 1$ ;IF NOT: BR
2099 013214 000207 RTS PC ;EXIT
2100
2101                                     ;ALL ONES*****
2102
2103 013216 012701 177777 DAT1: MOV #-1,R1 ;R1=DATA
2104 013222 012702 000640 DAT1A: MOV #640,R2 ;R2=WORD COUNT +2
2105 013226 010123 1$: MOV R1,(R3)+ ;LOAD BUFFER
2106 013230 005302 DEC R2 ;SEE IF DONE
2107 013232 001375 BNE 1$ ;IF NOT: BR
2108 013234 000207 RTS PC
2109
2110                                     ;ALL ZEROS*****
2111
2112 013236 005001 DAT2: CLR R1 ;R1=DATA
2113 013240 000770 BR DAT1A ;LOAD BUFFER
2114
2115                                     ;ONE/ZERO IN ALTERNATING CHARACTERS*****
2116
2117 013242 012701 125125 DAT3: MOV #125125,R1 ;R1=DATA
2118 013246 000765 BR DAT1A ;LOAD BUFFER
2119
2120                                     ;ALL BITS 0-377*****
2121
2122 013250 005001 DAT4: CLR R1 ;R1=STARTING DATA
2123 013252 012702 001500 MOV #1500,R2 ;R2=CHARACTER COUNT
2124 013256 110123 1$: MOV#B R1,(R3)+ ;LOAD BUFFER
2125 013260 105201 INCB R1 ;BUMP DATA
2126 013262 005302 DEC R2 ;SEE IF DONE
2127 013264 001374 BNE 1$ ;IF NOT: BR
2128 013266 000207 RTS PC
2129

```

```

2130
2131
2132
2133           ;SCOPE LOOP ON ERROR SUBROUTINE*****
2134
2135 013270 000240          SCOPE:  NOP
2136 013272 032777 040000 165250  BIT    #40000,@SWR      ;SEE IF LOOP ON ERROR
2137 013300 001001          BNE    1$              ;IF SO: BR
2138 013302 000207          RTS    PC              ;ELSE EXIT
2139 013304 000240          1$:    NOP
2140 013306 005737 000674    TST    SCOLP          ;SEE IF SCOPE ADDRESS
2141 013312 001001          BNE    2$              ;IF NOT: BR
2142 013314 000207          RTS    PC              ;ELSE EXIT
2143 013316 022626          2$:    CMP    (SP)+,(SP)+    ;RESET STACK
2144 013320 000177 165350    JMP    @SCOLP         ;LOOP ON ERROR
2145
2146           ;TEST ITERATION SUBROUTINE*****
2147
2148 013324 000240          ITER:  NOP
2149 013326 032777 004000 165214  BIT    #4000,@SWR      ;SEE IF ITERATIONS
2150 013334 001403          BEQ    2$              ;IF SO: BR
2151 013336 005037 000662    1$:    CLR    ITCNT        ;CLEAR ITERATION COUNTER
2152 013342 000207          RTS    PC              ;ELSE EXIT
2153 013344 005737 000730    2$:    TST    PCNTR        ;DO SINGLE SUBTEST ITERATION
2154 013350 001772          BEQ    1$              ;ON FIRST PASS
2155 013352 005237 000662    INC    ITCNT          ;BUMP COUNTER
2156 013356 023737 000662 000566  CMP    ITCNT,ITAMT    ;SEE IF DONE ALL
2157 013364 001764          BEQ    1$              ;IF SO: BR
2158 013366 005726          TST    (SP)+          ;RESET STACK
2159 013370 017700 165302    MOV    @ITRLP,R0      ;SET ITERATION POINTER
2160 013374 000110          JMP    (R0)           ;GO ITERATE
2161
2162
2163
2164           ;NON-STANDARD JUMPER HANDLER SUBROUTINE*****
2165
2166 013376 010046          NOST:  MOV    R0,-(SP)      ;+SAVE R0
2167 013400 012700 000120    MOV    #120,R0        ;+SET UP INDEX
2168 013404 012760 011622 000756  MOV    #FT26,TSTTBL(R0);+ADJUST SCHEDULAR TEST TABLE
2169 013412 005720          TST    (R0)+
2170 013414 012760 011622 000756  MOV    #FT26,TSTTBL(R0) ;+OVERLAY TEST LIST
2171
2172          TST    (R0)+
2173 013422 005720          MOV    #FT27,TSTTBL(R0)
2174 013432 005720          TST    (R0)+
2175 013434 012760 012060 000756  MOV    #FT27,TSTTBL(R0)
2176 013442 005720          TST    (R0)+
2177 013444 012760 003156 000756  MOV    #TEND,TSTTBL(R0)
2178 013452 005720          TST    (R0)+
2179 013454 012760 000027 000756  MOV    #27,TSTTBL(R0)
2180 013462 012737 000027 001120  MOV    #27,TLAST
2181 013470 012600          MOV    (SP)+,R0      ;RESTO R0
2182 013472 000207          RTS    PC
2183
2184           ;INITIALIZE SUBROUTINE*****
2185

```

```
2186 013474 000240          INIT1:  NOP
2187 013476 012777 000040 165014      MOV     #40,@CS      ;INIT
2188 013504 013777 000612 165006      MOV     DRVN,@CS    ;SELECT DRIVE
2189 013512 013777 060614 165022      MOV     SLVN,@TC    ;SELECT SLAVE
2190 013520 000207          RTS     PC           ;RETURN
2191
2192                          ;MAG TAPE INTERRUPT HANDLER*****
2193
2194 013522 000240          MTINT:  NOP
2195 013524 013716 000646      MOV     RTRN,(SP)   ;RETURN TO (RTRN)
2196 013530 000002          RTI                    ;RETURN
2197
```

2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250

013532 017746 165016  
013536 042716 000200  
013542 122716 000003  
013546 001010  
013550 005737 001662  
013554 001005  
013556 005077 164764  
013562 000005  
013564 000137 000200  
013570 122716 000001  
013574 001017  
013576 022737 000176 000550  
013604 001016  
013606 012737 177570 000550  
013614 004737 015174  
013620 012704 020100  
013624 004737 014316  
013630 004737 015216  
013634 122716 000007  
013640 001005  
013642 012737 000176 000550  
013650 004737 015076  
013654 005726  
013656 000002

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

;BUS ADDRESS TRAP HANDLER\*\*\*\*\*

