

TM03/TE16

BASIC FUNCTION TEST
CZTECBO

AH-A798B-MC
COPYRIGHT © 77-78
FICHE 1 OF 1

JAN 1979
digital
MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE: AC-A797B-MC
PRODUCT NAME: CZTECBO TM03-TE16/TU77 BASIC FUNCTION TEST
DATE CREATED: 15 NOV 78
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, 1978 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

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

1. ABSTRACT

THIS PROGRAM IS INTENDED TO TEST ALL OF THE BASIC FUNCTIONAL LEVEL OPERATIONS OF THE 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 JUMPED IN NON-STANDARD MODE
TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD: <3> [JUMPER:]
DRIVE NUMBER: <0> (CR) [DRVN:]
SLAVE NUMBER: <0> (CR) [SLVN:]
SERIAL NO: 12345
RH ONLY (NO-0,YES-1): <0> (0) [RHOF:]
IF THE SOFTWARE SWR IS INVOKED:
SWR <000000> NEW - (CR)

5. CONSOLE SWITCH SETTING

CONTROL:

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

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

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

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

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

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

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

SW15(100000): 1=HALT ON ERROR
0=CONTINUE
SW14(040000): 1=LOOP ON ERROR (SCOPE: 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 ERASE 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 REFLECT
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 3 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: HEAD 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 CZTECBO TMO3-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,.SEOP,$SCATCH,$SAVE,$RESTORE,$CHAIN,$SCHNMODE
.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 000046 .=46
621 000046 .WORD $ENDAD ;SET LOCATION 46
622 000052 .=52
623 000052 .WORD 0 ;SET LOCATION 52 = 0
624 000764 .=$SVPC ;RESTORE LOCATION CTR
625
626 ;TTY INTERRUPT VECTOR*****
627
628 000060 .=60
629 000060 013304 .WORD TTINT ;TTY INTERRUPT HEADER ADDRESS
630 000062 000340 .WORD 340 ;PRIORITY LEVEL 7
631
632 ;SOFTWARE SWITCH REGISTER*****
633 ;USED IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
634
635 000176 .=176
636 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
637
638
639 ;START ADDRESS*****
640
641 000200 .=200
642 000200 000137 001600 JMP START ;PROGRAM START
643
644 ;RESTART ADDRESS*****
645 000210 .=210
646 000210 000137 002540 JMP ST4
647
648 ;TM03 INTERRUPT VECTOR*****
649
650 000224 .=224
651 000224 013274 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
652 000226 000340 340
653
```



```
654
655          000510
656          :.=510
657          :MASS BUS REGISTER EQUIVS*****
658 000510 172440 C1: 172440
659 000512 172442 WC: 172442
660 000514 172444 BA: 172444
661 000516 172446 FC: 172446
662 000520 172450 CS: 172450
663 000522 172452 DS: 172452
664 000524 172454 ER: 172454
665 000526 172456 AS: 172456
666 000530 172460 CC: 172460
667 000532 172462 DB: 172462
668 000534 172464 MR: 172464
669 000536 172466 DT: 172466
670 000540 172470 UN: 172470
671 000542 172472 TC: 172472
672 000544 172474 BAE: 172474
673
674          :CONSTANTS*****
675
676 000546 177776 PSW: 177776 ;PROCESSOR STATUS
677 000550 177570 SWR: 177570 ;SWITCH REGISTER
678 000552 177560 TKS: 177560 ;TTY READER STATUS
679 000554 177562 TKB: 177562 ;TTY READ BUFFER
680 000556 177564 TPS: 177564 ;TTY PUNCH STATUS
681 000560 177566 TPB: 177566 ;TTY PUNCH BUFFER
682 000562 177777 SERNUM: 177777 ;SERIAL NUMBER
683 000564 000C11 DRVTP: 011 ;DRIVE TYPE
684 000566 000010 ITAMT: 10 ;ITERATION AMOUNT
685 000570 000224 VECT: 224 ;INTERRUPT VECTOR(RH)
686 000572 172440 REGS: 172440 ;STARTING REGISTER ADDRESS
687 000574 000004 BTRP: 4 ;BUS TRAP ADDRESS
688 000576 000006 BTRP2: 6 ;BUS TRAP PRIORITY LEVEL 7
```

```
689                                     ;FLAGS AND COUNTERS*****
690
691 000600 000000 TOB: 0
692 000602 000000 TIB: 0
693 000604 000000 RH17F: 0
694 000606 000000 HDRFL: 0
695 000610 000000 EMADDR: 0
696 000612 000000 DRVN: 0
697 000614 000000 SLVN: 0
698 000616 000000 BADDR: 0
699 000620 000000 FCNT: 0
700 000622 000000 WCNT: 0
701 000624 000000 RCNT: 0
702 000626 000000 EPRP: 0
703 000630 000000 ERRP1: 0
704 000632 000000 RRD: 0
705 000634 000000 RFD: 0
706 000636 000000 RDYDX: 0
707 000640 000000 OPDYX: 0
708 000642 000000 SCNT: 0
709 000644 000000 PFLG: 0
710 000646 000000 RTRN: 0
711 000650 000000 ERADD: 0
712 000652 000000 TEMP1: 0
713 000654 000000 TEMP2: 0
714 000656 000000 TEMP3: 0
715 000660 000000 STMSK: 0
716 000662 000000 ITCNT: 0
717 000664 000000 DSAV: 0
718 000666 000000 SAV1: 0
719 000670 000000 SAV2: 0
720 000672 000000 SAV3: 0
721 000674 000000 SCOLP: 0
722 000676 000000 ITRLP: 0
723 000700 000000 EXFL: 0
724 000702 000000 PEXFL: 0
725 000704 000000 STFLG: 0
726 000706 000000 LTADD: 0
727 000710 000000 FUN: 0
728 000712 000000 SERFL: 0
729 000714 000000 CRCNT: 0
730 000716 000000 UDES: 0
731 000720 000000 PATRN: 0
732 000722 000000 RMTF: 0
733 000724 000000 NRZOF: 0
734 000726 000000 RHOF: 0
735 000730 000000 PCNTR: 0
736 000732 000000 TEMPST: 0
737 000734 000000 COUNT: 0
738 000736 000000 RDSW: 0
739 000740 000000 NONSTD: 0
740 000742 000000 JUMPER: 0
741
```

742
743
744
745 000744 000000
746 000746 012770
747 000750 013010
748 000752 013014
749 000754 013022

;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

			:LOGIC TEST ENTRY TABLE*****	
	TSTTBL:			
750				
751				
752				
753	000756	000000	0	
754	000760	000000	0	
755	000762	003264	FT1	
756	000764	003264	FT1	
757	000766	003372	FT2	
758	000770	003372	FT2	
759	000772	003714	FT3	
760	000774	003714	FT3	
761	000776	004134	FT4	
762	001000	004134	FT4	
763	001002	004262	FT5	
764	001004	004262	FT5	
765	001006	004454	FT6	
766	001010	004454	FT6	
767	001012	004726	FT7	
768	001014	004726	FT7	
769	001016	005022	FT10	
770	001020	005022	FT10	
771	001022	005156	FT11	
772	001024	005156	FT11	
773	001026	005274	FT12	
774	001030	005274	FT12	
775	001032	005406	FT13	
776	001034	005406	FT13	
777	001036	005720	FT14	
778	001040	005720	FT14	
779	001042	006572	FT15	
780	001044	006572	FT15	
781	001046	006772	FT16	
782	001050	006772	FT16	
783	001052	007220	FT17	
784	001054	007220	FT17	
785	001056	007622	FT20	
786	001060	007622	FT20	
787	001062	010046	FT21	
788	001064	010046	FT21	
789	001066	010400	FT22	
790	001070	010400	FT22	
791	001072	010604	FT23	
792	001074	010604	FT23	
793	001076	011024	FT24	
794	001100	011024	FT24	
795	001102	011216	FT25	
796	001104	011216	FT25	
797	001106	011410	FT26	
798	001110	011410	FT26	
799	001112	011646	FT27	
800	001114	011646	FT27	
801	001116	003156	.WORD	TEND
802	001120	000027	.WORD	27

:CONTAINS # OF TESTS

```

803          001600          =1600
804          ;PROGRAM START AND HOUSEKEEPING*****
805
806 001600 012706 000500      START: MOV #500,SP          ;SET STACK POINTER
807 001604 013746 000006      MOV @#6,-(SP)          ;SAVE VECTORS
808 001610 013746 000004      MOV @#4,-(SP)
809 001614 012737 001640 000004  MOV #1$,@#4          ;SET UP FOR TIMEOUT
810 001622 005037 000006      CLR @#6
811 001626 022777 177777 176714  CMP #-1,@SWR          ;REFERENCE HARDWARE SWITCH REGISTER
812 001634 001402
813 001636 000404
814 001640 022626      1$: CMP (SP)+,(SP)+          ;ADJUST STACK
815 001642 012737 000176 000550  2$: MOV #SWREG,SWR          ;POINT TO SOFTWARE SWITCH REG
816 001650 012637 000004      3$: MOV (SP)+,@#4          ;RESTORE VECTORS
817 001654 012637 000006      MOV (SP)+,@#6
818 001660 005027      CLR (PC)+          ;;CLEAR CHAIN INDICATOR
819 001662 000000      (HNFLG: .WORD 0          ;;CHAIN MODE INDICATOR
820                                     ;;1/0 = CHAIN/NOT CHAIN MODE
821 001664 005737 000042      TST @#42          ;;BRANCH IF IN DUMP MODE
822 001670 001407
823 001672 012737 000176 000550  BEQ 50$
824 001700 005237 001662      MOV #SWREG,SWR          ;;INVOKE SOFTWARE SWR
825 001704 000137 001710      INC CHNFLG          ;;SET CHNFLG = CHAIN MODE
826 001710      JMP SCHN          ;;GO TO CHAIN ADDRESS
827 001710 000240      50$: NOP
828 001712 122737 000006 000041  4$: CMPB #6,@#41          ;BRANCH IF LOADED VIA TMDP (DUMP MODE)
829 001720 001005
830 001722 012704 017576      MOV #MSG69,R4          ;ADVISE USER TO REMOVE TMDP FROM UUT
831 001726 004737 014070      JSR PC,TTOUT
832 001732 000000      HALT
833 001734 012704 015153      5$: MOV #MSG3,R4
834 001740 004737 014070      JSR PC,TTOUT          ;PRINT TITLE
835 001744 005737 001662      TST CHNFLG          ;SEE IF IN CHAIN MODE
836 001750 001402      BEQ 6$          ;IF NOT: BR
837 001752 000137 002554      JMP TSCD          ;ELSE GO START TEST
838 001756 112737 000043 015153  6$: MOVB #'#,MSG3          ;DO NOT PRINT TITLE ON RESTART
839 001764 012704 015313      STOB: MOV #MSG4,R4
840 001770 004737 014070      JSR PC,TTOUT          ;REQUEST REGISTER ADDRESS
841 001774 013703 000572      MOV REGS,R3
842 002000 004737 014220      JSR PC,OCTP          ;PRINT CURRENT ADDRESS
843 002004 012705 000572      MOV #REGS,R5          ;SET ADDRESS SAVE LOC
844 002010 012701 000007      MOV #7,R1          ;SET SIZE OF RESPONSE
845 002014 012702 176400      MOV #176400,R2          ;SET UPPER LIMIT
846 002020 012703 172300      MOV #172300,R3          ;SET LOWER LIMIT
847 002024 004737 013546      JSR PC,TTR          ;GO GET RESPONSE
848 002030 012704 015336      MOV #MSG5,R4
849 002034 004737 014070      JSR PC,TTOUT          ;REQUEST VECTOR
850 002040 013703 000570      MOV VECT,R3
851 002044 004737 014220      JSR PC,OCTP          ;PRINT CURRENT VECTOR
852 002050 012705 000570      MOV #VECT,R5          ;SET ADDRESS SAVE LOC
853 002054 012701 000004      MOV #4,R1          ;SET SIZE OF RESPONSE
854 002060 012702 000224      MOV #224,R2          ;SET UPPER LIMIT
855 002064 012703 000150      MOV #150,R3          ;SET LOWER LIMIT
856 002070 004737 013546      JSR PC,TTR          ;GO GET RESPONSE
857 002074 013700 000570      MOV VECT,R0          ;GET VECTOR
858 002100 012720 013274      MOV #MTINT,(R0)+          ;LOAD INTERRUPT ADDRESS IN VECTOR
  
```

859	002104	012710	000340		MOV	#340,(R0)	:LOAD PRIORITY
860	002110	013700	000572		MOV	REGS,R0	:GET START OF REGS
861	002114	012701	000017		MOV	#17,R1	:SET NUMBER OF REGS
862	002120	012702	000510		MOV	#C1,R2	:GET START OF TABLE
863	002124	010022		ST0:	MOV	R0,(R2)+	:BUILD TABLE
864	002126	062700	000002		ADD	#2,R0	:BUMP ADDRESS
865	002132	005301			DEC	R1	:SEE IF DONE
866	002134	001373			BNE	ST0	:IF NOT: BR
867	002136	012702	000600		MOV	#TOB,R2	
868	002142	012700	000054		MOV	#54,R0	
869	002146	005022		ST1:	CLR	(R2)+	:CLEAR FLAGS + COUNTERS
870	002150	005300			DEC	R0	
871	002152	001375			BNE	ST1	
872	002154	012737	000001	000722	MOV	#1,RHTF	:SET ADDRESS TEST FLAG
873	002162	000137	003016		JMP	TSRH	:GO DO INITIAL ADDRESS TEST PASS
874	002166	012704	015415	ST1A:	MOV	#MSG10A,R4	
875	002172	004737	014070		JSR	PC,TTOUT	:REQUEST JUMPER CONFIGURATION
876	002176	012705	000742		MOV	#JUMPER,R5	:GET ADDRESS OF RESPONSE
877	002202	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
878	002206	012702	000004		MOV	#4,R2	:SET RANGE
879	002212	012703	000000		MOV	#0,R3	:LOWER LIMIT
880	002216	004737	013546		JSR	PC,TTR	:GET RESPONSE
881	002222	022737	000002	000742	CMP	#2,JUMPER	:TEST FOR NON-STANDARD MODE
882	002230	001002			BNE	1\$	
883	002232	004737	013150		JSR	PC,NOST	:MODIFY TEST SCHEDULE
884	002236	012704	015555	1\$:	MOV	#MSG10,R4	
885	002242	004737	014070		JSR	PC,TTOUT	:REQUEST DRIVE NUMBER
886	002246	013703	000612		MOV	DRVN,R3	:GET CURRENT DRIVE #
887	002252	004737	014220		JSR	PC,OCTP	:AND TYPE IT
888	002256	012705	000612		MOV	#DRVN,R5	:SET ADDRESS OF DRIVE NUMBER SAVE
889	002262	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
890	002266	012702	000007		MOV	#7,R2	:SET UPPER LIMIT
891	002272	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
892	002276	004737	013546		JSR	PC,TTR	:GO GET RESPONSE
893	002302	012777	000040	176210	MOV	#40,@CS	:SET INIT
894	002310	053777	000612	176202	BIS	DRVN,@CS	:SET DRIVE NUMBER
895	002316	005777	176166		TST	@C1	:ACCESS DRIVE
896	002322	032777	010000	176170	BIT	#10000,@CS	:SEE IF NED
897	002330	001405			BEQ	ST2	:IF NOT: BR
898	002332	012704	016546		MOV	#MSG41,R4	
899	002336	004737	014070		JSR	PC,TTOUT	:PRINT NOT AVAIL
900	002342	000711			BR	ST1A	:REDO DRIVE REQUEST
901	002344	012704	015575	ST2:	MOV	#MSG11,R4	
902	002350	004737	014070		JSR	PC,TTOUT	:REQUEST SLAVE NUMBER
903	002354	013703	000614		MOV	SLVN,R3	:GET CURRENT SLAVE #
904	002360	004737	014220		JSR	PC,OCTP	:AND TYPE IT
905	002364	012705	000614		MOV	#SLVN,R5	:SET ADDRESS OF SLAVE SAVE
906	002370	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
907	002374	012702	000007		MOV	#7,R2	:SET UPPER LIMIT
908	002400	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
909	002404	004737	013546		JSR	PC,TTR	:GO GET RESPONSE
910	002410	012777	000040	176102	MOV	#40,@CS	:INIT
911	002416	053777	000612	176074	BIS	DRVN,@CS	:SET DRIVE NUMBER
912	002424	013777	000614	176110	MOV	SLVN,@TC	:LOAD SLAVE NUMBER
913	002432	032777	002000	176076	BIT	#2000,@DT	:SEE IF SLAVE PRESENT
914	002440	001005			BNF	ST3	:IF SO: BR

```
915 002442 012704 016567      MOV    #MSG42,R4
916 002446 004737 014070      JSR    PC,TTOUT      ;PRINT NON-EXIST SLAVE
917 002452 000734              BR     ST2           ;REDO SLAVE REQUEST
918 002454 012704 016610      ST3:  MOV    #MSG43,R4
919 002460 004737 014070      JSR    PC,TTOUT      ;PRINT SERIAL NUMBER TAG
920 002464 017703 176050      MOV    @SN,R3
921 002470 004737 014546      JSR    PC,SNPT       ;PRINT SERIAL NUMBER
922 002474 012704 017453      MOV    #MSG62,R4     ;GET REQUEST
923 002500 004737 014070      JSR    PC,TTOUT      ;REQUEST RH11 ONLY RESPONSE
924 002504 013703 000726      MOV    RHOF,R3       ;GET CURRENT FLAG SETTING
925 002510 004737 014220      JSR    PC,OCTP       ;AND TYPE IT
926 002514 012705 000726      MOV    #RHOF,R5      ;SET FLAG ADDRESS
927 002520 012701 000002      MOV    #2,R1         ;SET SIZE OF RESPONSE
928 002524 012702 000001      MOV    #1,R2         ;SET UPPER LIMIT
929 002530 012703 000000      MOV    #0,R3         ;SET LOWER LIMIT
930 002534 004737 013546      JSR    PC,TTR        ;GO GET RESPONSE
931
932      ;START 210
933 002540 012706 000500      ST4:  MOV    #500,SP  ;SET STACK PTR
934 002544 005037 000730      CLR    PCNTR         ;CLEAR PASS COUNTER
935 002550 004737 014650      JSR    PC,GTSWR      ;GET SWITCHES
```

