

TM03, TE16
TU77

TM03/TE16, TU77 BFT
CZTECDO

AH-A798D-MC
FICHE 1 OF 1

OCT 1983
COPYRIGHT © 77-83
MADE IN USA



Microfiche grid containing multiple frames of data, including text and tables.



.REM %

IDENTIFICATION

PRODUCT CODE: AC-A797D-MC
PRODUCT NAME: CZTECDO TM03-TE16/TU77 BASIC FUNCTION TEST
DATE CREATED: 11 - JULY - 1983
MAINTAINER: TAPE 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) 1977,1983 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

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

1. ABSTRACT

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

2. REQUIREMENTS (HARDWARE)

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

3. LOADING PROCEDURE

USE STANDARD BINARY LOADING PROCEDURE

4. STARTING PROCEDURE

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

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

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

4.1 AUTOMATIC MODE OPERATION

IF THIS PROGRAM IS LOADED AND RUN IN AUTOMATIC (CHAIN) MODES
DEFAULT RESPONSES TO OPERATOR REQUESTS ARE USED, AND ALL AVAIL-
ABLE TM03/TE16 COMBINATIONS ARE TESTED. ADDITIONALLY THE SOFTWARE
SWR IS INVOKED WITH A SWITCH SETTING OF 000000
IF LOADED VIA ACT11 CHAIN MODE.

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

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

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

4.2 SAMPLE START AT 200

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

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

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

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

5. CONSOLE SWITCH SETTING

CONTROL:

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

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

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

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

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

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

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

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

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

6. ERROR PRINTOUTS

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

EXAMPLES:

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

FT13: WRITE-READ TEST

WRITE ERROR NRZ

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

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

FT14: SPACE TEST

SPACE REVERSE ERROR NRZ

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

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

FT14: SPACE TEST

POSITION ERROR:

REVERSE ERROR EXPT:20 RCVD:22

7. OPERATION

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

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE
****REFER TO SECTION 5 FOR DYNAMIC LOADING
OF SOFTWARE SWITCH REGISTER.***
3. PRESS START
4. ENTER APPROPRIATE RESPONSES TO THE TTY REQUESTS

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

SINGLE TEST SELECTION: (SW0-SW4)

WHEN SW0-4 ARE SET TO ZERO (00) THE SCHEDULAR WILL
EXECUTE ALL OF THE TESTS IN SEQUENCE . IF SW0-4 IS
SET TO SOME SPECIFIC TEST NUMBER THAT PARTICULAR TEST
WILL BE EXECUTED CONTINUOUSLY. ANY TEST MAY BE SINGLE
SELECTED IN ANY ORDER; HOWEVER, THE BEST WAY TO AFFECT
THE CHANGE IS TO USE SW10 TO HALT THE CURRENT TEST, THEN
CHANGE NUMBER AND PRESS CONTINUE.

8. SUBTEST SUMMARIES

THE FOLLOWING IS A LIST OF ALL TESTS IN THEIR PROPER SEQUENCE.
A BASIC DESCRIPTION OF EACH TEST IS PROVIDED TO AID IN UNDERSTANDING
OF THE ERROR MESSAGES ASSOCIATED WITH EACH ONE.

A. RH TESTS: THE FIRST TEN (10) TESTS WILL PERFORM BASIC RH
OPERATIONS AS FAR AS IS POSSIBLE WITHOUT REQUIRING
THE TM03-TE16/TU77 ITSELF. (SEE RH ONLY OPTION; PAR 7)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
.LIST BIN,LOC,SEQ
.TITLE CZTECDO TM03-TE16/TU77 BFT
:BASIC FUNCTION TEST
:AC-A797B-MC
:FEB 77
:J.G. ADAMS

:REVISED JUN 1977 BY J.G. ADAMS ;++B ADDED TU77 CAPABILITY

:REVISED NOV 1978 BY MIKE PAGE ;+ DESIGNATES CODE ADDED FOR
:NON-STANDARD JUMPER CONFIG.
:REVISED MAY 1983 BY B. LEBLANC :BL FIXEED CC0000810,CC0001688

.MCALL .SACT11,.SEOP,$SCATCH,$SSAVE,$RESTORE,$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.
```



```
625 ;REGISTER EQUIVS*****
626
627 000000 R0=%0
628 000001 R1=%1
629 000002 R2=%2
630 000003 R3=%3
631 000004 R4=%4
632 000005 R5=%5
633 000006 SP=%6
634 000007 PC=%7
635
637
638 ;ACT11 HOOK *****
(1) $SVPC= ;SAVE CURRENT LOCATION CTR
(1) 000764 .=42
(1) 000042 .WORD 0
(1) 000046 .=46
(1) 000046 .WORD $ENDAD ;SET LOCATION 46
(1) 000052 .=52
(1) 000052 .WORD 0 ;SET LOCATION 52 = 0
(1) 000764 .=$SVPC ;RESTORE LOCATION CTR
(1)
639 ;TTY INTERRUPT VECTOR*****
640
641 .=60
642 000060 .WORD TTINT ;TTY INTERRUPT HEADER ADDRESS
643 000062 .WORD 340 ;PRIORITY LEVEL 7
644
645 ;SOFTWARE SWITCH REGISTER*****
646 ;USED IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
647
648 .=176
649 000176 SWREG: 0 ;SOFTWARE SWITCH REGISTER
650
651
652 ;START ADDRESS*****
653
654 .=200
655 000200 JMP START ;PROGRAM START
656
657 ;RESTART ADDRESS*****
658 .=210
659 000210 JMP ST4
660
661 ;TM03 INTERRUPT VECTOR*****
662
663 .=224
664 000224 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
665 000226 013526
666 000226 000340
```

```
668
669          000510          .=510
670                                     :MASS BUS REGISTER EQUIVS*****
671
672 000510 172440          C1: 172440
673 000512 172442          WC: 172442
674 000514 172444          BA: 172444
675 000516 172446          FC: 172446
676 000520 172450          CS: 172450
677 000522 172452          DS: 172452
678 000524 172454          ER: 172454
679 000526 172456          AS: 172456
680 000530 172460          CC: 172460
681 000532 172462          DB: 172462
682 000534 172464          MR: 172464
683 000536 172466          DT: 172466
684 000540 172470          SN: 172470
685 000542 172472          TC: 172472
686 000544 172474          BAE: 172474
687
688                                     :CONSTANTS*****
689
690 000546 177776          PSW: 177776          :PROCESSOR STATUS
691 000550 177570          SWR: 177570          :SWITCH REGISTER
692 000552 177560          TKS: 177560          :TTY READER STATUS
693 000554 177562          TKB: 177562          :TTY READ BUFFER
694 000556 177564          TPS: 177564          :TTY PUNCH STATUS
695 000560 177566          TPB: 177566          :TTY PUNCH BUFFER
696 000562 177777          SERNUM: 177777       :SERIAL NUMBER
697 000564 000011          DRVTP: 011           :DRIVE TYPE
698 000566 000010          ITAMT: 10           :ITERATION AMOUNT
699 000570 000224          VECT: 224           :INTERRUPT VECTOR(RH)
700 000572 172440          REGS: 172440        :STARTING REGISTER ADDRESS
701 000574 000004          BTRP: 4             :BUS TRAP ADDRESS
702 000576 000006          BTRP2: 6            :BUS TRAP PRIORITY LEVEL 7
```

			;FLAGS AND COUNTERS*****	
704			TOB:	0
705			TIB:	0
706	000600	000000	RH17F:	0
707	000602	000000	HDRFL:	0
708	000604	000000	EMADDR:	0
709	000606	000000	DRVN:	0
710	000610	000000	SLVN:	0
711	000612	000000	BADDR:	0
712	000614	000000	FCNT:	0
713	000616	000000	WCNT:	0
714	000620	000000	RCNT:	0
715	000622	000000	ERRP:	0
716	000624	000000	ERRP1:	0
717	000626	000000	RRD:	0
718	000630	000000	RFD:	0
719	000632	000000	RDYDX:	0
720	000634	000000	OPDYX:	0
721	000636	000000	SCNT:	0
722	000640	000000	PFLG:	0
723	000642	000000	RTRN:	0
724	000644	000000	ERADD:	0
725	000646	000000	TEMP1:	0
726	000650	000000	TEMP2:	0
727	000652	000000	TEMP3:	0
728	000654	000000	STMSK:	0
729	000656	000000	ITCNT:	0
730	000660	000000	DSAV:	0
731	000662	000000	SAV1:	0
732	000664	000000	SAV2:	0
733	000666	000000	SAV3:	0
734	000670	000000	SCOLP:	0
735	000672	000000	ITRLP:	0
736	000674	000000	EXFL:	0
737	000676	000000	PEXFL:	0
738	000700	000000	STFLG:	0
739	000702	000000	LTADD:	0
740	000704	000000	FUN:	0
741	000706	000000	SERFL:	0
742	000710	000000	CRCNT:	0
743	000712	000000	UDES:	0
744	000714	000000	PATRI:	0
745	000716	000000	RHTF:	0
746	000720	000000	NRZOF:	0
747	000722	000000	RHOF:	0
748	000724	000000	PCNTR:	0
749	000726	000000	TEMPST:	0
750	000730	000000	COUNT:	0
751	000732	000000	RDSW:	0
752	000734	000000	NONSTD:	0
753	000736	000000	JUMPER:	0
754	000740	000000		
755	000742	000000		
756				

758
759
760
761 000744 000000
762 000746 013222
:63 000750 013242
764 000752 013246
765 000754 013254

;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

767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819

000756 000000
000760 000000
000762 003270
000764 003270
000766 003376
000770 003376
000772 003720
000774 003720
000776 004140
001000 004140
001002 004266
001004 004266
001006 004460
001010 004460
001012 004732
001014 004732
001016 005026
001020 005026
001022 005162
001024 005162
001026 005300
001030 005300
001032 005412
001034 005412
001036 005724
001040 005724
001042 006576
001044 006576
001046 007210
001050 007210
001052 007436
001054 007436
001056 010040
001060 010040
001062 010264
001064 010264
001066 010616
001070 010616
001072 011022
001074 011022
001076 011242
001100 011242
001102 011434
001104 011434
001106 011626
001110 011626
001112 012064
001114 012064
001116 003162
001120 000027

:LOGIC TEST ENTRY TABLE*****

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

TLAST: .WORD TEND
.WORD 27

:CONTAINS # OF TESTS

```

821          001600          . =1600
822          ;PROGRAM START AND HOUSEKEEPING*****
823
824 001600 012706 000500      START:  MOV    #500,SP          ;SET STACK POINTER
825 001604 013746 000006      MOV    @#6,-(SP)        ;SAVE VECTORS
826 001610 013746 000004      MOV    @#4,-(SP)
827 001614 012737 001640 000004  MOV    #1$,@#4          ;SET UP FOR TIMEOUT
828 001622 005037 000006      CLR    @#6
829 001626 022777 177777 176714  CMP    #-1,@SWR        ;REFERENCE HARDWARE SWITCH REGISTER
830 001634 001402      BEQ    2$
831 001636 000404      BR     3$
832 001640 022626      1$:  CMP    (SP)+,(SP)+      ;ADJUST STACK
833 001642 012737 000176 000550  2$:  MOV    #SWREG,SWR      ;POINT TO SOFTWARE SWITCH REG
834 001650 012637 000004      3$:  MOV    (SP)+,@#4        ;RESTORE VECTORS
835 001654 012637 000006      MOV    (SP)+,@#6
836 001660 005027      CLR    (PC)+
(1) 001662 000000      CHNFLG: .WORD 0        ;;CLEAR CHAIN INDICATOR
(1)          ;CHAIN MODE INDICATOR
(1)          ;;1/0 = CHAIN/NOT CHAIN MODE
(1)          ;;BRANCH IF IN DUMP MODE
(1) 001664 005737 000042      TST    @#42
(1) 001670 001407      BEQ    50$
(1) 001672 012737 000176 000550  MOV    #SWREG,SWR      ;;INVOKE SOFTWARE SWR
(1) 001700 005237 001662      INC    CHNFLG          ;;SET CHNFLG = CHAIN MODE
(1) 001704 000137 001710      JMP    SCHN            ;;GO TO CHAIN ADDRESS
(1) 001710
837 001710 000240      50$:  NOP
838 001712 122737 000006 000041  4$:  CMPB   #6,@#41        ;BRANCH IF LOADED VIA TMDP (DUMP MODE)
839 001720 001005      BNE    5$
840 001722 012704 020036      MOV    #MSG69,R4
841 001726 004737 014322      JSR    PC,TTOUT        ;ADVISE USER TO REMOVE TMDP FROM UUT
842 001732 000000      HALT
843 001734 012704 015405      5$:  MOV    #MSG3,R4
844 001740 004737 014322      JSR    PC,TTOUT        ;PRINT TITLE
845 001744 005737 001662      TST    CHNFLG          ;SEE IF IN CHAIN MODE
846 001750 001402      BEQ    6$              ;IF NOT: BR
847 001752 000137 002554      JMP    TSCD            ;ELSE GO START TEST
848 001756 112737 000043 015405  6$:  MOVB   #'#,MSG3        ;DO NOT PRINT TITLE ON RESTART
849 001764 012704 015545      STOB:  MOV    #MSG4,R4
850 001770 004737 014322      JSR    PC,TTOUT        ;REQUEST REGISTER ADDRESS
851 001774 013703 000572      MOV    REGS,R3
852 002000 004737 014452      JSR    PC,OCTP        ;PRINT CURRENT ADDRESS
853 002004 012705 000572      MOV    #REGS,R5        ;SET ADDRESS SAVE LOC
854 002010 012701 000007      MOV    #7,R1           ;SET SIZE OF RESPONSE
855 002014 012702 176400      MOV    #176400,R2      ;SET UPPER LIMIT
856 002020 012703 172300      MOV    #172300,R3      ;SET LOWER LIMIT
857 002024 004737 014000      JSR    PC,TTR          ;GO GET RESPONSE
858 002030 012704 015570      MOV    #MSG5,R4
859 002034 004737 014322      JSR    PC,TTOUT        ;REQUEST VECTOR
860 002040 013703 000570      MOV    VECT,R3
861 002044 004737 014452      JSR    PC,OCTP        ;PRINT CURRENT VECTOR
862 002050 012705 000570      MOV    #VECT,R5        ;SET ADDRESS SAVE LOC
863 002054 012701 000004      MOV    #4,R1           ;SET SIZE OF RESPONSE
864 002060 012702 000224      MOV    #224,R2         ;SET UPPER LIMIT
865 002064 012703 000150      MOV    #150,R3         ;SET LOWER LIMIT
866 002070 004737 014000      JSR    PC,TTR          ;GO GET RESPONSE
867 002074 013700 000570      MOV    VECT,R0        ;GET VECTOR
868 002100 012720 013526      MOV    #MTINT,(R0)+    ;LOAD INTERRUPT ADDRESS IN VECTOR
  
```

869	002104	012710	000340		MOV	#340,(R0)	:LOAD PRIORITY
870	002110	013700	000572		MOV	REGS,R0	:GET START OF REGS
871	002114	012701	000017		MOV	#17,R1	:SET NUMBER OF REGS
872	002120	012702	000510		MOV	#C1,R2	:GET START OF TABLE
873	002124	010022		ST0:	MOV	R0,(R2)+	:BUILD TABLE
874	002126	062700	000002		ADD	#2,R0	:BUMP ADDRESS
875	002132	005301			DEC	R1	:SEE IF DONE
876	002134	001373			BNE	ST0	:IF NOT: BR
877	002136	012702	000600		MOV	#TOB,R2	
878	002142	012700	000054		MOV	#54,R0	
879	002146	005022		ST1:	CLR	(R2)+	:CLEAR FLAGS + COUNTERS
880	002150	005300			DEC	R0	
881	002152	001375			BNE	ST1	
882	002154	012737	000001	000722	MOV	#1,RHTF	:SET ADDRESS TEST FLAG
883	002162	000137	003022		JMP	TSRH	:GO DO INITIAL ADDRESS TEST PASS
884	002166	012704	015647	ST1A:	MOV	#MSG10A,R4	
885	002172	004737	014322		JSR	PC,TTOUT	:REQUEST JUMPER CONFIGURATION
886	002176	012705	000742		MOV	#JUMPER,R5	:GET ADDRESS OF RESPONSE
887	002202	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
888	002206	012702	000004		MOV	#4,R2	:SET RANGE
889	002212	012703	000000		MOV	#0,R3	:LOWER LIMIT
890	002216	004737	014000		JSR	PC,TTR	:GET RESPONSE
891	002222	022737	000002	000742	CMP	#2,JUMPER	:TEST FOR NON-STANDARD MODE
892	002230	001002			BNE	1\$	
893	002232	004737	013402		JSR	PC,NOST	:MODIFY TEST SCHEDULE
894	002236	012704	016007	1\$:	MOV	#MSG10,R4	
895	002242	004737	014322		JSR	PC,TTOUT	:REQUEST DRIVE NUMBER
896	002246	013703	000612		MOV	DRVN,R3	:GET CURRENT DRIVE #
897	002252	004737	014452		JSR	PC,OCIP	:AND TYPE IT
898	002256	012705	000612		MOV	#DRVN,R5	:SET ADDRESS OF DRIVE NUMBER SAVE
899	002262	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
900	002266	012702	000007		MOV	#7,R2	:SET UPPER LIMIT
901	002272	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
902	002276	004737	014000		JSR	PC,TTR	:GO GET RESPONSE
903	002302	012777	000040	176210	MOV	#40,acs	:SET INIT
904	002310	053777	000612	176202	BIS	DRVN,acs	:SET DRIVE NUMBER
905	002316	005777	176166		TST	ac1	:ACCESS DRIVE
906	002322	032777	010000	176170	BIT	#10000,acs	:SEE IF NED
907	002330	001405			BEQ	ST2	:IF NOT: BR
908	002332	012704	017006		MOV	#MSG41,R4	
909	002336	004737	014322		JSR	PC,TTOUT	:PRINT NOT AVAIL
910	002342	000711			BR	ST1A	:REDO DRIVE REQUEST
911	002344	012704	016027	ST2:	MOV	#MSG11,R4	
912	002350	004737	014322		JSR	PC,TTOUT	:REQUEST SLAVE NUMBER
913	002354	013703	000614		MOV	SLVN,R3	:GET CURRENT SLAVE #
914	002360	004737	014452		JSR	PC,OCIP	:AND TYPE IT
915	002364	012705	000614		MOV	#SLVN,R5	:SET ADDRESS OF SLAVE SAVE
916	002370	012701	000002		MOV	#2,R1	:SET SIZE OF RESPONSE
917	002374	012702	000007		MOV	#7,R2	:SET UPPER LIMIT
918	002400	012703	000000		MOV	#0,R3	:SET LOWER LIMIT
919	002404	004737	014000		JSR	PC,TTR	:GO GET RESPONSE
920	002410	012777	000040	176102	MOV	#40,acs	:INIT
921	002416	053777	000612	176074	BIS	DRVN,acs	:SET DRIVE NUMBER
922	002424	013777	000614	176110	MOV	SLVN,ac	:LOAD SLAVE NUMBER
923	002432	032777	002000	176076	BIT	#2000,adT	:SEE IF SLAVE PRESENT
924	002440	001005			BNE	ST3	:IF SO: BR

925	002442	012704	017027	MOV	#MSG42,R4	
926	002446	004737	014322	JSR	PC,TTOUT	:PRINT NON-EXIST SLAVE
927	002452	000734		BR	ST2	:REDO SLAVE REQUEST
928	002454	012704	017050	ST3: MOV	#MSG43,R4	
929	002460	004737	014322	JSR	PC,TTOUT	:PRINT SERIAL NUMBER TAG
930	002464	017703	176050	MOV	@SN,R3	
931	002470	004737	015000	JSR	PC,SNPT	:PRINT SERIAL NUMBER
932	002474	012704	017713	MOV	#MSG62,R4	:GET REQUEST
933	002500	004737	014322	JSR	PC,TTOUT	:REQUEST RH11 ONLY RESPONSE
934	002504	013703	000726	MOV	RHOF,R3	:GET CURRENT FLAG SETTING
935	002510	004737	014452	JSR	PC,OCTP	:AND TYPE IT
936	002514	012705	000726	MOV	#RHOF,R5	:SET FLAG ADDRESS
937	002520	012701	000002	MOV	#2,R1	:SET SIZE OF RESPONSE
938	002524	012702	000001	MOV	#1,R2	:SET UPPER LIMIT
939	002530	012703	000000	MOV	#0,R3	:SET LOWER LIMIT
940	002534	004737	014000	JSR	PC,TTR	:GO GET RESPONSE
941						
942				:START 210		
943	002540	012706	000500	ST4: MOV	#500,SP	:SET STACK PTR
944	002544	005037	000730	CLR	PCNTR	:CLEAR PASS COUNTER
945	002550	004737	015102	JSR	PC,GTSWR	:GET SWITCHES