```
TRAP:  NOP  
        BIT    #20000,@SWR   ;SEE IF SHOULD PRINT ERRORS  
        BNE    TRAP2        ;IF NOT: BR  
        TST    HDRFL        ;SEE IF DONE HEADER  
        BNE    TRAP1        ;IF SO: BR  
        INC    HDRFL        ;ELSE SET HEADER FLAG  
        MOV    EMADDR,R4  
        JSR    PC,TTOUT     ;PRINT HEADER  
TRAP1: MOV    #MSG24,R4  
        JSR    PC,TTOUT     ;PRINT ERROR  
        MOV    R1,R3        ;GET ADDRESS THAT CAUSED THE TRAP  
        JSR    PC,OC7P      ;PRINT ADDRESS OF TRAP  
TRAP2: TST    @SWR         ;SEE IF HALT ON ERROR  
        BPL    TRAPX        ;IF NOT: BR  
        HALT  
TRAPX: CMP    (SP)+,(SP)+   ;RESET STACK  
        MOV    #FT1A,SCOLP  ;SET SCOPE ADDRESS  
        JSR    PC,SCOPE     ;GO SEE IF SCOPE LOOP  
        TST    RHTF        ;SEE IF INITIAL ADDRESS TEST  
        BEQ    TRAPXX      ;IF NOT: BR  
        JMP    STOB        ;ELSE REDO ADDRESS REQUEST  
TRAPXX: JMP   FT1B         ;RETURN TO TEST 1
```

```
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268 013774 010146
2269 013776 011601
2270 014000 005037 000652
2271 014004 005000
2272 014006 004737 014254
2273 014012 122737 000003 000602
2274 014020 001003
2275 014022 000005
2276 014024 000137 000200
2277 014030 122737 000015 000602
2278 014036 001004
2279 014040 005737 000652
2280 014044 001471
2281 014046 000457
2282 014050 122737 000025 000602
2283 014056 001005
2284 014060 012704 020020
2285 014064 004737 014316
2286 014070 000742
2287 014072 122737 000177 000602
2288 014100 001012
2289 014102 000241
2290 014104 006000
2291 014106 006200
2292 014110 006200
2293 014112 012704 020022
2294 014116 004737 014316
2295 014122 005201
2296 014124 000730
2297 014126 122737 000060 000602
2298 014134 101402
2299 014136 000137 014234
2300 014142 122737 000070 000602
2301 014150 101002
2302 014152 000137 014234
2303 014156 005237 000652
2304 014162 006300
2305 014164 006300
2306 014166 006300

:*****
:TTY ENTRY SUBROUTINE:
:
:THIS SUBROUTINE IS USED BY THE TEST CONDITION
:ENTRY ROUTINE TO READ THE RESPONSE ENTERED
:AT THE TTY AND CHECK THEM FOR LEGALITY AND
:LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
:(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
:THE CALLING ROUTINE.
:IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
:A QUESTION MARK IS TYPED (?) AND THE RESPONSE
:MAY BE REENTERED.
:ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
:MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
:CARRIAGE RETURN
:*****

TTR: MOV R1, -(SP) ;SAVE CHAR COUNT ON STACK
10$: MOV (SP), R1 ;RESTORE CHAR COUNT (FOR ^U)
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
CLR R0
1$: JSR PC, TTIN ;GO READ CHARACTER
CMPB #3, TIB ;BRANCH IF NOT ^C
BNE 11$
RESET ;RESET
JMP @#200 ;RESTART
11$: CMPB #15, TIB ;SEE IF CR
BNE 2$ ;IF NOT: BR
TST TEMP1 ;SEE IF FIRST CHARACTER
BEQ 9$ ;IF SO: BR
BR 6$ ;ELSE GO LOAD VALUE
2$: CMPB #25, TIB ;BRANCH IF NOT CONTROL U
BNE 21$
MOV #MSG65, R4 ;TYPE <CR><LF>
JSR PC, TTOUT
BR 10$ ;RESTART
21$: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'
BNE 3$
CLC ;REMOVE LAST CHARACTER
ROR R0
ASR R0
ASR R0
MOV #MSG66, R4 ;TYPE '\ '
JSR PC, TTOUT
INC R1 ;DECREMENT CHAR RECEIVED COUNT
BR 1$ ;GET NEXT CHARACTER
3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
BLOS 4$ ;IF NOT: BR
JMP T1NER ;ELSE GO TO ERROR
4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
BHI 5$ ;IF NOT: BR
JMP T1NER ;ELSE GO TO ERROR
5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
ASL R0
ASL R0 ;SHIFT 3 LEFT
ASL R0
```

J 5

```
2307 014170 042737 177770 000602      BIC      #177770,TIB      ;STRIP ASCII!
2308 014176 053700 000602      BIS      TIB,R0          ;LOAD CHARACTER
2309 014202 005301              DEC      R1              ;SEE IF DONE
2310 014204 001300              BNE      1$              ;IF NOT: BR
2311 014206 020002      6$:      CMP      R0,R2      ;SEE IF EXCEEDED MAXIMUM LIMIT
2312 014210 101402              BLOS     7$              ;IF NOT: BR
2313 014212 000137 014234      JMP      TINNER          ;ELSE GO TO ERROR
2314 014216 020300      7$:      CMP      R3,R0      ;SEE IF BELOW MINIMUM LIMIT
2315 014220 101402              BLOS     8$              ;IF NOT: BR
2316 014222 000137 014234      JMP      TINNER          ;ELSE GO TO ERROR
2317 014226 010015      8$:      MOV      R0,(R5)      ;LOAD VALUE
2318 014230 005726      9$:      TST      (SP)+         ;POP CHAR COUNT OFF STACK
2319 014232 000207      RTS      PC              ;EXIT
2320
2321
2322      ;TTY ENTRY ERROR SUBROUTINE*****
2323 014234 012704 015615      TINNER: MOV      #MSG7,R4
2324 014240 004737 014316      JSR      PC,TTOUT        ;PRINT?
2325 014244 005726      TST      (SP)+         ;POP CHAR COUNT OFF STACK
2326 014246 162716 000020      SUB      #20,(SP)       ;RESET SP TO START OF VALUE ROUTINE
2327 014252 000207      RTS      PC              ;REDO VALUE ENTRY
2328
2329      ;TTY READ SUBROUTINE*****
2330
2331 014254 005277 164272      TTIN:   INC      @TKS
2332 014260 105777 164266      1$:     TSTB     @TKS
2333 014264 100375              BPL      1$
2334 014266 117737 164262 000602      MOVB     @TKB,TIB
2335 014274 042737 000200 000602      BIC      #200,TIB      ;STRIP PARITY BIT
2336 014302 013737 000602 000600      MOV      TIB,TOB        ;MOVE CHAR TO OUTPUT BFR
2337 014310 004737 014416      JSR      PC,TOG          ;AND TYPE IT
2338 014314 000207      RTS      PC
2339
2340      ;TTY OUTPUT SUBROUTINE*****
2341
2342 014316 112437 000600      TTOUT:  MOVB     (R4)+,TOB
2343 014322 122737 000043 000600      CMPB     #43,TOB
2344 014330 001440              BEQ      TEX
2345 014332 122737 000045 000600      CMPB     #45,TOB
2346 014340 001403              BEQ      1$
2347 014342 004737 014416      JSR      PC,TOG
2348 014346 000763              BR       TTOUT
2349 014350 112737 000015 000600      1$:     MOVB     #15,TOB
2350 014356 004737 014416      JSR      PC,TOG
2351 014362 012703 000004              MOV      #4,R3
2352 014366 005037 000600      2$:     CLR      TOB
2353 014372 004737 014416      JSR      PC,TOG
2354 014376 005303              DEC      R3
2355 014400 001372              BNE      2$              ;DO FILLERS
2356 014402 112737 000012 000600      MOVB     #12,TOB
2357 014410 004737 014416      JSR      PC,TOG
2358 014414 000740              BR       TTOUT
2359 014416 105777 164134      TOG:    TSTB     @TPS
2360 014422 100375              BPL      TOG
2361 014424 113777 000600 164126      MOVB     TOB,@TPB
2362 014432 000207      TEX:    RTS      PC
```