```
936 ;TEST SCHEDULAR*****
937
938 002554 052777 000100 175770 TSCD: BIS #100,@TKS ;SET KEYBOARD IE BIT
939 002562 005037 000704 CLR STFLG ;CLEAR SINGLE TEST FLAG
940 002566 005037 000604 CLR RH17F ;SET RH INDICATOR = RH11
941 002572 013746 000004 MOV @#4,-(SP) ;SAVE ERROR TRAP VECTORS
942 002576 013746 000006 MOV @#6,-(SP) ;AND PRIORITY
943 002602 012737 002626 000004 MOV #1$,@#4 ;SET TIME OUT TRAP TO 1$ BELOW
944 002610 005037 000006 CLR @#6
945 002614 005777 175724 TST @BAE ;REFERENCE BAE REGISTER
946 002620 012737 000001 000604 MOV #1,RH17F ;SET FLAG = RH70
947 002626 012637 000006 1$: MOV (SP)+,@#6 ;RESTORE ERROR TRAP
948 002632 012637 000004 MOV (SP)+,@#4
949 002636 017700 175706 MOV @SWR,R0
950 002642 042700 177740 BIC #177740,R0
951 002646 001125 BNE STSCD ;GO SELECT SINGLE TEST
952 002650 005737 001662 TST CHNFLG ;BRANCH IF NOT IN CHAIN MODE
953 002654 001457 BEQ TSCDA
954 002656 012737 177777 000612 MOV #-1,DRVN ;:INITIALIZE DRIVE #
955 002664 012737 177777 000614 NXTDRV: MOV #-1,SLVN ;:INITIALIZE SLAVE #
956 002672 012777 000040 175620 1$: MOV #40,@CS ;:INIT CONTROLLER
957 002700 005237 000612 INC DRVN ;:STEP DRIVE #
958 002704 022737 000010 000612 CMP #10,DRVN ;:EXIT IF ALL DRIVES TESTED
959 002712 001524 BEQ $DONE ;:FOR AVAILABILITY
960 002714 013777 000612 175576 MOV DRVN,@CS ;:LOAD DRIVE #
961 002722 005777 175562 TST @C1 ;:ACCESS DRIVE
962 002726 032777 010000 175564 BIT #10000,@CS ;:BRANCH IF DRIVE NON EXISTANT
963 002734 001356 BNE 1$ ;:(NED = 1)
964 002736 005237 000614 NXTSLV: INC SLVN ;:STEP SLAVE # AND BRANCH
965 002742 001011 BNE 1$ ;:IF NOT SLAVE 0
966 002744 005737 000612 TST DRVN ;:BRANCH IF NOT DRIVE # 0
967 002750 001006 BNE 1$
968 002752 122737 000006 000041 CMPB #6,@#41 ;:BRANCH IF NOT TMDP
969 002760 001002 BNE 1$
970 002762 005237 000614 INC SLVN ;:STEP TO SLAVE # 1
971 002766 022737 000010 000614 1$: CMP #10,SLVN ;:BRANCH IF ALL SLAVES TESTED
972 002774 001733 BEQ NXTDRV ;:FOR AVAILABILITY
973 002776 013777 000614 175536 MOV SLVN,@TC ;:LOAD SLAVE UNIT #
974 003004 032777 002000 175524 BIT #2000,@DT ;:BRANCH IF SLAVE NOT
975 003012 001751 BEQ NXTSLV ;:PRESENT (SPR - 0)
976 003014 000240 TSCDA: NOP
977 003016 012737 000756 000706 TSRH: MOV #TSTTBL,LTADD
978 003024 062737 000004 000706 TSCD0: ADD #4,LTADD
979 003032 013737 000706 000676 TSCD1: MOV LTADD,IIRLP
980 003040 062737 000002 000676 ADD #2,IIRLP ;SET ITERATION ADDRESS
981 003046 005037 000660 CLR STMSK
982 003052 005037 000626 CLR ERRP
983 003056 005037 000606 CLR HDRFL ;CLEAR PRINT HEADER FLAG
984 003062 017700 175620 MOV @LTADD,R0 ;SET POINTER TO TEST
985 003066 000110 JMP (R0) ;GO TO TEST
986 003070 032777 002000 175452 TSCD2: BIT #2000,@SWR ;SEE IF HALT ON TEST
987 003076 001401 BEQ TSCD3 ;IF NOT: BR
988 003100 000000 HALT
989 003102 005737 000704 TSCD3: TST STFLG ;SE IF SINGLE TEST
990 003106 001746 BEQ TSCD0 ;IF NOT: BR
991 003110 017700 175434 MOV @SWR,R0
```


992	003114	042700	177740		BIC	#177740,R0	:BRANCH IF ALL TESTS SELECTED
993	003120	001615			BEQ	TSCD	
994	003122	012737	000001	000704	STSCD: MOV	#1,STFLG	:SET SINGLE TEST FLAG
995	003130	023700	001120		CMP	TLAST,R0	:SEE IF EXCEEDED TESTS
996	003134	002410			BLT	TEND	:IF SO: BR
997	003136	006300			ASL	R0	
998	003140	006100			ROL	R0	:SET TABLE MODIFIER
999	003142	012737	000756	000706	MOV	#TSTTBL,LTADD	
1000	003150	060037	000706		ADD	R0,LTADD	:SET TEST POINTER
1001	003154	000726			BR	TSCD1	
1002	003156	005737	001662		TEND: TST	CHNFLG	:BRANCH IF IN CHAIN MODE
1003	003162	001265			BNE	NXTSLV	
1004	003164	012704	015351		\$DONE: MOV	#MSG6,R4	
1005	003170	004737	014070		JSR	PC,TIOUT	:PRINT END OF PASS
1006	003174	013703	000730		MOV	PCNTR,R3	
1007	003200	004737	014220		JSR	PC,OCIP	:PRINT PASS NUMBER
1008	003204	005000			CLR	R0	
1009	003206	005300			1\$: DEC	R0	
1010	003210	001376			BNE	1\$	
1011	003212	013700	000042		MOV	@#42,R0	:GET ACT11 RETURN ADDRESS
1012	003216	001405			BEQ	HERE	:BRANCH IF NOT ACT11
1013	003220	000005			RESET		
1014	003222	004710			\$ENDAD: JSR	PC,(R0)	
1015	003224	000240			NOP		
1016	003226	000240			NOP		
1017	003230	000240			NOP		
1018	003232	000240			HERE: NOP		
1019	003234	005737	001662		TST	CHNFLG	:BRANCH IF IN CHAIN MODE
1020	003240	001005			BNE	TENDX	
1021	003242	032777	010000	175300	BIT	#10000,@SWR	:SEE IF HALT ON PASS
1022	003250	001001			BNE	TENDX	:IF NOT: BR
1023	003252	000000			HALT		
1024	003254	005237	000730		TENDX: INC	PCNTR	:BUMP PASS COUNTER
1025	003260	000137	002554		JMP	TSCD	:RESTART

```
1026  
1027  
1028  
1029 003264 012737 017672 000610 FT1: MOV #MSFT1,EMADDR ;SET HEADER  
1030 003272 012737 013432 000004 MOV #TRAP,@#4 ;SET TRAP HANDLER ADDRESS  
1031 003300 012737 000340 000006 MOV #340,@#6  
1032 003306 012700 000016 MOV #16,R0 ;SET NUMBER OF REGISTERS  
1033 003312 013701 000510 MOV C1,R1 ;GET FIRST ADDRESS (CS1)  
1034 003316 005711 FT1A: TST (R1) ;REFERENCE REGISTER  
1035 003320 000240 NOP ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR  
1036 003322 005300 FT1B: DEC R0 ;SEE IF DONE ALL  
1037 003324 001403 BEQ FT1X ;IF SO: BR  
1038 003326 062701 000002 ADD #2,R1 ;BUMP ADDRESS POINTER  
1039 003332 000771 BR FT1A ;CONTINUE  
1040 003334 012737 000006 000004 FT1X: MOV #6,@#4 ;RESET TRAP CATCHER  
1041 003342 012737 000000 000006 MOV #HALT,@#6  
1042 003350 005737 000722 TST RHTF ;SEE IF INITIAL ADDRESS TEST PASS  
1043 003354 001404 BEQ FT1XX ;IF NOT: BR  
1044 003356 005037 000722 CLR RHTF ;CLEAR FLAG  
1045 00336 000137 002166 JMP ST1A ;RETURN  
1046 003366 000137 003070 FT1XX: JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

1047
1048
1049
1050 003372 012737 017717 000610 FT2:  MOV    #MSFT2,EMADDR  ;SET TEST HEADER
1051 003400 012701 177777          MOV    #-1,R1        ;SET ALL ONES PATTERN
1052 003404 004737 013246          FT2A: JSR    PC,INIT1    ;GO INIT
1053 003410 013700 000512          MOV    WC,R0        ;GET ADDRESS OF WORD COUNT
1054 003414 010102          MOV    R1,R2        ;SET EXPT REGISTER BIT PATTERN
1055 003416 010110          MOV    R1,(R0)      ;LOAD PATTERN
1056 003420 021002          CMP    (R0),R2      ;SEE IF EXPT=RCVD
1057 003422 001410          BEQ    FT2B        ;IF SO: BR
1058 003424 012737 016074 000650  MOV    #MSG25,ERADD ;SET CODE
1059 003432 012737 003404 000674  MOV    #FT2A,SCOLP  ;SET SCOPE
1060 003440 004737 003560          JSR    PC,FT2ER     ;GO DO ERROR
1061 003444 013700 000514          FT2B: MOV    BA,R0        ;GET ADDRESS OF BUS ADDRESS
1062 003450 010102          MOV    R1,R2
1063 003452 042702 000001          BIC    #1,R2        ;SET EXPT PATTERN
1064 003456 010110          MOV    R1,(R0)      ;LOAD PATTERN
1065 003460 020210          CMP    R2,(R0)      ;SEE IF EXPT=RCVD
1066 003462 001410          BEQ    FT2C        ;IF SO:BR
1067 003464 012737 016102 000650  MOV    #MSG26,ERADD ;SET ERROR CODE
1068 003472 012737 003444 000674  MOV    #FT2B,SCOLP  ;SET SCOPE ADDRESS
1069 003500 004737 003560          JSR    PC,FT2ER     ;GO DO ERROR
1070 003504 013700 000532          FT2C: MOV    DB,R0        ;GET ADDRESS OF DATA BUFFER
1071 003510 010102          MOV    R1,R2
1072 003512 010110          MOV    R1,(R0)
1073 003514 012703 004000          MOV    #4000,R3     ;LOAD PATTERN
1074 003520 005303          FT2D: DEC    R3        ;DELAY
1075 003522 001376          BNE    FT2D
1076 003524 020210          CMP    R2,(R0)      ;SEE IF EXPT=RCVD
1077 003526 001410          BEQ    FT2E        ;IF SO: BR
1078 003530 012737 016110 000650  MOV    #MSG27,ERADD ;SET ERROR CODE
1079 003536 012737 003504 000674  MOV    #FT2C,SCOLP  ;SET SCOPE ADDRESS
1080 003544 004737 003560          JSR    PC,FT2ER     ;GO DO ERROR
1081 003550 005701          FT2E: TST    R1        ;SEE IF DONE RESET
1082 003552 001453          BEQ    FT2X        ;IF SO: BR
1083 003554 005001          CLR    R1          ;SET ZERO PATTERN
1084 003556 000712          BR    FT2A         ;DO ZERO BITS
1085 003560 000240          FT2ER: NOP
1086 003562 032777 020000 174760  BIT    #20000,@SWR  ;SEE IF PRINT ERROR
1087 003570 001034          BNE    FT2ERB      ;IF NOT: BR
1088 003572 005737 000606          TST    HDRFL       ;SEE ID DONE HEADER
1089 003576 001004          BNE    FT2ERA      ;IF SO: BR
1090 003600 013704 000610          MOV    EMADDR,R4
1091 003604 004737 014070          JSR    PC,TTOUT    ;DO HEADER
1092 003610 012737 000001 000606  FT2ERA: MOV    #1,HDRFL    ;SET FLAG
1093 003616 013704 000650          MOV    ERADD,R4
1094 003622 004737 014070          JSR    PC,TTOUT    ;PRINT ERROR CODE
1095 003626 012704 016040          MOV    #MSG22,R4
1096 003632 004737 014070          JSR    PC,TTOUT    ;PRINT EXPT TAG
1097 003636 010103          MOV    R1,R3
1098 003640 004737 014206          JSR    PC,OCTPE    ;PRINT EXPT
1099 003644 012704 016050          MOV    #MSG23,R4
1100 003650 004737 014070          JSR    PC,TTOUT    ;PRINT RCVD TAG
1101 003654 011003          MOV    (R0),R3
1102 003656 004737 014206          JSR    PC,OCTPE    ;PRINT RCVD

```

1103 003662 005777 174662
1104 003666 100001
1105 003670 000000
1106 003672 004737 013042
1107 003676 000240
1108 003700 000207
1109 003702 000240
1110 003704 004737 013076
1111 003710 000137 003070

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

```
1112
1113
1114 ;RH INITIALIZE TEST*****
1115 003714 012737 017754 000610 FT3: MOV #MSFT3,EMADDR ;SET TEST HEADER
1116 003722 012737 003714 000674 MOV #FT3,SCOLP
1117 003730 004737 013246 JSR PC,INIT1 ;GO INIT
1118 003734 052777 020000 174556 BIS #20000,@CS ;FORCE UPE =1
1119 003742 000240 NOP
1120 003744 004737 013246 JSR PC,INIT1 ;GO INIT
1121 003750 005777 174534 TST @C1 ;SEE IF SC IS RESET
1122 003754 100005 BPL FT3A ;IF SO: BR
1123 003756 012737 016146 000650 MOV #MSG29,ERADD ;SET ERROR CODE
1124 003764 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1125 003770 032777 040000 174512 FT3A: BIT #40000,@C1 ;SEE IF TRE IS RESET
1126 003776 001405 BEQ FT3B ;IF SO: BR
1127 004000 012737 016175 000650 MOV #MSG30,ERADD ;SET ERROR CODE.
1128 004006 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1129 004012 017701 174502 FT3B: MOV @CS,R1 ;GET CS2
1130 004016 042701 000307 BIC #307,R1 ;MARK IR/OR
1131 004022 005701 TST R1 ;SEE IF RESET
1132 004024 001405 BEQ FT3X ;IF SO: BR
1133 004026 012737 016225 000650 MOV #MSG31,ERADD ;SET ERROR CODE
1134 004034 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1135 004040 004737 013076 FT3X: JSR PC,ITER ;GO SEE IF ITERATION
1136 004044 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1137
1138 ;ERROR REPORT SUBROUTINE
1139 004050 000240 FT3ER: NOP
1140 004052 032777 020000 174470 BIT #20000,@SWR ;SEE IF PRINT ERROR
1141 004060 001015 BNE 2$ ;IF NOT: BR
1142 004062 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1143 004066 001006 BNE 1$ ;IF SO: BR
1144 004070 013704 000610 MOV EMADDR,R4
1145 004074 004737 014070 JSR PC,TTOUT ;PRINT HEADER
1146 004100 005237 000606 INC HDRFL
1147 004104 013704 000650 1$: MOV ERADD,R4
1148 004110 004737 014070 JSR PC,TTOUT ;PRINT ERROR CODE
1149 004114 005777 174430 2$: TST @SWR ;SEE IF HALT ON ERROR
1150 004120 100001 BPL 3$ ;IF NOT: BR
1151 004122 000000 HALT
1152 004124 000240 3$: NOP
1153 004126 004737 013042 JSR PC,SCOPE ;GO SEE IF SCOPE
1154 004132 000207 RTS PC ;IF NOT: BR
```

```
1155
1156
1157 ;RH11 SILO TEST 1: EPMTY SILO READ*****
1158 004134 005737 000604 FT4: TST RH17F
1159 004140 001141 BNE FT5X ;IF RH70: BR
1160 004142 012737 020006 000610 MOV #MSFT4,EMADDR ;SET TEST TEST HEADER
1161 004150 012777 000040 174342 MOV #40,@CS ;INIT
1162 004156 017700 174350 MOV @DB,R0 ;READ C
1163 004162 005777 174332 TST @CS ;SEE IF LIT IS SET
1164 004166 100013 BPL FT4ER ;IF NOT: BR
1165 004170 005777 174314 TST @C1 ;SEE IF SC IS SET
1166 004174 100014 BPL FT4ERA ;IF NOT: BR
1167 004176 032777 040000 174304 BIT #40000,@C1 ;SEE IF TRE IS SET
1168 004204 001414 BEQ FT4ERB ;IF NOT: BR
1169 004206 004737 013076 FT4X: JSR PC,ITER ;GO SEE IF ITERATION
1170 004212 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1171 004216 012737 016255 000650 FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE
1172 004224 000407 BR FT4ERC
1173 004226 012737 016273 000650 FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE
1174 004234 000403 BR FT4ERC
1175 004236 012737 016310 000650 FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.
1176 004244 000240 FT4ERC: NOP
1177 004246 012737 004134 000674 MOV #FT4,SCOLP ;SET SCOPE ADDRESS
1178 004254 004737 004050 JSR PC,FT3ER ;GO PRINT ERROR
1179 004260 000752 BR FT4X
```