```

947                                     ;TEST SCHEDULAR*****
948
949 002554 052777 000100 175770 TSCD: BIS #100,@TKS ;SET KEYBOARD IE BIT
950 002562 005037 000704 CLR STFLG ;CLEAR SINGLE TEST FLAG
951 002566 005037 000604 CLR RH17F ;SET RH INDICATOR = RH11
952 002572 013746 000004 MOV @#4,-(SP) ;SAVE ERROR TRAP VECTORS
953 002576 013746 000006 MOV @#6,-(SP) ;AND PRIORITY
954 002602 012737 002630 000004 MOV #1$,@#4 ;SET TIME OUT TRAP TO 1$ BELOW
955 002610 005037 000006 CLR @#6
956 002614 005777 175724 TST @BAE ;REFERENCE BAE REGISTER
957 002620 012737 000001 000604 MOV #1,RH17F ;SET FLAG = RH70
958 002626 000401 BR 2$
959 002630 022626 1$: CMP (SP)+,(SP)+ ;RESET STACK
960 002632 012637 000006 2$: MOV (SP)+,@#6 ;RESTORE ERROR TRAP
961 002636 012637 000004 MOV (SP)+,@#4
962 002642 017700 175702 MOV @SWR,RO
963 002646 042700 177740 BIC #177740,RO
964 002652 001125 BNE STSCD ;GO SELECT SINGLE TEST
965 002654 005737 001662 TST CHNFLG ;;BRANCH IF NOT IN CHAIN MODE
(1) 002660 001457 BEQ TSCDA
(1) 002662 012737 177777 000612 MOV #-1,DRVN ;;INITIALIZE DRIVE #
(1) 002670 012737 177777 000614 NXTDRV: MOV #-1,SLVN ;;INITIALIZE SLAVE #
(1) 002676 012777 000040 175614 1$: MOV #40,ACS ;;INIT CONTROLLER
(1) 002704 005237 000612 INC DRVN ;;STEP DRIVE #
(1) 002710 022737 000010 000612 CMP #10,DRVN ;;EXIT IF ALL DRIVES TESTED
(1) 002716 001524 BEQ $DONE ;;FOR AVAILABILITY
(1) 002720 013777 000612 175572 MOV DRVN,ACS ;;LOAD DRIVE #
(1) 002726 005777 175556 TST @C1 ;;ACCESS DRIVE
(1) 002732 032777 010000 175560 BIT #10000,ACS ;;BRANCH IF DRIVE NON EXISTANT
(1) 002740 001356 BNE 1$ ;;(NED = 1)
(1) 002742 005237 000614 NXTSLV: INC SLVN ;;STEP SLAVE # AND BRANCH
(1) 002746 001011 BNE 1$ ;;IF NOT SLAVE 0
(1) 002750 005737 000612 TST DRVN ;;BRANCH IF NOT DRIVE # 0
(1) 002754 001006 BNE 1$
(1) 002756 122737 000006 000041 CMPB #6,@#41 ;;BRANCH IF NOT TMDP
(1) 002764 001002 BNE 1$
(1) 002766 005237 000614 INC SLVN ;;STEP TO SLAVE # 1
(1) 002772 022737 000010 000614 1$: CMP #10,SLVN ;;BRANCH IF ALL SLAVES TESTED
(1) 003000 001733 BEQ NXTDRV ;;FOR AVAILABILITY
(1) 003002 013777 000614 175532 MOV SLVN,@TC ;;LOAD SLAVE UNIT #
(1) 003010 032777 002000 175520 BIT #2000,@DT ;;BRANCH IF SLAVE NOT
(1) 003016 001751 BEQ NXTSLV ;;PRESENT (SPR = 0)
966 003020 000240 TSCDA: NOP
967 003022 012737 000756 000706 TSRH: MOV #TSTTBL,LTADD
968 003030 062737 000004 000706 TSCD0: ADD #4,LTADD
969 003036 013737 000706 000676 TSCD1: MOV LTADD,ITRLP
970 003044 062737 000002 000676 ADD #2,ITRLP ;SET ITERATION ADDRESS
971 003052 005037 000660 CLR STMSK
972 003056 005037 000626 CLR ERRP
973 003062 005037 000606 CLR HDRFL ;CLEAR PRINT HEADER FLAG
974 003066 017700 175614 MOV @LTADD,RO ;SET POINTER TO TEST
975 003072 000110 JMP (RO) ;GO TO TEST
976 003074 032777 002000 175446 TSCD2: BIT #2000,@SWR ;SEE IF HALT ON TEST
977 003102 001401 BEQ TSCD3 ;IF NOT: BR
978 003104 000000 HALT
979 003106 005737 000704 TSCD3: TST STFLG ;SE IF SINGLE TEST
  
```

980	003112	001746				BEQ	TSCD0		;IF NOT: BR
981	003114	017700	175430			MOV	@SWR,R0		
982	003120	042700	177740			BIC	#177740,R0		;BRANCH IF ALL TESTS SELECTED
983	003124	001613				BEQ	TSCD		
984	003126	012737	000001	000704	STSCD:	MOV	#1,STFLG		;SET SINGLE TEST FLAG
985	003134	023700	001120			CMP	TLAST,R0		;SEE IF EXCEEDED TESTS
986	003140	002410				BLT	TEND		;IF SO: BR
987	003142	006300				ASL	R0		
988	003144	006100				ROL	R0		;SET TABLE MODIFIER
989	003146	012737	000756	000706		MOV	#TSTBL,LTADD		
990	003154	060037	000706			ADD	R0,LTADD		;SET TEST POINTER
991	003160	000726				BR	TSCD1		
992	003162	005737	001662		TEND:	TST	CHNFLG		;BRANCH IF IN CHAIN MODE
993	003166	001265				BNE	NXTSLV		
994	003170	012704	015603		\$DONE:	MOV	#MSG6,R4		
995	003174	004737	014322			JSR	PC,TTOUT		;PRINT END OF PASS
996	003200	013703	000730			MOV	PCNTR,R3		
997	003204	004737	014452			JSR	PC,OCTP		;PRINT PASS NUMBER
998	003210	005000				CLR	R0		
999	003212	005300			1\$:	DEC	R0		
1000	003214	001376				BNE	1\$		
1001	003216	013700	000042			MOV	@#42,R0		;GET ACT11 RETURN ADDRESS
(1)	003222	001405				BEQ	HERE		;BRANCH IF NOT ACT11
(1)	003224	000005				RESET			
(1)	003226	004710			\$ENDAD:	JSR	PC,(R0)		
(1)	003230	000240				NOP			
(1)	003232	000240				NOP			
(1)	003234	000240				NOP			
(1)	003236	000240			HERE:	NOP			
1002	003240	005737	001662			TST	CHNFLG		;BRANCH IF IN CHAIN MODE
1003	003244	001005				BNE	TENDX		
1004	003246	032777	010000	175274		BIT	#10000,@SWR		;SEE IF HALT ON PASS
1005	003254	001001				BNE	TENDX		;IF NOT: BR
1006	003256	000000				HALT			
1007	003260	005237	000730		TENDX:	INC	PCNTR		;BUMP PASS COUNTER
1008	003264	000137	002554			JMP	TSCD		;RESTART

```
1010
1011
1012
1013 003270 012737 020132 000610 FT1:  MOV #MSFT1,EMADDR ;SET HEADER
1014 003276 012737 013664 000004      MOV #TRAP,@#4 ;SET TRAP HANDLER ADDRESS
1015 003304 012737 000340 000006      MOV #340,@#6
1016 003312 012700 000016      MOV #16,R0 ;SET NUMBER OF REGISTERS
1017 003316 013701 000510      MOV C1,R1 ;GET FIRST ADDRESS (CS1)
1018 003322 005711      FT1A:  TST (R1) ;REFERENCE REGISTER
1019 003324 000240      NOP ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
1020 003326 005300      FT1B:  DEC R0 ;SEE IF DONE ALL
1021 003330 001403      BEQ FT1X ;IF SO: BR
1022 003332 062701 000002      ADD #2,R1 ;BUMP ADDRESS POINTER
1023 003336 000771      BR FT1A ;CONTINUE
1024 003340 012737 000006 000004 FT1X:  MOV #6,@#4 ;RESET TRAP CATCHER
1025 003346 012737 000000 000006      MOV #HALT,@#6
1026 003354 005737 000722      TST RHTF ;SEE IF INITIAL ADDRESS TEST PASS
1027 003360 001404      BEQ FT1XX ;IF NOT: BR
1028 003362 005037 000722      CLR RHTF ;CLEAR FLAG
1029 003366 000137 002166      JMP ST1A ;RETURN
1030 003372 000137 003074      FT1XX: JMP TSCD2 ;RETURN TO SCHEDULAR

;RH ADDRESSING TEST*****
```

```

1032
1033
1034
1035 003376 012737 020157 000610 FT2:  MOV    #MSFT2,EMADDR  ;SET TEST HEADER
1036 003404 012701 177777          MOV    #-1,R1      ;SET ALL ONES PATTERN
1037 003410 004737 013500          FT2A: JSR    PC,INIT1   ;GO INIT
1038 003414 013700 000512          MOV    WC,R0      ;GET ADDRESS OF WORD COUNT
1039 003420 010102          MOV    R1,R2      ;SET EXPT REGISTER BIT PATTERN
1040 003422 010110          MOV    R1,(R0)    ;LOAD PATTERN
1041 003424 021002          CMP    (R0),R2    ;SEE IF EXPT=RCVD
1042 003426 001410          BEQ    FT2B       ;IF SO: BR
1043 003430 012737 016334 000650          MOV    #MSG25,ERADD ;SET CODE
1044 003436 012737 003410 000674          MOV    #FT2A,SCOLP ;SET SCOPE
1045 003444 004737 003564          JSR    PC,FT2ER   ;GO DO ERROR
1046 003450 013700 000514          FT2B: MOV    BA,R0      ;GET ADDRESS OF BUS ADDRESS
1047 003454 010102          MOV    R1,R2
1048 003456 042702 000001          BIC    #1,R2      ;SET EXPT PATTERN
1049 003462 010110          MOV    R1,(R0)    ;LOAD PATTERN
1050 003464 020210          CMP    R2,(R0)    ;SEE IF EXPT=RCVD
1051 003466 001410          BEQ    FT2C       ;IF SO:BR
1052 003470 012737 016342 000650          MOV    #MSG26,ERADD ;SET ERROR CODE
1053 003476 012737 003450 000674          MOV    #FT2B,SCOLP ;SET SCOPE ADDRESS
1054 003504 004737 003564          JSR    PC,FT2ER   ;GO DO ERROR
1055 003510 013700 000532          FT2C: MOV    DB,R0      ;GET ADDRESS OF DATA BUFFER
1056 003514 010102          MOV    R1,R2
1057 003516 010110          MOV    R1,(R0)    ;LOAD PATTERN
1058 003520 012703 004000          MOV    #4000,R3
1059 003524 005303          FT2D: DEC    R3        ;DELAY
1060 003526 001376          BNE    FT2D
1061 003530 020210          CMP    R2,(R0)    ;SEE IF EXPT=RCVD
1062 003532 001410          BEQ    FT2E       ;IF SO: BR
1063 003534 012737 016350 000650          MOV    #MSG27,ERADD ;SET ERROR CODE
1064 003542 012737 003510 000674          MOV    #FT2C,SCOLP ;SET SCOPE ADDRESS
1065 003550 004737 003564          JSR    PC,FT2ER   ;GO DO ERROR
1066 003554 005701          FT2E: TST    R1        ;SEE IF DONE RESET
1067 003556 001453          BEQ    FT2X       ;IF SO: BR
1068 003560 005001          CLR    R1        ;SET ZERO PATTERN
1069 003562 000712          BR     FT2A       ;DO ZERO BITS
1070 003564 000240          FT2ER: NOP
1071 003566 032777 020000 174754          BIT    #20000,@SWR ;SEE IF PRINT ERROR
1072 003574 001034          BNE    FT2ERB     ;IF NOT: BR
1073 003576 005737 000606          TST    HDRFL      ;SEE ID DONE HEADER
1074 003602 001004          BNE    FT2ERA     ;IF SO: BR
1075 003604 013704 000610          MOV    EMADDR,R4
1076 003610 004737 014322          JSR    PC,TTOUT   ;DO HEADER
1077 003614 012737 000001 000606 FT2ERA: MOV    #1,HDRFL   ;SET FLAG
1078 003622 013704 000650          MOV    ERADD,R4
1079 003626 004737 014322          JSR    PC,TTOUT   ;PRINT ERROR CODE
1080 003632 012704 016300          MOV    #MSG22,R4
1081 003636 004737 014322          JSR    PC,TTOUT   ;PRINT EXPT TAG
1082 003642 010103          MOV    R1,R3
1083 003644 004737 014440          JSR    PC,OCTPE   ;PRINT EXPT
1084 003650 012704 016310          MOV    #MSG23,R4
1085 003654 004737 014322          JSR    PC,TTOUT   ;PRINT RCVD TAG
1086 003660 011003          MOV    (R0),R3
1087 003662 004737 014440          JSR    PC,OCTPE   ;PRINT RCVD

```

CZTECDO TM03-TE16/TU77 BFT
CZTECD.P11 06-JUL-83 14:41

MACY11 30(1046) 06-JUL-83 21:02 ^{B 3} PAGE 23-1

SEQ 0027

1088	003666	005777	174656
1089	003672	100001	
1090	003674	000000	
1091	003676	004737	013274
1092	003702	000240	
1093	003704	000207	
1094	003706	000240	
1095	003710	004737	013330
1096	003714	000137	003074

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

```

1098
1099
1100 ;RH INITIALIZE TEST*****
1101 003720 012737 020214 000610 FT3: MOV #MSFT3,EMADDR ;SET TEST HEADER
1102 003726 012737 003720 000674 MOV #FT3,SCOLP
1103 003734 004737 013500 JSR PC,INIT1 ;GO INIT
1104 003740 052777 020000 174552 BIS #20000,@CS ;FORCE UPE =1
1105 003746 000240 NOP
1106 003750 004737 013500 JSR PC,INIT1 ;GO INIT
1107 003754 005777 174530 TST @C1 ;SEE IF SC IS RESET
1108 003760 100005 BPL FT3A ;IF SO: BR
1109 003762 012737 016406 000650 MOV #MSG29,ERADD ;SET ERROR CODE
1110 003770 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1111 003774 032777 040000 174506 FT3A: PIT #40000,@C1 ;SEE IF TRE IS RESET
1112 004002 001405 BEQ FT3B ;IF SO: BR
1113 004004 012737 016435 000650 MOV #MSG30,ERADD ;SET ERROR CODE.
1114 004012 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1115 004016 017701 174476 FT3B: MOV @CS,R1 ;GET CS2
1116 004022 042701 000307 BIC #307,R1 ;MARK IR/OR
1117 004026 005701 TST R1 ;SEE IF RESET
1118 004030 001405 BEQ FT3X ;IF SO: BR
1119 004032 012737 016465 000650 MOV #MSG31,ERADD ;SET ERROR CODE
1120 004040 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1121 004044 004737 013330 FT3X: JSR PC,ITER ;GO SEE IF ITERATION
1122 004050 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
1123
1124 ;ERROR REPORT SUBROUTINE
1125 004054 000240 FT3ER: NOP
1126 004056 032777 020000 174464 BIT #20000,@SWR ;SEE IF PRINT ERROR
1127 004064 001015 BNE 2$ ;IF NOT: BR
1128 004066 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1129 004072 001006 BNE 1$ ;IF SO: BR
1130 004074 013704 000610 MOV EMADDR,R4
1131 004100 004737 014322 JSR PC,TTOUT ;PRINT HEADER
1132 004104 005237 000606 INC HDRFL
1133 004110 013704 000650 1$: MOV ERADD,R4
1134 004114 004737 014322 JSR PC,TTOUT ;PRINT ERROR CODE
1135 004120 005777 174424 2$: TST @SWR ;SEE IF HALT ON ERROR
1136 004124 100001 BPL 3$ ;IF NOT: BR
1137 004126 000000 HALT
1138 004130 000240 3$: NOP
1139 004132 004737 013274 JSR PC,SCOPE ;GO SEE IF SCOPE
1140 004136 000207 RTS PC ;IF NOT: BR

```

```
1142  
1143 ;RH11 SILO TEST 1: EPMTY SILO READ*****  
1144  
1145 004140 005737 000604 FT4: TST RH17F  
1146 004144 001141 BNE FT5X ;IF RH70: BR  
1147 004146 012737 020246 000610 MOV #MSFT4,EMADDR ;SET TEST TEST HEADER  
1148 004154 012777 000040 174336 MOV #40,@CS ;INIT  
1149 004162 017700 174344 MOV @DB,R0 ;READ DB  
1150 004166 005777 174326 TST @CS ;SEE IF DLT IS SET  
1151 004172 100013 BPL FT4ER ;IF NOT: BR  
1152 004174 005777 174310 TST @C1 ;SEE IF SC IS SET  
1153 004200 100014 BPL FT4ERA ;IF NOT: BR  
1154 004202 032777 040000 174300 BIT #40000,@C1 ;SEE IF TRE IS SET  
1155 004210 001414 BEQ FT4ERB ;IF NOT: BR  
1156 004212 004737 013330 FT4X: JSR PC,ITER ;GO SEE IF ITERATION  
1157 004216 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR  
1158 004222 012737 016515 000650 FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE  
1159 004230 000407 BR FT4ERC  
1160 004232 012737 016533 000650 FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE  
1161 004240 000403 BR FT4ERC  
1162 004242 012737 016550 000650 FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.  
1163 004250 000240 FT4ERC: NOP  
1164 004252 012737 004140 000674 MOV #FT4,SCOLP ;SET SCOPE ADDRESS  
1165 004260 004737 004054 JSR PC,FT3ER ;GO PRINT ERROR  
1166 004264 000752 BR FT4X
```