```

2363                                     ;OCTAL OUTPUT SUBROUTINE*****
2364
2365 014434 012737 000001 014664 OCTPE: MOV #1,OFL
2366 014442 010304          MOV R3,R4
2367 014444 000410          BR OCTP0
2368 014446 005037 014664 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
2369 014452 010304          OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
2370 014454 001004          BNE OCTP0 ;IF NOT ZERO: BR
2371 014456 004737 014644          JSR PC,OCTPG1 ;ELSE PRINT ZERO
2372 014462 000137 014606          JMP OCTP3 ;SPACE AND EXIT
2373 014466 032704 100000          OCTP0: BIT #10000C,R4 ;SEE IF MSD = 1
2374 014472 001406          BEQ OCTP1 ;IF NOT: BR
2375 014474 012704 000001          MOV #1,R4
2376 014500 004737 014622          JSR PC,OCTPG ;PRINT 1
2377 014504 000137 014516          JMP OCTP2
2378 014510 005004          OCTP1: CLR R4
2379 014512 004737 014622          JSR PC,OCTPG ;PRINT 0
2380 014516 010304          OCTP2: MOV R3,R4
2381 014520 006004          ROR R4
2382 014522 006004          ROR R4
2383 014524 006004          ROR R4 ;POSITION DIGIT
2384 014526 006004          ROR R4
2385 014530 000304          SWAB R4
2386 014532 004737 014622          JSR PC,OCTPG ;PRINT DIGIT 2
2387 014536 010304          MOV R3,R4
2388 014540 006004          ROR R4
2389 014542 000304          SWAB R4
2390 014544 004737 014622          JSR PC,OCTPG ;PRINT DIGIT 3
2391 014550 010304          MOV R3,R4
2392 014552 006104          ROL R4
2393 014554 006104          ROL R4
2394 014556 000304          SWAB R4
2395 014560 004737 014622          JSR PC,OCTPG ;PRINT DIGIT 4
2396 014564 010304          MOV R3,R4
2397 014566 006004          ROR R4
2398 014570 006004          ROR R4
2399 014572 006004          ROR R4
2400 014574 004737 014622          JSR PC,OCTPG
2401 014600 010304          MOV R3,R4
2402 014602 004737 014622          JSR PC,OCTPG ;PRINT DIGIT 5
2403 014606 012737 000240 000600 OCTP3: MOV #240,TOB
2404 014614 004737 014416          JSR PC,TOG ;PRINT SPACE
2405 014620 000207          RTS PC ;EXIT
2406 014622 042704 177770          OCTPG: BIC #177770,R4
2407 014626 001004          BNE OCTPG0
2408 014630 005737 014664          TST OFL
2409 014634 001001          BNE OCTPG0
2410 014636 000207          RTS PC
2411
2412 014640 005237 014664          OCTPG0: INC OFL
2413 014644 052704 000260          OCTPG1: BIS #260,R4
2414 014650 010437 000600          MOV R4,TOB
2415 014654 004737 014416          JSR PC,TOG
2416 014660 010304          MOV R3,R4
2417 014662 000207          RTS PC
2418 014664 000000          OFL: 0 ;FIRST CHAR FLAG
  
```

```

2419
2420 ;DATA CHARACTER OUTPUT SUBROUTINE*****
2421
2422 014666 005037 000600 DOUT: CLR TOB
2423 014672 012704 000010 MOV #10,R4 ;SET NUMBER TO PRINT
2424 014676 110337 000600 MOVB R3,TOB
2425 014702 105777 163650 1$: TSTB @TPS
2426 014706 100375 BPL 1$
2427 014710 132737 000200 000600 BITB #200,TOB
2428 014716 001404 BEQ 2$
2429 014720 012777 000061 163632 MOV #061,@TPB
2430 014726 000403 BR 3$
2431 014730 012777 000060 163622 2$: MOV #060,@TPB
2432 014736 006137 000600 3$: ROL TOB
2433 014742 005304 DEC R4
2434 014744 001356 BNE 1$
2435 014746 000207 RTS PC
2436
2437 014750 013703 000656 DOUTD: MOV TEMP3,R3
2438 014754 000303 SWAB R3
2439 014756 004737 014666 JSR PC,DOUT
2440 014762 013703 000656 MOV TEMP3,R3
2441 014766 004737 014666 JSR PC,DOUT
2442 014772 000207 RTS PC
2443
2444 ;SERIAL NUMBER PRINT SUBROUTINE*****
2445
2446 014774 010304 SNPT: MOV R3,R4
2447 014776 000304 SWAB R4
2448 015000 006004 ROR R4
2449 015002 006004 ROR R4
2450 015004 006004 ROR R4
2451 015006 006004 ROR R4 ;GET FIRST DIGIT
2452 015010 004737 015052 JSR PC,SNPG ;GO PRINT
2453 015014 010304 MOV R3,R4
2454 015016 000304 SWAB R4 ;GET SECOND DIGIT
2455 015020 004737 015052 JSR PC,SNPG ;GO PRINT
2456 015024 010304 MOV R3,R4
2457 015026 006004 ROR R4
2458 015030 006004 ROR R4
2459 015032 006004 ROR R4
2460 015034 006004 ROR R4 ;GET THIRD DIGIT
2461 015036 004737 015052 JSR PC,SNPG ;GO PRINT
2462 015042 010304 MOV R3,R4 ;GET FOURTH DIGIT
2463 015044 004737 015052 JSR PC,SNPG ;GO PRINT
2464 015050 000207 RTS PC ;EXIT
2465 015052 012737 000260 000600 SNPG: MOV #260,TOB ;SET BASE = 0
2466 015060 042704 177760 BIC #177760,R4 ;MASK DIGIT
2467 015064 050437 000600 BIS R4,TOB ;SET ASCII
2468 015070 004737 014416 JSR PC,TOG ;TYPE DIGIT
2469 015074 000207 RTS PC ;RETURN
2470

```