```
1180
1181
1182 ;RH11 SILO TEST 2· IR/OR CHECK*****
1183 004262 005737 000604 FT5: TST RH17F ;SEE IF RH70
1184 004266 001066 BNE FT5X ;IF SO: BR
1185 004270 012737 020036 000610 MOV #MSFT5,EMADDR ;SET TEST HEADER
1186 004276 012737 004304 000674 MOV #FT5A,SCOLP ;SET SCOPE ADDRESS
1187 004304 004737 013246 FT5A: JSR PC,INIT1 ;GO INIT
1188 004310 032777 000100 174202 BIT #100,@CS ;SEE IF IR IS SET
1189 004316 001005 BNE FT5B ;IF SO: BR
1190 004320 012737 016326 000650 MOV #MSG35,ERADD ;SET ERROR CODE
1191 004326 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1192 004332 032777 000200 174160 FT5B: BIT #200,@CS ;SEE IF OR IS RESET
1193 004340 001405 BEQ FT5C ;IF SO: BR
1194 004342 012737 016353 000650 MOV #MSG36,ERADD ;SET ERROR CODE
1195 004350 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1196 004354 012777 000000 174150 FT5C: MOV #0,@DB ;LOAD ZERO INTO SILO
1197 004362 032777 000200 174130 BIT #200,@CS ;SEE THAT OR RESET
1198 004370 001405 BEQ FT5D ;IF IT DOES: BR
1199 004372 012737 016402 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1200 004400 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1201 004404 012777 177777 174120 FT5D: MOV #-1,@DB ;LOAD SILO WITH -1
1202 004412 012700 004000 MOV #400,R0
1203 004416 032777 000200 174074 FT5E: BIT #200,@CS ;SEE IF OR IS SET
1204 004424 001007 BNE FT5X ;IF SO: BR
1205 004426 005300 DEC R0
1206 004430 001372 BNE FT5E ;AWAIT OR
1207 004432 012737 016402 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1208 004440 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1209 004444 004737 013076 FT5X: JSR PC,ITER ;GO SEE IF ITERATION
1210 004450 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
1211
1212
1213
1214 004454 005737 000604 FT6: TST RH17F
1215 004460 001052 BNE FT6X
1216 004462 012737 020066 000610 MOV #MSFT6,EMADDR ;IF RH70: BR
1217 004470 012737 004476 000674 MOV #FT6A,SCOLP ;SET TEST HEADER
1218 004476 004737 013246 FT6A: JSR PC,INIT1 ;SET SCOPE ADDRESS
1219 004502 005000 CLR R0 ;GO INIT
1220 004504 010077 174022 FT6B: MOV R0,@DB ;PRESET DATA
1221 004510 005200 INC R0 ;LOAD SILO
1222 004512 022700 000102 CMP #102,R0 ;BUMP DATA
1223 004516 001372 BNE FT6B ;SEE IF FILLED ALL
1224 004520 032777 000100 173772 BIT #100,@CS ;IF NOT: BR
1225 004526 001405 BEQ FT6C ;SEE IF IR IS RESET.
1226 004530 012737 016513 000650 MOV #MSG40,ERADD ;IF SO: BR
1227 004536 004737 004050 JSR PC,FT3ER ;SET ERROR CODE
1228 004542 032777 000200 173750 FT6C: BIT #200,@CS ;GO DO ERROR
1229 004550 001005 BNE FT6D ;SEE IF OR IS SET
1230 004552 012737 016441 000650 MOV #MSG38,ERADD ;IF SO: BR
1231 004560 004737 004050 JSR PC,FT3ER ;SET ERROR CODE
1232 004564 005000 FT6D: CLR R0 ;GO DO ERROR
1233 004566 017701 173740 FT6E: MOV @DB,R1 ;PRESET DATA
1234 004572 020001 CMP R0,R1 ;READ SILO
1235 004574 001010 BNE FT6DE ;SEE IF EXPT=RCVD
1236 004576 005200 INC R0 ;IF NOT: BR
1237 004600 022700 000102 CMP #102,R0 ;BUMP DATA
1238 004604 001370 BNE FT6E ;SEE IF DONE ALL
1239 004606 004737 013076 FT6X: JSR PC,ITER ;IF NOT: BR
1240 004612 000137 003070 JMP TSCD2 ;GO SEE IF ITERATION
1241 ;RETURN TO SCHEDULAR
1242 004616 000240 FT6DE: NOP
1243 004620 032777 020000 173722 BIT #20000,@SWR ;SEE IF PRINT ERROR
1244 004626 001032 BNE FT6DEB ;IF NOT: BR
1245 004630 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1246 004634 013701 000610 MOV EMADDR,R1
1247 004640 004737 014070 JSR PC,TTOUT ;PRINT HEADER
1248 004644 005237 000606 INC HDRFL ;SET FLAG
1249 004650 012704 016473 FT6DEA: MOV #MSG39,R4
1250 004654 004737 014070 JSR PC,TTOUT ;PRINT SILO READ ERROR
1251 004660 012704 016040 MOV #MSG22,R4
1252 004664 004737 014070 JSR PC,TTOUT ;PRINT EXPT TAG
1253 004670 010003 MOV R0,R3
1254 004672 004737 014220 JSR PC,OCTP ;PRINT EXPT
1255 004676 012704 016050 MOV #MSG23,R4
1256 004702 004737 014070 JSR PC,TTOUT ;PRINT RCVD TAG
1257 004706 010103 MOV R1,R3
1258 004710 004737 014220 JSR PC,OCTP ;PRINT RCVD
1259 004714 005777 173630 FT6DEB: TST @SWR ;SEE IF HALT ON ERROR
1260 004720 100001 BPL FT6DEX ;IF NOT: BR
1261 004722 000000 HALT
1262 004724 000207 FT6DEX: RTS PC ;RETURN TO TEST
```



```
1263
1264 ;RH11 SILO TEST 4: SILO OVERFLOW*****
1265
1266 004726 005737 000604 FT7: TST RH17F
1267 004732 001021 BNE FT7X ;IF RH70: BR
1268 004734 012737 020116 000610 MOV #MSFT7,EMADDR ;SET TEST HEADER
1269 004742 012737 004726 000674 MOV #FT7,SCOLP ;SET SCOPE ADDRESS
1270 004750 004737 013246 JSR PC,INIT1 ;GO INIT
1271 004754 012700 000103 MOV #103,R0 ;SET SIZE OF SILO +1
1272 004760 010077 173546 FT7A: MOV R0,@DB ;LOAD SILO
1273 004764 005300 DEC R0 ;SEE IF DONE
1274 004766 001374 BNE FT7A ;IF NOT: BR
1275 004770 005777 173524 TST @CS ;SEE IF DLT IS SET
1276 004774 100004 BPL FT7ER ;IF NOT: BR
1277 004776 004737 013076 FT7X: JSR PC,ITER ;GO SEE IF ITERATION
1278 005002 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1279 005006 012737 016255 000650 FT7ER: MOV #MSG32,ERADD ;SET ERROR CODE
1280 005014 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1281 005020 000766 BR FT7X
```

```
1282  
1283  
1284 ;RH11 SILO TEST 5: SILO RESET*****  
1285 005022 005737 000604 FT10: TST RH17F  
1286 005026 001034 BNE FT10X ;IF RH70: BR  
1287 005030 012737 020146 000610 MOV #MSGFT10,EMADDR ;SET TEST HEADER  
1288 005036 012737 005022 000674 MOV #FT10,SCOLP ;SET SCOPE ADDRESS  
1289 005044 012777 000040 173446 MOV #4,@CS ;INITIALIZE  
1290 005052 012700 000004 MOV #4,R0 ;SET NUMBER OF SILO WRITER  
1291 005056 010077 173450 FT10A: MOV R0,@DB ;WRITE SILO  
1292 005062 005300 DEC R0 ;SEE IF DONE  
1293 005064 001374 BNE FT10A ;IF NOT: BR  
1294 005066 052777 000040 173424 BIS #4,@CS ;INITIALIZE  
1295 005074 012777 177777 *173430 MOV #-1,@DB ;WRITE SILO  
1296 005102 017701 173424 MOV @DB,R1 ;READ SILO 1  
1297 005106 017701 173420 MOV @DB,R1 ;READ SILO 2  
1298 005112 005777 173402 TST @CS ;SEE IF DLT IS SET  
1299 005116 100011 BPL FT10ER ;IF NOT: PC  
1300 005120 004737 013076 FT10X: JSR PC,ITER ;GO SEE * ITERATION  
1301 005124 005737 000726 TST RHOF ;SEE IF RH11 ONLY  
1302 005130 001402 BEQ FT10XX ;IF NOT: BR  
1303 005132 000137 003156 JMP TEND ;ELSE GO TO END  
1304 005136 000137 003070 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR  
1305 005142 012737 016255 000650 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE  
1306 005150 004737 004050 JSR PC,FT3ER ;GO DO ERROR  
1307 005154 000761 BR FT10X
```

```
1308                                     ;NOP TEST*****
1309
1310 005156 000240                               FT11: NOP
1311 005160 012737 005156 000674             MOV #FT11,SCOLP ;SET SCOPE ADDRESS
1312 005166 004737 013246                     JSR PC,INIT1
1313 005172 012737 000300 000716             MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
1314 005200 012737 177777 000620             MOV #-1,FCNT ;SET FC= ALL OVER
1315 005206 012737 177777 000622             MOV #-1,WCNT ;SET WC= ALL OVER
1316 005214 012737 177777 000616             MOV #-1,BADDR ;SET BA= ALL OVER
1317 005222 012737 000001 000636             MOV #1,RDYDX ;SET DELAY
1318 005230 012737 000001 000640             MOV #1,OPDYX ;SET OP DELAY
1319 005236 012737 000001 000710             MOV #1,FUN ;SET NOP FUNCTIONS CODE
1320 005244 004737 012072                     JSR PC,EXEC ;GO EXECUTE COMMAND
1321 005250 000240                               NOP
1322 005252 012737 020177 000610             MOV #MSFT11,EMADDR
1323 005260 004737 012272                     JSR PC,ERCHK ;GO CHECK REGISTER
1324 005264 004737 013076                     JSR PC,ITER ;GO SEE IF ITERATIONS
1325 005270 000137 003070                     JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
1326                                     ;REWIND TEST*****
1327
1328 005274 000240          FT12:  NOP
1329 005276 012737 005274 000674  MOV    #FT12,SCOLP
1330 005304 004737 013246          JSR    PC,INIT1      ;GO INITIALIZE
1331 005310 052777 001700 173224  BIS    #1700,@TC     ;SET TO NRZ,NORMAL
1332 005316 012737 177760 000620  MOV    #-20,FCNT    ;SET FC=20
1333 005324 012737 177770 000622  MOV    #-10,WCNT    ;SET WC=10
1334 005332 012737 021054 000616  MOV    #WDATA,BADDR ;SET BA=WRITE BUFFER
1335 005340 012737 000007 000710  MOV    #7,FUN       ;SET REWIND OP CODE
1336 005346 004737 012072          JSR    PC,EXEC       ;GO EXECUTE COMMAND
1337 005352 000240          NOP
1338 005354 032777 020000 173140  FT12A: BIT    #20000,@DS
1339 005362 001374          BNE    FT12A        ;AWAIT PIP
1340 005364 012737 020217 000610  MOV    #MSFT12,EMADDR
1341 005372 004737 012272          JSR    PC,ERCHK     ;GO CHECK FOR ERROR
1342 005376 004737 013076          JSR    PC,ITER      ;GO SEE IF ITERATION
1343 005402 000137 003070          JMP    TSCD2        ;RETURN TO SCHEDULAR
1344
```

```

1345                                     :WRITE/READ TEST*****
1346
1347 005406 000240 FT13: NOP
1348 005410 012737 000001 000636 MOV #1,RDYDX
1349 005416 012737 000001 000640 MOV #1,OPDYX
1350 005424 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
1351 005432 012737 020242 000610 MOV #MSFT13,EMADDR ;SET TEST HEADER
1352 005440 012737 000001 000720 MOV #1,PATRN
1353 005446 004737 012730 JSR PC,DSUP ;SET UP ALL ONES DATA PATTERN
1354 005452 012737 001700 000716 MOV #1700,UDES ;SET TO 800 BPI NORMAL
1355 005460 004737 012224 FT13A: JSR PC,RWIND ;GO REWIND
1356 005464 012737 177600 000620 MOV #-200,FCNT ;SET FC
1357 005472 012737 177700 000622 MOV #-100,WCNT ;SET WC
1358 005500 012737 021054 000616 MOV #WDATA,BADDR ;SET BA
1359 005506 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1360 005514 012737 015615 000626 MOV #MSG12,ERRP
1361 005522 004737 012072 FT13B: JSR PC,EXEC ;GO EXECUTE COMMAND
1362 005526 005037 000674 CLR SCOLP ;NO SCOPE LOOP
1363 005532 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR
1364 005536 005337 000624 DEC RCNT ;SEE IF DONE ALL
1365 005542 001367 BNE FT13B ;IF NOT:BR
1366 005544 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
1367 005552 012737 022566 000616 MOV #RDATA,BADDR
1368 005560 062737 000200 000616 ADD #200,BADDR ;SET BA
1369 005566 012737 000077 000710 MOV #77,FUN ;SET READ REVERSE OP-CPDE
1370 005574 012737 015633 000626 MOV #MSG13,ERRP
1371 005602 004737 012072 FT13C: JSR PC,EXEC ;GO EXECUTE COMMAND
1372 005606 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR
1373 005612 005337 000624 DEC RCNT ;SEE IF READ ALL
1374 005616 001371 BNE FT13C ;IF NOT:BR
1375 005620 162737 000200 000616 SUB #200,BADDR ;SET BA
1376 005626 012737 000071 000710 MOV #71,FUN ;SET READ FORWARD OP-CODE
1377 005634 012737 015660 000626 MOV #MSG14,ERRP
1378 005642 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
1379 005650 004737 012072 FT13D: JSR PC,EXEC ;GO EXECUTE COMMAND
1380 005654 004737 012272 JSR PC,ERCHK ;GO CHECK ERRORS
1381 005660 005337 000624 DEC RCNT ;SEE IF DONE ALL
1382 005664 001371 BNE FT13D ;IF NOT:BR
1383 005666 032737 002000 000716 BIT #2000,UDES ;SEE IF DONE PE
1384 005674 001007 BNE FT13X ;IF SO: BR
1385 005676 012737 002300 000716 MOV #2300,UDES ;SET PE MODE
1386 005704 012737 000100 000624 MOV #100,RCNT ;RESET RECORD COUNT
1387 005712 000662 BR FT13A ;GO DO NEXT DENSITY
1388 005714 000137 003070 FT13X: JMP *SCD2 ;RETURN TO SCHEDULAR
  
```

```

1389                                     :SPACE TEST*****
1390
1391 005720 000240          FT14:  NOP
1392 005722 012737 020271 000610  MOV      #MSFT14,EMADDR  ;SET TEST HEADER
1393 005730 012737 001700 000716  MOV      #1700,UDES    ;SET NRZ,NORMAL
1394 005736 004737 012224          FT14A1: JSR      PC,RWIND   ;GO INITIALIZE
1395 005742 012737 000100 000624  MOV      #100,RCNT    ;SET NUMBER OF RECORDER
1396 005750 012737 177777 021054  MOV      #-1,WDATA    ;SET DATA PATTERN
1397 005756 012737 177700 000620  MOV      #-100,FCNT   ;PRESET FRAME CNT
1398 005764 012737 177740 000622  MOV      #-40,WCNT    ;PRESET WORD CNT
1399 005772 004737 013246          FT14A:  JSR      PC,INIT1  ;GO REWIND
1400 005776 012737 001000 000640  MOV      #1000,OPDYX
1401 006004 012737 040000 000636  MOV      #40000,RDYDX
1402 006012 012737 000061 000710  MOV      #61,FUN      ;SET WRITE OP-CODE
1403 006020 012737 102300 000660  MOV      #102300,STMSK ;MASK DATA RELATED ERRORS
1404 006026 052777 000010 172464  BIS      #10,@CS      ;INHIBIT BUS ADDRESS INCREMENT
1405 006034 004737 012072          JSR      PC,EXEC      ;GO EXECUTE COMMAND
1406 006040 012737 016732 000626  MOV      #MSG46,ERRP  ;SET ERROR CODE
1407 006046 004737 012272          JSR      PC,ERCHK     ;GO CHECK ERRORS
1408 006052 005737 000712          TST      SERFL        ;SEE IF ERROR
1409 006056 001402          BEQ      FT14A2       ;IF NOT: BR
1410 006060 000137 006544          JMP      FT14X        ;ELSE EXIT
1411 006064 005337 000620          FT14A2: DEC      FCNT    ;BUMP FC
1412 006070 032737 000001 000620  BIT      #1,FCNT      ;SEE IF SHOULD BUMP WC
1413 006076 001403          BEQ      FT14A3       ;IF NOT: BR
1414 006100 162737 000001 000622  SUB      #1,WCNT      ;BUMP WC
1415 006106 005337 000624          FT14A3: DEC      RCNT    ;SEE IF DONE ALL
1416 006112 001327          BNE      FT14A        ;WRITE ALL RECORDS
1417 006114 012737 000100 000632  MOV      #100,RRD     ;PRESET RECORD POSITION
1418 006122 012737 000176 000634  MOV      #176,RFD
1419 006130 012737 177701 000642  MOV      #-77,SCNT   ;SET SPACE AMOUNT
1420 006136 012737 000033 000710  FT14B:  MOV      #33,FUN  ;SET OP-CODE SPACE REVERSE
1421 006144 004737 012072          JSR      PC,EXEC      ;GO EXECUTE COMMAND
1422 006150 012737 017003 000626  MOV      #MSG48,ERRP  ;SET ERROR CODE
1423 006156 004737 012272          JSR      PC,ERCHK     ;GO CHECK ERRORS
1424 006162 005737 000712          TST      SERFL        ;SEE IF ERROR
1425 006166 001166          BNE      FT14X        ;IF SO: BR
1426 006170 004737 006264          JSR      PC,FT14RR    ;GO READ REVERSE + CHECK DATA
1427 006174 000240          NOP
1428 006176 012737 000031 000710  MOV      #31,FUN      ;SET SPACE FORWARD OP-CODE
1429 006204 005237 000642          INC      SCNT         ;SET SPACE AMOUNT
1430 006210 001555          BEQ      FT14X        ;IF DONE: BR
1431 006212 004737 012072          JSR      PC,EXEC      ;GO EXECUTE COMMAND
1432 006216 012737 016756 000626  MOV      #MSG47,ERRP  ;SET ERROR CODE
1433 006224 004737 012272          JSR      PC,ERCHK     ;GO CHECK ERROR
1434 006230 005737 000712          TST      SERFL        ;SEE IF ERROR FLAG
1435 006234 001143          BNE      FT14X        ;IF NO: BR
1436 006236 004737 006326          JSR      PC,FT14RF    ;GO READ FORWARD FOR POSITION CHECK
1437 006242 000240          NOP
1438 006244 005237 000642          INC      SCNT         ;DECREMENT SPACE AMOUNT
1439 006250 001535          BEQ      FT14X        ;IF DONE: BR
1440 006252 005237 000632          INC      RRD         ;BUMP DATA EXPT
1441 006256 005337 000634          DEC      RFD         ;BUMP DATA EXPT
1442 006262 000725          BR       FT14B
1443 006264 000240          FT14RR: NOP
1444 006266 012737 022566 000616  MOV      #RDATA,BADDR ;SET BA
  
```

1445	006274	012737	000077	000710	MOV	#77,FUN	:SET READ REVERSE OP-CODE
1446	006302	004737	012072		JSR	PC,EXEC	:GO EXECUTE COMMAND
1447	006306	000240			NOP		
1448	006310	013705	000632		MOV	RRD,R5	
1449	006314	020577	172176		CMP	R5,@FC	:SEE IF CORRECT RECORD
1450	006320	001020			BNE	FT14RER	:IF NOT: BR
1451	006322	000137	006354		JMP	FT14EC	:GO CLEAR RH11 ERROR BIT
1452	006326	000240			FT14RF: NOP		
1453	006330	012737	000071	000710	MOV	#71,FUN	:SET READ FORWARD OP-CODE
1454	006336	004737	012072		JSR	PC,EXEC	:GO EXECUTE COMMAND
1455	006342	013705	000634		MOV	RFD,R5	
1456	006346	020577	172144		CMP	R5,@FC	:SEE IF CORRECT RECORD
1457	006352	001003			BNE	FT14RER	:IF NOT: BR
1458	006354	004737	013246		FT14EC: JSR	PC,INIT1	:CLEAR RH
1459	006360	000207			RTS	PC	:RETURN
1460	006362	000240			FT14RER: NOP		
1461	006364	032777	020000	172156	BIT	#2000,@SWR	:SEE IF PRINT INHIBITED
1462	006372	001060			BNE	FT14R3	:IF SO: BR
1463	006374	012704	020271		MOV	#MSG14,R4	
1464	006400	004737	014070		JSR	PC,TTOUT	:PRINT HEADER
1465	006404	012704	015373		MOV	#MSG9,R4	
1466	006410	004737	014070		JSR	PC,TTOUT	:PRINT ERROR TYPE
1467	006414	012704	016025		MOV	#MSG20,R4	:SET NRZ TAG POINTER
1468	006420	032737	002000	000716	BIT	#2000,UDES	:SEE IF PE
1469	006426	001402			BEQ	FT14R0	:IF NOT: BR
1470	006430	012704	016033		MOV	#MSG21,R4	:ELSE SET PE TAG POINTER
1471	006434	004737	014070		FT14R0: JSR	PC,TTOUT	:PRINT TAG
1472	006440	032737	000002	000710	BIT	#2,FUN	:SEE IF READ REVERSE
1473	006446	001003			BNE	FT14R1	:IF SO: BR
1474	006450	012704	016005		MOV	#MSG17,R4	
1475	006454	000402			BR	FT14R2	:GO PRINT
1476	006456	012704	015765		FT14R1: MOV	#MSG16,R4	
1477	006462	004737	014070		FT14R2: JSR	PC,TTOUT	:PRINT FRWD/REV
1478	006466	012704	016040		MOV	#MSG22,R4	
1479	006472	004737	014070		JSR	PC,TTOUT	:PRINT EXPT TAG
1480	006476	010503			MOV	R5,R3	
1481	006500	042703	177700		BIC	#177700,R3	:MASK RECORD NUMBER
1482	006504	004737	014220		JSR	PC,OCTP	:PRINT EXPT RECORD NUMBER
1483	006510	012704	016050		MOV	#MSG23,R4	
1484	006514	004737	014070		JSR	PC,TTOUT	:PRINT RCVD TAG
1485	006520	017703	171772		MOV	@FC,R3	
1486	006524	042703	177700		BIC	#177700,R3	:MASK RECORD NUMBER
1487	006530	004737	014220		JSR	PC,OCTP	:PRINT ACTUAL RECORD NUMBER
1488	006534	005777	172010		FT14R3: TST	@SWR	:SEE IF HALT ON ERROR
1489	006540	100001			BPL	FT14X	:IF NOT: BR
1490	006542	000000			HALT		
1491	006544	032737	002000	000716	FT14X: BIT	#2000,UDES	:SEE IF DONE PE
1492	006552	001005			BNE	FT14XX	:IF SO: BR
1493	006554	012737	002300	000716	MOV	#2300,UDES	:SET TO PE
1494	006562	000137	005736		JMP	FT14A1	:DO IN PE
1495	006566	000137	003070		FT14XX: JMP	TSCD2	:RETURN TO SCHEDULAR