```
1168
1169
1170 ;RH11 SILO TEST 2: IR/OR CHECK*****
1171 004266 005737 000604 FT5: TST RH17F ;SEE IF RH70
1172 004272 001066 BNE FT5X ;IF SO: BR
1173 004274 012737 020276 000610 MOV #MSFT5,EMADDR ;SET TEST HEADER
1174 004302 012737 004310 000674 MOV #FT5A,SCOLP ;SET SCOPE ADDRESS
1175 004310 004737 013500 FT5A: JSR PC,INIT1 ;GO INIT
1176 004314 032777 000100 174176 BIT #100,ACS ;SEE IF IR IS SET
1177 004322 001005 BNE FT5B ;IF SO: BR
1178 004324 012737 016566 000650 MOV #MSG35,ERADD ;SET ERROR CODE
1179 004332 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1180 004336 032777 000200 174154 FT5B: BIT #200,ACS ;SEE IF OR IS RESET
1181 004344 001405 BEQ FT5C ;IF SO: BR
1182 004346 012737 016613 000650 MOV #MSG36,ERADD ;SET ERROR CODE
1183 004354 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1184 004360 012777 000000 174144 FT5C: MOV #0,ADB ;LOAD ZERO INTO SILO
1185 004366 032777 000200 174124 BIT #200,ACS ;SEE THAT OR RESET
1186 004374 001405 BEQ FT5D ;IF IT DOES: BR
1187 004376 012737 016642 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1188 004404 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1189 004410 012777 177777 174114 FT5D: MOV #-1,ADB ;LOAD SILO WITH -1
1190 004416 012700 004000 MOV #4000,R0
1191 004422 032777 000200 174070 FT5E: BIT #200,ACS ;SEE IF OR IS SET
1192 004430 001007 BNE FT5X ;IF SO: BR
1193 004432 005300 DEC R0
1194 004434 001372 BNE FT5E ;AWAIT OR
1195 004436 012737 016642 000650 MOV #MSG37,ERADD ;SET ERROR CODE
1196 004444 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1197 004450 004737 013330 FT5X: JSR PC,ITER ;GO SEE IF ITERATION
1198 004454 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

1200
1201 ;RH11 SILO TEST 3: SILO DATA TEST*****
1202
1203 004460 005737 000604 FT6: TST RH17F
1204 004464 001052 BNE FT6X ;IF RH70: BR
1205 004466 012737 020326 000610 MOV #MSG40,EMADDR ;SET TEST HEADER
1206 004474 012737 004502 000674 MOV #FT6A,SCOLP ;SET SCOPE ADDRESS
1207 004502 004737 013500 FT6A: JSR PC,INIT1 ;GO INIT
1208 004506 005000 CLR R0 ;PRESET DATA
1209 004510 010077 174016 FT6B: MOV R0,@DB ;LOAD SILO
1210 004514 005200 INC R0 ;BUMP DATA
1211 004516 022700 000102 CMP #102,R0 ;SEE IF FILLED ALL
1212 004522 001372 BNE FT6B ;IF NOT: BR
1213 004524 032777 000100 173766 BIT #100,@CS ;SEE IF IR IS RESET.
1214 004532 001405 BEQ FT6C ;IF SO: BR
1215 004534 012737 016753 000650 MOV #MSG40,ERADD ;SET ERROR CODE
1216 004542 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1217 004546 032777 000200 173744 FT6C: BIT #200,@CS ;SEE IF OR IS SET
1218 004554 001005 BNE FT6D ;IF SO: BR
1219 004556 012737 016701 000650 MOV #MSG38,ERADD ;SET ERROR CODE
1220 004564 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1221 004570 005000 FT6D: CLR R0 ;PRESET DATA
1222 004572 017701 173734 FT6E: MOV @DB,R1 ;READ SILO
1223 004576 020001 CMP R0,R1 ;SEE IF EXPT=RCVD
1224 004600 001010 BNE FT6DE ;IF NOT: BR
1225 004602 005200 INC R0 ;BUMP DATA
1226 004604 022700 000102 CMP #102,R0 ;SEE IF DONE ALL
1227 004610 001370 BNE FT6E ;IF NOT: BR
1228 004612 004737 013330 FT6X: JSR PC,ITER ;GO SEE IF ITERATION
1229 004616 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
1230
1231 004622 000240 FT6DE: NOP
1232 004624 032777 020000 173716 BIT #20000,@SWR ;SEE IF PRINT ERROR
1233 004632 001032 BNE FT6DEB ;IF NOT: BR
1234 004634 005737 000606 TST HDRFL ;SEE IF DONE HEADER
1235 004640 013701 000610 MOV EMADDR,R1
1236 004644 004737 014322 JSR PC,TTOUT ;PRINT HEADER
1237 004650 005237 000606 INC HDRFL ;SET FLAG
1238 004654 012704 016733 FT6DEA: MOV #MSG39,R4
1239 004660 004737 014322 JSR PC,TTOUT ;PRINT SILO READ ERROR
1240 004664 012704 016300 MOV #MSG22,R4
1241 004670 004737 014322 JSR PC,TTOUT ;PRINT EXPT TAG
1242 004674 010003 MOV R0,R3
1243 004676 004737 014452 JSR PC,OCTP ;PRINT EXPT
1244 004702 012704 016310 MOV #MSG23,R4
1245 004706 004737 014322 JSR PC,TTOUT ;PRINT RCVD TAG
1246 004712 010103 MOV R1,R3
1247 004714 004737 014452 JSR PC,OCTP ;PRINT RCVD
1248 004720 005777 173624 FT6DEB: TST @SWR ;SEE IF HALT ON ERROR
1249 004724 100001 BPL FT6DEX ;IF NOT: BR
1250 004726 000000 HALT
1251 004730 000207 FT6DEX: RTS PC ;RETURN TO TEST

```

```
1253  
1254  
1255  
1256 004732 005737 000604 FT7: TST RH17F  
1257 004736 001021 BNE FT7X ;IF RH70: BR  
1258 004740 012737 020356 000610 MOV #MSFT7,EMADDR ;SET TEST HEADER  
1259 004746 012737 004732 000674 MOV #FT7,SCOLP ;SET SCOPE ADDRESS  
1260 004754 004737 013500 JSR PC,INIT1 ;GO INIT  
1261 004760 012700 000103 MOV #103,R0 ;SET SIZE OF SILO +1  
1262 004764 010077 173542 FT7A: MOV R0,ADB ;LOAD SILO  
1263 004770 005300 DEC R0 ;SEE IF DONE  
1264 004772 001374 BNE FT7A ;IF NOT: BR  
1265 004774 005777 173520 TST @CS ;SEE IF DLT IS SET  
1266 005000 100004 BPL FT7ER ;IF NOT: BR  
1267 005002 004737 013330 FT7X: JSR PC,ITER ;GO SEE IF ITERATION  
1268 005006 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR  
1269 005012 012737 016515 000650 FT7ER: MOV #MSG32,ERADD ;SET ERROR CODE  
1270 005020 004737 004054 JSR PC,FT3ER ;GO DO ERROR  
1271 005024 000766 BR FT7X
```

```
1273  
1274  
1275  
1276 005026 005737 000604 FT10: TST RH17F  
1277 005032 001034 BNE FT10X ;IF RH70: BR  
1278 005034 012737 020406 000610 MOV #MSFT10,EMADDR ;SET TEST HEADER  
1279 005042 012737 005026 000674 MOV #FT10,SCOLP ;SET SCOPE ADDRESS  
1280 005050 012777 000040 173442 MOV #4,@CS ;INITIALIZE  
1281 005056 012700 000004 MOV #4,R0 ;SET NUMBER OF SILO WRITER  
1282 005062 010077 173444 FT10A: MOV R0,@DB ;WRITE SILO  
1283 005066 005300 DEC R0 ;SEE IF DONE  
1284 005070 001374 BNE FT10A ;IF NOT: BR  
1285 005072 052777 000040 173420 BIS #4,@CS ;INITIALIZE  
1286 005100 012777 177777 173424 MOV #-1,@DB ;WRITE SILO  
1287 005106 017701 173420 MOV @DB,R1 ;READ SILO 1  
1288 005112 017701 173414 MOV @DB,R1 ;READ SILO 2  
1289 005116 005777 173376 TST @CS ;SEE IF DLT IS SET  
1290 005122 100011 BPL FT10ER ;IF NOT: BR  
1291 005124 004737 013330 FT10X: JSR PC,ITER ;GO SEE IF ITERATION  
1292 005130 005737 000726 TST RHOF ;SEE IF RH11 ONLY  
1293 005134 001402 BEQ FT10XX ;IF NOT: BR  
1294 005136 000137 003162 JMP TEND ;ELSE GO TO END  
1295 005142 000137 003074 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR  
1296 005146 012737 016515 000650 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE  
1297 005154 004737 004054 JSR PC,FT3ER ;GO DO ERROR  
1298 005160 000761 BR FT10X
```

```
1300 ;NOP TEST*****
1301
1302 005162 000240 FT11: NOP
1303 005164 012737 005162 000674 MOV #FT11,SCOLP ;SET SCOPE ADDRESS
1304 005172 004737 013500 JSR PC,INIT1
1305 005176 012737 000300 000716 MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
1306 005204 012737 177777 000620 MOV #-1,FCNT ;SET FC= ALL OVER
1307 005212 012737 177777 000622 MOV #-1,WCNT ;SET WC= ALL OVER
1308 005220 012737 177777 000616 MOV #-1,BADDR ;SET BA= ALL OVER
1309 005226 012737 000001 000636 MOV #1,RDYDX ;SET DELAY
1310 005234 012737 000001 000640 MOV #1,OPDYX ;SET OP DELAY
1311 005242 012737 000001 000710 MOV #1,FUN ;SET NOP FUNCTIONS CODE
1312 005250 004737 012310 JSR PC,EXEC ;GO EXECUTE COMMAND
1313 005254 000240 NOP
1314 005256 012737 020437 000610 MOV #MSFT11,EMADDR
1315 005264 004737 012510 JSR PC,ERCHK ;GO CHECK REGISTER
1316 005270 004737 013330 JSR PC,ITER ;GO SEE IF ITERATIONS
1317 005274 000137 003074 JMP TSCD2 ;RETURN TO SCHEDULAR
```

```
1319                                     ;REWIND TEST*****
1320
1321 005300 000240                                     FT12: NOP
1322 005302 012737 005300 000674                   MOV #FT12,SCOLP
1323 005310 004737 013500                               JSR PC,INIT1           ;GO INITIALIZE
1324 005314 052777 001700 173220                   BIS #1700,@TC         ;SET TO NRZ,NORMAL
1325 005322 012737 177760 000620                   MOV #-20,FCNT        ;SET FC=20
1326 005330 012737 177770 000622                   MOV #-10,WCNT        ;SET WC=10
1327 005336 012737 021314 000616                   MOV #WDATA,BADDR    ;SET BA=WRITE BUFFER
1328 005344 012737 000007 000710                   MOV #7,FUN           ;SET REWIND OP CODE
1329 005352 004737 012310                               JSR PC,EXEC          ;GO EXECUTE COMMAND
1330 005356 000240                                     NOP
1331 005360 032777 020000 173134 FT12A: BIT #20000,@DS
1332 005366 001374                               BNE FT12A            ;AWAIT PIP
1333 005370 012737 020457 000610                   MOV #MSFT12,EMADDR
1334 005376 004737 012510                               JSR PC,FPCHK         ;GO CHECK FOR ERROR
1335 005402 004737 013330                               JSR PC,ITER         ;GO SEE IF ITERATION
1336 005406 000137 003074                               JMP TSCD2           ;RETURN TO SCHEDULAR
1337
```

```
1339                                     ;WRITE/READ TEST*****
1340
1341 005412 000240 FT13: NOP
1342 005414 012737 000001 000636 MOV #1,RDYDX
1343 005422 012737 000001 000640 MOV #1,OPDYX
1344 005430 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
1345 005436 012737 020502 000610 MOV #MSFT13,EMADDR ;SET TEST HEADER
1346 005444 012737 000001 000720 MOV #1,PATRN
1347 005452 004737 013162 JSR PC,DSUP ;SET UP ALL ONES DATA PATTERN
1348 005456 012737 001700 000716 MOV #1700,UDES ;SET TO 800 BPI NORMAL
1349 005464 004737 012442 FT13A: JSR PC,RWIND ;GO REWIND
1350 005470 012737 177600 000620 MOV #-200,FCNT ;SET FC
1351 005476 012737 177700 000622 MOV #-100,WCNT ;SET WC
1352 005504 012737 021314 000616 MOV #WDATA,BADDR ;SET BA
1353 005512 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
1354 005520 012737 016047 000626 MOV #MSG12,ERRP
1355 005526 004737 012310 FT13B: JSR PC,EXEC ;GO EXECUTE COMMAND
1356 005532 005037 000674 CLR SCOLP ;NO SCOPE LOOP
1357 005536 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1358 005542 005337 000624 DEC RCNT ;SEE IF DONE ALL
1359 005546 001367 BNE FT13B ;IF NOT: BR
1360 005550 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
1361 005556 012737 023026 000616 MOV #RDATA,BADDR
1362 005564 062737 000200 000616 ADD #200,BADDR ;SET BA
1363 005572 012737 000077 000710 MOV #77,FUN ;SET READ REVERSE OP-CPDE
1364 005600 012737 016065 000626 MOV #MSG13,ERRP
1365 005606 004737 012310 FT13C: JSR PC,EXEC ;GO EXECUTE COMMAND
1366 005612 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1367 005616 005337 000624 DEC RCNT ;SEE IF READ ALL
1368 005622 001371 BNE FT13C ;IF NOT:BR
1369 005624 162737 000200 000616 SUB #200,BADDR ;SET BA
1370 005632 012737 000071 000710 MOV #71,FUN ;SET READ FORWARD OP-CODE
1371 005640 012737 016112 000626 MOV #MSG14,ERRP
1372 005646 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
1373 005654 004737 012310 FT13D: JSR PC,EXEC ;GO EXECUTE COMMAND
1374 005660 004737 012510 JSR PC,ERCHK ;GO CHECK ERRORS
1375 005664 005337 000624 DEC RCNT ;SEE IF DONE ALL
1376 005670 001371 BNE FT13D ;IF NOT:BR
1377 005672 032737 002000 000716 BIT #2000,UDES ;SEE IF DONE PE
1378 005700 001007 BNE FT13X ;IF SO: BR
1379 005702 012737 002300 000716 MOV #2300,UDES ;SET PE MODE
1380 005710 012737 000100 000624 MOV #100,RCNT ;RESET RECORD COUNT
1381 005716 000662 BR FT13A ;GO DO NEXT DENSITY
1382 005720 000137 003074 FT13X: JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

1384                                     ;SPACE TEST*****
1385
1386 005724 000240 FT14:  NOP
1387 005726 012737 020531 000610  MOV      #MSFT14,EMADDR ;SET TEST HEADER
1388 005734 012737 001700 000716  MOV      #1700,UDES  ;SET NRZ,NORMAL
1389 005742 004737 012442          FT14A1: JSR      PC,RWND   ;GO INITIALIZE
1390 005746 012737 000100 000624  MOV      #100,RCNT  ;SET NUMBER OF RECORDER
1391 005754 012737 177777 021314  MOV      #-1,WDATA  ;SET DATA PATTERN
1392 005762 012737 177700 000620  MOV      #-100,FCNT ;PRESET FRAME CNT
1393 005770 012737 177740 000622  MOV      #-40,WCNT  ;PRESET WORD CNT
1394 005776 004737 013500          FT14A:  JSR      PC,INIT1 ;GO REWIND
1395 006002 012737 001000 000640  MOV      #1000,OPDYX
1396 006010 012737 040000 000636  MOV      #40000,RDYDX
1397 006016 012737 000061 000710  MOV      #61,FUN    ;SET WRITE OP-CODE
1398 006024 012737 102300 000660  MOV      #102300,STMSK ;MASK DATA RELATED ERRORS
1399 006032 052777 000010 172460  BIS      #10,ACS   ;INHIBIT BUS ADDRESS INCREMENT
1400 006040 004737 012310          JSR      PC,EXEC   ;GO EXECUTE COMMAND
1401 006044 012737 017172 000626  MOV      #MSG46,ERRP ;SET ERROR CODE
1402 006052 004737 012510          JSR      PC,ERCHK  ;GO CHECK ERRORS
1403 006056 005737 000712          TST      SERFL    ;SEE IF ERROR
1404 006062 001402          BEQ      FT14A2   ;IF NOT: BR
1405 006064 000137 006550          JMP      FT14X    ;ELSE EXIT
1406 006070 005337 000620          FT14A2: DEC      FCNT  ;BUMP FC
1407 006074 032737 000001 000620  BIT      #1,FCNT   ;SEE IF SHOULD BUMP WC
1408 006102 001403          BEQ      FT14A3   ;IF NOT: BR
1409 006104 162737 000001 000622  SUB      #1,WCNT   ;BUMP WC
1410 006112 005337 000624          FT14A3: DEC      RCNT  ;SEE IF DONE ALL
1411 006116 001327          BNE      FT14A    ;WRITE ALL RECORDS
1412 006120 012737 000100 000632  MOV      #100,RRD  ;PRESET RECORD POSITION
1413 006126 012737 000176 000634  MOV      #176,RFD
1414 006134 012737 177701 000642  MOV      #-77,SCNT ;SET SPACE AMOUNT
1415 006142 012737 000033 000710  FT14B:  MOV      #33,FUN ;SET OP-CODE SPACE REVERSE
1416 006150 004737 012310          JSR      PC,EXEC  ;GO EXECUTE COMMAND
1417 006154 012737 017243 000626  MOV      #MSG48,ERRP ;SET ERROR CODE
1418 006162 004737 012510          JSR      PC,ERCHK ;GO CHECK ERRORS
1419 006166 005737 000712          TST      SERFL   ;SEE IF ERROR
1420 006172 001166          BNE      FT14X    ;IF SO: BR
1421 006174 004737 006270          JSR      PC,FT14RR ;GO READ REVERSE + CHECK DATA
1422 006200 000240          NOP
1423 006202 012737 000031 000710  MOV      #31,FUN  ;SET SPACE FORWARD OP-CODE
1424 006210 005237 000642          INC      SCNT    ;SET SPACE AMOUNT
1425 006214 001555          BEQ      FT14X   ;IF DONE: BR
1426 006216 004737 012310          JSR      PC,EXEC  ;GO EXECUTE COMMAND
1427 006222 012737 017216 000626  MOV      #MSG47,ERRP ;SET ERROR CODE
1428 006230 004737 012510          JSR      PC,ERCHK ;GO CHECK ERROR
1429 006234 005737 000712          TST      SERFL   ;SEE IF ERROR FLAG
1430 006240 001143          BNE      FT14X   ;IF NO: BR
1431 006242 004737 006332          JSR      PC,FT14RF ;GO READ FORWARD FOR POSITION CHECK
1432 006246 000240          NOP
1433 006250 005237 000642          INC      SCNT    ;DECREMENT SPACE AMOUNT
1434 006254 001535          BEQ      FT14X   ;IF DONE: BR
1435 006256 005237 000632          INC      RRD    ;BUMP DATA EXPT
1436 006262 005337 000634          DEC      RFD    ;BUMP DATA EXPT
1437 006266 000725          BR      FT14B
1438 006270 000240          FT14RR: NOP
1439 006272 012737 023026 000616  MOV      #RDATA,BADDR ;SET BA
  
```

1440	006300	012737	000077	000710	MOV	#77,FUN	:SET READ REVERSE OP-CODE
1441	006306	004737	012310		JSR	PC,EXEC	:GO EXECUTE COMMAND
1442	006312	000240			NOP		
1443	006314	013705	000632		MOV	RRD,R5	
1444	006320	020577	172172		CMP	R5,@FC	:SEE IF CORRECT RECORD
1445	006324	001020			BNE	FT14RER	:IF NOT: BR
1446	006326	000137	006360		JMP	FT14EC	:GO CLEAR RH11 ERROR BIT
1447	006332	000240			NOP		
1448	006334	012737	000071	000710	MOV	#71,FUN	:SET READ FORWARD OP-CODE
1449	006342	004737	012310		JSR	PC,EXEC	:GO EXECUTE COMMAND
1450	006346	013705	000634		MOV	RFD,R5	
1451	006352	020577	172140		CMP	R5,@FC	:SEE IF CORRECT RECORD
1452	006356	001003			BNE	FT14RER	:IF NOT: BR
1453	006360	004737	013500		JSR	PC,INIT1	:CLEAR RH
1454	006364	000207			RTS	PC	:RETURN
1455	006366	000240			NOP		
1456	006370	032777	020000	172152	BIT	#2000,@SWR	:SEE IF PRINT INHIBITED
1457	006376	001060			BNE	FT14R3	:IF SO: BR
1458	006400	012704	020531		MOV	#MSFT14,R4	
1459	006404	004737	014322		JSR	PC,TTOUT	:PRINT HEADER
1460	006410	012704	015625		MOV	#MSG9,R4	
1461	006414	004737	014322		JSR	PC,TTOUT	:PRINT ERROR TYPE
1462	006420	012704	016265		MOV	#MSG20,R4	:SET NRZ TAG POINTER
1463	006424	032737	002000	000716	BIT	#2000,UDES	:SEE IF PE
1464	006432	001402			BEQ	FT14R0	:IF NOT: BR
1465	006434	012704	016273		MOV	#MSG21,R4	:ELSE SET PE TAG POINTER
1466	006440	004737	014322		JSR	PC,TTOUT	:PRINT TAG
1467	006444	032737	000002	000710	BIT	#2,FUN	:SEE IF READ REVERSE
1468	006452	001003			BNE	FT14R1	:IF SO: BR
1469	006454	012704	016245		MOV	#MSG17,R4	
1470	006460	000402			BR	FT14R2	:GO PRINT
1471	006462	012704	016225		MOV	#MSG16,R4	
1472	006466	004737	014322		JSR	PC,TTOUT	:PRINT FRWD/REV
1473	006472	012704	016300		MOV	#MSG22,R4	
1474	006476	004737	014322		JSR	PC,TTOUT	:PRINT EXPT TAG
1475	006502	010503			MOV	R5,R3	
1476	006504	042703	177700		BIC	#177700,R3	:MASK RECORD NUMBER
1477	006510	004737	014452		JSR	PC,OCTP	:PRINT EXPT RECORD NUMBER
1478	006514	012704	016310		MOV	#MSG23,R4	
1479	006520	004737	014322		JSR	PC,TTOUT	:PRINT RCVD TAG
1480	006524	017703	171766		MOV	@FC,R3	
1481	006530	042703	177700		BIC	#177700,R3	:MASK RECORD NUMBER
1482	006534	004737	014452		JSR	PC,OCTP	:PRINT ACTUAL RECORD NUMBER
1483	006540	005777	172004		TST	@SWR	:SEE IF HALT ON ERROR
1484	006544	100001			BPL	FT14X	:IF NOT: BR
1485	006546	000000			HALT		
1486	006550	032737	002000	000716	BIT	#2000,UDES	:SEE IF DONE PE
1487	006556	001005			BNE	FT14XX	:IF SO: BR
1488	006560	012737	002300	000716	MOV	#2300,UDES	:SET TO PE
1489	006566	000137	005742		JMP	FT14A1	:DO IN PE
1490	006572	000137	003074		JMP	TSCD2	:RETURN TO SCHEDULAR