```
2471
2472          ;ROUTINE TO LOAD NEW VALUE INTO SWITCHES
2473 015076 022737 000176 000550 GTSWR:  CMP    #SWREG,SWR    ;BRANCH IF SOFTWARE SWR
2474 015104 001032          BNE    1$          ;NOT INVOKED
2475 015106 004737 015174          JSR    PC,,SAVE    ;SAVE REGISTERS ON THE STACK
2476 015112 012704 021265          MOV    #SMSWR,R4
2477 015116 004737 014316          JSR    PC,TTOUT
2478 015122 017703 163422          MOV    @SWR,R3
2479 015126 004737 014434          JSR    PC,OCTPE
2480 015132 012704 021274          MOV    #SMNEW,R4
2481 015136 004737 014316          JSR    PC,TTOUT
2482 015142 013705 000550          MOV    SWR,R5    ;TTR ROUTINE RETURNS NEW VALUE TO (R5)
2483 015146 012701 000007          MOV    #7,R1    ;LIMIT RESPONSE TO 7 CHARS
2484 015152 012702 177777          MOV    #177777,R2 ;BETWEEN 0 AND 177777
2485 015156 012703 000000          MOV    #0,R3
2486 015162 004737 013774          JSR    PC,TTR
2487 015166 004737 015216          JSR    PC,,RESTORE ;RESTORE REGISTERS
2488 015172 000207          1$:   RTS    PC
2489
2490          ;;ROUTINE TO SAVE REGISTERS ON THE STACK
2491 015174 010546          .SAVE: MOV    %5,-(SP)    ;;R5 IS SAVED AT 12(SP)
2492 015176 010446          MOV    %4,-(SP)    ;;R4 IS SAVED AT 10(SP)
2493 015200 010346          MOV    %3,-(SP)    ;;R3 IS SAVED AT 6(SP)
2494 015202 010246          MOV    %2,-(SP)    ;;R2 IS SAVED AT 4(SP)
2495 015204 010146          MOV    %1,-(SP)    ;;R1 IS SAVED AT 2(SP)
2496 015206 010046          MOV    %0,-(SP)    ;;R0 IS SAVED AT (SP)
2497 015210 016646 000014          MOV    14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
2498 015214 000207          RTS    PC          ;;RETURN TO CALLER
2499
2500          ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
2501 015216 012666 000014          .RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
2502 015222 012600          MOV    (SP)+,%0
2503 015224 012601          MOV    (SP)+,%1
2504 015226 012602          MOV    (SP)+,%2
2505 015230 012603          MOV    (SP)+,%3
2506 015232 012604          MOV    (SP)+,%4
2507 015234 012605          MOV    (SP)+,%5
2508 015236 000207          RTS    PC          ;;RETURN
2509
2510
```