```
1496                                     ;ERASE TEST*****
1497
1498 006572 000240          FT15:  NOP
1499 006574 005037 000660          CLR      STMSK
1500 006600 012737 000100 000636  MOV      #100,RDYDX
1501 006606 012737 000010 000640  MOV      #10,OPDYX
1502 006614 012737 020313 000610  MOV      #MSFT15,EMADDR ;SET TEST HEADER
1503 006622 004737 012224          JSR      PC,RWIND ;REWIND
1504 006626 012737 022566 000616  MOV      #RDATA,BADDR ;SET BA
1505 006634 012737 001700 000716  MOV      #1700,UDES ;SET NRZ, NORMAL
1506 006642 012737 000025 000710  FT15A: MOV      #25,FUN ;SET ERASE OP-CODE
1507 006650 012737 000454 000624  MOV      #300,RCNT ;++B SET TO ERASE 300 TIMES
1508 006656 004737 012072          FT15B: JSR      PC,EXEC ;GO EXECUTE COMMAND
1509 006662 012737 016732 000626  MOV      #MSG46,ERRP ;SET ERROR CODE
1510 006670 004737 012272          JSR      PC,ERCHK ;GO CHECK ERRORS
1511 006674 005737 000712          TST      SERFL ;SEE IF ANY ERRORS
1512 006700 001032          BNE      FT15X ;IF SO EXIT
1513 006702 005337 000624          DEC      RCNT ;SEE IF DONE ERASING
1514 006706 001363          BNE      FT15B ;IF NOT: BR
1515 006710 000240          NOP
1516 006712 004737 012224          JSR      PC,RWIND ;REWIND
1517 006716 012737 177600 000622  MOV      #-200,WCNT ;SET WC
1518 006724 012737 000071 000710  MOV      #71,FUN ;SET READ FORWARD OP-CODE
1519 006732 012737 000040 000636  MOV      #40,RDYDX ;SET DELAY
1520 006740 004737 012072          JSR      PC,EXEC ;GO EXECUTE COMMAND
1521 006744 000240          NOP
1522 006746 012737 017404 000626  MOV      #MSG60,ERRP ;SET ERROR CODE
1523 006754 012737 020000 000660  MOV      #20000,STMSK
1524 006762 004737 012272          JSR      PC,ERCHK ;GO CHECK ERRORS
1525 006766 000137 003070          FT15X: JMP      TSCD2 ;RETURN TO SCHEDULAR
```



```
1526                                     ;TAPE MARK WRITE/READ TEST*****
1527
1528 006772 000240                      FT16:  NOP
1529 006774 012737 000001 000636      MOV    #1,RDYDX
1530 007002 012737 001000 000640      MOV    #1000,OPDYX
1531 007010 012737 020335 000610      MOV    #MSFT16,EMADDR ;SET HEADER
1532 007016 012737 001700 000716      MOV    #1700,UDES    ;SET TO NRZ,NORMAL,ODD
1533 007024 004737 012224                      FT16A: JSR    PC,RWND    ;INIT AND REWIND SLAVE
1534 007030 012737 177760 000620      FT16B: MOV    #-20,FCNT  ;FC=20
1535 007036 012737 177770 000622      MOV    #-10,WCNT    ;WC=10
1536 007044 012737 000027 000710      MOV    #27,FUN      ;SET WRITE TAPE MARK OP-CODE
1537 007052 004737 012072                      JSR    PC,EXEC      ;GO EXECUTE COMMAND
1538 007056 012737 001000 000660      MOV    #1000,STMSK  ;SET FOR FCE MASK
1539 007064 012737 015705 000626      MOV    #MSG15,ERRP  ;SET ERROR CODE
1540 007072 004737 012272                      JSR    PC,ERCHK     ;GO CHECK ERROR
1541 007076 004737 012672                      JSR    PC,TMCHK     ;GO SEE IF TM SET
1542 007102 012737 000077 000710      MOV    #77,FUN      ;SET READ REVERSE OP-CODE
1543 007110 004737 012072                      JSR    PC,EXEC      ;GO EXECUTE COMMAND
1544 007114 012737 001000 000660      MOV    #1000,STMSK  ;SET FCE ERROR MASK
1545 007122 012737 015633 000626      MOV    #MSG13,ERRP  ;SET ERROR CODE
1546 007130 004737 012272                      JSR    PC,ERCHK     ;GO CHECK ERRORS
1547 007134 004737 012672                      JSR    PC,TMCHK     ;GO SEE IF TM SET
1548 007140 012737 000071 000710      MOV    #71,FUN      ;SET READ FORWARD OP-CODE
1549 007146 004737 012072                      JSR    PC,EXEC      ;GO EXECUTE COMMAND
1550 007152 012737 015660 000626      MOV    #MSG14,ERRP  ;SET ERROR CODE
1551 007160 004737 012272                      JSR    PC,ERCHK     ;TO CHECK ERRORS
1552 007164 004737 012672                      JSR    PC,TMCHK     ;GO SEE IF TM SET
1553 007170 032737 002000 000716      BIT    #2000,UDES   ;SEE IF DONE PE
1554 007176 001004                      BNE    FT16X        ;IF SO: BR
1555 007200 012737 002300 000716      MOV    #2300,UDES   ;SET PE, NORMAI
1556 007206 000706                      BR     FT16A        ;DO IN PE
1557 007210 004737 013076      FT16X: JSR    PC,ITER  ;DO ITERATIONS
1558 007214 000137 003070      JMP    TSCD2        ;RETURN TO SCHEDULAR
1559
```

```

1560
1561
1562
1563 007220 005037 000624 FT17: CLR RCNT
1564 007224 012737 020376 000610 MOV #MSFT17,EMADDR ;SET HEADER
1565 007232 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ
1566 007240 004737 012224 FT17A: JSR PC,RWND ;REWIND TAPE
1567 007244 012737 000027 000710 FT17B: MOV #27,FUN
1568 007252 012737 040000 0G0636 MOV #40000,RDYDX ;SET DRY DELAY
1569 007260 012737 040000 000640 MOV #40000,OPDYX ;SET OP DELAY
1570 007266 004737 012072 JSR PC,EXEC ;GO WRITE TM
1571 007272 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1572 007300 012737 015705 000626 MOV #MSG15,ERRP ;SET ERROR TYPE
1573 007306 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR
1574 007312 005737 000712 TST SERFL ;SEE IF ERROR
1575 007316 001137 BNE FT17X ;IF SO: BR
1576 007320 004737 012672 JSR PC,TMCHK ;GO SEE IF TM SET
1577 007324 000240 NOP
1578 007326 000240 NOP
1579 007330 032737 000100 000624 BIT #100,RCNT ;SEE IF DONE PATTERN
1580 007336 001045 BNE FT17D ;IF SO: BR
1581 007340 062737 000020 000624 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1582 007346 013737 000624 000652 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1583 007354 012737 177600 000622 MOV #-200,WCNT ;WC=128
1584 007362 012737 177400 000620 MOV #-400,FCNT ;FC=256
1585 007370 012737 021054 000616 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1586 007376 012737 000061 000710 MOV #61,FUN ;SET WRITE OP CODE
1587 007404 000240 FT17C: NOP
1588 007406 000240 NOP
1589 007410 004737 012072 JSR PC,EXEC ;GO WRITE
1590 007414 012737 015615 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1591 007422 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1592 007430 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR
1593 007434 005737 000712 TST SERFL ;SEE IF ERROR
1594 007440 001066 BNE FT17X ;IF SO: BR
1595 007442 005337 000652 DEC TEMP1 ;SEE IF DONE ALL
1596 007446 001356 BNE FT17C ;IF NOT: BR
1597 007450 000675 BR FT17B ;ELSE GO DO TM
1598 007452 000240 FT17D: NOP
1599 007454 012737 000033 000710 MOV #33,FUN ;SET SPACE REVERSE
1600 007462 012737 015765 000626 MOV #MSG16,ERRP ;SET ERROR CODE
1601 007470 012737 177600 000642 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1602 007476 012737 000005 000624 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1603 007504 004737 013246 FT17E: JSR PC,INIT1 ;GO INIT
1604 007510 004737 012072 JSR PC,EXEC ;GO SPACE
1605 007514 012737 001000 000660 MOV #1000,STMSK ;SET ERROR MASK
1606 007522 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR
1607 007526 005737 000712 TST SERFL ;SEE IF ERROR
1608 007532 001031 BNE FT17X ;IF SO: BR
1609 007534 004737 012672 JSR PC,TMCHK ;GO SEE IF TM SET
1610 007540 005337 000624 DEC RCNT ;SEE IF DONE SPACES
1611 007544 001357 BNE FT17E ;IF NOT: BR
1612 007546 022737 000031 000710 CMP #31,FUN ;SEE IF DONE FORWARD
1613 007554 001407 BEQ FT17F ;IF SO: BR
1614 007556 012737 016005 000626 MOV #MSG17,ERRP ;SET ERROR CODE
1615 007564 012737 000031 000710 MOV #31,FUN ;SET TO SPACE FORWARD
  
```

1616	007572	000736			BR	FT17D1		:DO FORWARD
1617	007574	032737	002000	000716	FT17F:	BIT	#2000,UDES	:SEE IF DONE PE
1618	007602	001005			BNE	FT17X		:IF SO: BR
1619	007604	012737	002300	000716	MOV	#2300,UDES		:SET TO PE
1620	007612	000137	007240		JMP	FT17A		:GO PE
1621	007616	000137	003070		FT17X:	JMP	TSCD2	:RETURN TO SCHEDULAR

```
1622  
1623  
1624 ;WRITE CHECK TEST*****  
1625 007622 000240 FT20: NOP  
1626 007624 012737 020424 000610 MOV #MSFT20,EMADDR ;SET HEADER  
1627 007632 012737 001700 000716 MOV #1700,UDES ;SET UNIT DESCRIPTION  
1628 007640 004737 012224 FT20A: JSR PC,RWIND ;INIT AND REWIND SLAVE  
1629 007644 012737 000003 000720 MOV #3,PATRN  
1630 007652 004737 012730 JSR PC,DSUP ;GO SET PATTERN 3  
1631 007656 012737 021054 000616 MOV #WDATA,BADDR ;SET BA  
1632 007664 012737 177400 000620 MOV #-400,FCNT ;SET FC  
1633 007672 012737 177600 000622 MOV #-200,WCNT ;SET WC  
1634 007700 012737 000061 000710 MOV #61,FUN ;SET WRITE OP CODE  
1635 007706 004737 012072 JSR PC,EXEC ;GO WRITE RECORD  
1636 007712 012737 016732 000626 MOV #MSG46,ERRP ;SET ERROR CODE  
1637 007720 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR  
1638 007724 005737 000712 TST SERFL ;SEE IF ERORR  
1639 007730 001042 BNE FT20X ;IF SO: BR  
1640 007732 012737 015765 000626 MOV #MSG16,ERRP ;SET REVERSE ERROR TAG  
1641 007740 012737 000057 000710 MOV #57,FUN ;SET REVERSE WRITE CHECK OP-CODE  
1642 007746 062737 000376 000616 ADD #376,BADDR ;SET BA FOR REVERSE CHECK  
1643 007754 004737 012072 JSR PC,EXEC ;GO DO REVERSE CHECK  
1644 007760 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR  
1645 007764 012737 016005 000626 FT20B: MOV #MSG17,ERRP ;SET FORWARD TAG  
1646 007772 012737 000051 000710 MOV #51,FUN ;SET FORWARD CHECK OP CODE  
1647 010000 162737 000376 000616 SUB #376,BADDR ;SET BA FOR FORWARD CHECK  
1648 010006 004737 012072 JSR PC,EXEC ;GO DO FORWARD CHECK  
1649 010012 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR  
1650 010016 032737 002000 000716 FT20C: BIT #2000,UDES ;SEE IF DONE PE  
1651 010024 001004 BNE FT20X ;IF SO: BR  
1652 010026 012737 002300 000716 MOV #2300,UDES ;ELSE SET PE  
1653 010034 000701 BR FT20A ;DO IN PE  
1654 010036 004737 013076 FT20X: JSR PC,ITER ;DO ITERATIONS  
1655 010042 000137 003070 JMP TSC02 ;RETURN TO SCHEDULAR
```

```

1656
1657
1658 ;ERASE HEAD TEST*****
1659 010046 012737 020455 000610 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1660 010054 004737 012224 FT21A: JSR PC,RWND ;GO REWIND
1661 010060 012737 000003 000720 MOV #3,PATRN
1662 010066 004737 012730 JSR PC,DSUP ;GO SET PATTERN 3
1663 010072 012737 021054 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1664 010100 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1665 010106 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1666 010114 012737 001700 000716 MOV #1700,UDES ;SET NRZ, NORMAL
1667 010122 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1668 010130 004737 012072 JSR PC,EXEC ;GO DO WRITE 1
1669 010134 012737 015615 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1670 010142 004737 012272 JSR PC,ERCHK ;GO CHECK FOR ERROR
1671 010146 004737 012072 JSR PC,EXEC ;YES DO WRITE 2
1672 010152 004737 012272 JSR PC,ERCHK ;YES CHECK FOR ERROR
1673 010156 000240 NOP
1674 010160 004737 012224 JSR PC,RWND ;GO REWIND
1675 010164 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400(10)
1676 010172 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200(10)
1677 010200 004737 012072 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1678 010204 000240 FT21SCP:NOP
1679 010206 004737 012224 JSR PC,RWND ;REWIND
1680 010212 012737 022566 000616 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1681 010220 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400
1682 010226 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200
1683 010234 012737 000071 000710 MOV #71,FUN ;SET READ OP-CODE
1684 010242 004737 012072 JSR PC,EXEC ;GO READ RECORD 1
1685 010246 012737 015660 000626 MOV #MSG14,ERRP ;SET ERROR CODE
1686 010254 004737 012272 JSR PC,ERCHK ;GO CHECK FOR ERROR
1687 010260 000240 NOP
1688 010262 052777 000010 170230 BIS #10,@CS ;INHIBIT BA INCREMENT
1689 010270 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
1690 010276 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
1691 010304 004737 012072 JSR PC,EXEC ;GO READ RECORD 2
1692 010310 022777 001440 170200 CMP #800.,@FC ;SEE IF READ RECORD 2 OK
1693 010316 001424 BEQ FT21X ;IF SO: BR
1694 010320 022777 001441 170170 CMP #801.,@FC ;BRANCH IF IN GREY AREA
1695 010326 001420 BEQ FT21X
1696 010330 022777 001440 170160 1$: CMP #800.,@FC ;BRANCH IF ERASE HEAD REVERSED
1697 010336 101404 BLOS FT21B ;IF SO: BR
1698 010340 012737 016625 000650 MOV #MSG44,ERADD ;SET ERASE HEAD INOPERATIVE ERROR CODE
1699 010346 000403 BR FT21C
1700 010350 012737 016655 000650 FT21B: MOV #MSG45,ERADD ;SET ERASE HEAD REVERSED ERROR CODE
1701 010356 012737 010204 000674 FT21C: MOV #FT21SCP,SCOLP ;SET SCOPE ADDRESS
1702 010364 004737 004050 JSR PC,FT3ER ;GO PRINT ERROR
1703 010370 004737 013076 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1704 010374 000137 003070 JMP TS:CD2 ;RETURN TO SCHEDULAR
1705
1706

```