```

1492                                     ;ERASE TEST*****
1493
1494 006576 000240          FT15:  NOP
1495 006600 005037 000660      CLR      STMSK
1496 006604 012737 000100 000636  MOV      #100,RDYDX
1497 006612 012737 000010 000640  MOV      #10,OPDYX
1498 006620 012737 020553 000610  MOV      #MSFT15,EMADDR ;SET TEST HEADER
1499 006626 004737 012442      JSR      PC,RWND ;REWIND
1500 006632 012737 023026 000616  MOV      #RDATA,BADDR ;SET BA
1501 006640 012737 001700 000716  MOV      #1700,UDES ;SET NRZ, NORMAL
1502 006646 012737 000025 000710  FT15A:  MOV      #25,FUN ;SET ERASE OP-CODE
1503 006654 012737 000454 000624  MOV      #300,RCNT ;++B SET TO ERASE 300 TIMES
1504 006662 004737 012310      FT15B:  JSR      PC,EXEC ;GO EXECUTE COMMAND
1505 006666 012737 017172 000626  MOV      #MSG46,ERRP ;SET ERROR CODE
1506 006674 004737 012510      JSR      PC,ERCHK ;GO CHECK ERRORS
1507 006700 005737 000712      TST      SERFL ;SEE IF ANY ERRORS
1508 006704 001032          BNE      FT15X ;IF SO EXIT
1509 006706 005337 000624      DEC      RCNT ;SEE IF DONE ERASING
1510 006712 001363          BNE      FT15B ;IF NOT: BR
1511 006714 000240          NOP
1512 006716 004737 012442      JSR      PC,RWND ;REWIND
1513 006722 012737 177600 000622  MOV      #-200,WCNT ;SET WC
1514 006730 012737 000071 000710  MOV      #71,FUN ;SET READ FORWARD OP-CODE
1515 006736 012737 000040 000636  MOV      #40,RDYDX ;SET DELAY
1516 006744 004737 012310      JSR      PC,EXEC ;GO EXECUTE COMMAND
1517 006750 000240          NOP
1518 006752 012737 017644 000626  MOV      #MSG60,ERRP ;SET ERROR CODE
1519 006760 012737 020000 000660  MOV      #20000,STMSK
1520 006766 004737 012510      JSR      PC,ERCHK ;GO CHECK ERRORS
1521                                     ;*****
1522
1523                                     ;THIS CODE ADDED TO FORM REV C
1524
1525                                     ;THE SSC BIT AND THE PIP BIT IN THE DRIVE STATUS REG
1526                                     ;SHOULD NOT BE SET CONCURRENTLY
1527
1528                                     ;*****
1529
1530 006772 012737 000100 000636  FT15X:  MOV      #100,RDYDX ;SET DELAY
1531 007000 012737 000010 000640      MOV      #10,OPDYX
1532 007006 012737 000020 000624      MOV      #20,RCNT ;SET UP FOR 20 ERASES
1533 007014 012737 023026 000616      MOV      #RDATA,BADDR ;SET UP BUSS ADDRS
1534 007022 012737 001700 000716      MOV      #1700,UDES ;SET UP TAPE CONTROL
1535 007030 012737 000025 000710      MOV      #25,FUN ;SET FUN FOR ERASE
1536 007036 004737 012310      JSR      PC,EXEC ;GO EXECUTE CMD
1537 007042 005337 000624      DEC      RCNT ;DECREMENT THE NUMBER OF EXECUTES
1538 007046 001362          BNE      1$ ;BRANCH IF MORE LEFT
1539 007050 052777 001700 171464      BIS      #1700,@TC
1540 007056 012777 177760 171432      MOV      #-20,@FC
1541 007064 012777 177770 171420      MOV      #-10,@WC
1542 007072 012777 021314 171414      MOV      #WDATA,@BA
1543 007100 012777 000007 171402      MOV      #7,@C1 ;DO REWIND
1544 007106 000240          NOP
1545 007110 032777 000100 171404      2$:  BIT      #100,@DS ;WAIT FOE SSC
1546 007116 001774          BEQ      2$
1547 007120 017737 171376 000652      MOV      @DS,TEMP1 ;READ DRIVE STATUS REG IMMEDIATELY

```

```

1548 007126 032737 020000 000652      BIT      #20000,TEMP1      ;CHECK FOR PIP
1549 007134 001420                      BEQ      FT15XX          ;BRANCH IF NOT SET
1550 007136 052737 000001 007206      BIS      #1,TAG          ;SET FLAG FOR ERROR
1551 007144 012704 016217                      MOV      #MSG15B,R4
1552 007150 004737 014322                      JSR      PC,TTOUT
1553 007154 010703                      MOV      PC,R3
1554 007156 062703 000010                      ADD      #10,R3
1555 007162 004737 014440                      JSR      PC,OCTPE
1556
1557 007166 004737 012732                      JSR      PC,ERPTB1      ;GO PRINT ERROR
1558                                ;DS REG REPORTED IS ITS CONTENTS
1559                                ;AT THE TIME OF THE ERROR (5 LINES OF CODE BACK)
1560 007172 005037 007206                      CLR      TAG            ;CLEAR FLAG
1561 007176 004737 013330      FT15XX: JSR      PC,ITER    ;CHECK FOR ITERATIONS
1562 007202 000137 003074                      JMP      TSCD2          ;GO TO SCHEDULAR
1563
1564 007206 000000      TAG:      .WORD    0      ;++C FLAG FOR ERROR ROUTINE
  
```



```

1601
1602
1603
1604 007436 005037 000624 FT17: CLR RCNT
1605 007442 012737 020636 000610 MOV #MSFT17,EMADDR ;SET HEADER
1606 007450 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ
1607 007456 004737 012442 FT17A: JSR PC,RWND ;REWIND TAPE
1608 007462 012737 000027 000710 FT17B: MOV #27,FUN
1609 007470 012737 040000 000636 MOV #40000,RDYDX ;SET DRY DELAY
1610 007476 012737 040000 000640 MOV #40000,OPDYX ;SET OP DELAY
1611 007504 004737 012310 JSR PC,EXEC ;GO WRITE TM
1612 007510 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1613 007516 012737 016137 000626 MOV #MSG15,ERRP ;SET ERROR TYPE
1614 007524 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1615 007530 005737 000712 TST SERFL ;SEE IF ERROR
1616 007534 001137 BNE FT17X ;IF SO: BR
1617 007536 004737 013124 JSR PC,TMCHK ;GO SEE IF TM SET
1618 007542 000240 NOP
1619 007544 000240 NOP
1620 007546 032737 000100 000624 BIT #100,RCNT ;SEE IF DONE PATTEFN
1621 007554 001045 BNE FT17D ;IF SO: BR
1622 007556 062737 000020 000624 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1623 007564 013737 000624 000652 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1624 007572 012737 177600 000622 MOV #-200,WCNT ;WC=128
1625 007600 012737 177400 000620 MOV #-400,FCNT ;FC=256
1626 007606 012737 021314 000616 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1627 007614 012737 000061 000710 MOV #61,FUN ;SET WRITE OP CODE
1628 007622 000240 FT17C: NOP
1629 007624 000240 NOP
1630 007626 004737 012310 JSR PC,EXEC ;GO WRITE
1631 007632 012737 016047 000626 MOV #MSG12,ERRP ;SET ERROR CODE
1632 007640 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1633 007646 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1634 007652 005737 000712 TST SERFL ;SEE IF ERROR
1635 007656 001066 BNE FT17X ;IF SO: BR
1636 007660 005337 000652 DEC TEMP1 ;SEE IF DONE ALL
1637 007664 001356 BNE FT17C ;IF NOT: BR
1638 007666 000675 BR FT17B ;ELSE GO DO TM
1639 007670 000240 FT17D: NOP
1640 007672 012737 000033 000710 MOV #33,FUN ;SET SPACE REVERSE
1641 007700 012737 016225 000626 MOV #MSG16,ERRP ;SET ERROR CODE
1642 007706 012737 177600 000642 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1643 007714 012737 000005 000624 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1644 007722 004737 013500 FT17E: JSR PC,INIT1 ;GO INIT
1645 007726 004737 012310 JSR PC,EXEC ;GO SPACE
1646 007732 012737 001000 000660 MOV #1000,STMSK ;SET ERROR MASK
1647 007740 004737 012510 JSR PC,ERCHK ;GO CHECK ERROR
1648 007744 005737 000712 TST SERFL ;SEE IF ERROR
1649 007750 001031 BNE FT17X ;IF SO: BR
1650 007752 004737 013124 JSR PC,TMCHK ;GO SEE IF TM SET
1651 007756 005337 000624 DEC RCNT ;SEE IF DONE SPACES
1652 007762 001357 BNE FT17E ;IF NOT: BR
1653 007764 022737 000031 000710 CMP #31,FUN ;SEE IF DONE FORWARD
1654 007772 001407 BEQ FT17F ;IF SO: BR
1655 007774 012737 016245 000626 MOV #MSG17,ERRP ;SET ERROR CODE
1656 010002 012737 000031 000710 MOV #31,FUN ;SET TO SPACE FORWARD
  
```

CZTECDO TM03-TE16/TU77 BFT
CZTECD.P11 06-JUL-83 14:41

MACY11 30(1046) 06-JUL-83 21:02 E 4
PAGE 36-1

SEQ 0043

1657	010010	000736				BR	FT17D1	:DO FORWARD
1658	010012	032737	002000	000716	FT17F:	BIT	#2000,UDES	:SEE IF DONE PE
1659	010020	001005				BNE	FT17X	:IF SO: BR
1660	010022	012737	002300	000716		MOV	#2300,UDES	:SET TO PE
1661	010030	000137	007456			JMP	FT17A	:GO PE
1662	010034	000137	003074		FT17X:	JMP	TSCD2	:RETURN TO SCHEDULAR

```
1664  
1665  
1666  
1667 010040 000240  
1668 010042 012737 020664 000610  
1669 010050 012737 001700 000716  
1670 010056 004737 012442  
1671 010062 012737 000003 000720  
1672 010070 004737 013162  
1673 010074 012737 021314 000616  
1674 010102 012737 177400 000620  
1675 010110 012737 177600 000622  
1676 010116 012737 000061 000710  
1677 010124 004737 012310  
1678 010130 012737 017172 000626  
1679 010136 004737 012510  
1680 010142 005737 000712  
1681 010146 001042  
1682 010150 012737 016225 000626  
1683 010156 012737 000057 000710  
1684 010164 062737 000376 000616  
1685 010172 004737 012310  
1686 010176 004737 012510  
1687 010202 012737 016245 000626 FT20B:  
1688 010210 012737 000051 000710  
1689 010216 162737 000376 000616  
1690 010224 004737 012310  
1691 010230 004737 012510  
1692 010234 032737 002000 000716 FT20C:  
1693 010242 001004  
1694 010244 012737 002300 000716  
1695 010252 000701  
1696 010254 004737 013330 FT20X:  
1697 010260 000137 003074
```

:WRITE CHECK TEST*****

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

```

1699
1700                                     :ERASE HEAD TEST*****
1701
1702 010264 012737 020715 000610 FT21:  MOV  #MSFT21,EMADDR  :SET TEST HEADER
1703 010272 004737 012442          FT21A: JSR  PC,RWND      :GO REWIND
1704 010276 012737 000003 000720      MOV  #3,PATRN
1705 010304 004737 013162          JSR  PC,DSUP      :GO SET PATTERN 3
1706 010310 012737 021314 000616      MOV  #WDATA,BADDR :SET BA=WRITE BUFFER
1707 010316 012737 176340 000620      MOV  #-800.,FCNT  :SET FC=800(10)
1708 010324 012737 177160 000622      MOV  #-400.,WCNT  :SET WC=400(10)
1709 010332 012737 001700 000716      MOV  #1700,UDES   :SET NRZ, NORMAL
1710 010340 012737 000061 000710      MOV  #61,FUN      :SET WRITE OP-CODE
1711 010346 004737 012310          JSR  PC,EXEC      :GO DO WRITE 1
1712 010352 012737 016047 000626      MOV  #MSG12,ERRP  :SET ERROR CODE
1713 010360 004737 012510          JSR  PC,ERCHK     :GO CHECK FOR ERROR
1714 010364 004737 012310          JSR  PC,EXEC      :YES DO WRITE 2
1715 010370 004737 012510          JSR  PC,ERCHK     :YES CHECK FOR ERROR
1716 010374 000240          NOP
1717 010376 004737 012442          JSR  PC,RWND      :GO REWIND
1718 010402 012737 177160 000620      MOV  #-400.,FCNT  :SET FC=400(10)
1719 010410 012737 177470 000622      MOV  #-200.,WCNT  :SET WC=200(10)
1720 010416 004737 012310          JSR  PC,EXEC      :GO REWRITE RECORD 1-WH TO EH
1721 010422 000240          FT21SCP:NOP
1722 010424 004737 012442          JSR  PC,RWND      :REWIND
1723 010430 012737 023026 000616      MOV  #RDATA,BADDR :SET BA=READ BUFFER
1724 010436 012737 177160 000620      MOV  #-400.,FCNT  :SET FC=400
1725 010444 012737 177470 000622      MOV  #-200.,WCNT  :SET WC=200
1726 010452 012737 000071 000710      MOV  #71,FUN      :SET READ OP-CODE
1727 010460 004737 012310          JSR  PC,EXEC      :GO READ RECORD 1
1728 010464 012737 016112 000626      MOV  #MSG14,ERRP  :SET ERROR CODE
1729 010472 004737 012510          JSR  PC,ERCHK     :GO CHECK FOR ERROR
1730 010476 000240          NOP
1731 010500 052777 000010 170012      BIS  #10,ACS      :INHIBIT BA INCREMENT
1732 010506 012737 176340 000620      MOV  #-800.,FCNT  :SET FC=800(10)
1733 010514 012737 177160 000622      MOV  #-400.,WCNT  :SET WC=400(10)
1734 010522 004737 012310          JSR  PC,EXEC      :GO READ RECORD 2
1735 010526 022777 001440 167762      CMP  #800.,@FC    :SEE IF READ RECORD 2 OK
1736 010534 001424          BEQ  FT21X        :IF SO: BR
1737 010536 022777 001441 167752      CMP  #801.,@FC    :BRANCH IF IN GREY AREA
1738 010544 001420          BEQ  FT21X
1739 010546 022777 001440 167742 1$:  CMP  #800.,@FC    :BRANCH IF ERASE HEAD REVERSED
1740 010554 101404          BLOS FT21B        :IF SO: BR
1741 010556 012737 017065 000650      MOV  #MSG44,ERADD :SET ERASE HEAD INOPERATIVE ERROR CODE
1742 010564 000403          BR  FT21C
1743 010566 012737 017115 000650 FT21B: MOV  #MSG45,ERADD  :SET ERASE HEAD REVERSED ERROR CODE
1744 010574 012737 010422 000674 FT21C: MOV  #FT21SCP,SCOLP :SET SCOPE ADDRESS
1745 010602 004737 004054          JSR  PC,FT3ER     :GO PRINT ERROR
1746 010606 004737 013330          FT21X: JSR  PC,ITER   :GO SEE IF ITERATION
1747 010612 000137 003074          JMP  TSCD?        :RETURN TO SCHEDULAR
1748
1749

```