```
2511 ;MESSAGE TABLE*****
2512
2513 015240 041445 030523 020040 MSG1: .ASCII /%CS1 WC BA FC CS2 /
2514 015246 020040 041527 020040
2515 015254 020040 041040 020101
2516 015262 020040 020040 041506
2517 015270 020040 020040 041440
2518 015276 031123 020040 020040
2519 015304 051504 020040 020040 .ASCII /DS ER TC%#/
2520 015312 042440 020122 020040
2521 015320 020040 041524 021445
2522 015326 051045 053505 047111 MSG2: .ASCII /%REWIND ERROR-BOT NOT SET WHEN PIP CLEARED#/
2523 015334 020104 051105 047522
2524 015342 026522 047502 020124
2525 015350 047516 020124 042523
2526 015356 020124 044127 047105
2527 015364 050040 050111 041440
2528 015372 042514 051101 042105
2529 015400 043
2530 015401 045 052045 030115 MSG3: .ASCII '%TM03-TE16/TU77 BASIC FUNCTION TEST (CZTECCO)%';++B
2531 015406 026463 042524 033061
2532 015414 052057 033525 020067
2533 015422 040502 044523 020103
2534 015430 052506 041516 044524
2535 015436 047117 052040 051505
2536 015444 020124 041450 052132
2537 015452 041505 030103 022451
2538 015460 054524 042520 036040 .ASCII /TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%#/
2539 015466 051103 020076 047524
2540 015474 052040 051105 044515
2541 015502 040516 042524 051040
2542 015510 051505 047520 051516
2543 015516 020105 020046 041536
2544 015524 052040 020117 042522
2545 015532 052123 051101 022524
2546 015540 043
2547 015541 045 042522 044507 MSG4: .ASCII /%REGISTER START = #/
2548 015546 052123 051105 051440
2549 015554 040524 052122 036440
2550 015562 021440
2551 015564 053045 041505 047524 MSG5: .ASCII /%VECTOR = #/
2552 015572 020122 020075 043
2553 015577 045 047105 020104 MSG6: .ASCII /%END OF PASS #/
2554 015604 043117 050040 051501
2555 015612 020123 043
2556 015615 040 020077 043 MSG7: .ASCII / ? #/
2557 015621 045 047520 044523 MSG9: .ASCII /%POSITION ERROR: #/
2558 015626 044524 047117 042440
2559 015634 051122 051117 020072
2560 015642 043
2561 015643 045 051511 041440 MSG10A: .ASCII /%IS CONTROLLER JUMPERED IN NON-STANDARD MODE/<15><12>
2562 015650 047117 051124 046117
2563 015656 042514 020122 052512
2564 015664 050115 051105 042105
2565 015672 044440 020116 047516
2566 015700 026516 052123 047101
```

2567	015706	040504	042122	046440		
2568	015714	042117	006505	012		
2569	015721	124	050131	020105	.ASCII	/TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD: #/
2570	015726	020062	047506	020122		
2571	015734	047516	026516	052123		
2572	015742	047101	040504	042122		
2573	015750	047440	020122	051103		
2574	015756	043040	051117	051440		
2575	015764	040524	042116	051101		
2576	015772	035104	020040	020040		
2577	016000	020040	043			
2578	016003	045	051104	053111	MSG10: .ASCII	/%DRIVE NUMBER: #/
2579	016010	020105	052516	041115		
2580	016016	051105	020072	043		
2581	016023	045	046123	053101	MSG11: .ASCII	/%SLAVE NUMBER: #/
2582	016030	020105	052516	041115		
2583	016036	051105	020072	043		
2584	016043	045	051127	052111	MSG12: .ASCII	/%WRITE ERROR #/
2585	016050	020105	051105	047522		
2586	016056	020122	043			
2587	016061	045	042522	042101	MSG13: .ASCII	/%READ REVERSE ERROR #/
2588	016066	051040	053105	051105		
2589	016074	042523	042440	051122		
2590	016102	051117	021440			
2591	016106	051045	040505	020104	MSG14: .ASCII	/%READ FORWARD ERROR #/
2592	016114	047506	053522	051101		
2593	016122	020104	051105	047522		
2594	016130	020122	043			
2595	016133	045	051127	052111	MSG15: .ASCII	/%WRITE TM ERROR #/
2596	016140	020105	046524	042440		
2597	016146	051122	051117	021440		
2598	016154	020045	051127	052111	MSG15A: .ASCII	/% WRITE TM ERROR ON SECOND TM #/
2599	016162	020105	046524	042440		
2600	016170	051122	051117	047440		
2601	016176	020116	042523	047503		
2602	016204	042116	052040	020115		
2603	016212	043				
2604	016213	045	041520	020040	MSG15B: .ASCII	/%PC #/
2605	016220	043				
2606	016221	045	042522	042526	MSG16: .ASCII	/%REVERSE ERROR #/
2607	016226	051522	020105	051105		
2608	016234	047522	020122	043		
2609	016241	045	047506	053522	MSG17: .ASCII	/%FORWARD ERROR #/
2610	016246	051101	020104	051105		
2611	016254	047522	020122	043		
2612	016261	040	051116	020132	MSG20: .ASCII	/ NRZ #/
2613	016266	043				
2614	016267	040	042520	021440	MSG21: .ASCII	/ PE #/
2615	016274	042440	050130	035124	MSG22: .ASCII	/ EXPT: #/
2616	016302	021440				
2617	016304	051040	053103	035104	MSG23: .ASCII	/ RCVD: #/
2618	016312	021440				
2619	016314	041045	051525	052040	MSG24: .ASCII	/%BUS TRAP: #/
2620	016322	040522	035120	021440		
2621	016330	053445	035103	021440	MSG25: .ASCII	/%WC: #/
2622	016336	041045	035101	021440	MSG26: .ASCII	/%BA: #/

2623	016344	042045	035102	021440	MSG27:	.ASCII	/%DB: #/
2624	016352	044445	044516	020124	MSG28:	.ASCII	/%INIT DID NOT CLEAR RH #/
2625	016360	044504	020104	047516			
2626	016366	020124	046103	040505			
2627	016374	020122	044122	021440			
2628	016402	051445	020103	047516	MSG29:	.ASCII	/%SC NOT RESET BY INIT #/
2629	016410	020124	042522	042523			
2630	016416	020124	054502	044440			
2631	016424	044516	020124	043			
2632	016431	045	051124	020105	MSG30:	.ASCII	/%TRE NOT RESET BY INIT #/
2633	016436	047516	020124	042522			
2634	016444	042523	020124	054502			
2635	016452	044440	044516	020124			
2636	016460	043					
2637	016461	045	051503	020062	MSG31:	.ASCII	/%CS2 NOT RESET BY INIT #/
2638	016466	047516	020124	042522			
2639	016474	042523	020124	054502			
2640	016502	044440	044516	020124			
2641	016510	043					
2642	016511	045	046104	020124	MSG32:	.ASCII	/%DLT NOT SET #/
2643	016516	047516	020124	042523			
2644	016524	020124	043				
2645	016527	045	041523	047040	MSG33:	.ASCII	/%SC NOT SET #/
2646	016534	052117	051440	052105			
2647	016542	021440					
2648	016544	052045	042522	047040	MSG34:	.ASCII	/%TRE NOT SET #/
2649	016552	052117	051440	052105			
2650	016560	021440					