```
1707 ;BUFFERED COMMAND TEST*****
1708
1709 010400 012737 020504 000610 FT22: MOV #MSFT22,EMADDR ;SET TEST HEADER
1710 010406 004737 012224 JSR PC,RWIND ;GO REWIND
1711 010412 012700 000003 MOV #3,R0 ;SET NUMBER OF WRITES
1712 010416 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ NORMAL
1713 010424 012737 021054 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1714 010432 012737 177000 000620 MOV #-1000,FCNT ;SET FC=1000
1715 010440 012737 177400 000622 MOV #-400,WCNT ;SET WC=400
1716 010446 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1717 010454 004737 012072 FT22A: JSR PC,EXEC ;GO DO WRITE
1718 010460 005300 DEC R0 ;SEE IF DONE ALL
1719 010462 001374 BNE FT22A ;IF NOT: BR
1720 010464 000240 NOP
1721 010466 012777 000007 170014 MOV #7,@C1 ;START REWIND
1722 010474 032777 000200 170020 FT22B: BIT #200,@DS
1723 010502 001774 BEQ FT22B
1724 010504 004737 013246 JSR PC,INIT1 ;INITIALIZE
1725 010510 012737 000010 000636 MOV #10,RDYDX ;SET LONG READY DELAY
1726 010516 004737 012072 JSR PC,EXEC ;ISSUE BUFFERED WRITE
1727 010522 000240 NOP
1728 010524 012737 017030 000626 MOV #MSG49,ERRP ;SET ERROR CODE
1729 010532 012737 102300 000660 MOV #102300,STMSK ;MARK DATA ERROR
1730 010540 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR
1731 010544 032777 000002 167750 BIT #2,@DS ;SEE IF BOT IS SET
1732 010552 001410 BEQ FT22X ;IF NOT: BR
1733 010554 012737 017056 000650 MOV #MSG50,ERADD ;SET ERROR CODE
1734 010562 012737 010400 000674 MOV #FT22,SCOLP
1735 010570 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1736 010574 004737 013076 FT22X: JSR PC,ITER ;GO SEE IF ITERATION
1737 010600 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULAR
1738
1739
```

```

1740 ;READ-IN PRESET TEST*****
1741
1742 010604 005737 000614 FT23: TST SLVN ;SEE IF SLAVE SELECT=0
1743 010610 001103 BNE FT23X ;IF NOT:BR
1744 010612 012737 020541 000610 MOV #MSFT23,EMADDR ;SET TEST HEADER
1745 010620 004737 013246 JSR PC,INIT1 ;GO INIT
1746 010624 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ NORMAL
1747 010632 012737 021054 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1748 010640 012737 177400 000620 MOV #-400,FCNT ;SET FC=400
1749 010646 012737 177600 000622 MOV #-200,WCNT ;SET WC=200
1750 010654 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1751 010662 004737 012072 JSR PC,EXEC ;GO DO WRITE
1752 010666 000240 NOP
1753 010670 004737 013246 JSR PC,INIT1 ;INITIALIZE
1754 010674 012737 000021 000710 MOV #21,FUN ;SET READ-IN PRESET OP CODE
1755 010702 004737 012072 JSR PC,EXEC ;GO DO COMMAND
1756 010706 005000 CLR R0
1757 010710 012703 000004 MOV #4,R3 ;SET MULT
1758 010714 032777 020000 167600 FT23A: BIT #20000,ADS ;SEE IF PIP RESET
1759 010722 001404 BEQ FT23B ;IF SO: BR
1760 010724 005300 DEC R0
1761 010726 001372 BNE FT23A ;AWAIT PIP RESET
1762 010730 005303 DEC R3
1763 010732 001370 BNE FT23A ;DELAY
1764 010734 032777 000002 167560 FT23B: BIT #2,ADS ;SEE IF BOT
1765 010742 001010 BNE FT23C ;IF SO: BR
1766 010744 012737 017114 000650 MOV #MSG51,ERADD ;SET ERROR CODE
1767 010752 012737 010604 000674 MOV #FT23,SCOLP
1768 010760 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1769 010764 012701 141000 FT23C: MOV #141000,R1 ;SET EXPT TC
1770 010770 013700 000542 MOV TC,R0 ;SET TC ADDRESS
1771 010774 020110 CMP R1,(R0) ;SEE IF EXPT=RCVD
1772 010776 001410 BEQ FT23X ;IF SO: BR
1773 011000 012737 017150 000650 MOV #MSG52,ERADD ;SET ERROR CODE
1774 011006 012737 010604 000674 MOV #FT23,SCOLP ;CLEAR SCOPE ADDRESS
1775 011014 004737 003560 JSR PC,FT2ER ;GO DO ERROR
1776 011020 000137 003070 FT23X: JMP TSCD2 ;RETURN TO SCHEDULAR
1777
1778

```

```
1779
1780
1781          ;AUTO-DENSITY SELECT TEST: WRITE-NRZ,READ-PE
1782 011024 012737 020574 000610 FT24:  MOV  #MSFT24,EMADDR ;SET ERROR MSG HEADER
1783 011032 004737 012224          JSR  PC,RWIND ;REWIND SLAVE
1784 011036 012737 000001 000720  MOV  #1,PATRN ;SELECT PATTERN
1785 011044 004737 012730          JSR  PC,DSUP ;GO DO DATA SETUP
1786 011050 012737 021054 000616  MOV  #WDATA,BADDR ;SET BUS ADDRESS,
1787 011056 012737 177400 000620  MOV  #-400,FCNT ;FRAME COUNT,
1788 011064 012737 177600 000622  MOV  #-200,WCNT ;WORD COUNT,
1789 011072 012737 001700 000716  MOV  #1700,UDES ;& SLAVE DESC = NRZ NORMAL
1790 011100 012737 000061 000710  MOV  #61,FUN ;LOAD OP CODE WRITE FWD
1791 011106 004737 012072          JSR  PC,EXEC ;GO EXECUTE COMMAND
1792 011112 012737 016732 000626  MOV  #MSG46,ERRP ;SET ERROR MSG ADDRESS
1793 011120 004737 012272          JSR  PC,ERCHK ;GO CHECK ERRORS
1794 011124 005737 000712          TST  SERFL ;BRANCH IF AN ERROR OCCURRED
1795 011130 001026          BNE  FT24X
1796 011132 004737 012224          JSR  PC,RWIND ;REWIND SLAVE
1797 011136 012737 022566 000616  MOV  #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1798 011144 012737 002300 000716  MOV  #2300,UDES ;SET SLAVE DESC PE,NORMAL
1799 011152 012737 000071 000710  MOV  #71,FUN ;SET OP CODE = READ FWD
1800 011160 004737 012072          JSR  PC,EXEC ;GO READ RECORD
1801 011164 032777 000040 167330  BIT  #40,ADS ;BRANCH IF PES BIT CLEARED
1802 011172 001405          BEQ  FT24X
1803 011174 012737 017503 000650  MOV  #MSG63,ERADD
1804 011202 004737 004050          JSR  PC,FT3ER ;GO PROCESS ERROR
1805 011206 004737 013076          FT24X: JSR  PC,ITER
1806 011212 000137 003070          JMP  TSCD2 ;RETURN TO SCHEDULER
1807
```



```

1808
1809
1810 011216 012737 020652 000610 ;AUTO-DENSITY SELECT TEST: WRITE-PE,READ-NRZ
1811 011224 004737 012224 FT25: MOV #MSFT25,EMADDR ;SET ERROR MESSAGE ADDRESS
1812 011230 012737 000001 000720 JSR PC,RWIND ;REWIND SLAVE
1813 011236 004737 012730 JSR #1,PATRN ;SELECT PATTERN
1814 011242 012737 021054 000616 JSR PC,DSUP ;GO DO DATA SETUP
1815 011250 012737 177400 000620 MOV #WCDATA,BADDR ;SET BUS ADDRESS
1816 011256 012737 177600 000622 MOV #-400,FCNT ;FRAME COUNT,
1817 011264 012737 002300 000716 MOV #-200,WCNT ;WORD COUNT,
1818 011272 012737 000061 000710 MOV #2300,UDES ;& SLAVE DESC = PE,NORMAL
1819 011300 004737 012072 JSR #61,FUN ;LOAD WRITE OP CODE
1820 011304 012737 016732 000626 JSR PC,EXEC ;GO EXECUTE WRITE
1821 011312 004737 012272 JSR #MSG46,ERRP ;SET ERROR MSG HDR
1822 011316 005737 000712 JSR PC,ERCHK ;GO CHECK FOR ERRORS
1823 011322 001026 TST SERFL ;BRANCH IF ERROR OCURRED
1824 011324 004737 012224 BNE FT25X
1825 011330 012737 022566 000616 JSR PC,RWIND ;REWIND SLAVE
1826 011336 012737 001700 000716 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
1827 011344 012737 000071 000710 MOV #1700,UDES ;SET SLAVE DESC = NRZ,NORMAL
1828 011352 004737 012072 JSR #71,FUN ;SET READ FWD OP CODE
1829 011356 032777 000040 167136 JSR PC,EXEC ;GO EXECUTE
1830 011364 001005 BNE #40,ADS ;BRANCH ID PES BIT GOT SET
1831 011366 012737 017534 000650 MOV #MSG64,ERADD
1832 011374 004737 004050 JSR PC,FT3ER ;GO PROCESS ERROR
1833 011400 004737 013076 FT25X: JSR PC,ITER ;ITERATION LOOP
1834 011404 000137 003070 JMP TSCD2 ;RETURN TO SCHEDULER
1835

```

```
1836 ;++B SEQUENTIAL TAPE MARK TEST
1837
1838 011410 000240 FT26: NOP
1839 011412 012737 020730 000610 MOV #MSFT26,EMADDR ;SET TEST ERROR MSG HEADER
1840 011420 012737 001700 000716 MOV #1700,UDES ;SET NRZ
1841 011426 004737 012224 1$: JSR PC,RWIND ;REWIND SLAVE
1842 011432 012737 000027 000710 MOV #27,FUN ;SET WRITE TAPE MARK FUNCTION CODE
1843 011440 004737 012072 JSR PC,EXEC ;GO DO TAPE MARK
1844 011444 005037 000660 CLR STMSK ;CLEAR EXPECTED ERROR MASK
1845 011450 012737 015705 000626 MOV #MSG15,ERRP ;SET ERROR MESSAGE
1846 011456 004737 012272 JSR PC,ERCHK ;GO CHECK FOR ERRORS
1847 011462 004737 012672 JSR PC,TMCHK ;GO CHECK FOR TAPE MARK
1848 011466 005737 000712 TST SERFL ;EXIT TEST IF ERROR DETECTED
1849 011472 001061 BNE FT26X
1850 011474 004737 012072 JSR PC,EXEC ;WRITE SECOND TAPE MARK
1851 011500 012737 015726 000626 MOV #MSG15A,ERRP ;SET ERROR MESSAGE
1852 011506 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR
1853 011512 004737 012672 JSR PC,TMCHK
1854 011516 005737 000712 TST SERFL ;EXIT TEST IF ERROR DETECTED
1855 011522 001045 BNE FT26X
1856 011524 004737 012224 JSR PC,RWIND ;REWIND
1857 011530 012737 000031 000710 MOV #31,FUN ;SET SPACE FORWARD OP CODE
1858 011536 012737 177777 000642 MOV #-1,SCNT ;SET # OF RECORDS TO SPACE
1859 011544 004737 012072 JSR PC,EXEC ;GO SPACE FORWARD
1860 011550 012737 016756 000626 MOV #MSG47,ERRP ;SET SPACE FORWARD ERROR
1861 011556 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR BITS
1862 011562 004737 012672 JSR PC,TMCHK ;GO CHECK IF TAPE MARK DETECTED
1863 011566 005737 000712 TST SERFL ;EXIT TEST IF ERROR DETECTED
1864 011572 001021 BNE FT26X
1865 011574 004737 012072 JSR PC,EXEC ;SPACE TO SECOND TAPE MARK
1866 011600 004737 012272 JSR PC,ERCHK ;GO CHECK ERROR BITS
1867 011604 004737 012672 JSR PC,TMCHK ;CHECK IF TAPE MARK DETECTED
1868 011610 005737 000712 TST SERFL ;EXIT TEST IF ERROR DETECTED
1869 011614 001010 BNE FT26X
1870 011616 032737 002000 000716 BIT #2000,UDES ;EXIT TEST IF PE COMPLETED
1871 011624 001004 BNE FT26X
1872 011626 012737 002300 000716 MOV #2300,UDES ;SET PE MODE
1873 011634 000674 BR 1$
1874 011636 004737 013076 FT26X: JSR PC,ITER
1875 011642 000137 003070 JMP TSCD2
```

```

1876                                     ;REWIND: OFF LINE TEST*****
1877
1878 011646 032777 010000 166674 FT27: BIT #10000,@SWR ;SEE IF IN CONTINUOUS MODE
1879 011654 001104 BNE FT27XX ;IF SO: BR
1880 011656 005737 001662 TST CHNFLG ;BRANCH IF CHAIN MODE
1881 011662 001101 BNE FT27XX
1882 011664 012737 020771 000610 MOV #MSFT27,EMADDR ;SET TEST HEADER
1883 011672 004737 012224 JSR PC,RWIND ;REWIND & SELECT SLAVE
1884 011676 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN (ALL 1'S)
1885 011704 004737 012730 JSR PC,DSUP ;FILL WRITE BUFFER
1886 011710 012737 021054 000616 MOV #WDATA,BADDR ;SET WRITE BUFFER BUS ADDRESS
1887 011716 012737 177400 000620 MOV #-400,FCNT ;SET FRAME COUNT
1888 011724 012737 177600 000622 MOV #-200,WCNT ;SET WORD COUNT
1889 011732 012737 001700 000716 MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
1890 011740 012737 000061 000710 MOV #61,FUN ;SET WRITE COMMAND
1891 011746 004737 012072 JSR PC,EXEC ;GO WRITE A RECORD
1892 011752 004737 013246 JSR PC,INIT1 ;++B CLEAR ANY ERROR BITS
1893 011756 012777 000003 166524 MOV #3,@C1 ;ISSUE REWIND: OFF LINE COMMAND
1894 011764 005037 000674 CLR SCOLP ;CLEAR SCOPE LOOP
1895 011770 012700 000042 MOV #42,R0
1896 011774 005001 1$: CLR R1 ;CLEAR TIMER
1897 011776 005301 2$: DEC R1
1898 012000 001376 BNE R0 ;IF NOT TIMED OUT: BR
1899 012002 005300 DEC R0
1900 012004 001373 BNE 1$ ;IF NOT ALL TIMED OUT: BR
1901 012006 032777 010000 166500 BIT #10000,@DS ;SEE IF MOL IS RESET
1902 012014 001406 BEQ 3$ ;IF SO: BR
1903 012016 012737 017167 000650 MOV #MSG53,ERADD ;SET ERROR CODE
1904 012024 004737 004050 JSR PC,FT3ER ;GO DO ERROR
1905 012030 000412 BR FT27X
1906 012032 013700 000524 3$: MOV ER,R0 ;GET ADDRESS OF ERROR REG
1907 012036 005001 CLR R1 ;RESULT SHOULD BE 0
1908 012040 020110 CMP R1,(R0) ;BRANCH IF ERROR REG 0
1909 012042 001405 BEQ FT27X
1910 012044 012737 017570 000650 MOV #MSG67,ERADD ;SET ERROR MSG HEADER
1911 012052 004737 003560 JSR PC,FT2ER ;GO TYPE ERROR
1912 012056 012704 017214 FT27X: MOV #MSG54,R4
1913 012062 004737 014070 JSR PC,TOUT ;PRINT ON LINE REQUEST
1914 012066 000137 003070 FT27XX: JMP TSCD2 ;RETURN TO SCHEDUL R

```

```

1915
1916
1917
1918 012072 000240 EXEC: NOP
1919 012074 053777 000716 166440 BIS UDES,@TC ;LOAD TAPE CONT
1920 012102 013777 000622 166402 MOV WCNT,@WC ;LOAD WC
1921 012110 013777 000620 166400 MOV FCNT,@FC ;LOAD FC
1922 012116 013777 000616 166370 MOV BADDR,@BA ;LOAD BA
1923 012124 022737 000031 000710 CMP #31,FUN ;SEE IF SPACE FORWARD
1924 012132 001404 BEQ EXECA ;IF SO: BR
1925 012134 022737 000033 000710 CMP #33,FUN ;SEE IF SPACE REVERSE
1926 012142 001003 BNE EXECB ;IF NOT: BR
1927 012144 013777 000642 166344 EXECA: MOV SCNT,@FC ;SET SPACE COUNT
1928 012152 000240 EXECB: NOP
1929 012154 013777 000710 166326 MOV FUN,@C1 ;LOAD OP-CODE + GO
1930 012162 000240 NOP
1931 012164 013703 000636 MOV RDYDX,R3 ;SET DELAY
1932 012170 005004 CLR R4
1933 012172 032777 000200 166322 EXECC: BIT #200,@DS ;SEE IF DRY
1934 012200 001004 BNE EXECX ;IF SO: BR
1935 012202 005304 DEC R4
1936 012204 001372 BNE EXECC
1937 012206 005303 DEC R3 ;DELAY FOR DRY
1938 012210 001370 BNE EXECC
1939 012212 013703 000640 EXECX: MOV OPDYX,R3
1940 012216 005303 EXECXA: DEC R3 ;DELAY
1941 012220 001376 BNE EXECXA
1942 012222 000207 EXECXX: RTS PC ;RETURN TO CALLER
1943

```

```
1944  
1945  
1946 012224 004737 013246  
1947 012230 012777 000007 166252  
1948 012236 032777 000002 166256  
1949 012244 001774  
1950 012246 032777 020000 166246  
1951 012254 001374  
1952 012256 105777 166240  
1953 012262 100375  
1954 012264 004737 013246  
1955 012270 000207  
1956
```

:REWIND SUBROUTINE*****

RWIND:	JSR	PC,INIT1	:INIT SLAVE
	MOV	#7,@C1	:START REWIND
1\$:	BIT	#2,@DS	:WAIT FOR BOT TO SET
	BEQ	1\$	
2\$:	BIT	#20000,@DS	:WAIT FOR PIP TO CLEAR
	BNE	2\$	
3\$:	TSTB	@DS	:++B WAIT FOR DRY
	BPL	3\$:++B
	JSR	PC,INIT1	:INIT
	RTS	PC	:RETURN TO CALLER