```
1751                                     :BUFFERED COMMAND TEST*****
1752
1753 010616 012737 020744 000610 FT22: MOV #MSFT22,EMADDR :SET TEST HEADER
1754 010624 004737 012442 JSR PC,RWIND :GO REWIND
1755 010630 012700 000003 MOV #3,R0 :SET NUMBER OF WRITES
1756 010634 012737 001700 000716 MOV #1700,UDES :SET TO NRZ NORMAL
1757 010642 012737 021314 000616 MOV #WDATA,BADDR :SET BA=WRITE BUFFER
1758 010650 012737 177000 000620 MOV #-1000,FCNT :SET FC=1000
1759 010656 012737 177400 000622 MOV #-400,WCNT :SET WC=400
1760 010664 012737 000061 000710 MOV #61,FUN :SET WRITE OP-CODE
1761 010672 004737 012310 FT22A: JSR PC,EXEC :GO DO WRITE
1762 010676 005300 DEC R0 :SEE IF DONE ALL
1763 010700 001374 BNE FT22A :IF NOT: BR
1764 010702 000240 NOP
1765 010704 012777 000007 167576 MOV #7,ac1 :START REWIND
1766 010712 032777 000200 167602 FT22B: BIT #200,ads
1767 010720 001774 BEQ FT22B
1768 010722 004737 013500 JSR PC,INIT1 :INITIALIZE
1769 010726 012737 000010 000636 MOV #10,RDYDX :SET LONG READY DELAY
1770 010734 004737 012310 JSR PC,EXEC :ISSUE BUFFERED WRITE
1771 010740 000240 NOP
1772 010742 012737 017270 000626 MOV #MSG49,ERRP :SET ERROR CODE
1773 010750 012737 102300 000660 MOV #102300,STMSK :MARK DATA ERROR
1774 010756 004737 012510 JSR PC,ERCHK :GO CHECK ERROR
1775 010762 032777 000002 167532 BIT #2,ads :SEE IF BOT IS SET
1776 010770 001410 BEQ FT22X :IF NOT: BR
1777 010772 012737 017316 000650 MOV #MSG50,ERADD :SET ERROR CODE
1778 011000 012737 010616 000674 MOV #FT22,SCOLP
1779 011006 004737 004054 JSR PC,FT3ER :GO DO ERROR
1780 011012 004737 013330 FT22X: JSR PC,ITER :GO SEE IF ITERATION
1781 011016 000137 003074 JMP TSCD2 :RETURN TO SCHEDULAR
1782
1783
```

```

1785                                     ;READ-IN PRESET TEST*****
1786
1787 011022 005737 000614          FT23: TST      SLVN      ;SEE IF SLAVE SELECT=0
1788 011026 001103                BNE      FT23X    ;IF NOT:BR
1789 011030 012737 021001 000610  MOV      #MSFT23,EMADDR ;SET TEST HEADER
1790 011036 004737 013500          JSR      PC,INIT1  ;GO INIT
1791 011042 012737 001700 000716  MOV      #1700,UDES   ;SET TO NRZ NORMAL
1792 011050 012737 021314 000616  MOV      #WDATA,BADDR ;SET BA=WRITE BUFFER
1793 011056 012737 177400 000620  MOV      #-400,FCNT   ;SET FC=400
1794 011064 012737 177600 000622  MOV      #-200,WCNT   ;SET WC=200
1795 011072 012737 000061 000710  MOV      #61,FUN     ;SET WRITE OP-CODE
1796 011100 004737 012310          JSR      PC,EXEC    ;GO DO WRITE
1797 011104 000240                NOP
1798 011106 004737 013500          JSR      PC,INIT1  ;INITIALIZE
1799 011112 012737 000021 000710  MOV      #21,FUN     ;SET READ-IN PRESET OP CODE
1800 011120 004737 012310          JSR      PC,EXEC    ;GO DO COMMAND
1801 011124 005000                CLR      R0
1802 011126 012703 000004          MOV      #4,R3      ;SET MULT
1803 011132 032777 020000 167362  FT23A: BIT      #20000,ADS ;SEE IF PIP RESET
1804 011140 001404                BEQ      FT23B     ;IF SO: BR
1805 011142 005300                DEC      R0
1806 011144 001372                BNE      FT23A     ;AWAIT PIP RESET
1807 011146 005303                DEC      R3
1808 011150 001370                BNE      FT23A     ;DELAY
1809 011152 032777 000002 167342  FT23B: BIT      #2,ADS  ;SEE IF BOT
1810 011160 001010                BNE      FT23C     ;IF SO: BR
1811 011162 012737 017354 000650  MOV      #MSG51,ERADD ;SET ERROR CODE
1812 011170 012737 011022 000674  MOV      #FT23,SCOLP
1813 011176 004737 004054          JSR      PC,FT3ER   ;GO DO ERROR
1814 011202 012701 141000          FT23C: MOV      #141000,R1 ;SET EXPT TC
1815 011206 013700 000542          MOV      TC,R0     ;SET TC ADDRESS
1816 011212 020110                CMP      R1,(R0)   ;SEE IF EXPT=RCVD
1817 011214 001410                BEQ      FT23X     ;IF SO: BR
1818 011216 012737 017410 000650  MOV      #MSG52,ERADD ;SET ERROR CODE
1819 011224 012737 011022 000674  MOV      #FT23,SCOLP ;CLEAR SCOPE ADDRESS
1820 011232 004737 003564          JSR      PC,FT2ER   ;GO DO ERROR
1821 011236 000137 003074          FT23X: JMP      TSCD2 ;RETURN TO SCHEDULAR
1822
1823

```

```
1825
1826
1827
1828 011242 012737 021034 000610 FT24: MOV #MSFT24,EMADDR :SET ERROR MSG HEADER
1829 011250 004737 012442 JSR PC,RWND :REWIND SLAVE
1830 011254 012737 000001 000720 MOV #1,PATRN :SELECT PATTERN
1831 011262 004737 013162 JSR PC,DSUP :GO DO DATA SETUP
1832 011266 012737 021314 000616 MOV #WDATA,BADDR :SET BUS ADDRESS,
1833 011274 012737 177400 000620 MOV #-400,FCNT :FRAME COUNT,
1834 011302 012737 177600 000622 MOV #-200,WCNT :WORD COUNT,
1835 011310 012737 001700 000716 MOV #1700,UDES :& SLAVE DESC = NRZ NORMAL
1836 011316 012737 000061 000710 MOV #61,FUN :LOAD OP CODE WRITE FWD
1837 011324 004737 012310 JSR PC,EXEC :GO EXECUTE COMMAND
1838 011330 012737 017172 000626 MOV #MSG46,ERRP :SET ERROR MSG ADDRESS
1839 011336 004737 012510 JSR PC,ERCHK :GO CHECK ERRORS
1840 011342 005737 000712 TST SERFL :BRANCH IF AN ERROR OCCURRED
1841 011346 001026 BNE FT24X
1842 011350 004737 012442 JSR PC,RWND :REWIND SLAVE
1843 011354 012737 023026 000616 MOV #RDATA,BADDR :SET BUS ADDRESS FOR READ
1844 011362 012737 002300 000716 MOV #2300,UDES :SET SLAVE DESC = PE,NORMAL
1845 011370 012737 000071 000710 MOV #71,FUN :SET OP CODE = READ FWD
1846 011376 004737 012310 JSR PC,EXEC :GO READ RECORD
1847 011402 032777 000040 167112 BIT #40,ADS :BRANCH ID PES BIT CLEARED
1848 011410 001405 BEQ FT24X
1849 011412 012737 017743 000650 MOV #MSG63,ERADD
1850 011420 004737 004054 JSR PC,FT3ER :GO PROCESS ERROR
1851 011424 004737 013330 FT24X: JSR PC,ITER
1852 011430 000137 003074 JMP TSCD2 :RETURN TO SCHEDULER
1853
```

```
1855  
1856  
1857 011434 012737 021112 000610 ;AUTO-DENSITY SELECT TEST: WRITE-PE,READ-NRZ  
1858 011442 004737 012442 FT25: MOV #MSFT25,EMADDR ;SET ERROR MESSAGE ADDRESS  
1859 011446 012737 000001 000720 JSR PC,RWIND ;REWIND SLAVE  
1860 011454 004737 013162 JSR #1,PATRN ;SELECT PATTERN  
1861 011460 012737 021314 000616 JSR PC,DSUP ;GO DO DATA SETUP  
1862 011466 012737 177400 000620 MOV #WDATA,BADDR ;SET BUS ADDRESS  
1863 011474 012737 177600 000622 MOV #-400,FCNT ;FRAME COUNT,  
1864 011502 012737 002300 000716 MOV #-200,WCNT ;WORD COUNT,  
1865 011510 012737 000061 000710 MOV #2300,UDES ;& SLAVE DESC = PE,NORMAL  
1866 011516 004737 012310 JSR #61,FUN ;LOAD WRITE OP CODE  
1867 011522 012737 017172 000626 JSR PC,EXEC ;GO EXECUTE WRITE  
1868 011530 004737 012510 JSR #MSG46,ERRP ;SET ERROR MSG HDR  
1869 011534 005737 000712 TST PC,ERCHK ;GO CHECK FOR ERRORS  
1870 011540 001026 BNE SERFL ;BRANCH IF ERROR OCURRED  
1871 011542 004737 012442 JSR FT25X ;REWIND SLAVE  
1872 011546 012737 023026 000616 MOV PC,RWIND ;SET BUS ADDRESS FOR READ  
1873 011554 012737 001700 000716 MOV #RDATA,BADDR ;SET SLAVE DESC = NRZ,NORMAL  
1874 011562 012737 000071 000710 MOV #1700,UDES ;SET READ FWD OP CODE  
1875 011570 004737 012310 JSR #71,FUN ;GO EXECUTE  
1876 011574 032777 000040 166720 BIT PC,EXEC ;BRANCH ID PES BIT GOT SET  
1877 011602 001005 BNE #40,ADS  
1878 011604 012737 017774 000650 MOV FT25X ;GO PROCESS ERROR  
1879 011612 004737 004054 JSR #MSG64,ERADD ;ITERATION LOOP  
1880 011616 004737 013330 FT25X: JSR PC,FT3ER ;RETURN TO SCHEDULER  
1881 011622 000137 003074 JSR PC,ITER  
1882 JMP TSCD2
```

;++B SEQUENTIAL TAPE MARK TEST

1885									
1886									
1887	011626	000240			FT26:	NOP			
1888	011630	012737	021170	000610		MOV	#MSFT26,EMADDR	:SET TEST ERROR MSG HEADER	
1889	011636	012737	001700	000716		MOV	#1700,UDES	:SET NRZ	
1890	011644	004737	012442		1\$:	JSR	PC,RWND	:REWIND SLAVE	
1891	011650	012737	000027	000710		MOV	#27,FUN	:SET WRITE TAPE MARK FUNCTION CODE	
1892	011656	004737	012310			JSR	PC,EXEC	:GO DO TAPE MARK	
1893	011662	005037	000660			CLR	STMSK	:CLEAR EXPECTED ERROR MASK	
1894	011666	012737	016137	000626		MOV	#MSG15,ERRP	:SET ERROR MESSAGE	
1895	011674	004737	012510			JSR	PC,ERCHK	:GO CHECK FOR ERRORS	
1896	011700	004737	013124			JSR	PC,TMCHK	:GO CHECK FOR TAPE MARK	
1897	011704	005737	000712			TST	SERFL	:EXIT TEST IF ERROR DETECTED	
1898	011710	001061				BNE	FT26X		
1899	011712	004737	012310			JSR	PC,EXEC	:WRITE SECOND TAPE MARK	
1900	011716	012737	016160	000626		MOV	#MSG15A,ERRP	:SET ERROR MESSAGE	
1901	011724	004737	012510			JSR	PC,ERCHK	:GO CHECK ERROR	
1902	011730	004737	013124			JSR	PC,TMCHK		
1903	011734	005737	000712			TST	SERFL	:EXIT TEST IF ERROR DETECTED	
1904	011740	001045				BNE	FT26X		
1905	011742	004737	012442			JSR	PC,RWND	:REWIND	
1906	011746	012737	000031	000710		MOV	#31,FUN	:SET SPACE FORWARD OP CODE	
1907	011754	012737	177777	000642		MOV	#-1,SCNT	:SET # OF RECORDS TO SPACE	
1908	011762	004737	012310			JSR	PC,EXEC	:GO SPACE FORWARD	
1909	011766	012737	017216	000626		MOV	#MSG47,ERRP	:SET SPACE FORWARD ERROR	
1910	011774	004737	012510			JSR	PC,ERCHK	:GO CHECK ERROR BITS	
1911	012000	004737	013124			JSR	PC,TMCHK	:GO CHECK IF TAPE MARK DETECTED	
1912	012004	005737	000712			TST	SERFL	:EXIT TEST IF ERROR DETECTED	
1913	012010	001021				BNE	FT26X		
1914	012012	004737	012310			JSR	PC,EXEC	:SPACE TO SECOND TAPE MARK	
1915	012016	004737	012510			JSR	PC,ERCHK	:GO CHECK ERROR BITS	
1916	012022	004737	013124			JSR	PC,TMCHK	:CHECK IF TAPE MARK DETECTED	
1917	012026	005737	000712			TST	SERFL	:EXIT TEST IF ERROR DETECTED	
1918	012032	001010				BNE	FT26X		
1919	012034	032737	002000	000716		BIT	#2000,UDES	:EXIT TEST IF PE COMPLETED	
1920	012042	001004				BNE	FT26X		
1921	012044	012737	002300	000716		MOV	#2300,UDES	:SET PE MODE	
1922	012052	000674				BR	1\$		
1923	012054	004737	013330		FT26X:	JSR	PC,ITER		
1924	012060	000137	003074			JMP	TSCD2		

```

1926                                     ;REWIND: OFF LINE TEST*****
1927
1928 012064 032777 010000 166456 FT27: BIT #10000,@SWR ;SEE IF IN CONTINUOUS MODE
1929 012072 001104 BNE FT27XX ;IF SO: BR
1930 012074 005737 001662 TST CHNFLG ;BRANCH IF CHAIN MODE
1931 012100 001101 BNE FT27XX
1932 012102 012737 021231 000610 MOV #MSFT27,EMADDR ;SET TEST HEADER
1933 012110 004737 012442 JSR PC,RWND ;REWIND & SELECT SLAVE
1934 012114 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN (ALL 1'S)
1935 012122 004737 013162 JSR PC,DSUP ;FILL WRITE BUFFER
1936 012126 012737 021314 000616 MOV #WDATA,BADDR ;SET WRITE BUFFER BUS ADDRESS
1937 012134 012737 177400 000620 MOV #-400,FCNT ;SET FRAME COUNT
1938 012142 012737 177600 000622 MOV #-200,WCNT ;SET WORD COUNT
1939 012150 012737 001700 000716 MOV #1700,UDES ;SET UNIT DESCRIPTION = NRZ
1940 012156 012737 000061 000710 MOV #61,FUN ;SET WRITE COMMAND
1941 012164 004737 012310 JSR PC,EXEC ;GO WRITE A RECORD
1942 012170 004737 013500 JSR PC,INIT1 ;++B CLEAR ANY ERROR BITS
1943 012174 012777 000003 166306 MOV #3,@C1 ;ISSUE REWIND: OFF LINE COMMAND
1944 012202 005037 000674 CLR SCOLP ;CLEAR SCOPE LOOP
1945 012206 012700 000042 MOV #42,R0
1946 012212 005001 1$: CLR R1 ;CLEAR TIMER
1947 012214 005301 2$: DEC R1
1948 012216 001376 BNE 2$ ;IF NOT TIMED OUT: BR
1949 012220 005300 DEC R0
1950 012222 001373 BNE 1$ ;IF NOT ALL TIMED OUT: BR
1951 012224 032777 010000 166270 BIT #10000,@DS ;SEE IF MOL IS RESET
1952 012232 001406 BEQ 3$ ;IF SO: BR
1953 012234 012737 017427 000650 MOV #MSG53,ERADD ;SET ERROR CODE
1954 012242 004737 004054 JSR PC,FT3ER ;GO DO ERROR
1955 012246 000412 BR FT27X
1956 012250 013700 000524 3$: MOV ER,R0 ;GET ADDRESS OF ERROR REG
1957 012254 005001 CLR R1 ;RESULT SHOULD BE 0
1958 012256 020110 CMP R1,(R0) ;BRANCH IF ERROR REG = 0
1959 012260 001405 BEQ FT27X
1960 012262 012737 020030 000650 MOV #MSG67,ERADD ;SET ERROR MSG HEADER
1961 012270 004737 003564 JSR PC,FT2ER ;GO TYPE ERROR
1962 012274 012704 017454 FT27X: MOV #MSG54,R4
1963 012300 004737 014322 JSR PC,TTOUT ;PRINT ON LINE REQUEST
1964 012304 000137 003074 FT27XX: JMP TSCD2 ;RETURN TO SCHEDULER
  
```

```
1966  
1967  
1968  
1969 012310 000240  
1970 012312 053777 000716 166222  
1971 012320 013777 000622 166164  
1972 012326 013777 000620 166162  
1973 012334 013777 000616 166152  
1974 012342 022737 000031 000710  
1975 012350 001404  
1976 012352 022737 000033 000710  
1977 012360 001003  
1978 012362 013777 000642 166126  
1979 012370 000240  
1980 012372 013777 000710 166110  
1981 012400 000240  
1982 012402 013703 000636  
1983 012406 005004  
1984 012410 032777 000200 166104  
1985 012416 001004  
1986 012420 005304  
1987 012422 001372  
1988 012424 005303  
1989 012426 001370  
1990 012430 013703 000640  
1991 012434 005303  
1992 012436 001376  
1993 012440 000207  
1994
```

;COMMAND EXECUTE SUBROUTINE*****

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

```
1996                                     ;REWIND SUBROUTINE*****
1997
1998 012442 004737 013500          RWND: JSR   PC,INIT1      ;INIT SLAVE
1999 012446 012777 000007 166034  MOV   #7,AC1      ;START REWIND
2000 012454 032777 000002 166040  1$:  BIT   #2,ADS     ;WAIT FOR BOT TO SET
2001 012462 001774                      BEQ   1$
2002 012464 032777 020000 166030  2$:  BIT   #20000,ADS ;WAIT FOR PIP TO CLEAR
2003 012472 001374                      BNE   2$
2004 012474 105777 166022          3$:  TSTB  ADS        ;++B WAIT FOR DRY
2005 012500 100375                      BPL   3$          ;++B
2006 012502 004737 013500          JSR   PC,INIT1    ;INIT
2007 012506 000207                      RTS   PC          ;RETURN TO CALLER
2008
```