2651	016562	044445	020122	047516	MSG35:	.ASCII	/%IR NOT SET BY INIT #/
2652	016570	020124	042523	020124			
2653	016576	054502	044440	044516			
2654	016604	020124	043				
2655	016607	045	051117	047040	MSG36:	.ASCII	/%OR NOT RESET BY INIT #/
2656	016614	052117	051040	051505			
2657	016622	052105	041040	020131			
2658	016630	047111	052111	021440			
2659	016636	047445	020122	047516	MSG37:	.ASCII	/%OR NOT RESET BY 1 SILO ENTRY #/
2660	016644	020124	042522	042523			
2661	016652	020124	054502	030440			
2662	016660	051440	046111	020117			
2663	016666	047105	051124	020131			
2664	016674	043					
2665	016675	045	051117	047040	MSG38:	.ASCII	/%OR NOT SET BY SILO FULL #/
2666	016702	052117	051440	052105			
2667	016710	041040	020131	044523			
2668	016716	047514	043040	046125			
2669	016724	020114	043				
2670	016727	045	040502	020104	MSG39:	.ASCII	/%BAD SILO READ #/
2671	016734	044523	047514	051040			
2672	016742	040505	020104	043			
2673	016747	045	051111	047040	MSG40:	.ASCII	/%IR NOT RESET BY SILO FULL #/
2674	016754	052117	051040	051505			
2675	016762	052105	041040	020131			
2676	016770	044523	047514	043040			
2677	016776	046125	021514				
2678	017002	047045	047117	042455	MSG41:	.ASCII	/%NON-EXIST DRIVE #/

2679	017010	044530	052123	042040	
2680	017016	044522	042526	043	
2681	017023	045	047516	026516	MSG42: .ASCII /%NON-EXIST SLAVE#/
2682	017030	054105	051511	020124	
2683	017036	046123	053101	021505	
2684	017044	051445	051105	040511	MSG43: .ASCII /%SERIAL NO: #/
2685	017052	020114	047516	020072	
2686	017060	043			
2687	017061	045	051105	051501	MSG44: .ASCII /%ERASE HEAD INOPERATIVE#/
2688	017066	020105	042510	042101	
2689	017074	044440	047516	042520	
2690	017102	040522	044524	042526	
2691	017110	043			
2692	017111	045	047520	051523	MSG45: .ASCII /%POSSIBLE ERASE HEAD PROBLEM: /
2693	017116	041111	042514	042440	
2694	017124	040522	042523	044040	
2695	017132	040505	020104	051120	
2696	017140	041117	042514	035115	
2697	017146	040			
2698	017147	103	042510	045503	.ASCII /CHECK POLARITY#/
2699	017154	050040	046117	051101	
2700	017162	052111	021531		
2701	017166	051445	052105	052455	MSG46: .ASCII /%SET-UP WRITE ERROR#/
2702	017174	020120	051127	052111	
2703	017202	020105	051105	047522	
2704	017210	021522			
2705	017212	051445	040520	042503	MSG47: .ASCII /%SPACE FORWARD ERROR#/
2706	017220	043040	051117	040527	
2707	017226	042122	042440	051122	
2708	017234	051117	043		
2709	017237	045	050123	041501	MSG48: .ASCII /%SPACE REVERSE ERROR#/
2710	017244	020105	042522	042526	
2711	017252	051522	020105	051105	
2712	017260	047522	021522		
2713	017264	041045	043125	042506	MSG49: .ASCII /%BUFFERED WRITE ERROR#/
2714	017272	042522	020104	051127	
2715	017300	052111	020105	051105	
2716	017306	047522	021522		
2717	017312	041045	052117	051440	MSG50: .ASCII /%BOT SET AFTER BUFFERED WRITE#/
2718	017320	052105	040440	052106	
2719	017326	051105	041040	043125	
2720	017334	042506	042522	020104	
2721	017342	051127	052111	021505	
2722	017350	047045	020117	047502	MSG51: .ASCII /%NO BOT FROM READ IN PRESET#/
2723	017356	020124	051106	046517	
2724	017364	051040	040505	020104	
2725	017372	047111	050040	042522	
2726	017400	042523	021524		
2727	017404	052045	020103	047111	MSG52: .ASCII /%TC INCORRECT #/
2728	017412	047503	051122	041505	
2729	017420	020124	043		
2730	017423	045	047515	020114	MSG53: .ASCII /%MOL FAILED TO CLEAR#/
2731	017430	040506	046111	042105	
2732	017436	052040	020117	046103	
2733	017444	040505	021522		
2734	017450	022445	042522	042523	MSG54: .ASCII /%%RESET SLAVE TO ON LINE BEFORE CONTINUING/

2735	017456	020124	046123	053101	
2736	017464	020105	047524	047440	
2737	017472	020116	044514	042516	
2738	017500	041040	043105	051117	
2739	017506	020105	047503	052116	
2740	017514	047111	044525	043516	
2741	017522	051445	052105	051440	.ASCII /%SET SW12=1 IF YOU DOT WISH TO REPEAT REWIND OFFLINE TEST#/
2742	017530	030527	036462	020061	
2743	017536	043111	054440	052517	
2744	017544	042040	052117	053440	
2745	017552	051511	020110	047524	
2746	017560	051040	050105	040505	
2747	017566	020124	042522	044527	
2748	017574	042116	047440	043106	
2749	017602	044514	042516	052040	
2750	017610	051505	021524		
2751	017614	044440	042524	035122	MSG56: .ASCII / ITER: #/
2752	017622	021440			
2753	017624	052045	020115	047516	MSG57: .ASCII /%TM NOT SET#/
2754	017632	020124	042523	021524	
2755	017640	042445	052111	042510	MSG60: .ASCII /%EITHER TAPE NOT ERASED OR OPI PROBLEM#/
2756	017646	020122	040524	042520	
2757	017654	047040	052117	042440	
2758	017662	040522	042523	020104	
2759	017670	051117	047440	044520	
2760	017676	050040	047522	046102	
2761	017704	046505	043		
2762	017707	045	044122	047440	MSG62: .ASCII /%RH ONLY (NO=0,YES=1): #/
2763	017714	046116	020131	047050	
2764	017722	036517	026060	042531	
2765	017730	036523	024461	020072	
2766	017736	043			
2767	017737	045	044504	020104	MSG63: .ASCII /%DID NOT AUTO SELECT NRZ#/
2768	017744	047516	020124	052501	
2769	017752	047524	051440	046105	
2770	017760	041505	020124	051116	
2771	017766	021532			
2772	017770	042045	042111	047040	MSG64: .ASCII /%DID NOT AUTO SELECT PE#/
2773	017776	052117	040440	052125	
2774	020004	020117	042523	042514	
2775	020012	052103	050040	021505	
2776	020020	021445			MSG65: .ASCII /%#/
2777	020022	021534			MSG66: .ASCII /\#/
2778	020024	042445	035122	021440	MSG67: .ASCII /%ER: #/
2779	020032	051045	046505	053117	MSG69: .ASCII /%REMOVE TMDP FROM SLAVE TO BE TESTED%#/
2780	020040	020105	046524	050104	
2781	020046	043040	047522	020115	
2782	020054	046123	053101	020105	
2783	020062	047524	041040	020105	
2784	020070	042524	052123	042105	
2785	020076	021445			
2786	020100	044045	051101	053504	MSG70: .ASCII /%HARDWARE SWR IN USE%#/
2787	020106	051101	020105	053523	
2788	020114	020122	047111	052440	
2789	020122	042523	021445		
2790					