```

1957                                     ;ERROR CHECK SUBROUTINE*****
1958
1959 012272 0C5037 000712          ERCHK: CLR      SERFL      ;CLEAR FLAG
1960 012276 017737 166220 000664  MOV     @DS,DSAV    ;SAVE DRIVE STATUS REGISTER
1961 012304 032777 040000 166210  BIT     #40000,@DS ;SEE IF ERROR
1962 012312 001001                    BNE     ERPT        ;IF SO: BR
1963 012314 000207                    RTS     PC           ;RETURN
1964 012316 017704 166202          ERPT:  MOV     @ER,R4   ;GET ERROR REGISTER
1965 012322 032737 002000 0C0716  BIT     #2000,UDES  ;SEE IF PE
1966 012330 001403                    BEQ     2$          ;IF SO: BR
1967 012332 042737 000200 000660  BIC     #200,STMSK  ;RESET PEF MASK
1968 012340 022737 000003 000742  2$:    CMP     #3,JUMPER ;+TEST FOR NON-STANDARD JUMPER
1969 012346 001413                    BEQ     ERPTA1     ;+BRANCH IF STANDARD
1970 012350 022777 011024 166330  CMP     #FT24,@LTADD ;+CHECK FOR TEST 24
1971 012356 001404                    BEQ     1$          ;+BRANCH IF TST24
1972 012360 022777 011216 166320  CMP     #FT25,@LTADD ;+CHECK FOR TEST 25
1973 012366 001003                    BNE     ERPTA1
1974 012370 052737 020000 000660  1$:    BIS     #20000,STMSK ;+SET OPI BIT IN ERROR MASK
1975 012376 043704 000660          ERPTA1: BIC     STMSK,R4  ;MASK DONT CARE BITS
1976 012402 001530                    BEQ     ERPTX      ;IF NO UNEXPECTED ERRORS: BR
1977 012404 012737 000001 000712  ERPTG: MOV     #1,SERFL ;SET FLAG
1978 012412 032777 020000 166130  BIT     #20000,@SWR ;SEE IF SHCULD PRINT ERRORS
1979 012420 001115                    BNE     ERPTD      ;IF NOT: BR
1980 012422 005737 000606          TST     HDRFL      ;SEE IF DONE HEADER
1981 012426 001006                    BNE     ERPTA      ;IF SO: BR
1982 012430 005237 000606          INC     HDRFL      ;SET HEADER FLAG
1983 012434 013704 000610          MOV     EMADDR,R4
1984 012440 004737 014070          JSR     PC,TTOUT   ;PRINT HEADER
1985 012444 013704 000626          ERPTA: MOV     ERRP,R4 ;GET ERROR CODE
1986 012450 001414                    BEQ     ERPTB      ;IF NONE: BR
1987 012452 004737 014070          JSR     PC,TTOUT   ;PRINT ERROR CODE
1988 012456 012704 016025          MOV     #MSG20,R4  ;SET NRZ TAG
1989 012462 032777 002000 166052  BIT     #2000,@TC  ;SEE IF PE
1990 012470 001402                    BEQ     ERPT1A     ;IF NOT: BR
1991 012472 012704 016033          MOV     #MSG21,R4  ;ELSE SET PE TAG
1992 012476 004737 014070          ERPT1A: JSR     PC,TTOUT ;PRINT TAG
1993 012502 013704 000630          ERPTB: MOV     ERRP1,R4 ;SEE IF CODE 2
1994 012506 001402                    BEQ     ERPTB1     ;IF NOT: BR
1995 012510 004737 014070          JSR     PC,TTOUT   ;PRINT CODE 2
1996 012514 032777 004000 166026  ERPTB1: BIT     #4000,@SWR ;SEE IF ITERATION
1997 012522 001010                    BNE     ERPTC      ;IF NOT: BR
1998 012524 012704 017360          MOV     #MSG56,R4
1999 012530 004737 014070          JSR     PC,TTOUT   ;PRINT ITER TAG
2000 012534 013703 000662          MOV     ITCNT,R3
2001 012540 004737 014220          JSR     PC,OC1P    ;PRINT ITERATION
2002 012544 012704 015012          ERPTC: MOV     #MSG1,R4 ;PRINT REGISTER TAG
2003 012550 004737 014070          JSR     PC,TTOUT
2004 012554 017703 165730          MOV     @C1,R3
2005 012560 004737 014206          JSR     PC,OC1PE   ;PRINT CS1
2006 012564 017703 165722          MOV     @WC,R3
2007 012570 004737 014206          JSR     PC,OC1PE   ;PRINT WC
2008 012574 017703 165714          MOV     @BA,R3
2009 012600 004737 014206          JSR     PC,OC1PE   ;PRINT BA
2010 012604 017703 165706          MOV     @FC,R3
2011 012610 004737 014206          JSR     PC,OC1PE   ;PRINT FC
2012 012614 017703 165700          MOV     @CS,R3

```

2013	012620	004737	014206		JSR	PC,OCTPE		;PRINT CS2
2014	012624	017703	165672		MOV	@DS,R3		
2015	012630	004737	014206		JSR	PC,OCTPE		;PRINT DS
2016	012634	017703	165664		MOV	@ER,R3		
2017	012640	004737	014206		JSR	PC,OCTPE		;PRINT ER
2018	012644	017703	165672		MOV	@TC,R3		
2019	012650	004737	014206		JSR	PC,OCTPE		;PRINT TC
2020	012654	005777	165670	ERPTD:	TST	@SWR		;SEE IF HALT ON ERROR
2021	012660	100001			BPL	ERPTX		;IF NOT: BR
2022	012662	000000			HALT			
2023	012664	004737	013246	ERPTX:	JSR	PC,INIT1		;INIT
2024	012670	000207		ERPTXX:	RTS	PC		;RETURN
2025								
2026								

```
2027 ;TAPE MARK STATUS CHECK*****
2028
2029 012672 032737 000004 000664 TMCHK: BIT #4, DSAV ;BRANCH IF TM SET
2030 012700 001012 BNE 1$
2031 012702 005737 000712 TST SERFL ;SEE IF HAD ERROR
2032 012706 001007 BNE 1$ ;IF SO: BR
2033 012710 012737 017370 000630 MOV #MSG57, ERRP1 ;SET ERROR CODE 2
2034 012716 004737 012404 JSR PC, EPPTG ;GO PRINT TM ERROR
2035 012722 005037 000630 CLR ERRP1 ;CLEAR CODE 2 FLAG
2036 012726 000207 1$: RTS PC ;RETURN
2037
2038 ;DATA SETUP ROUTINE*****
2039
2040 012730 000240 DSUP: NOP
2041 012732 012703 021054 DSO: MOV #WDATA, R3 ;R3 = ADDRS OF WRITE BUFFER
2042 012736 013701 000720 MOV PATRN, R1 ;R1 = PATTERN SELECTOR
2043 012742 006301 ASL R1 ;MAKE PATTERN SELECTOR EVEN
2044 012744 004771 000744 JSR PC, @DATBL(R1) ;GO GENERATE PATTERN
2045 012750 012702 000640 MOV #640, R2 ;R2=BUFFER SIZE +2
2046 012754 012701 022566 MOV #RDATA, R1 ;R1=READ DATA START
2047 012760 005021 1$: CLR (R1)+ ;CLEAR BUFFER
2048 012762 005302 DEC R2 ;SEE IF DONE ALL
2049 012764 001375 BNE 1$ ;IF NOT: BR
2050 012766 000207 RTS PC ;EXIT
2051
2052 ;ALL ONES*****
2053
2054 012770 012701 177777 DAT1: MOV #-1, R1 ;R1=DATA
2055 012774 012702 000640 DAT1A: MOV #640, R2 ;R2=WORD COUNT +2
2056 013000 010123 1$: MOV R1, (R3)+ ;LOAD BUFFER
2057 013002 005302 DEC R2 ;SEE IF DONE
2058 013004 001375 BNE 1$ ;IF NOT: BR
2059 013006 000207 RTS PC
2060
2061 ;ALL ZEROS*****
2062
2063 013010 005001 DAT2: CLR R1 ;R1=DATA
2064 013012 000770 BR DAT1A ;LOAD BUFFER
2065
2066 ;ONE/ZERO IN ALTERNATING CHARACTERS*****
2067
2068 013014 012701 125125 DAT3: MOV #125125, R1 ;R1=DATA
2069 013020 000765 BR DAT1A ;LOAD BUFFER
2070
2071 ;ALL BITS 0-377*****
2072
2073 013022 005001 DAT4: CLR R1 ;R1=STARTING DATA
2074 013024 012702 001500 MOV #1500, R2 ;R2=CHARACTER COUNT
2075 013030 110123 1$: MOV R1, (R3)+ ;LOAD BUFFER
2076 013032 105201 INCB R1 ;BUMP DATA
2077 013034 005302 DEC R2 ;SEE IF DONE
2078 013036 001374 BNE 1$ ;IF NOT: BR
2079 013040 000207 RTS PC
2080
```



```
2081
2082
2083
2084
2085 ;SCOPE LOOP ON ERROR SUBROUTINE*****
2086 013042 000240 SCOPE: NOP
2087 013044 032777 040000 165476 BIT #40000,@SWR ;SEE IF LOOP ON ERROR
2088 013052 001001 BNE 1$ ;IF SO: BR
2089 013054 000207 RTS PC ;ELSE EXIT
2090 013056 000240 1$: NOP
2091 013060 005737 000674 TST SCOLP ;SEE IF SCOPE ADDRESS
2092 013064 001001 BNE 2$ ;IF NOT: BR
2093 013066 000207 RTS PC ;ELSE EXIT
2094 013070 022626 2$: CMP (SP)+,(SP)+ ;RESET STACK
2095 013072 000177 165576 JMP @SCOLP ;LOOP ON ERROR
2096
2097 ;TEST ITERATION SUBROUTINE*****
2098
2099 013076 000240 ITER: NOP
2100 013100 032777 004000 165442 BIT #4000,@SWR ;SEE IF ITERATIONS
2101 013106 001403 BEQ 2$ ;IF SO: BR
2102 013110 005037 000662 1$: CLR ITCNT ;CLEAR ITERATION COUNTER
2103 013114 000207 RTS PC ;ELSE EXIT
2104 013116 005737 000730 2$: TST PCNTR ;DO SINGLE SUBTEST ITERATION
2105 013122 001772 BEQ 1$ ;ON FIRST PASS
2106 013124 005237 000662 INC ITCNT ;BUMP COUNTER
2107 013130 023737 000662 000566 CMP ITCNT,ITAMT ;SEE IF DONE ALL
2108 013136 001764 BEQ 1$ ;IF SO: BR
2109 013140 005726 TST (SP)+ ;RESET STACK
2110 013142 017700 16553C MOV @ITRLP,R0 ;SET ITERATION POINTER
2111 013146 000110 JMP (R0) ;GO ITERATE
2112
2113
2114
2115 ;NON-STANDARD JUMPER HANDLER SUBROUTINE*****
2116
2117 013150 010046 NOST: MOV R0,-(SP) ;+SAVF R0
2118 013152 012700 000120 MOV #120,R0 ;+SET UP INDEX
2119 013156 012760 011410 000756 MOV #FT26,TSTTBL(R0) ;+ADJUST SCHEDULAR TEST TABLE
2120 013164 005720 TST (R0)+
2121 013166 012760 011410 000756 MOV #FT26,TSTTBL(R0) ;+OVERLAY TEST LIST
2122
2123 TST (R0)+
2124 013176 012760 011646 000756 MOV #FT27,TSTTBL(R0)
2125 013204 005720 TST (R0)+
2126 013206 012760 011646 000756 MOV #FT27,TSTTBL(R0)
2127 013214 005720 TST (R0)+
2128 013216 012760 003156 000756 MOV #TEND,TSTTBL(R0)
2129 013224 005720 TST (R0)+
2130 013226 012760 000027 000756 MOV #27,TSTTBL(R0)
2131 013234 012737 000027 001120 MOV #27,TLAST
2132 013242 012600 MOV (SP)+,R0 ;RESTO R0
2133 013244 000207 RTS PC
2134
2135 ;INITIALIZE SUBROUTINE*****
2136
```

```
2137 013246 000240
2138 013250 012777 000040 165242
2139 013256 013777 000612 165234
2140 013264 013777 000614 165250
2141 013272 000207
2142
2143
2144
2145 013274 000240
2146 013276 013716 000646
2147 013302 000002
2148

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

;MAG TAPE INTERRUPT HANDLER*****

MTINT: NOP
      MOV RTRN,(SP) ;RETURN TO (RTRN)
      RTI ;RETURN
```

```

2149
2150
2151 ;TTY INTERRUPT HANDLER*****
2152 013304 017746 165244 TTINT: MOV @TKB,-(SP) ;GET CHARACTER
2153 013310 042716 000200 BIC #200,(SP) ;CLEAR PARITY BIT
2154 013314 122716 000003 CMPB #3,(SP) ;BRANCH IF NOT CONTROL C
2155 013320 001010 BNE 1$
2156 013322 005737 001662 TST CHNFLG ;INHIBIT ^C IF CHAIN MODE
2157 013326 001005 BNE 1$
2158 013330 005077 165212 CLR @PSW
2159 013334 000005 RESET
2160 013336 000137 000200 JMP @#200 ;RESTART PROGRAM
2161 013342 122716 000001 1$: CMPB #1,(SP) ;BRANCH IF NOT ^A
2162 013346 001017 BNE 2$
2163 013350 022737 000176 000550 CMP #SWREG,SWR ;BRANCH IF HARDWARE SWR IS INVOKED
2164 013356 001016 BNE 3$
2165 013360 012737 177570 000550 MOV #177570,SWR ;INVOKE HARDWARE SWR
2166 013366 004737 014746 JSR PC,,SAVE ;SAVE REGISTERS ON THE STACK
2167 013372 012704 017644 MOV #MSG70,R4 ;TYPE 'HARDWARE SWR IN USE'
2168 013376 004737 014070 JSR PC,TTOUT
2169 013402 004737 014770 JSR PC,,RESTORE
2170 013406 122716 000007 2$: CMPB #7,(SP) ;BRANCH IF NOT ^G
2171 013412 001005 BNE 4$
2172 013414 012737 000176 000550 3$: MOV #SWREG,SWR ;INVOKE SOFTWARE SWR
2173 013422 004737 014650 JSR PC,GTSWR ;GET SOFTWARE SWITCHES
2174 013426 005726 4$: TST (SP)+ ;POP CHARACTER OFF THE STACK
2175 013430 000002 RTI
2176
2177 ;BUS ADDRESS TRAP HANDLER*****
2178
2179 013432 000240 TRAP: NOP
2180 013434 032777 020000 165106 BIT #20000,@SWR ;SEE IF SHOULD PRINT ERRORS
2181 013442 001020 BNE TRAP2 ;IF NOT: BR
2182 013444 005737 000606 TST HDRFL ;SEE IF DONE HEADER
2183 013450 001006 BNE TRAP1 ;IF SO: BR
2184 013452 005237 000606 INC HDRFL ;ELSE SET HEADER FLAG
2185 013456 013704 000610 MOV EMADDR,R4
2186 013462 004737 014070 JSR PC,TTOUT ;PRINT HEADER
2187 013466 012704 016060 TRAP1: MOV #MSG24,R4
2188 013472 004737 014070 JSR PC,TTOUT ;PRINT ERROR
2189 013476 010103 MOV R1,R3 ;GET ADDRESS THAT CAUSED THE TRAP
2190 013500 004737 014220 JSR PC,OCTP ;PRINT ADDRESS OF TRAP
2191 013504 005777 165040 TRAP2: TST @SWR ;SEE IF HALT ON ERROR
2192 013510 100001 BPL TRAPX ;IF NOT: BR
2193 013512 000000 HALT
2194 013514 022626 TRAPX: CMP (SP)+,(SP)+ ;RESET STACK
2195 013516 012737 003316 000674 MOV #FT1A,SCOLP ;SET SCOPE ADDRESS
2196 013524 004737 013042 JSR PC,SCOPE ;GO SEE IF SCOPE LOOP
2197 013530 005737 000722 TST RHIF ;SEE IF INITIAL ADDRESS TEST
2198 013534 001402 BEQ TRAPXX ;IF NOT: BR
2199 013536 000137 001764 JMP STOB ;ELSE REDO ADDRESS REQUEST
2200 013542 000137 003322 TRAPXX: JMP FT1B ;RETURN TO TEST 1
2201

```

```
2202 :*****
2203 :TTY ENTRY SUBROUTINE:
2204 :
2205 :THIS SUBROUTINE IS USED BY THE TEST CONDITION
2206 :ENTRY ROUTINE TO READ THE RESPONSE ENTERED
2207 :AT THE TTY AND CHECK THEM FOR LEGALITY AND
2208 :LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
2209 :(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
2210 :THE CALLING ROUTINE.
2211 :IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
2212 :A QUESTION MARK IS TYPED (?) AND THE RESPONSE
2213 :MAY BE REENTERED.
2214 :ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
2215 :MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
2216 :CARRIAGE RETURN
2217 :*****
2218
2219 013546 010146 TTR: MOV R1, -(SP) ;SAVE CHAR COUNT ON STACK
2220 013550 011601 10$: MOV (SP), R1 ;RESTORE CHAR COUNT (FOR ^U)
2221 013552 005037 000652 CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
2222 013556 005000 CLR R0
2223 013560 004737 014026 1$: JSR PC, TTIN ;GO READ CHARACTER
2224 013564 122737 000003 000602 CMPB #3, TIB ;BRANCH IF NOT ^C
2225 013572 001003 BNE 11$
2226 013574 000005 RESET ;RESET
2227 013576 000137 000200 JMP @#200 ;RESTART
2228 013602 122737 000015 000602 11$: CMPB #15, TIB ;SEE IF CR
2229 013610 001004 BNE 2$ ;IF NOT: BR
2230 013612 005737 000652 TST TEMP1 ;SEE IF FIRST CHARACTER
2231 013616 001471 BEQ 9$ ;IF SO: BR
2232 013620 000457 BR 6$ ;ELSE GO LOAD VALUE
2233 013622 122737 000025 000602 2$: CMPB #25, TIB ;BRANCH IF NOT CONTROL U
2234 013630 001005 BNE 21$
2235 013632 012704 017564 MOV #MSG65, R4 ;TYPE <CR><LF>
2236 013636 004737 014070 JSR PC, TTOUT
2237 013642 000742 BR 10$ ;RESTART
2238 013644 122737 000177 000602 21$: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'
2239 013652 001012 BNE 3$
2240 013654 000241 CLC ;REMOVE LAST CHARACTER
2241 013656 006000 ROR R0
2242 013660 006200 ASR R0
2243 013662 006200 ASR R0
2244 013664 012704 017566 MOV #MSG66, R4 ;TYPE '\ '
2245 013670 004737 014070 JSR PC, TTOUT
2246 013674 005201 INC R1 ;DECREMENT CHAR RECEIVED COUNT
2247 013676 000730 BR 1$ ;GET NEXT CHARACTER
2248 013700 122737 000060 000602 3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
2249 013706 101402 BLOS 4$ ;IF NOT: BR
2250 013710 000137 014006 JMP T1NER ;ELSE GO TO ERROR
2251 013714 122737 000070 000602 4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
2252 013722 101002 BHI 5$ ;IF NOT: BR
2253 013724 000137 014006 JMP T1NER ;ELSE GO TO ERROR
2254 013730 005237 000652 5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
2255 013734 006300 ASL R0
2256 013736 006300 ASL R0 ;SHIFT 3 LEFT
2257 013740 006300 ASL R0
```