```

2010                                     :ERROR CHECK SUBROUTINE*****
2011
2012 012510 005037 000712                ERCHK: CLR      SERFL      :CLEAR FLAG
2013 012514 017737 166002 000664        MOV      @DS,DSAV    :SAVE DRIVE STATUS REGISTER
2014 012522 032777 040000 165772        BIT      #40000,@DS :SEE IF ERROR
2015 012530 001001                       BNE      ERPT        :IF SO: BR
2016 012532 000207                       RTS      PC           :RETURN
2017 012534 017704 165764                ERPT:  MOV      @ER,R4  :GET ERROR REGISTER
2018 012540 032737 002000 000716        BIT      #2000,UDES  :SEE IF PE
2019 012546 001403                       BEQ      2$          :IF SO: BR
2020 012550 042737 000200 000660        BIC      #200,STMSK  :RESET PEF MASK
2021 012556 022737 000003 000742        2$:   CMP      #3,JUMPER :+TEST FOR NON-STANDARD JUMPER
2022 012564 001413                       BEQ      ERPTA1     :+BRANCH IF STANDARD
2023 012566 022777 011242 166112        CMP      #FT24,@LTADD :+CHECK FOR TEST 24
2024 012574 001404                       BEQ      1$          :+BRANCH IF TST24
2025 012576 022777 011434 166102        CMP      #FT25,@LTADD :+CHECK FOR TEST 25
2026 012604 001003                       BNE      ERPTA1
2027 012606 052737 020000 000660        1$:   BIS      #20000,STMSK :+SET OPI BIT IN ERROR MASK
2028 012614 043704 000660                ERPTA1: BIC      STMSK,R4 :MASK DONT CARE BITS
2029 012620 001536                       BEQ      ERPTX      :IF NO UNEXPECTED ERRORS: BR
2030 012622 012737 000001 000712        ERPTG: MOV      #1,SERFL :SET FLAG
2031 012630 032777 020000 165712        BIT      #20000,@SWR :SEE IF SHOULD PRINT ERRORS
2032 012636 001123                       BNE      ERPTD      :IF NOT: BR
2033 012640 005737 000606                TST      HDRFL      :SEE IF DONE HEADER
2034 012644 001006                       BNE      ERPTA      :IF SO: BR
2035 012646 005237 000606                INC      HDRFL      :SET HEADER FLAG
2036 012652 013704 000610                MOV      EMADDR,R4
2037 012656 004737 014322                JSR      PC,TTOUT    :PRINT HEADER
2038 012662 013704 000626                ERPTA: MOV      ERRP,R4  :GET ERROR CODE
2039 012666 001414                       BEQ      ERPTB      :IF NONE: BR
2040 012670 004737 014322                JSR      PC,TTOUT    :PRINT ERROR CODE
2041 012674 012704 016265                MOV      #MSG20,R4  :SET NRZ TAG
2042 012700 032777 002000 165634        BIT      #2000,@TC  :SEE IF PE
2043 012706 001402                       BEQ      ERPT1A     :IF NOT: BR
2044 012710 012704 016273                MOV      #MSG21,R4  :ELSE SET PE TAG
2045 012714 004737 014322                ERPT1A: JSR      PC,TTOUT :PRINT TAG
2046 012720 013704 000630                ERPTB: MOV      ERRP1,R4 :SEE IF CODE 2
2047 012724 001402                       BEQ      ERPTB1     :IF NOT: BR
2048 012726 004737 014322                JSR      PC,TTOUT    :PRINT CODE 2
2049 012732 032777 004000 165610        ERPTB1: BIT      #4000,@SWR :SEE IF ITERATION
2050 012740 001010                       BNE      ERPTC      :IF NOT: BR
2051 012742 012704 017620                MOV      #MSG56,R4
2052 012746 004737 014322                JSR      PC,TTOUT    :PRINT ITER TAG
2053 012752 013703 000662                MOV      ITCNT,R3
2054 012756 004737 014452                JSR      PC,OCTP     :PRINT ITERATION
2055 012762 012704 015244                ERPTC: MOV      #MSG1,R4
2056 012766 004737 014322                JSR      PC,TTOUT    :PRINT REGISTER TAG
2057 012772 017703 165512                MOV      @C1,R3
2058 012776 004737 014440                JSR      PC,OCTPE    :PRINT CS1
2059 013002 017703 165504                MOV      @WC,R3
2060 013006 004737 014440                JSR      PC,OCTPE    :PRINT WC
2061 013012 017703 165476                MOV      @BA,R3
2062 013016 004737 014440                JSR      PC,OCTPE    :PRINT BA
2063 013022 017703 165470                MOV      @FC,R3
2064 013026 004737 014440                JSR      PC,OCTPE    :PRINT FC
2065 013032 017703 165462                MOV      @CS,R3

```

2066	013036	004737	014440		JSR	PC,OCTPE		:PRINT CS2
2067	013042	005737	007206		TST	TAG		:++C CHECK FOR SPECIAL DS
2068	013046	001403			BEQ	1\$		
2069	013050	013703	000652		MOV	TEMP1,R3		:++C PRINT DS READ INTO TEMP1 AT CRITICAL TIME
2070	013054	000402			BR	2\$		
2071	013056	017703	165440	1\$:	MOV	@DS,R3		
2072	013062	004737	014440	2\$:	JSR	PC,OCTPE		:PRINT DS
2073	013066	017703	165432		MOV	@ER,R3		
2074	013072	004737	014440		JSR	PC,OCTPE		:PRINT ER
2075	013076	017703	165440		MOV	@TC,R3		
2076	013102	004737	014440		JSR	PC,OCTPE		:PRINT TC
2077	013106	005777	165436	ERPTD:	TST	@SWR		:SEE IF HALT ON ERROR
2078	013112	100001			BPL	ERPTX		:IF NOT: BR
2079	013114	000000			HALT			
2080	013116	004737	013500	ERPTX:	JSR	PC,INIT1		:INIT
2081	013122	000207		ERPTXX:	RTS	PC		:RETURN
2082								
2083								

```
2085                                     ;TAPE MARK STATUS CHECK*****
2086
2087 013124 032737 000004 000664 TMCHK: BIT #4, DSAV ;BRANCH IF TM SET
2088 013132 001012 BNE 1$
2089 013134 005737 000712 TST SERFL ;SEE IF HAD ERROR
2090 013140 001007 BNE 1$ ;IF SO: BR
2091 013142 012737 017630 000630 MOV #MSG57, ERRP1 ;SET ERROR CODE 2
2092 013150 004737 012622 JSR PC, ERPTG ;GO PRINT TM ERROR
2093 013154 005037 000630 CLR ERRP1 ;CLEAR CODE 2 FLAG
2094 013160 000207 1$: RTS PC ;RETURN
2095
2096                                     ;DATA SETUP ROUTINE*****
2097
2098 013162 000240 DSUP: NOP
2099 013164 012703 DSO: MOV #WDATA, R3 ;R3 = ADDRS OF WRITE BUFFER
2100 013170 013701 MOV PATRN, R1 ;R1 = PATTERN SELECTOR
2101 013174 006301 ASL R1 ;MAKE PATTERN SELECTOR EVEN
2102 013176 004771 JSR PC, @DATBL(R1) ;GO GENERATE PATTERN
2103 013202 012702 MOV #640, R2 ;R2=BUFFER SIZE +2
2104 013206 012701 MOV #RDATA, R1 ;R1=READ DATA START
2105 013212 005021 1$: CLR (R1)+ ;CLEAR BUFFER
2106 013214 005302 DEC R2 ;SEE IF DONE ALL
2107 013216 001375 BNE 1$ ;IF NOT: BR
2108 013220 000207 RTS PC ;EXIT
2109
2110                                     ;ALL ONES*****
2111
2112 013222 012701 DAT1: MOV #-1, R1 ;R1=DATA
2113 013226 012702 DAT1A: MOV #640, R2 ;R2=WORD COUNT +2
2114 013232 010123 1$: MOV R1, (R3)+ ;LOAD BUFFER
2115 013234 005302 DEC R2 ;SEE IF DONE
2116 013236 001375 BNE 1$ ;IF NOT: BR
2117 013240 000207 RTS PC
2118
2119                                     ;ALL ZEROS*****
2120
2121 013242 005001 DAT2: CLR R1 ;R1=DATA
2122 013244 000770 BR CAT1A ;LOAD BUFFER
2123
2124                                     ;ONE/ZERO IN ALTERNATING CHARACTERS*****
2125
2126 013246 012701 DAT3: MOV #125125, R1 ;R1=DATA
2127 013252 000765 BR DAT1A ;LOAD BUFFER
2128
2129                                     ;ALL BITS 0-377*****
2130
2131 013254 005001 DAT4: CLR R1 ;R1=STARTING DATA
2132 013256 012702 001500 MOV #1500, R2 ;R2=CHARACTER COUNT
2133 013262 110123 1$: MOV R1, (R3)+ ;LOAD BUFFER
2134 013264 105201 INCB R1 ;BUMP DATA
2135 013266 005302 DEC R2 ;SEE IF DONE
2136 013270 001374 BNE 1$ ;IF NOT: BR
2137 013272 000207 RTS PC
2138
```

2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195

:SCOPE LOOP ON ERROR SUBROUTINE*****

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

:TEST ITERATION SUBROUTINE*****

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

:NON-STANDARD JUMPER HANDLER SUBROUTINE*****

```
NOST:   MOV     R0,-(SP)           ;+SAVE R0
        MOV     #120,R0          ;+SET UP INDEX
        MOV     #FT26,TSTTBL(R0);+ADJUST SCHEDULAR TEST TABLE
        TST     (R0)+
        MOV     #FT26,TSTTBL(R0) ;+OVERLAY TEST LIST
        TST     (R0)+
        MOV     #FT27,TSTTBL(R0)
        TST     (R0)+
        MOV     #TEND,TSTTBL(R0)
        TST     (R0)+
        MOV     #27,TSTTBL(R0)
        MOV     #27,TLAST
        MOV     (SP)+,R0         ;RESTO R0
        RTS     PC
```

:INITIALIZE SUBROUTINE*****

```
2196 013500 000240          INIT1:  NOP
2197 013502 012777 000040 165010  MOV      #40,@CS      :INIT
2198 013510 013777 000612 165002  INIT2:  MOV      DRVN,@CS  :SELECT DRIVE
2199 013516 013777 000614 165016  MOV      SLVN,@TC     :SELECT SLAVE
2200 013524 000207          RTS      PC           :RETURN
2201
2202                          :MAG TAPE INTERRUPT HANDLER*****
2203
2204 013526 000240          MTINT:  NOP
2205 013530 013716 000646  MOV      RTRN,(SP)    :RETURN TO (RTRN)
2206 013534 000002          RTI
2207
```

```

2209
2210 ;TTY INTERRUPT HANDLER*****
2211
2212 013536 017746 165012 TTINT: MOV @TKB,-(SP) ;GET CHARACTER
2213 013542 042716 000200 BIC #200,(SP) ;CLEAR PARITY BIT
2214 013546 122716 000003 CMPB #3,(SP) ;BRANCH IF NOT CONTROL C
2215 013552 001010 BNE 1$
2216 013554 005737 001662 TST CHNFLG ;INHIBIT ^C IF CHAIN MODE
2217 013560 001005 BNE 1$
2218 013562 005077 164760 CLR @PSW
2219 013566 000005 RESET
2220 013570 000137 000200 JMP @#200 ;RESTART PROGRAM
2221 013574 122716 000001 1$: CMPB #1,(SP) ;BRANCH IF NOT ^A
2222 013600 001017 BNE 2$
2223 013602 022737 000176 000550 CMP #SWREG,SWR ;BRANCH IF HARDWARE SWR IS INVOKED
2224 013610 001016 BNE 3$
2225 013612 012737 177570 000550 MOV #177570,SWR ;INVOKE HARDWARE SWR
2226 013620 004737 015200 JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
2227 013624 012704 020104 MOV #MSG70,R4 ;TYPE 'HARDWARE SWR IN USE'
2228 013630 004737 014322 JSR PC,TTOUT
2229 013634 004737 015222 JSR PC,RESTORE
2230 013640 122716 000007 2$: CMPB #7,(SP) ;BRANCH IF NOT ^G
2231 013644 001005 BNE 4$
2232 013646 012737 000176 000550 3$: MOV #SWREG,SWR ;INVOKE SOFTWARE SWR
2233 013654 004737 015102 JSR PC,GTSWR ;GET SOFTWARE SWITCHES
2234 013660 005726 4$: TST (SP)+ ;POP CHARACTER OFF THE STACK
2235 013662 000002 RTI

```

```

2236
2237 ;BUS ADDRESS TRAP HANDLER*****
2238
2239 013664 000240 TRAP: NOP
2240 013666 032777 020000 164654 BIT #20000,@SWR ;SEE IF SHOULD PRINT ERRORS
2241 013674 001020 BNE TRAP2 ;IF NOT: BR
2242 013676 005737 000606 TST HDRFL ;SEE IF DONE HEADER
2243 013702 001006 BNE TRAP1 ;IF SO: BR
2244 013704 005237 000606 INC HDRFL ;ELSE SET HEADER FLAG
2245 013710 013704 000610 MOV EMADDR,R4
2246 013714 004737 014322 JSR PC,TTOUT ;PRINT HEADER
2247 013720 012704 016320 TRAP1: MOV #MSG24,R4
2248 013724 004737 014322 JSR PC,TTOUT ;PRINT ERROR
2249 013730 010103 MOV R1,R3 ;GET ADDRESS THAT CAUSED THE TRAP
2250 013732 004737 014452 JSR PC,OCTP ;PRINT ADDRESS OF TRAP
2251 013736 005777 164606 TRAP2: TST @SWR ;SEE IF HALT ON ERROR
2252 013742 100001 BPL TRAPX ;IF NOT: BR
2253 013744 000000 HALT
2254 013746 022626 TRAPX: CMP (SP)+,(SP)+ ;RESET STACK
2255 013750 012737 003322 000674 MOV #FT1A,SCOLP ;SET SCOPE ADDRESS
2256 013756 004737 013274 JSR PC,SCOPE ;GO SEE IF SCOPE LOOP
2257 013762 005737 000722 TST RHTF ;SEE IF INITIAL ADDRESS TEST
2258 013766 001402 BEQ TRAPXX ;IF NOT: BR
2259 013770 000137 001764 JMP STOB ;ELSE REDO ADDRESS REQUEST
2260 013774 000137 003326 TRAPXX: JMP FT1B ;RETURN TO TEST 1
2261

```

2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318

014000 010146
014002 011601
014004 005037 000652
014010 005000
014012 004737 014260
014016 122737 000003 000602
014024 001003
014026 000005
014030 000137 000200
014034 122737 000015 000602 11\$:
014042 001004
014044 005737 000652
014050 001471
014052 000457
014054 122737 000025 000602 2\$:
014062 001005
014064 012704 020024
014070 004737 014322
014074 000742
014076 122737 000177 000602 21\$:
014104 001012
014106 000241
014110 006000
014112 006200
014114 006200
014116 012704 020026
014122 004737 014322
014126 005201
014130 000730
014132 122737 000060 000602 3\$:
014140 101402
014142 000137 014240
014146 122737 000070 000602 4\$:
014154 101002
014156 000137 014240
014162 005237 000652 5\$:
014166 006300
014170 006300
014172 006300

```
*****  
: TTY ENTRY SUBROUTINE:  
: THIS SUBROUTINE IS USED BY THE TEST CONDITION  
: ENTRY ROUTINE TO READ THE RESPONSE ENTERED  
: AT THE TTY AND CHECK THEM FOR LEGALITY AND  
: LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL  
: (0-7) AND MUST FALL WITHIN THE LIMITS SET BY  
: THE CALLING ROUTINE.  
: IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,  
: A QUESTION MARK IS TYPED (?) AND THE RESPONSE  
: MAY BE REENTERED.  
: ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND  
: MAY BE TERMINATED AT LESS THAN SIX BY TYPING A  
: CARRIAGE RETURN  
*****  
TTR: MOV R1, -(SP) ;SAVE CHAR COUNT ON STACK  
10$: MOV (SP), R1 ;RESTORE CHAR COUNT (FOR ^U)  
CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG  
CLR R0  
1$: JSR PC, TTIN ;GO READ CHARACTER  
CMPB #3, TIB ;BRANCH IF NOT ^C  
BNE 11$  
RESET ;RESET  
JMP @#200 ;RESTART  
CMPB #15, TIB ;SEE IF CR  
BNE 2$ ;IF NOT: BR  
TST TEMP1 ;SEE IF FIRST CHARACTER  
BEQ 9$ ;IF SO: BR  
BR 6$ ;ELSE GO LOAD VALUE  
CMPB #25, TIB ;BRANCH IF NOT CONTROL U  
BNE 21$  
MOV #MSG65, R4 ;TYPE <CR><LF>  
JSR PC, TTOUT  
BR 10$ ;RESTART  
CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'  
BNE 3$  
CLC ;REMOVE LAST CHARACTER  
ROR R0  
ASR R0  
ASR R0  
MOV #MSG66, R4 ;TYPE '\'  
JSR PC, TTOUT  
INC R1 ;DECREMENT CHAR RECEIVED COUNT  
BR 1$ ;GET NEXT CHARACTER  
CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0  
BLOS 4$ ;IF NOT: BR  
JMP TNER ;ELSE GO TO ERROR  
CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7  
BHI 5$ ;IF NOT: BR  
JMP TNER ;ELSE GO TO ERROR  
INC TEMP1 ;SET FIRST CHARACTER FLAG  
ASL R0  
ASL R0 ;SHIFT 3 LEFT  
ASL R0
```

```
2319 014174 042737 177770 000602      BIC      #177770,TIB      :STRIP ASCII
2320 014202 053700 000602      BIS      TIB,R0          :LOAD CHARACTER
2321 014206 005301          DEC      R1              :SEE IF DONE
2322 014210 001300          BNE      1$              :IF NOT: BR
2323 014212 020002      6$:      CMP      R0,R2        :SEE IF EXCEEDED MAXIMUM LIMIT
2324 014214 101402      BLOS     7$              :IF NOT: BR
2325 014216 000137 014240      JMP      TINNER          :ELSE GO TO ERROR
2326 014222 020300      7$:      CMP      R3,R0        :SEE IF BELOW MINIMUM LIMIT
2327 014224 101402      BLOS     8$              :IF NOT: BR
2328 014226 000137 014240      JMP      TINNER          :ELSE GO TO ERROR
2329 014232 010015      8$:      MOV      R0,(R5)       :LOAD VALUE
2330 014234 005726      9$:      TST      (SP)+         :POP CHAR COUNT OFF STACK
2331 014236 000207      RTS      PC              :EXIT
2332
2333      :TTY ENTRY ERROR SUBROUTINE*****
2334
2335 014240 012704 015621      TINNER:  MOV      #MSG7,R4
2336 014244 004737 014322      JSR      PC,TTOUT        :PRINT?
2337 014250 005726          TST      (SP)+         :POP CHAR COUNT OFF STACK
2338 014252 162716 000020      SUB      #20,(SP)       :RESET SP TO START OF VALUE ROUTINE
2339 014256 000207      RTS      PC              :REDO VALUE ENTRY
2340
2341      :TTY READ SUBROUTINE*****
2342
2343 014260 005277 164266      TTIN:    INC      @TKS
2344 014264 105777 164262      1$:      TSTB     @TKS
2345 014270 100375          BPL      1$
2346 014272 117737 164256 000602      MOV      @TKB,TIB
2347 014300 042737 000200 000602      BIC      #200,TIB        :STRIP PARITY BIT
2348 014306 013737 000602 000600      MOV      TIB,TOB        :MOVE CHAR TO OUTPUT BFR
2349 014314 004737 014422      JSR      PC,TOG          :AND TYPE IT
2350 014320 000207      RTS      PC
2351
2352      :TTY OUTPUT SUBROUTINE*****
2353
2354 014322 112437 000600      TTOUT:   MOV      (R4)+,TOB
2355 014326 122737 000043 000600      CMP      #43,TOB
2356 014334 001440          BEQ      TEX
2357 014336 122737 000045 000600      CMP      #45,TOB
2358 014344 001403          BEQ      1$
2359 014346 004737 014422      JSR      PC,TOG
2360 014352 000763          BR       TTOUT
2361 014354 112737 000015 000600      1$:      MOV      #15,TOB
2362 014362 004737 014422      JSR      PC,TOG
2363 014366 012703 000004          MOV      #4,R3
2364 014372 005037 000600      2$:      CLR      TOB
2365 014376 004737 014422      JSR      PC,TOG
2366 014402 005303          DEC      R3
2367 014404 001372          BNE      2$              :DO FILLERS
2368 014406 112737 000012 000600      MOV      #12,TOB
2369 014414 004737 014422      JSR      PC,TOG
2370 014420 000740          BR       TTOUT
2371 014422 105777 164130      TOG:    TSTB     @TPS
2372 014426 00375          BPL      TOG
2373 014430 113777 000600 164122      MOV      TOB,@TPB
2374 014436 000207      RTS      PC
```