```
2791 ;TEST HEADERS*****
2792
2793 020126 022445 052106 035061 MSFT1: .ASCII /%FT1:RH ADDRESSING #/
2794 020134 044122 040440 042104
2795 020142 042522 051523 047111
2796 020150 020107 043
2797 020153 045 043045 031124 MSFT2: .ASCII /%FT2:RH REGISTER BITS TEST #/
2798 020160 051072 020110 042522
2799 020166 044507 052123 051105
2800 020174 041040 052111 020123
2801 020202 042524 052123 021440
2802 020210 022445 052106 035063 MSFT3: .ASCII /%FT3:RH INITIALIZE TEST #/
2803 020216 044122 044440 044516
2804 020224 044524 046101 055111
2805 020232 020105 042524 052123
2806 020240 021440
2807 020242 022445 052106 035064 MSFT4: .ASCII /%FT4:RH11 SILO TEST 1 #/
2808 020250 044122 030461 051440
2809 020256 046111 020117 042524
2810 020264 052123 030440 021440
2811 020272 022445 052106 035065 MSFT5: .ASCII /%FT5:RH11 SILO TEST 2 #/
2812 020300 044122 030461 051440
2813 020306 046111 020117 042524
2814 020314 052123 031040 021440
2815 020322 022445 052106 035066 MSFT6: .ASCII /%FT6:RH11 SILO TEST 3 #/
2816 020330 044122 030461 051440
2817 020336 046111 020117 042524
2818 020344 052123 031440 021440
2819 020352 022445 052106 035067 MSFT7: .ASCII /%FT7:RH11 SILO TEST 4 #/
2820 020360 044122 030461 051440
2821 020366 046111 020117 042524
2822 020374 052123 032040 021440
2823 020402 022445 052106 030061 MSFT10: .ASCII /%FT10:RH11 SILO TEST 5 #/
2824 020410 051072 030510 020061
2825 020416 044523 047514 052040
2826 020424 051505 020124 020065
2827 020432 043
2828 020433 045 043045 030524 MSFT11: .ASCII /%FT11:NOP TEST#/
2829 020440 035061 047516 020120
2830 020446 042524 052123 043
2831 020453 045 043045 030524 MSFT12: .ASCII /%FT12:REWIND TEST#/
2832 020460 035062 042522 044527
2833 020466 042116 052040 051505
2834 020474 021524
2835 020476 022445 052106 031461 MSFT13: .ASCII /%FT13:WRITE-READ TEST#/
2836 020504 053472 044522 042524
2837 020512 051055 040505 020104
2838 020520 042524 052123 043
2839 020525 045 043045 030524 MSFT14: .ASCII /%FT14:SPACE TEST#/
2840 020532 035064 050123 041501
2841 020540 020105 042524 052123
2842 020546 043
2843 020547 045 043045 030524 MSFT15: .ASCII /%FT15:ERASE TEST#/
2844 020554 035065 051105 051501
2845 020562 020105 042524 052123
2846 020570 043
```

2847	020571	045	043045	030524	MSFT16: .ASCII /%FT16:TAPE MARK WRITE-READ TEST#/
2848	020576	035066	040524	042520	
2849	020604	046440	051101	020113	
2850	020612	051127	052111	026505	
2851	020620	042522	042101	052040	
2852	020626	051505	021524		
2853	020632	022445	052106	033461	MSFT17: .ASCII /%FT17:TM SPACE TEST #/
2854	020640	052072	020115	050123	
2855	020646	041501	020105	042524	
2856	020654	052123	021440		
2857	020660	022445	052106	030062	MSFT20: .ASCII /%FT20:WRITE CHECK TEST #/
2858	020666	053472	044522	042524	
2859	020674	041440	042510	045503	
2860	020702	052040	051505	020124	
2861	020710	043			
2862	020711	045	043045	031124	MSFT21: .ASCII /%FT21:ERASE HEAD TEST#/
2863	020716	035061	051105	051501	
2864	020724	020105	042510	042101	
2865	020732	052040	051505	021524	
2866	020740	022445	052106	031062	MSFT22: .ASCII /%FT22:BUFFERED COMMAND TEST#/
2867	020746	041072	043125	042506	
2868	020754	042522	020104	047503	
2869	020762	046515	047101	020104	
2870	020770	042524	052123	043	
2871	020775	045	043045	031124	MSFT23: .ASCII /%FT23:READ IN PRESET TEST#/
2872	021002	035063	042522	042101	
2873	021010	044440	020116	051120	
2874	021016	051505	052105	052040	
2875	021024	051505	021524		
2876	021030	022445	052106	032062	MSFT24: .ASCII /%FT24:AUTO DENSITY SELECT: WRITE-NRZ,READ-PE#/
2877	021036	040472	052125	020117	
2878	021044	042504	051516	052111	
2879	021052	020131	042523	042514	
2880	021060	052103	020072	051127	
2881	021066	052111	026505	051116	
2882	021074	026132	042522	042101	
2883	021102	050055	021505		
2884	021106	022445	052106	032462	MSFT25: .ASCII /%FT25:AUTO DENSITY SELECT: WRITE-PE,READ-NRZ#/
2885	021114	040472	052125	020117	
2886	021122	042504	051516	052111	
2887	021130	020131	042523	042514	
2888	021136	052103	020072	051127	
2889	021144	052111	026505	042520	
2890	021152	051054	040505	026504	
2891	021160	051116	021532		
2892	021164	022445	052106	033062	MSFT26: .ASCII /%FT26:SEQUENTIAL TAPE MARK TEST#/
2893	021172	051472	050505	042525	
2894	021200	052116	040511	020114	
2895	021206	040524	042520	046440	
2896	021214	051101	020113	042524	
2897	021222	052123	043		
2898	021225	045	043045	031124	MSFT27: .ASCII /%FT27:REWIND-OFF LINE TEST#/
2899	021232	035067	042522	044527	
2900	021240	042116	047455	043106	
2901	021246	046040	047111	020105	
2902	021254	042524	052123	043	

2903	021261	045	043536	043	\$CNTG:	.ASCII	/%^G#/ / %SWR= #/
2904	021265	045	053523	036522	\$MSWR:	.ASCII	
2905	021272	021440					
2906	021274	020040	042516	036527	\$MNEW:	.ASCII	/ NEW= #/
2907	021302	021440					
2908	021304	022477	043		\$QUEST:	.ASCII	/?%#/ .EVEN
2909							
2910							
2911		021310			WDATA:	0	
2912	021310	000000					
2913		023022					
2914	023022	000000			RDATA:	0	.+.1510
2915							
2916		000001					.END





FT15A	006642	1508#						
FT15B	006656	1510#	1516					
FT15X	006766	1514	1536#					
FT15XX	007172	1555	1567#					
FT16	007204	783	784	1573#				
FT16A	007236	1578#	1601					
FT16B	007242	1579#						
FT16X	007422	1599	1602#					
FT17	007432	785	786	1608#				
FT17A	007452	1611#	1665					
FT17B	007456	1612#	1642					
FT17C	007616	1632#	1641					
FT17D	007664	1625	1643#					
FT17D1	007702	1646#	1661					
FT17E	007716	1648#	1656					
FT17F	010006	1658	1662#					
FT17X	010030	1620	1639	1653	1663	1666#		
FT2	003372	759	760	1052#				
FT2A	003404	1054#	1061	1086				
FT2B	003444	1059	1063#	1070				
FT2C	003504	1068	1072#	1081				
FT2D	003520	1076#	1077					