```
2258 013742 042737 177770 000602 BIC #177770,TIB ;STRIP ASCII
2259 013750 053700 000602 BIS TIB,R0 ;LOAD CHARACTER
2260 013754 005301 DEC R1 ;SEE IF DONE
2261 013756 001300 BNE 1$ ;IF NOT: BR
2262 013760 020002 6$: CMP R0,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
2263 013762 101402 BLOS 7$ ;IF NOT: BR
2264 013764 000137 014006 JMP TINNER ;ELSE GO TO ERROR
2265 013770 020300 7$: CMP R3,R0 ;SEE IF BELOW MINIMUM LIMIT
2266 013772 101402 BLOS 8$ ;IF NOT: BR
2267 013774 000137 014006 JMP TINNER ;ELSE GO TO ERROR
2268 014000 010015 8$: MOV R0,(R5) ;LOAD VALUE
2269 014002 005726 9$: TST (SP)+ ;POP CHAR COUNT OFF STACK
2270 014004 000207 RTS PC ;EXIT
2271
2272 ;TTY ENTRY ERROR SUBROUTINE*****
2273
2274 014006 012704 015367 TINNER: MOV #MSG7,R4
2275 014012 004737 014070 JSR PC,TTOUT ;PRINT?
2276 014016 005726 TST (SP)+ ;POP CHAR COUNT OFF STACK
2277 014020 162716 000020 SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
2278 014024 000207 RTS PC ;REDO VALUE ENTRY
2279
2280 ;TTY READ SUBROUTINE*****
2281
2282 014026 005277 164520 TTIN: INC @TKS
2283 014032 105777 164514 1$: TSTB @TKS
2284 014036 100375 BPL 1$
2285 014040 117737 164510 000602 MOVB @TKB,TIB
2286 014046 042737 000200 000602 BIC #200,TIB ;STRIP PARITY BIT
2287 014054 013737 000602 000600 MOV TIB,TOB ;MOVE CHAR TO OUTPUT BFR
2288 014062 004737 014170 JSR PC,TOG ;AND TYPE IT
2289 014066 000207 RTS PC
2290
2291 ;TTY OUTPUT SUBROUTINE*****
2292
2293 014070 112437 000600 TTOUT: MOVB (R4)+,TOB
2294 014074 122737 000043 000600 CMPB #43,TOB
2295 014102 001440 BEQ TEX
2296 014104 122737 000045 000600 CMPB #45,TOB
2297 014112 001403 BEQ 1$
2298 014114 004737 014170 JSR PC,TOG
2299 014120 000763 BR TTOUT
2300 014122 112737 000015 000600 1$: MOVB #15,TOB
2301 014130 004737 014170 JSR PC,TOG
2302 014134 012703 000004 MOV #4,R3
2303 014140 005037 000600 2$: CLR TOB
2304 014144 004737 014170 JSR PC,TOG
2305 014150 005303 DEC R3
2306 014152 001372 BNE 2$ ;DO FILLERS
2307 014154 112737 000012 000600 MOVB #12,TOB
2308 014162 004737 014170 JSR PC,TOG
2309 014166 000740 BR TTOUT
2310 014170 105777 164362 TOG: TSTB @TPS
2311 014174 100375 BPL TOG
2312 014176 113777 000600 164354 MOVB TOB,@TPB
2313 014204 000207 TEX: RTS PC
```

```

2314                                     ;OCTAL OUTPUT SUBROUTINE*****
2315
2316 014206 012737 000001 014436 OCTPE: MOV #1,OFL
2317 014214 010304          MOV R3,R4
2318 014216 000410          BR OCTP0
2319 014220 005037 014436 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
2320 014224 010304          OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
2321 014226 001004          BNE OCTP0 ;IF NOT ZERO: BR
2322 014230 004737 014416 JSR PC,OCTPG1 ;ELSE PRINT ZERO
2323 014234 000137 014360 JMP OCTP3 ;SPACE AND EXIT
2324 014240 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
2325 014244 001406          BEQ OCTP1 ;IF NOT: BR
2326 014246 012704 000001 MOV #1,R4
2327 014252 004737 014374 JSR PC,OCTPG ;PRINT 1
2328 014256 000137 014270 JMP OCTP2
2329 014262 005004          OCTP1: CLR R4
2330 014264 004737 014374 JSR PC,OCTPG ;PRINT 0
2331 014270 010304          OCTP2: MOV R3,R4
2332 014272 006004          ROR R4
2333 014274 006004          ROR R4
2334 014276 006004          ROR R4 ;POSITION DIGIT
2335 014300 006004          ROR R4
2336 014302 000304          SWAB R4
2337 014304 004737 014374 JSR PC,OCTPG ;PRINT DIGIT 2
2338 014310 010304          MOV R3,R4
2339 014312 006004          ROR R4
2340 014314 000304          SWAB R4
2341 014316 004737 014374 JSR PC,OCTPG ;PRINT DIGIT 3
2342 014322 010304          MOV R3,R4
2343 014324 006104          ROL R4
2344 014326 006104          ROL R4
2345 014330 000304          SWAB R4
2346 014332 004737 014374 JSR PC,OCTPG ;PRINT DIGIT 4
2347 014336 010304          MOV R3,R4
2348 014340 006004          ROR R4
2349 014342 006004          ROR R4
2350 014344 006004          ROR R4
2351 014346 004737 014374 JSR PC,OCTPG
2352 014352 010304          MOV R3,R4
2353 014354 004737 014374 JSR PC,OCTPG ;PRINT DIGIT 5
2354 014360 012737 000240 000600 OCTP3: MOV #240,TOB
2355 014366 004737 014170 JSR PC,TOG ;PRINT SPACE
2356 014372 000207          RTS PC ;EXIT
2357 014374 042704 177770 OCTPG: BIC #177770,R4
2358 014400 001004          BNE OCTPG0
2359 014402 005737 014436 TST OFL
2360 014406 001001          BNE OCTPG0
2361 014410 000207          RTS PC
2362
2363 014412 005237 014436 OCTPG0: INC OFL
2364 014416 052704 000260 OCTPG1: BIS #260,R4
2365 014422 010437 000600 MOV R4,TOB
2366 014426 004737 014170 JSR PC,TOG
2367 014432 010304          MOV R3,R4
2368 014434 000207          RTS PC
2369 014436 000000          OFL: 0 ;FIRST CHAR FLAG
  
```

```

2370
2371
2372 ;DATA CHARACTER OUTPUT SUBROUTINE*****
2373 014440 005037 000600 DOUT: CLR TOB
2374 014444 012704 000010 MOV #10,R4 ;SET NUMBER TO PRINT
2375 014450 110337 000600 MOVB R3,TOB
2376 014454 105777 164076 1$: TSTB @TPS
2377 014460 100375 BPL 1$
2378 014462 132737 000200 000600 BITB #200,TOB
2379 014470 001404 BEQ 2$
2380 014472 012777 000061 164060 MOV #061,@TPB
2381 014500 000403 BR 3$
2382 014502 012777 000060 164050 2$: MOV #060,@TPB
2383 014510 006137 000600 3$: ROL TOB
2384 014514 005304 DEC R4
2385 014516 001356 BNE 1$
2386 014520 000207 RTS PC
2387
2388 014522 013703 000656 DOUTD: MOV TEMP3,R3
2389 014526 000303 SWAB R3
2390 014530 004737 014440 JSR PC,DOUT
2391 014534 013703 000656 MOV TEMP3,R3
2392 014540 004737 014440 JSR PC,DOUT
2393 014544 000207 RTS PC
2394
2395 ;SERIAL NUMBER PRINT SUBROUTINE*****
2396
2397 014546 010304 SNPT: MOV R3,R4
2398 014550 000304 SWAB R4
2399 014552 006004 ROR R4
2400 014554 006004 ROR R4
2401 014556 006004 ROR R4
2402 014560 006004 ROR R4
2403 014562 004737 014624 JSR PC,SNPG ;GET FIRST DIGIT
2404 014566 010304 MOV R3,R4 ;GO PRINT
2405 014570 000304 SWAB R4 ;GET SECOND DIGIT
2406 014572 004737 014624 JSR PC,SNPG ;GO PRINT
2407 014576 010304 MOV R3,R4
2408 014600 006004 ROR R4
2409 014602 006004 ROR R4
2410 014604 006004 ROR R4
2411 014606 006004 ROR R4
2412 014610 004737 014624 JSR PC,SNPG ;GET THIRD DIGIT
2413 014614 010304 MOV R3,R4 ;GO PRINT
2414 014616 004737 014624 JSR PC,SNPG ;GET FOURTH DIGIT
2415 014622 000207 RTS PC ;GO PRINT
2416 014624 012737 000260 000600 SNPG: MOV #260,TOB ;EXIT
2417 014632 042704 177760 BIC #177760,R4 ;SET BASE = 0
2418 014636 050437 000600 BIS R4,TOB ;MASK DIGIT
2419 014642 004737 014170 JSR PC,TOG ;SET ASCII
2420 014646 000207 RTS PC ;TYPE DIGIT
2421 ;RETURN

```

```
2422
2423 ;ROUTINE TO LOAD NEW VALUE INTO SWITCHES
2424 014650 022737 000176 000550 GTSWR: CMP #SWREG,SWR ;BRANCH IF SOFTWARE SWR
2425 014656 001032 BNE 1$ ;NOT INVOKED
2426 014660 004737 014746 JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
2427 014664 012704 021031 MOV #SMSWR,R4
2428 014670 004737 014070 JSR PC,TTOUT
2429 014674 017703 163650 MOV @SWR,R3
2430 014700 004737 014206 JSR PC,OCTPE
2431 014704 012704 021040 MOV #SMNEW,R4
2432 014710 004737 014070 JSR PC,TTOUT
2433 014714 013705 000550 MOV SWR,R5 ;TTR ROUTINE RETURNS NEW VALUE TO (R5)
2434 014720 012701 000007 MOV #7,R1 ;LIMIT RESPONSE TO 7 CHARS
2435 014724 012702 177777 MOV #177777,R2 ;BETWEEN 0 AND 177777
2436 014730 012703 000000 MOV #0,R3
2437 014734 004737 013546 JSR PC,TTR
2438 014740 004737 014770 JSR PC,.RESTORE ;RESTORE REGISTERS
2439 014744 000207 1$: RTS PC
2440
2441 ;;ROUTINE TO SAVE REGISTERS ON THE STACK
2442 014746 010546 .SAVE: MOV %5,-(SP) ;;R5 IS SAVED AT 12(SP)
2443 014750 010446 MOV %4,-(SP) ;;R4 IS SAVED AT 10(SP)
2444 014752 010346 MOV %3,-(SP) ;;R3 IS SAVED AT 6(SP)
2445 014754 010246 MOV %2,-(SP) ;;R2 IS SAVED AT 4(SP)
2446 014756 010146 MOV %1,-(SP) ;;R1 IS SAVED AT 2(SP)
2447 014760 010046 MOV %0,-(SP) ;;R0 IS SAVED AT (SP)
2448 014762 016646 000014 MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
2449 014766 000207 RTS PC ;;RETURN TO CALLER
2450
2451 ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
2452 014770 012666 000014 .RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
2453 014774 012600 MOV (SP)+,%0
2454 014776 012601 MOV (SP)+,%1
2455 015000 012602 MOV (SP)+,%2
2456 015002 012603 MOV (SP)+,%3
2457 015004 012604 MOV (SP)+,%4
2458 015006 012605 MOV (SP)+,%5
2459 015010 000207 RTS PC ;;RETURN
2460
2461
```



```
2462 ;MESSAGE TABLE*****
2463
2464 015012 041445 030523 020040 MSG1: .ASCII /%CS1 WC BA FC CS2 /
2465 015020 020040 041527 020040
2466 015026 020040 041040 020101
2467 015034 020040 020040 041506
2468 015042 020040 020040 041440
2469 015050 031123 020040 020040
2470 015056 051504 020040 020040 .ASCII /DS ER TC%/
2471 015064 042440 020122 020040
2472 015072 020040 041524 021445
2473 015100 051045 053505 047111 MSG2: .ASCII /%REWIND ERROR-BOT NOT SET WHEN PIP CLEARED#/
2474 015106 020104 051105 047522
2475 015114 026522 047502 020124
2476 015122 047516 020124 042523
2477 015130 020124 044127 047105
2478 015136 050040 050111 041440
2479 015144 042514 051101 042105
2480 015152 043
2481 015153 045 052045 030115 MSG3: .ASCII '%TM03-TE16/TU77 BASIC FUNCTION TEST (CZTECBO)%';++B
2482 015160 026463 042524 033061
2483 015166 052057 033525 020067
2484 015174 040502 044523 020103
2485 015202 052506 041516 044524
2486 015210 047117 052040 051505
2487 015216 020124 041450 052132
2488 015224 041505 030102 022451
2489 015232 054524 042520 036040 .ASCII /TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%/
2490 015240 051103 020076 047524
2491 015246 052040 051105 044515
2492 015254 040516 042524 051040
2493 015262 051505 047520 051516
2494 015270 020105 020046 041536
2495 015276 052040 020117 042522
2496 015304 052123 051101 022524
2497 015312 043
2498 015313 045 042522 044507 MSG4: .ASCII /%REGISTER START = #/
2499 015320 052123 051105 051440
2500 015326 040524 052122 036440
2501 015334 021440
2502 015336 053045 041505 047524 MSG5: .ASCII /%VECTOR = #/
2503 015344 020122 020075 043
2504 015351 045 047105 020104 MSG6: .ASCII /%END OF PASS #/
2505 015356 043117 050040 051501
2506 015364 020123 043
2507 015367 040 020077 043 MSG7: .ASCII / ? #/
2508 015373 045 047520 044523 MSG9: .ASCII /%POSITION ERROR: #/
2509 015400 044524 047117 042440
2510 015406 051122 051117 020072
2511 015414 043
2512 015415 045 051511 041440 MSG10A: .ASCII /%IS CONTROLLER JUMPERED IN NON-STANDARD MODE/<15><12>
2513 015422 047117 051124 046117
2514 015430 042514 020122 052512
2515 015436 050115 051105 042105
2516 015444 044440 020116 047516
2517 015452 026516 052123 047101
```

2518	015460	040504	042122	046440		
2519	015466	042117	006505	012		
2520	015473	124	050131	020105	.ASCII	/TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD: #/
2521	015500	020062	047506	020122		
2522	015506	047516	026516	052123		
2523	015514	047101	040504	042122		
2524	015522	047440	020122	051103		
2525	015530	043040	051117	051440		
2526	015536	040524	042116	051101		
2527	015544	035104	020040	020040		
2528	015552	020040	043			
2529	015555	045	051104	053111	MSG10:	.ASCII /%DRIVE NUMBER: #/
2530	015562	020105	052516	041115		
2531	015570	051105	020072	043		
2532	015575	045	046123	053101	MSG11:	.ASCII /%SLAVE NUMBER: #/
2533	015602	020105	052516	041115		
2534	015610	051105	020072	043		
2535	015615	045	051127	052111	MSG12:	.ASCII /%WRITE ERROR #/
2536	015622	020105	051105	047522		
2537	015630	020122	043			
2538	015633	045	042522	042101	MSG13:	.ASCII /%READ REVERSE ERROR #/
2539	015640	051040	053105	051105		
2540	015646	042523	042440	051122		
2541	015654	051117	021440			
2542	015660	051045	040505	020104	MSG14:	.ASCII /%READ FORWARD ERROR #/
2543	015666	047506	053522	051101		
2544	015674	020104	051105	047522		
2545	015702	020122	043			
2546	015705	045	051127	052111	MSG15:	.ASCII /%WRITE TM ERROR #/
2547	015712	020105	046524	042440		
2548	015720	051122	051117	021440		
2549	015726	020045	051127	052111	MSG15A:	.ASCII /%WRITE TM ERROR ON SECOND TM #/
2550	015734	020105	046524	042440		
2551	015742	051122	051117	047440		
2552	015750	020116	042523	047503		
2553	015756	042116	052040	020115		
2554	015764	043				
2555	015765	045	042522	042526	MSG16:	.ASCII /%REVERSE ERROR #/
2556	015772	051522	020105	051105		
2557	016000	047522	020122	043		
2558	016005	045	047506	053522	MSG17:	.ASCII /%FORWARD ERROR #/
2559	016012	051101	020104	051105		
2560	016020	047522	020122	043		
2561	016025	040	051116	020132	MSG20:	.ASCII / NRZ #/
2562	016032	043				
2563	016033	040	042520	021440	MSG21:	.ASCII / PE #/
2564	016040	042440	050130	035124	MSG22:	.ASCII / EXPT: #/
2565	016046	021440				
2566	016050	051040	053103	035104	MSG23:	.ASCII / RCVD: #/
2567	016056	021440				
2568	016060	041045	051525	052040	MSG24:	.ASCII /%BUS TRAP: #/
2569	016066	040522	035120	021440		
2570	016074	053445	035103	021440	MSG25:	.ASCII /%WC: #/
2571	016102	041045	035101	021440	MSG26:	.ASCII /%BA: #/
2572	016110	042045	035102	021440	MSG27:	.ASCII /%DB: #/
2573	016116	044445	044516	020124	MSG28:	.ASCII /%INIT DID NOT CLEAR HW #/

2574	016124	044504	020104	047516	
2575	016132	020124	046103	040505	
2576	016140	020122	044122	021440	
2577	016146	051445	020103	047516	MSG29: .ASCII /%SC NOT RESET BY INIT #/
2578	016154	020124	042522	042523	
2579	016162	020124	054502	044440	
2580	016170	044516	020124	043	
2581	016175	045	051124	020105	MSG30: .ASCII /%TRE NOT RESET BY INIT #/
2582	016202	047516	020124	042522	
2583	016210	042523	020124	054502	
2584	016216	044440	044516	020124	
2585	016224	043			
2586	016225	045	051503	020062	MSG31: .ASCII /%CS2 NOT RESET BY INIT #/
2587	016232	047516	020124	042522	
2588	016240	042523	020124	054502	
2589	016246	044440	044516	020124	
2590	016254	043			
2591	016255	045	046104	020124	MSG32: .ASCII /%DLT NOT SET #/
2592	016262	047516	020124	042523	
2593	016270	020124	043		
2594	016273	045	041523	047040	MSG33: .ASCII /%SC NOT SET #/
2595	016300	052117	051440	052105	
2596	016306	021440			
2597	016310	052045	042522	047040	MSG34: .ASCII /%TRE NOT SET #/
2598	016316	052117	051440	052105	
2599	016324	021440			
2600	016326	044445	020122	047516	MSG35: .ASCII /%IR NOT SET BY INIT #/
2601	016334	020124	042523	020124	
2602	016342	054502	044440	044516	
2603	016350	020124	043		
2604	016353	045	051117	047040	MSG36: .ASCII /%OR NOT RESET BY INIT #/
2605	016360	052117	051040	051505	
2606	016366	052105	041040	020131	
2607	016374	047111	052111	021440	
2608	016402	047445	020122	047516	MSG37: .ASCII /%OR NOT RESET BY 1 SILO ENTRY #/
2609	016410	020124	042522	042523	
2610	016416	020124	054502	030440	
2611	016424	051440	046111	020117	
2612	016432	047105	051124	020131	
2613	016440	043			
2614	016441	045	051117	047040	MSG38: .ASCII /%OR NOT SET BY SILO FULL #/
2615	016446	052117	051440	052105	
2616	016454	041040	020131	044523	
2617	016462	047514	043040	046125	
2618	016470	020114	043		
2619	016473	045	040502	020104	MSG39: .ASCII /%BAD SILO READ #/
2620	016500	044523	047514	051040	
2621	016506	040505	020104	043	
2622	016513	045	051111	047040	MSG40: .ASCII /%IR NOT RESET BY SILO FULL #/
2623	016520	052117	051040	051505	
2624	016526	052105	041040	020131	
2625	016534	044523	047514	043040	
2626	016542	046125	021514		
2627	016546	047045	047117	042455	MSG41: .ASCII /%NON-EXIST DRIVE #/
2628	016554	044530	052123	042040	
2629	016562	044522	042526	043	

2630	016567	045	047516	026516	MSG42:	.ASCII	/%NON-EXIST SLAVE#/
2631	016574	054105	051511	020124			
2632	016602	046123	053101	021505			
2633	016610	051445	051105	040511	MSG43:	.ASCII	/%SERIAL NO: #/
2634	016616	020114	047516	020072			
2635	016624	043					
2636	016625	045	051105	051501	MSG44:	.ASCII	/%ERASE HEAD INOPERATIVE#/
2637	016632	020105	042510	042101			
2638	016640	044440	047516	042520			
2639	016646	040522	044524	042526			
2640	016654	043					
2641	016655	045	047520	051523	MSG45:	.ASCII	/%POSSIBLE ERASE HEAD PROBLEM: /
2642	016662	041111	042514	042440			
2643	016670	040522	042523	044040			
2644	016676	040505	020104	051120			
2645	016704	041117	042514	035115			
2646	016712	040					
2647	016713	103	042510	045503		.ASCII	/%CHECK POLARITY#/
2648	016720	050040	046117	051101			
2649	016726	052111	021531				
2650	016732	051445	052105	052455	MSG46:	.ASCII	/%SET-UP WRITE ERROR#/
2651	016740	020120	051127	052111			
2652	016746	020105	051105	047522			
2653	016754	021522					
2654	016756	051445	040520	042503	MSG47:	.ASCII	/%SPACE FORWARD ERROR#/
2655	016764	043040	051117	040527			
2656	016772	042122	042440	051122			
2657	017000	051117	043				
2658	017003	045	050123	041501	MSG48:	.ASCII	/%SPACE REVERSE ERROR#/
2659	017010	020105	042522	042526			
2660	017016	051522	020105	051105			
2661	017024	047522	021522				
2662	017030	041045	043125	042506	MSG49:	.ASCII	/%BUFFERED WRITE ERROR#/
2663	017036	042522	020104	051127			
2664	017044	052111	020105	051105			
2665	017052	047522	021522				
2666	017056	041045	052117	051440	MSG50:	.ASCII	/%BOT SET AFTER BUFFERED WRITE#/
2667	017064	052105	040440	052106			
2668	017072	051105	041040	043125			
2669	017100	042506	042522	020104			
2670	017106	051127	052111	021505			
2671	017114	047045	020117	047502	MSG51:	.ASCII	/%NO BOT FROM READ IN PRESET#/
2672	017122	020124	051106	046517			
2673	017130	051040	040505	020104			
2674	017136	047111	050040	042522			
2675	017144	042523	021524				
2676	017150	052045	020103	047111	MSG52:	.ASCII	/%TC INCORRECT #/
2677	017156	047503	051122	041505			
2678	017164	020124	043				
2679	017167	045	047515	020114	MSG53:	.ASCII	/%MOL FAILED TO CLEAR#/
2680	017174	040506	046111	042105			
2681	017202	052040	020117	046103			
2682	017210	040505	021522				
2683	017214	022445	042522	042523	MSG54:	.ASCII	/%RESET SLAVE TO ON LINE BEFORE CONTINUING/
2684	017222	020124	046123	053101			
2685	017230	020105	047524	047440			

2686	017236	020116	044514	042516	
2687	017244	041040	043105	051117	
2688	017252	020105	047503	052116	
2689	017260	047111	044525	043516	
2690	017266	051445	052105	051440	.ASCII /%SET SW12=1 IF YOU DOT WISH TO REPEAT REWIND OFFLINE TEST#/
2691	017274	030527	036462	020061	
2692	017302	043111	054440	052517	
2693	017310	042040	052117	053440	
2694	017316	051511	020110	047524	
2695	017324	051040	050105	040505	
2696	017332	020124	042522	044527	
2697	017340	042116	047440	043106	
2698	017346	044514	042516	052040	
2699	017354	051505	021524		
2700	017360	044440	042524	035122	MSG56: .ASCII / ITER: #/
2701	017366	021440			
2702	017370	052045	020115	047516	MSG57: .ASCII /%TM NOT SET#/
2703	017376	020124	042523	021524	
2704	017404	042445	052111	042510	MSG60: .ASCII /%FITHER TAPE NOT ERASED OR OPI PROBLEM#/
2705	017412	020122	040524	042520	
2706	017420	047040	052117	042440	
2707	017426	040522	042523	020104	
2708	017434	051117	047440	044520	
2709	017442	050040	047522	046102	
2710	017450	046505	043		
2711	017453	045	044122	047440	MSG62: .ASCII /%RH ONLY (NO=0,YES-1): #/
2712	017460	046116	020131	047050	
2713	017466	036517	026060	042531	
2714	017474	036523	024461	020072	
2715	017502	043			
2716	017503	045	044504	020104	MSG63: .ASCII /%DID NOT AUTO SELECT NRZ#/
2717	017510	047516	020124	052501	
2718	017516	047524	051440	046105	
2719	017524	041505	020124	051116	
2720	017532	021532			
2721	017534	042045	042111	047040	MSG64: .ASCII /%DID NOT AUTO SELECT PE#/
2722	017542	052117	040440	052125	
2723	017550	020117	042523	042514	
2724	017556	052103	050040	021505	
2725	017564	021445			MSG65: .ASCII /%#/
2726	017566	021534			MSG66: .ASCII /%#/
2727	017570	042445	035122	021440	MSG67: .ASCII /%ER: #/
2728	017576	051045	046505	053117	MSG69: .ASCII /%REMOVE TMDP FROM SLAVE TO BE TESTED%/
2729	017604	020105	046524	050104	
2730	017612	043040	047522	020115	
2731	017620	046123	053101	020105	
2732	017626	047524	041040	020105	
2733	017634	042524	052123	042105	
2734	017642	021445			
2735	017644	044045	051101	053504	MSG70: .ASCII /%HARDWARE SWR IN USE%/
2736	017652	051101	020105	053523	
2737	017660	020122	047111	052440	
2738	017666	042523	021445		
2739					