```

2376                                     :OCIAL OUTPUT SUBROUTINE*****
2377
2378 014440 012737 000001 014670 OCTPE: MOV #1,OFL
2379 014446 010304          MOV R3,R4
2380 014450 000410          BR OCTP0
2381 014452 005037 014670 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
2382 014456 010304          OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
2383 014460 001004          BNE OCTP0 ;IF NOT ZERO: BR
2384 014462 004737 014650 JSR PC,OCTPG1 ;ELSE PRINT ZERO
2385 014466 000137 014612 JMP OCTP3 ;SPACE AND EXIT
2386 014472 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
2387 014476 001406          BEQ OCTP1 ;IF NOT: BR
2388 014500 012704 000001 MOV #1,R4
2389 014504 004737 014626 JSR PC,OCTPG ;PRINT 1
2390 014510 000137 014522 JMP OCTP2
2391 014514 005004          OCTP1: CLR R4
2392 014516 004737 014626 JSR PC,OCTPG ;PRINT 0
2393 014522 010304          OCTP2: MOV R3,R4
2394 014524 006004          ROR R4
2395 014526 006004          ROR R4 ;POSITION DIGIT
2396 014530 006004          ROR R4
2397 014532 006004          ROR R4
2398 014534 000304          SWAB R4
2399 014536 004737 014626 JSR PC,OCTPG ;PRINT DIGIT 2
2400 014542 010304          MOV R3,R4
2401 014544 006004          ROR R4
2402 014546 000304          SWAB R4
2403 014550 004737 014626 JSR PC,OCTPG ;PRINT DIGIT 3
2404 014554 010304          MOV R3,R4
2405 014556 006104          ROL R4
2406 014560 006104          ROL R4
2407 014562 000304          SWAB R4
2408 014564 004737 014626 JSR PC,OCTPG ;PRINT DIGIT 4
2409 014570 010304          MOV R3,R4
2410 014572 006004          ROR R4
2411 014574 006004          ROR R4
2412 014576 006004          ROR R4
2413 014600 004737 014626 JSR PC,OCTPG
2414 014604 010304          MOV R3,R4
2415 014606 004737 014626 JSR PC,OCTPG ;PRINT DIGIT 5
2416 014612 012737 000240 000600 OCTP3: MOV #240,TOB
2417 014620 004737 014422 JSR PC,TOG ;PRINT SPACE
2418 014624 000207          RTS PC ;EXIT
2419 014626 042704 177770 OCTPG: BIC #177770,R4
2420 014632 001004          BNE OCTPG0
2421 014634 005737 014670 TST OFL
2422 014640 001001          BNE OCTPG0
2423 014642 000207          RTS PC
2424
2425 014644 005237 014670 OCTPG0: INC OFL
2426 014650 052704 000260 OCTPG1: BIS #260,R4
2427 014654 010437 000600 MOV R4,TOB
2428 014660 004737 014422 JSR PC,TOG
2429 014664 010304          MOV R3,R4
2430 014666 000207          RTS PC
2431 014670 000000          OFL: 0 ;FIRST CHAR FLAG
  
```

```
2432
2433
2434
2435 014672 005037 000600 DOUT: CLR TOB
2436 014676 012704 000010 MOV #10,R4 ;SET NUMBER TO PRINT
2437 014702 110337 000600 MOVB R3,TOB
2438 014706 105777 163644 1$: TSTB @TPS
2439 014712 100375 BPL 1$
2440 014714 132737 000200 000600 BITB #200,TOB
2441 014722 001404 BEQ 2$
2442 014724 012777 000061 163626 MOV #061,@TPB
2443 014732 000403 BR 3$
2444 014734 012777 000060 163616 2$: MOV #060,@TPB
2445 014742 006137 000600 3$: ROL TOB
2446 014746 005304 DEC R4
2447 014750 001356 BNE 1$
2448 014752 000207 RTS PC
2449
2450 014754 013703 000656 DOUTD: MOV TEMP3,R3
2451 014760 000303 SWAB R3
2452 014762 004737 014672 JSR PC,DOUT
2453 014766 013703 000656 MOV TEMP3,R3
2454 014772 004737 014672 JSR PC,DOUT
2455 014776 000207 RTS PC
2456
2457 ;SERIAL NUMBER PRINT SUBROUTINE*****
2458
2459 015000 010304 SNPT: MOV R3,R4
2460 015002 000304 SWAB R4
2461 015004 006004 ROR R4
2462 015006 006004 ROR R4
2463 015010 006004 ROR R4
2464 015012 006004 ROR R4 ;GET FIRST DIGIT
2465 015014 004737 015056 JSR PC,SNPG ;GO PRINT
2466 015020 010304 MOV R3,R4
2467 015022 000304 SWAB R4 ;GET SECOND DIGIT
2468 015024 004737 015056 JSR PC,SNPG ;GO PRINT
2469 015030 010304 MOV R3,R4
2470 015032 006004 ROR R4
2471 015034 006004 ROR R4
2472 015036 006004 ROR R4
2473 015040 006004 ROR R4 ;GET THIRD DIGIT
2474 015042 004737 015056 JSR PC,SNPG ;GO PRINT
2475 015046 010304 MOV R3,R4 ;GET FOURTH DIGIT
2476 015050 004737 015056 JSR PC,SNPG ;GO PRINT
2477 015054 000207 RTS PC ;EXIT
2478 015056 012737 000260 000600 SNPG: MOV #260,TOB ;SET BASE = 0
2479 015064 042704 177760 BIC #177760,R4 ;MASK DIGIT
2480 015070 050437 000600 BIS R4,TOB ;SET ASCII
2481 015074 004737 014422 JSR PC,TOG ;TYPE DIGIT
2482 015100 000207 RTS PC ;RETURN
2483
```

```

2485
2486
2487 015102 022737 000176 000550 :ROUTINE TO LOAD NEW VALUE INTO SWITCHES
2488 015110 001032 GTSWR: CMP #SWREG,SWR :BRANCH IF SOFTWARE SWR
2489 015112 004737 015200 BNE 1$ :NOT INVOKED
2490 015116 012704 021271 JSR PC,.SAVE :SAVE REGISTERS ON THE STACK
2491 015122 004737 014322 MOV #SMSWR,R4
2492 015126 017703 163416 JSR PC,TTOUT
2493 015132 004737 014440 MOV @SWR,R3
2494 015136 012704 021300 JSR PC,OCPE
2495 015142 004737 014322 MOV #SMNEW,R4
2496 015146 013705 000550 JSR PC,TTOUT
2497 015152 012701 000007 MOV SWR,R5 :TTR ROUTINE RETURNS NEW VALUE TO (R5)
2498 015156 012702 177777 MOV #7,R1 :LIMIT RESPONSE TO 7 CHARS
2499 015162 012703 000000 MOV #177777,R2 :BETWEEN 0 AND 177777
2500 015166 004737 014000 JSR PC,TTR
2501 015172 004737 015222 JSR PC,.RESTORE :RESTORE REGISTERS
2502 015176 000207 1$: RTS PC
2503
2504 :ROUTINE TO SAVE REGISTERS ON THE STACK
(1) 015200 010546 .SAVE: MOV %5,-(SP) ;;R5 IS SAVED AT 12(SP)
(1) 015202 010446 MOV %4,-(SP) ;;R4 IS SAVED AT 10(SP)
(1) 015204 010346 MOV %3,-(SP) ;;R3 IS SAVED AT 6(SP)
(1) 015206 010246 MOV %2,-(SP) ;;R2 IS SAVED AT 4(SP)
(1) 015210 010146 MOV %1,-(SP) ;;R1 IS SAVED AT 2(SP)
(1) 015212 010046 MOV %0,-(SP) ;;R0 IS SAVED AT (SP)
(1) 015214 016646 000014 MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
(1) 015220 000207 RTS PC ;;RETURN TO CALLER
2505 :ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
(1) 015222 012666 000014 .RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
(1) 015226 012600 MOV (SP)+,%0
(1) 015230 012601 MOV (SP)+,%1
(1) 015232 012602 MOV (SP)+,%2
(1) 015234 012603 MOV (SP)+,%3
(1) 015236 012604 MOV (SP)+,%4
(1) 015240 012605 MOV (SP)+,%5
(1) 015242 000207 RTS PC ;;RETURN
2506
  
```

2508

:MESSAGE TABLE*****

2509

2510	015244	041445	030523	020040	MSG1:	.ASCII	/%CS1	WC	BA	FC	CS2	/
	015252	020040	041527	020040								
	015260	020040	041040	020101								
	015266	020040	020040	041506								
	015274	020040	020040	041440								
	015302	031123	020040	020040								
2511	015310	051504	020040	020040		.ASCII	/DS	ER	TC%#			
	015316	042440	020122	020040								
	015324	020040	041524	021445								
2512	015332	051045	053505	047111	MSG2:	.ASCII	/%REWIND ERROR-BOT	NOT SET	WHEN PIP	CLEARED#		
	015340	020104	051105	047522								
	015346	026522	047502	020124								
	015354	047516	020124	042523								
	015362	020124	044127	047105								
	015370	050040	050111	041440								
	015376	042514	051101	042105								
	015404	043										
2513	015405	045	052045	030115	MSG3:	.ASCII	'%XTM03-TE16/TU77	BASIC FUNCTION	TEST (CZTECDO)%'	;++B		
	015412	026463	042524	033061								
	015420	052057	033525	020067								
	015426	040502	044523	020103								
	015434	052506	041516	044524								
	015442	047117	052040	051505								
	015450	020124	041450	052132								
	015456	041505	030104	022451								
2514	015464	054524	042520	036040		.ASCII	/TYPE <CR>	TO TERMINATE	RESPONSE &	^C TO RESTART%#		
	015472	051103	020076	047524								
	015500	052040	051105	044515								
	015506	040516	042524	051040								
	015514	051505	047520	051516								
	015522	020105	020046	041536								
	015530	052040	020117	042522								
	015536	052123	051101	022524								
	015544	043										
2515	015545	045	042522	044507	MSG4:	.ASCII	/%REGISTER	START = #				
	015552	052123	051105	051440								
	015560	040524	052122	036440								
	015566	021440										
2516	015570	053045	041505	047524	MSG5:	.ASCII	/%VECTOR = #					
	015576	020122	020075	043								
2517	015603	045	047105	020104	MSG6:	.ASCII	/%END OF	PASS #				
	015610	043117	050040	051501								
	015616	020123	043									
2518	015621	040	020077	043	MSG7:	.ASCII	/ ? #					
2519	015625	045	047520	044523	MSG9:	.ASCII	/%POSITION	ERROR: #				
	015632	044524	047117	042440								
	015640	051122	051117	020072								
	015646	043										
2520	015647	045	051511	041440	MSG10A:	.ASCII	/%IS CONT..	OLLER JUMPERED	IN NON-STANDARD	MODE/<15><12>		
	015654	047117	051124	046117								
	015662	042514	020122	052512								
	015670	050115	051105	042105								
	015676	044440	020116	047516								
	015704	026516	052123	047101								

2521	015712	040504	042122	046440			
	015720	042117	006505	012			
	015725	124	050131	020105	.ASCII	/TYPE 2 FOR NON-STANDARD OR CR FOR STANDARD:	/
	015732	020062	047506	020122			
	015740	047516	026516	052123			
	015746	047101	040504	042122			
	015754	047440	020122	051103			
	015762	043040	051117	051440			
	015770	040524	042116	051101			
	015776	035104	020040	020040			
2522	016004	020040	043				
	016007	045	051104	053111	MSG10:	.ASCII	/%DRIVE NUMBER: #/
	016014	020105	052516	041115			
2523	016022	051105	020072	043	MSG11:	.ASCII	/%SLAVE NUMBER: #/
	016027	045	046123	053101			
	016034	020105	052516	041115			
2524	016042	051105	020072	043	MSG12:	.ASCII	/%WRITE ERROR #/
	016047	045	051127	052111			
	016054	020105	051105	047522			
2525	016062	020122	043				
	016065	045	042522	042101	MSG13:	.ASCII	/%READ REVERSE ERROR #/
	016072	051040	053105	051105			
	016100	042523	042440	051122			
2526	016106	051117	021440		MSG14:	.ASCII	/%READ FORWARD ERROR #/
	016112	051045	040505	020104			
	016120	047506	053522	051101			
	016126	020104	051105	047522			
2527	016134	020122	043				
	016137	045	051127	052111	MSG15:	.ASCII	/%WRITE TM ERROR #/
	016144	020105	046524	042440			
2528	016152	051122	051117	021440	MSG15A:	.ASCII	/% WRITE TM ERROR ON SECOND TM #/
	016160	020045	051127	052111			
	016166	020105	046524	042440			
	016174	051122	051117	047440			
	016202	020116	042523	047503			
	016210	042116	052040	020115			
2529	016216	043					
	016217	045	041520	020040	MSG15B:	.ASCII	/%PC #/
	016224	043					
2530	016225	045	042522	042526	MSG16:	.ASCII	/%REVERSE ERROR #/
	016232	051522	020105	051105			
	016240	047522	020122	043			
2531	016245	045	047506	053522	MSG17:	.ASCII	/%FORWARD ERROR #/
	016252	051101	020104	051105			
	016260	047522	020122	043			
2532	016265	040	051116	020132	MSG20:	.ASCII	/ NRZ #/
	016272	043					
2533	016273	040	042520	021440	MSG21:	.ASCII	/ PE #/
2534	016300	042440	050130	035124	MSG22:	.ASCII	/ EXPT: #/
	016306	021440					
2535	016310	051040	053103	035104	MSG23:	.ASCII	/ RCVD: #/
	016316	021440					
2536	016320	041045	051525	052040	MSG24:	.ASCII	/%BUS TRAP: #/
	016326	040522	035120	021440			
2537	016334	053445	035103	021440	MSG25:	.ASCII	/%WC: #/
2538	016342	041045	035101	021440	MSG26:	.ASCII	/%BA: #/

2539	016350	042045	035102	021440	MSG27:	.ASCII	/XDB: #/
2540	016356	044445	044516	020124	MSG28:	.ASCII	/XINIT DID NOT CLEAR RH #/
	016364	044504	020104	047516			
	016372	020124	046103	040505			
	016400	020122	044122	021440			
2541	016406	051445	020103	047516	MSG29:	.ASCII	/XSC NOT RESET BY INIT #/
	016414	020124	042522	042523			
	016422	020124	054502	044440			
	016430	044516	020124	043			
2542	016435	045	051124	020105	MSG30:	.ASCII	/XTRE NOT RESET BY INIT #/
	016442	047516	020124	042522			
	016450	042523	020124	054502			
	016456	044440	044516	020124			
	016464	043					
2543	016465	045	051503	020062	MSG31:	.ASCII	/XCS2 NOT RESET BY INIT #/
	016472	047516	020124	042522			
	016500	042523	020124	054502			
	016506	044440	044516	020124			
	016514	043					
2544	016515	045	046104	020124	MSG32:	.ASCII	/XDLT NOT SET #/
	016522	047516	020124	042523			
	016530	020124	043				
2545	016533	045	041523	047040	MSG33:	.ASCII	/XSC NOT SET #/
	016540	052117	051440	052105			
	016546	021440					
2546	016550	052045	042522	047040	MSG34:	.ASCII	/XTRE NOT SET #/
	016556	052117	051440	052105			
	016564	021440					
2547	016566	044445	020122	047516	MSG35:	.ASCII	/XIR NOT SET BY INIT #/
	016574	020124	042523	020124			
	016602	054502	044440	044516			
	016610	020124	043				
2548	016613	045	051117	047040	MSG36:	.ASCII	/XOR NOT RESET BY INIT #/
	016620	052117	051040	051505			
	016626	052105	041040	020131			
	016634	047111	052111	021440			
2549	016642	047445	020122	047516	MSG37:	.ASCII	/XOR NOT RESET BY 1 SILO ENTRY #/
	016650	020124	042522	042523			
	016656	020124	054502	030440			
	016664	051440	046111	020117			
	016672	047105	051124	020131			
	016700	043					
2550	016701	045	051117	047040	MSG38:	.ASCII	/XOR NOT SET BY SILO FULL #/
	016706	052117	051440	052105			
	016714	041040	020131	044523			
	016722	047514	043040	046125			
	016730	020114	043				
2551	016733	045	040502	020104	MSG39:	.ASCII	/XBAD SILO READ #/
	016740	044523	047514	051040			
	016746	040505	020104	043			
2552	016753	045	051111	047040	MSG40:	.ASCII	/XIR NOT RESET BY SILO FULL#/
	016760	052117	051040	051505			
	016766	052105	041040	020131			
	016774	044523	047514	043040			
	017002	046125	021514				
2553	017006	047045	047117	042455	MSG41:	.ASCII	/XNON-EXIST DRIVE#/

	017014	044530	052123	042040			
	017022	044522	042526	043			
2554	017027	045	047516	026516	MSG42:	.ASCII	/%NON-EXIST SLAVEN#
	017034	054105	051511	020124			
	017042	046123	053101	021505			
2555	017050	051445	051105	040511	MSG43:	.ASCII	/%SERIAL NO: #/
	017056	020114	047516	020072			
	017064	043					
2556	017065	045	051105	051501	MSG44:	.ASCII	/%ERASE HEAD INOPERATIVE#
	017072	020105	042510	042101			
	017100	044440	047516	042520			
	017106	040522	044524	042526			
	017114	043					
2557	017115	045	047520	051523	MSG45:	.ASCII	/%POSSIBLE ERASE HEAD PROBLEM: /
	017122	041111	042514	042440			
	017130	040522	042523	044040			
	017136	040505	020104	051120			
	017144	041117	042514	035115			
	017152	040					
2558	017153	103	042510	045503		.ASCII	/%CHECK POLARITY#
	017160	050040	046117	051101			
	017166	052111	021531				
2559	017172	051445	052105	052455	MSG46:	.ASCII	/%SET-UP WRITE ERROR#
	017200	020120	051127	052111			
	017206	020105	051105	047522			
	017214	021522					
2560	017216	051445	040520	042503	MSG47:	.ASCII	/%SPACE FORWARD ERROR#
	017224	043040	051117	040527			
	017232	042122	042440	051122			
	017240	051117	043				
2561	017243	045	050123	041501	MSG48:	.ASCII	/%SPACE REVERSE ERROR#
	017250	020105	042522	042526			
	017256	051522	020105	051105			
	017264	047522	021522				
2562	017270	041045	043125	042506	MSG49:	.ASCII	/%BUFFERED WRITE ERROR#
	017276	042522	020104	051127			
	017304	052111	020105	051105			
	017312	047522	021522				
2563	017316	041045	052117	051440	MSG50:	.ASCII	/%BOT SET AFTER BUFFERED WRITE#
	017324	052105	040440	052106			
	017332	051105	041040	043125			
	017340	042506	042522	020104			
	017346	051127	052111	021505			
2564	017354	047045	020117	047502	MSG51:	.ASCII	/%NO BOT FROM READ IN PRESET#
	017362	020124	051106	046517			
	017370	051040	040505	020104			
	017376	047111	050040	042522			
	017404	042523	021524				
2565	017410	052045	020103	047111	MSG52:	.ASCII	/%TC INCORRECT #/
	017416	047503	051122	041505			
	017424	020124	043				
2566	017427	045	047515	020114	MSG53:	.ASCII	/%MOL FAILED TO CLEAR#
	017434	040506	046111	042105			
	017442	052040	020117	046103			
	017450	040505	021522				
2567	017454	022445	042522	042523	MSG54:	.ASCII	/%RESET SLAVE TO ON LINE BEFORE CONTINUING/

	017462	020124	046123	053101	
	017470	020105	047524	047440	
	017476	020116	044514	042516	
	017504	041040	043105	051117	
	017512	020105	047503	052116	
	017520	047111	044525	043516	
2568	017526	051445	052105	051440	.ASCII /%SET SW12=1 IF YOU DOT WISH TO REPEAT REWIND OFFLINE TEST#/
	017534	030527	036462	020061	
	017542	043111	054440	052517	
	017550	042040	052117	053440	
	017556	051511	020110	047524	
	017564	051040	050105	040505	
	017572	020124	042522	044527	
	017600	042116	047440	043106	
	017606	044514	042516	052040	
	017614	051505	021524		
2569	017620	044440	042524	035122	MSG56: .ASCII / ITER: #/
	017626	021440			
2570	017630	052045	020115	047516	MSG57: .ASCII /%TM NOT SET#/
	017636	020124	042523	021524	
2571	017644	042445	052111	042510	MSG60: .ASCII /%EITHER TAPE NOT ERASED OR OPI PROBLEM#/
	017652	020122	040524	042520	
	017660	047040	052117	042440	
	017666	040522	042523	020104	
	017674	051117	047440	044520	
	017702	050040	047522	046102	
	017710	046505	043		
2572	017713	045	044122	047440	MSG62: .ASCII /%RH ONLY (NO=0,YES=1): #/
	017720	046116	020131	047050	
	017726	036517	026060	042531	
	017734	036523	024461	020072	
	017742	043			
2573	017743	045	044504	020104	MSG63: .ASCII /%DID NOT AUTO SELECT NRZ#/
	017750	047516	020124	052501	
	017756	047524	051440	046105	
	017764	041505	020124	051116	
	017772	021532			
2574	017774	042045	042111	047040	MSG64: .ASCII /%DID NOT AUTO SELECT PE#/
	020002	052117	040440	052125	
	020010	020117	042523	042514	
	020016	052103	050040	021505	
2575	020024	021445			MSG65: .ASCII /%#/
2576	020026	021534			MSG66: .ASCII /\#/
2577	020030	042445	035122	021440	MSG67: .ASCII /%ER: #/
2578	020036	051045	046505	053117	MSG69: .ASCII /%REMOVE TMDP FROM SLAVE TO BE TESTED%/
	020044	020105	046524	050104	
	020052	043040	047522	020115	
	020060	046123	053101	020105	
	020066	047524	041040	020105	
	020074	042524	052123	042105	
	020102	021445			
2579	020104	044045	051101	053504	MSG70: .ASCII /%HARDWARE SWR IN USE%/
	020112	051101	020105	053523	
	020120	020122	047111	052440	
2580	020126	042523	021445		