FT2E	003550	1079	1083#					
FT2ER	003560	1062	1071	1082	1087#	1820	1956	
FT2ERA	003610	1091	1094#					
FT2ERB	003662	1089	1105#					
FT2ERC	003672	1106	1108#					
FT2X	003702	1084	1111#					
FT20	010034	787	788	1670#				
FT20A	010052	1673#	1698					
FT20B	010176	1690#						
FT20C	010230	1695#						
FT20X	010250	1684	1696	1699#				
FT21	010260	789	790	1704#				
FT21A	010266	1705#						
FT21B	010562	1742	1745#					
FT21C	010570	1744	1746#					
FT21SC	010416	1723#	1746					
FT21X	010602	1738	1740	1748#				
FT22	010612	791	792	1754#	1779			
FT22A	010666	1762#	1764					
FT22B	010706	1767#	1768					
FT22X	011006	1777	1781#					
FT23	011016	793	794	1787#	1812	1819		
FT23A	011126	1803#	1806	1808				
FT23B	011146	1804	1809#					
FT23C	011176	1810	1814#					
FT23X	011232	1788	1817	1821#				
FT24	011236	795	796	1827#	2015			
FT24X	011420	1840	1847	1850#				
FT25	011430	797	798	1855#	2017			
FT25X	011612	1868	1875	1878#				
FT26	011622	799	800	1883#	2168	2170		
FT26X	012050	1894	1900	1909	1914	1916	1919#	
FT27	012060	801	802	1923#	2173	2175		
FT27X	012270	1950	1954	1957#				



MSFT12	020453	1342	2831#		
MSFT13	020476	1353	2835#		
MSFT14	020525	1394	1465	2839#	
MSFT15	020547	1504	2843#		
MSFT16	020571	1576	2847#		
MSFT17	020632	1609	2853#		
MSFT2	020153	1052	2797#		
MSFT20	020660	1671	2857#		
MSFT21	020711	1704	2862#		
MSFT22	020740	1754	2866#		
MSFT23	020775	1789	2871#		
MSFT24	021030	1827	2876#		
MSFT25	021106	1855	2884#		
MSFT26	021164	1884	2892#		
MSFT27	021225	1927	2898#		
MSFT3	020210	1117	2802#		
MSFT4	020242	1162	2807#		
MSFT5	020272	1187	2811#		
MSFT6	020322	1218	2815#		
MSFT7	020352	1270	2819#		
MSG1	015240	2047	2513#		
MSG10	016003	886	2578#		
MSG10A	015643	876	2561#		
MSG11	016023	903	2581#		
MSG12	016043	1362	1635	1714	2584#
MSG13	016061	1372	1590	2587#	
MSG14	016106	1379	1595	1730	2591#
MSG15	016133	1584	1617	1890	2595#
MSG15A	016154	1896	2598#		
MSG15B	016213	1557	2604#		
MSG16	016221	1478	1645	1685	2606#
MSG17	016241	1476	1659	1690	2609#
MSG2	015326	2522#			
MSG20	016261	1469	2033	2612#	
MSG21	016267	1472	2036	2614#	
MSG22	016274	1097	1253	1480	2615#
MSG23	016304	1101	1257	1485	2617#
MSG24	016314	2236	2619#		
MSG25	016330	1060	2621#		
MSG26	016336	1069	2622#		
MSG27	016344	1080	2623#		
MSG28	016352	2624#			
MSG29	016402	1125	2628#		
MSG3	015401	835	840*	2530#	
MSG30	016431	1129	2632#		
MSG31	016461	1135	2637#		
MSG32	016511	1173	1281	1307	2642#
MSG33	016527	1175	2645#		
MSG34	016544	1177	2648#		
MSG35	016562	1192	2651#		
MSG36	016607	1196	2655#		
MSG37	016636	1201	1209	2659#	
MSG38	016675	1232	2665#		
MSG39	016727	1251	2670#		
MSG4	015541	841	2547#		
MSG40	016747	1228	2673#		



RDYDX	000636	708#	1319*	1350*	1403*	1502*	1521*	1536*	1574*	1613*	1770*	1976		
REGS	000572	688#	843	845	862									
RFD	000634	707#	1420*	1443*	1457									
RHOF	000726	736#	926	928	1303									
RHTF	000722	734#	874*	1044	1046*	2246								
RH17F	000604	695#	942*	948*	1160	1185	1216	1268	1287					
RRD	000632	706#	1419*	1442*	1450									
RTRN	000646	712#	2195											
RWND	012436	1357	1396	1505	1518	1578	1611	1673	1705	1719	1724	1755	1828	1841
		1856	1869	1886	1901	1928	1991#							
SAV1	000666	720#												
SAV2	000670	721#												
SAV3	000672	722#												
SCHN	001710	827	829#											
SCNT	000642	710#	1421*	1431*	1440*	1646*	1903*	1972						
SCOLP	000674	723#	1061*	1070*	1081*	1118*	1179*	1188*	1219*	1271*	1290*	1313*	1331*	1364*
		1746*	1779*	1812*	1819*	1939*	2140	2144	2244*					
SCOPE	013270	1108	1155	2135#	2245									
SERFL	000712	730#	1410	1426	1436	1513	1619	1638	1652	1683	1839	1867	1893	1899
		1908	1913	2004*	2022*	2080								
SERNUM	000562	684#												
SLVN	000614	699#	905	907	914	957*	966*	972*	973	975	1787	2189		
SN	000540	672#	922											
SNPG	015052	2452	2455	2461	2463	2465#								
SNPT	014774	923	2446#											
START	001600	644	808#											
STFLG	000704	727#	941*	991	996*									
STMSK	000660	717#	983*	1405*	1501*	1525*	1583*	1589*	1616*	1636*	1650*	1774*	1889*	2012*
		2019*	2020											
STSCD	003122	953	996#											
STO	002124	865#	868											
STOB	001764	841#	2248											
ST1	002146	871#	873											
ST1A	002166	876#	902	1047										
ST2	002344	899	903#	919										
ST3	002454	916	920#											
ST4	002540	648	935#											
SWR	000550	679#	813	817*	825*	951	988	993	1023	1088	1105	1142	1151	1245
		1261	1463	1490	1923	2023	2041	2069	2136	2149	2212	2214*	2221*	2229
		2240	2473	2478	2482									
SWREG	000176	638#	817	825	2212	2221	2473							
TAG	007202	1556*	1566*	1570#	2059									
TC	000542	673#	914*	975*	1333*	1545*	1815	1964*	2034	2067	2189*			
TEMPST	000732	738#												
TEMP1	000652	714#	1553*	1554	1627*	1640*	2061	2270*	2279	2303*				
TEMP2	000654	715#												
TEMP3	000656	716#	2437	2440										
TEND	003156	803	998	1004#	1305	2177								
TENDX	003254	1022	1024	1026#										
TEX	014432	2344	2362#											
TIB	000602	694#	2273	2277	2282	2287	2297	2300	2307*	2308	2334*	2335*	2336	
TINER	014234	2299	2302	2313	2316	2323#								
TKB	000554	681#	2201	2334										
TKS	000552	680#	940*	2331*	2332									
TLAST	001120	804#	997	2180*										
TMCHK	013120	1586	1592	1597	1621	1654	1892	1898	1907	1912	2078#			



CZTECCO TM03-TE16/TU77 BFT  
CZTECC.P11 24-JUL-79 14:10

MACY11 30A(1052) 24-JUL-79 14:15 PAGE 83  
CROSS REFERENCE TABLE -- MACRO NAMES

D 7

SEQ 0081

\$CATCH	537#	616
\$CHAIN	537#	820
\$CHNMO	537#	954
\$RESTO	537#	2500
\$SAVE	537#	2490
.\$ACT1	537#	617
.\$EOP	537#	1013

. ABS. 023024 000

ERRORS DETECTED: 0

CZTECC.CZTECC.SEQ/CRF/SOL=CZTEAC.SML/ML.CZTECC.P11  
RUN-TIME: 10 18 2 SECONDS  
RUN-TIME RATIO: 82/32=2.5  
CORE USED: 9K (17 PAGES)