```
2740 ;TEST HEADERS*****
2741
2742 017672 022445 052106 035061 MSFT1: .ASCII /%%FT1:RH ADDRESSING #/
2743 017700 044122 040440 042104
2744 017706 042522 051523 047111
2745 017714 020107 043
2746 017717 045 043045 031124 MSFT2: .ASCII /%%FT2:RH REGISTER BITS TEST #/
2747 017724 051072 020110 042522
2748 017732 044507 052123 051105
2749 017740 041040 052111 020123
2750 017746 042524 052123 021440
2751 017754 022445 052106 035063 MSFT3: .ASCII /%%FT3:RH INITIALIZE TEST #/
2752 017762 044122 044440 044516
2753 017770 044524 046101 055111
2754 017776 020105 042524 052123
2755 020004 021440
2756 020006 022445 052106 035064 MSFT4: .ASCII /%%FT4:RH11 SILO TEST 1 #/
2757 020014 044122 030461 051440
2758 020022 046111 020117 042524
2759 020030 052123 030440 021440
2760 020036 022445 052106 035065 MSFT5: .ASCII /%%FT5:RH11 SILO TEST 2 #/
2761 020044 044122 030461 051440
2762 020052 046111 020117 042524
2763 020060 052123 031040 021440
2764 020066 022445 052106 035066 MSFT6: .ASCII /%%FT6:RH11 SILO TEST 3 #/
2765 020074 044122 030461 051440
2766 020102 046111 020117 042524
2767 020110 052123 031440 021440
2768 020116 022445 052106 035067 MSFT7: .ASCII /%%FT7:RH11 SILO TEST 4 #/
2769 020124 044122 030461 051440
2770 020132 046111 020117 042524
2771 020140 052123 032040 021440
2772 020146 022445 052106 030061 MSFT10: .ASCII /%%FT10:RH11 SILO TEST 5 #/
2773 020154 051072 030510 020061
2774 020162 044523 047514 052040
2775 020170 051505 020124 020065
2776 020176 043
2777 020177 045 043045 030524 MSFT11: .ASCII /%%FT11:NOP TEST#/
2778 020204 035061 047516 020120
2779 020212 042524 052123 043
2780 020217 045 043045 030524 MSFT12: .ASCII /%%FT12:REWIND TEST#/
2781 020224 035062 042522 044527
2782 020232 042116 052040 051505
2783 020240 021524
2784 020242 022445 052106 031461 MSFT13: .ASCII /%%FT13:WRITE-READ TEST#/
2785 020250 053472 044522 042524
2786 020256 051055 040505 020104
2787 020264 042524 052123 043
2788 020271 045 043045 030524 MSFT14: .ASCII /%%FT14:SPACE TEST#/
2789 020276 035064 050123 041501
2790 020304 020105 042524 052123
2791 020312 043
2792 020313 045 043045 030524 MSFT15: .ASCII /%%FT15:ERASE TEST#/
2793 020320 035065 051105 051501
2794 020326 020105 042524 052123
2795 020334 043
```

2796	020335	045	043045	030524	MSFT16: .ASCII /%%FT16:TAPE MARK WRITE-READ TEST#/
2797	020342	035066	040524	042520	
2798	020350	046440	051101	020113	
2799	020356	051127	052111	026505	
2800	020364	042522	042101	052040	
2801	020372	051505	021524		
2802	020376	022445	052106	033461	MSFT17: .ASCII /%%FT17:TM SPACE TEST #/
2803	020404	052072	020115	050123	
2804	020412	041501	020105	042524	
2805	020420	052123	021440		
2806	020424	022445	052106	030062	MSFT20: .ASCII /%%FT20:WRITE CHECK TEST #/
2807	020432	053472	044522	042524	
2808	020440	041440	042510	045503	
2809	020446	052040	051505	020124	
2810	020454	043			
2811	020455	045	043045	031124	MSFT21: .ASCII /%%FT21:ERASE HEAD TEST#/
2812	020462	035061	051105	051501	
2813	020470	020105	042510	042101	
2814	020476	052040	051505	021524	
2815	020504	022445	052106	031062	MSFT22: .ASCII /%%FT22:BUFFERED COMMAND TEST#/
2816	020512	041072	043125	042506	
2817	020520	042522	020104	047503	
2818	020526	046515	047101	020104	
2819	020534	042524	052123	043	
2820	020541	045	043045	031124	MSFT23: .ASCII /%%FT23:READ IN PRESET TEST#/
2821	020546	035063	042522	042101	
2822	020554	044440	020116	051120	
2823	020562	051505	052105	052040	
2824	020570	051505	021524		
2825	020574	022445	052106	032062	MSFT24: .ASCII /%%FT24:AUTO DENSITY SELECT: WRITE-NRZ,READ-PE#/
2826	020602	040472	052125	020117	
2827	020610	042504	051516	052111	
2828	020616	020131	042523	042514	
2829	020624	052103	020072	051127	
2830	020632	052111	026505	051116	
2831	020640	026132	042522	042101	
2832	020646	050055	021505		
2833	020652	022445	052106	032462	MSFT25: .ASCII /%%FT25:AUTO DENSITY SELECT: WRITE-PE,READ-NRZ#/
2834	020660	040472	052125	020117	
2835	020666	042504	051516	052111	
2836	020674	020131	042523	042514	
2837	020702	052103	020072	051127	
2838	020710	052111	026505	042520	
2839	020716	051054	040505	026504	
2840	020724	051116	021532		
2841	020730	022445	052106	033062	MSFT26: .ASCII /%%FT26:SEQUENTIAL TAPE MARK TEST#/
2842	020736	051472	050505	042525	
2843	020744	052116	040511	020114	
2844	020752	040524	042520	046440	
2845	020760	051101	020113	042524	
2846	020766	052123	043		
2847	020771	045	043045	031124	MSFT27: .ASCII /%%FT27:REWIND-OFF LINE TEST#/
2848	020776	035067	042522	044527	
2849	021004	042116	047455	043106	
2850	021012	046040	047111	020105	
2851	021020	042524	052123	043	

2852	021025	045	043536	043	\$CNTG:	.ASCII	/%^GA/
2853	021031	045	053523	036522	\$MSWR:	.ASCII	/%SWR #/
2854	021036	021440					
2855	021040	020040	042516	036527	\$MNEW:	.ASCII	/ NEW= #/
2856	021046	021440					
2857	021050	022477	043		\$QUEST:	.ASCII	/?%#/
2858							
2859							
2860		021054			WDATA:	.EVEN	
2861	021054	000000				0	
2862		022566			RDATA:	.+1510	
2863	022566	000000				0	
2864							
2865		000001				.END	

FT15A	006642	1506#						
FT15B	006656	1508#	1514					
FT15X	006766	1512	1525#					
FT16	006772	781	782	1528#				
FT16A	007024	1533#	1556					
FT16B	007030	1534#						
FT16X	007210	1554	1557#					
FT17	007220	783	784	1563#				
FT17A	007240	1566#	1620					
FT17B	007244	1567#	1597					
FT17C	007404	1587#	1596					
FT17D	007452	1580	1598#					
FT17D1	007470	1601#	1616					
FT17E	007504	1603#	1611					
FT17F	007574	1613	1617#					
FT17X	007616	1575	1594	1608	1618	1621#		
FT2	003372	757	758	1050#				
FT2A	003404	1052#	1059	1084				
FT2B	003444	1057	1061#	1068				
FT2C	003504	1066	1070#	1079				
FT2D	003520	1074#	1075					
FT2E	003550	1077	1081#					
FT2ER	003560	1060	1069	1080	1085#	1775	1911	
FT2ERA	003610	1089	1092#					
FT2ERB	003662	1087	1103#					
FT2ERC	003672	1104	1106#					
FT2X	003702	1082	1109#					
FT20	007622	785	786	1625#				
FT20A	007640	1628#	1653					
FT20B	007764	1645#						
FT20C	010016	1650#						
FT20X	010036	1639	1651	1654#				
FT21	010046	787	788	1659#				
FT21A	010054	1660#						
FT21B	010350	1697	1700#					
FT21C	010356	1699	1701#					
FT21SC	010204	1678#	1701					
FT21X	010370	1693	1695	1703#				
FT22	010400	789	790	1709#	1734			
FT22A	010454	1717#	1719					
FT22B	010474	1722#	1723					
FT22X	010574	1732	1736#					
FT23	010604	791	792	1742#	1767	1774		
FT23A	010714	1758#	1761	1763				
FT23B	010734	1759	1764#					
FT23C	010764	1765	1769#					
FT23X	011020	1743	1772	1776#				
FT24	011024	793	794	1782#	1970			
FT24X	011206	1795	1802	1805#				
FT25	011216	795	796	1810#	1972			
FT25X	011400	1823	1830	1833#				
FT26	011410	797	798	1838#	2119	2121		
FT26X	011636	1849	1855	1864	1869	1871	1874#	
FT27	011646	799	800	1878#	2124	2126		
FT27X	012056	1905	1909	1912#				
FT27XX	012066	1879	1881	1914#				

MSFT13	020242	1351	2784#		
MSFT14	020271	1392	1463	2788#	
MSFT15	020313	1502	2792#		
MSFT16	020335	1531	2796#		
MSFT17	020376	1564	2802#		
MSFT2	017717	1050	2746#		
MSFT20	020424	1626	2806#		
MSFT21	020455	1659	2811#		
MSFT22	020504	1709	2815#		
MSFT23	020541	1744	2820#		
MSFT24	020574	1782	2825#		
MSFT25	020652	1810	2833#		
MSFT26	020730	1839	2841#		
MSFT27	020771	1882	2847#		
MSFT3	017754	1115	2751#		
MSFT4	020006	1160	2756#		
MSFT5	020036	1185	2760#		
MSFT6	020066	1216	2764#		
MSFT7	020116	1268	2768#		
MSG1	015012	2002	2464#		
MSG10	015555	884	2529#		
MSG10A	015415	874	2512#		
MSG11	015575	901	2532#		
MSG12	015615	1360	1590	1669	2535#
MSG13	015633	1370	1545	2538#	
MSG14	015660	1377	1550	1685	2542#
MSG15	015705	1539	1572	1845	2546#
MSG15A	015726	1851	2549#		
MSG16	015765	1476	1600	1640	2555#
MSG17	016005	1474	1614	1645	2558#
MSG2	015100	2473#			
MSG20	016025	1467	1988	2561#	
MSG21	016033	1470	1991	2563#	
MSG22	016040	1095	1251	1478	2564#
MSG23	016050	1099	1255	1483	2566#
MSG24	016060	2187	2568#		
MSG25	016074	1058	2570#		
MSG26	016102	1067	2571#		
MSG27	016110	1078	2572#		
MSG28	016116	2573#			
MSG29	016146	1123	2577#		
MSG3	015153	833	838*	2481#	
MSG30	016175	1127	2581#		
MSG31	016225	1133	2586#		
MSG32	016255	1171	1279	1305	2591#
MSG33	016273	1173	2594#		
MSG34	016310	1175	2597#		
MSG35	016326	1190	2600#		
MSG36	016353	1194	2604#		
MSG37	016402	1199	1207	2608#	
MSG38	016441	1230	2614#		
MSG39	016473	1249	2619#		
MSG4	015313	839	2498#		
MSG40	016513	1226	2622#		
MSG41	016546	898	2627#		
MSG42	016567	915	2630#		

CZTECBO TMO3-TE16/TU77 BFT
CZTECB.P11 22-NOV-78 07:51

MACY11 30A(1052) 21-DEC-78 13:15 PAGE 82
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0080

\$CATCH	537#	616
\$CHAIN	537#	818
\$CHNO	537#	952
\$RESTO	537#	2451
\$SAVE	537#	2441
.\$ACT1	537#	617
.\$EOP	537#	1011

. ABS. 022570 000

ERRORS DETECTED: 0

DSKZ:CZTECB,DSKZ:CZTECB.SEQ/CRF/SOL=CZTEAB.SML/ML,DSKM:CZTECB.P11
RUN-TIME: 4 6 1 SECONDS
RUN-TIME RATIO: 97/12=7.9
CORE USED: 9K (17 PAGES)