```
2582                                     :TEST HEADERS*****
2583
2584 020132 022445 052106 035061 MSFT1: .ASCII /%FT1:RH ADDRESSING #/
      020140 044122 040440 042104
      020146 042522 051523 047111
      020154 020107 043
2585 020157 045 043045 031124 MSFT2: .ASCII /%FT2:RH REGISTER BITS TEST #/
      020164 051072 020110 042522
      020172 044507 052123 051105
      020200 041040 052111 020123
      020206 042524 052123 021440
2586 020214 022445 052106 035063 MSFT3: .ASCII /%FT3:RH INITIALIZE TEST #/
      020222 044122 044440 044516
      020230 044524 046101 055111
      020236 020105 042524 052123
      020244 021440
2587 020246 022445 052106 035064 MSFT4: .ASCII /%FT4:RH11 SILO TEST 1 #/
      020254 044122 030461 051440
      020262 046111 020117 042524
      020270 052123 030440 021440
2588 020276 022445 052106 035065 MSFT5: .ASCII /%FT5:RH11 SILO TEST 2 #/
      020304 044122 030461 051440
      020312 046111 020117 042524
      020320 052123 031040 021440
2589 020326 022445 052106 035066 MSFT6: .ASCII /%FT6:RH11 SILO TEST 3 #/
      020334 044122 030461 051440
      020342 046111 020117 042524
      020350 052123 031440 021440
2590 020356 022445 052106 035067 MSFT7: .ASCII /%FT7:RH11 SILO TEST 4 #/
      020364 044122 030461 051440
      020372 046111 020117 042524
      020400 052123 032040 021440
2591 020406 022445 052106 030061 MSFT10: .ASCII /%FT10:RH11 SILO TEST 5 #/
      020414 051072 030510 020061
      020422 044523 047514 052040
      020430 051505 020124 020065
      020436 043
2592 020437 045 043045 030524 MSFT11: .ASCII /%FT11:NOP TEST#/
      020444 035061 047516 020120
      020452 042524 052123 043
2593 020457 045 043045 030524 MSFT12: .ASCII /%FT12:REWIND TEST#/
      020464 035062 042522 044527
      020472 042116 052040 051505
      020500 021524
2594 020502 022445 052106 031461 MSFT13: .ASCII /%FT13:WRITE-READ TEST#/
      020510 053472 044522 042524
      020516 051055 040505 020104
      020524 042524 052123 043
2595 020531 045 043045 030524 MSFT14: .ASCII /%FT14:SPACE TEST#/
      020536 035064 050123 041501
      020544 020105 042524 052123
      020552 043
2596 020553 045 043045 030524 MSFT15: .ASCII /%FT15:ERASE TEST#/
      020560 035065 051105 051501
      020566 020105 042524 052123
      020574 043
```

2597	020575	045	043045	030524	MSFT16: .ASCII /%FT16:TAPE MARK WRITE-READ TEST#/ /
	020602	035066	040524	042520	
	020610	046440	051101	020113	
	020616	051127	052111	026505	
	020624	042522	042101	052040	
	020632	051505	021524		
2598	020636	022445	052106	033461	MSFT17: .ASCII /%FT17:TM SPACE TEST #/ /
	020644	052072	020115	050123	
	020652	041501	020105	042524	
	020660	052123	021440		
2599	020664	022445	052106	030062	MSFT20: .ASCII /%FT20:WRITE CHECK TEST #/ /
	020672	053472	044522	042524	
	020700	041440	042510	045503	
	020706	052040	051505	020124	
	020714	043			
2600	020715	045	043045	031124	MSFT21: .ASCII /%FT21:ERASE HEAD TEST#/ /
	020722	035061	051105	051501	
	020730	020105	042510	042101	
	020736	052040	051505	021524	
2601	020744	022445	052106	031062	MSFT22: .ASCII /%FT22:BUFFERED COMMAND TEST#/ /
	020752	041072	043125	042506	
	020760	042522	020104	047503	
	020766	046515	047101	020104	
	020774	042524	052123	043	
2602	021001	045	043045	031124	MSFT23: .ASCII /%FT23:READ IN PRESET TEST#/ /
	021006	035063	042522	042101	
	021014	044440	020116	051120	
	021022	051505	052105	052040	
	021030	051505	021524		
2603	021034	022445	052106	032062	MSFT24: .ASCII /%FT24:AUTO DENSITY SELECT: WRITE-NRZ,READ-PE#/ /
	021042	040472	052125	020117	
	021050	042504	051516	052111	
	021056	020131	042523	042514	
	021064	052103	020072	051127	
	021072	052111	026505	051116	
	021100	026132	042522	042101	
	021106	050055	021505		
2604	021112	022445	052106	032462	MSFT25: .ASCII /%FT25:AUTO DENSITY SELECT: WRITE-PE,READ-NRZ#/ /
	021120	040472	052125	020117	
	021126	042504	051516	052111	
	021134	020131	042523	042514	
	021142	052103	020072	051127	
	021150	052111	026505	042520	
	021156	051054	040505	026504	
	021164	051116	021532		
2605	021170	022445	052106	033062	MSFT26: .ASCII /%FT26:SEQUENTIAL TAPE MARK TEST#/ /
	021176	051472	050505	042525	
	021204	052116	040511	020114	
	021212	040524	042520	046440	
	021220	051101	020113	042524	
	021226	052123	043		
2606	021231	045	043045	031124	MSFT27: .ASCII /%FT27:REWIND-OFF LINE TEST#/ /
	021236	035067	042522	044527	
	021244	042116	047455	043106	
	021252	046040	047111	020105	
	021260	042524	052123	043	

CZTECDO TM03-TE16/TU77 BFT
CZTECD.P11 06-JUL-83 14:41

MACY11 30(1046) 06-JUL-83 21:02 H 6
PAGE 52-2

SEQ 0072

2607	021265	045	043536	043	SCNTG:	.ASCII	/X^G#/ /XSWR= #/
2608	021271	045	053523	036522	SMSWR:	.ASCII	
2609	021276	021440			SMNEW:	.ASCII	/ NEW= #/
	021300	020040	042516	036527			
	021306	021440			SQUEST:	.ASCII	/?%#/ .EVEN
2610	021310	022477	043		WDATA:	0	
2611							
2612							
2613		021314					
2614	021314	000000			RDATA:	0	
2615		023026					
2616	023026	000000					
2617							
2618		000001					.END

FT15B	006662	1504#	1510					
FT15X	006772	1508	1530#					
FT15XX	007176	1549	1561#					
FT16	007210	798	799	1568#				
FT16A	007242	1573#	1596					
FT16B	007246	1574#						
FT16X	007426	1594	1597#					
FT17	007436	800	801	1604#				
FT17A	007456	1607#	1661					
FT17B	007462	1608#	1638					
FT17C	007622	1628#	1637					
FT17D	007670	1621	1639#					
FT17D1	007706	1642#	1657					
FT17E	007722	1644#	1652					
FT17F	010012	1654	1658#					
FT17X	010034	1616	1635	1649	1659	1662#		
FT2	003376	774	775	1035#				
FT2A	003410	1037#	1044	1069				
FT2B	003450	1042	1046#	1053				
FT2C	003510	1051	1055#	1064				
FT2D	003524	1059#	1060					
FT2E	003554	1062	1066#					
FT2ER	003564	1045	1054	1065	1070#	1820	1961	
FT2ERA	003614	1074	1077#					
FT2ERB	003666	1072	1088#					
FT2ERC	003676	1089	1091#					
FT2X	003706	1067	1094#					
FT20	010040	802	803	1667#				
FT20A	010056	1670#	1695					
FT20B	010202	1687#						
FT20C	010234	1692#						
FT20X	010254	1681	1693	1696#				
FT21	010264	804	805	1702#				
FT21A	010272	1703#						
FT21B	010566	1740	1743#					
FT21C	010574	1742	1744#					
FT21SC	010422	1721#	1744					
FT21X	010606	1736	1738	1746#				
FT22	010616	806	807	1753#	1778			
FT22A	010672	1761#	1763					
FT22B	010712	1766#	1767					
FT22X	011012	1776	1780#					
FT23	011022	808	809	1787#	1812	1819		
FT23A	011132	1803#	1806	1808				
FT23B	011152	1804	1809#					
FT23C	011202	1810	1814#					
FT23X	011236	1788	1817	1821#				
FT24	011242	810	811	1828#	2023			
FT24X	011424	1841	1848	1851#				
FT25	011434	812	813	1857#	2025			
FT25X	011616	1870	1877	1880#				
FT26	011626	814	815	1887#	2178	2180		
FT26X	012054	1898	1904	1913	1918	1920	1923#	
FT27	012064	816	817	1928#	2183	2185		
FT27X	012274	1955	1959	1962#				
FT27XX	012304	1929	1931	1964#				

MSFT13	020502	1345	2594#		
MSFT14	020531	1387	1458	2595#	
MSFT15	020553	1498	2596#		
MSFT16	020575	1571	2597#		
MSFT17	020636	1605	2598#		
MSFT2	020157	1035	2585#		
MSFT20	020664	1668	2599#		
MSFT21	020715	1702	2600#		
MSFT22	020744	1753	2601#		
MSFT23	021001	1789	2602#		
MSFT24	021034	1828	2603#		
MSFT25	021112	1857	2604#		
MSFT26	021170	1888	2605#		
MSFT27	021231	1932	2606#		
MSFT3	020214	1101	2586#		
MSFT4	020246	1147	2587#		
MSFT5	020276	1173	2588#		
MSFT6	020326	1205	2589#		
MSFT7	020356	1258	2590#		
MSG1	015244	2055	2510#		
MSG10	016007	894	2522#		
MSG10A	015647	884	2520#		
MSG11	016027	911	2523#		
MSG12	016047	1354	1631	1712	2524#
MSG13	016065	1364	1585	2525#	
MSG14	016112	1371	1590	1728	2526#
MSG15	016137	1579	1613	1894	2527#
MSG15A	016160	1900	2528#		
MSG15B	016217	1551	2529#		
MSG16	016225	1471	1641	1682	2530#
MSG17	016245	1469	1655	1687	2531#
MSG2	015332	2512#			
MSG20	016265	1462	2041	2532#	
MSG21	016273	1465	2044	2533#	
MSG22	016300	1080	1240	1473	2534#
MSG23	016310	1084	1244	1478	2535#
MSG24	016320	2247	2536#		
MSG25	016334	1043	2537#		
MSG26	016342	1052	2538#		
MSG27	016350	1063	2539#		
MSG28	016356	2540#			
MSG29	016406	1109	2541#		
MSG3	015405	843	848*	2513#	
MSG30	016435	1113	2542#		
MSG31	016465	1119	2543#		
MSG32	016515	1158	1269	1296	2544#
MSG33	016533	1160	2545#		
MSG34	016550	1162	2546#		
MSG35	016566	1178	2547#		
MSG36	016613	1182	2548#		
MSG37	016642	1187	1195	2549#	
MSG38	016701	1219	2550#		
MSG39	016733	1238	2551#		
MSG4	015545	849	2515#		
MSG40	016753	1215	2552#		
MSG41	017006	908	2553#		

CZTECDO TM03-TE16/TU77 BFT
CZTECD.P11 06-JUL-83 14:41

MACY11 30(1046) 06-JUL-83 21:02 PAGE 53-6
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0079

REGS	000572	700#	851	853	870									
RFD	000634	720#	1413*	1436*	1450									
RHOF	000726	749#	934	936	1292									
RHTF	000722	747#	882*	1026	1028*	2257								
RH17F	000604	708#	951*	957*	1145	1171	1203	1256	1276					
RRD	000632	719#	1412*	1435*	1443									
RTRN	000646	725#	2205											
RWND	012442	1349	1389	1499	1512	1573	1607	1670	1703	1717	1722	1754	1829	1842
		1858	1871	1890	1905	1933	1998#							
SAV1	000666	733#												
SAV2	000670	734#												
SAV3	000672	735#												
SCHN	001710	836	837#											
SCNT	000642	723#	1414*	1424*	1433*	1642*	1907*	1978						
SCOLP	000674	736#	1044*	1053*	1064*	1102*	1164*	1174*	1206*	1259*	1279*	1303*	1322*	1356*
		1744*	1778*	1812*	1819*	1944*	2150	2154	2255*					
SCOPE	013274	1091	1139	2145#	2256									
SERFL	000712	743#	1403	1419	1429	1507	1615	1634	1648	1680	1840	1869	1897	1903
		1912	1917	2012*	2030*	2089								
SERNUM	000562	696#												
SLVN	000614	712#	913	915	922	965*	1787	2199						
SN	000540	684#	930											
SNPG	015056	2465	2468	2474	2476	2478#								
SNPT	015000	931	2459#											
START	001600	655	824#											
STFLG	000704	740#	950*	979	984*									
STMSK	000660	730#	971*	1398*	1495*	1519*	1578*	1584*	1612*	1632*	1646*	1773*	1893*	2020*
		2027*	2028											
STSCD	003126	964	984#											
STO	002124	873#	876											
STOB	001764	849#	2259											
ST1	002146	879#	881											
ST1A	002166	884#	910	1029										
ST2	002344	907	911#	927										
ST3	002454	924	928#											
ST4	002540	659	943#											
SWR	000550	691#	829	833*	836*	962	976	981	1004	1071	1088	1126	1135	1232
		1248	1456	1483	1928	2031	2049	2077	2146	2159	2223	2225*	2232*	2240
		2251	2487	2492	2496									
SWREG	000176	649#	833	836	2223	2232	2487							
TAG	007206	1550*	1560*	1564#	2067									
TC	000542	685#	922*	965*	1324*	1539*	1815	1970*	2042	2075	2199*			
TEMPST	000732	751#												
TEMP1	000652	727#	1547*	1548	1623*	1636*	2069	2282*	2291	2315*				
TEMP2	000654	728#												
TEMP3	000656	729#	2450	2453										
TEND	003162	818	986	992#	1294	2187								
TENDX	003260	1003	1005	1007#										
TEX	014436	2356	2374#											
TIB	000602	707#	2285	2289	2294	2299	2309	2312	2319*	2320	2346*	2347*	2348	
TINER	014240	2311	2314	2325	2328	2335#								
TKB	000554	693#	2212	2346										
TKS	000552	692#	949*	2343*	2344									
TLAST	001120	819#	985	2190*										
TMCHK	013124	1581	1587	1592	1617	1650	1896	1902	1911	1916	2087#			
TOB	000600	706#	877	2348*	2354*	2355	2357	2361*	2364*	2368*	2373	2416*	2427*	2435*

CZTECD0 TMO3-TE16/TU77 BFT
CZTECD.P11 06-JUL-83 14:41

MACY11 30(1046) 06-JUL-83 21:02 ^{D 7} PAGE 54
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0081

\$SCATCH	555#	636
\$CHAIN	555#	836
\$CHNMO	555#	965
\$RESTO	555#	2505
\$SAVE	555#	2504
.\$ACT1	555#	638
.\$EOP	555#	1001

. ABS. 023030 000

ERRORS DETECTED: 0

CZTECD,CZTECD/CRF=CZTEAD.SML/ML,CZTECD.P11
RUN-TIME: 3 6 1 SECONDS
RUN-TIME RATIO: 16/11=1.4
CORE USED: 11K (22 PAGES)