

TM02/TU45

BASIC FUNCTION TEST
CZTUJA0

AH-E473A-MC

COPYRIGHT © 75-78

FICHE 1 OF 1

JUL 1978

digital

MADE IN USA

.NLIST SEQ,LOC,BIN
.REM_

IDENTIFICATION

PRODUCT CODE: AC-E472A-MC
PRODUCT NAME: CZTUJA0 TM02/TU45 BASIC FUNCTION TEST
DATE CREATED: 25 MAY 1978
MAINTAINER: COMPUTER SPECIAL SYSTEMS
AUTHOR: R.B. BARNES/R. J. COLLINS

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT (C) 1975, 1976, 1977, 1978 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	1
2.	REQUIREMENTS	1
3.	LOADING PROCEEDURE	1
4.	STARTING PROCEEDURE	1
5.	SWITCH SETTINGS	2
6.	ERROR PRINTOUTS	3
7.	OPERATION	4
8.	TEST DESCRIPTION	5
9.	LISTING	

(PAGE 1)

1. ABSTRACT

THIS PROGRAM IS INTENDED TO TEST ALL OF THE BASIC FUNCTIONAL LEVEL OPERATIONS OF THE TM02/TU45 MAG TAPE SYSTEM. ALL FUNCTIONS; WRITE, READ, SPACE, ERASE, REWIND, ETC; WILL BE TESTED. IN ADDITION TO THE TM02/TU45 TESTS, THE RH WILL BE TESTED SEPARATELY IN SO FAR AS IT IS POSSIBLE TO SEPARATE THE RH FROM THE TM02/TU45 ITSELF.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR
- B. 8K OF CORE
- C. CONSOLE TTY
- D. TM02 MAGTAPE CONTROLLER
- E. MASS BUS CONTROLLER
- F. TU45 MAG TAPE TRANSPORT

3. LOADING PROCEEDURE

USE STANDARD BINARY LOADING PROCEEDURE

4. STARTING PROCEEDURE

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.

4.1 SAMPLE START AT 200(8): OPERATOR RESPONSES ARE IN PARENS.

TM02-TU45 BASIC FUNCTION TEST
ENTER CONDITIONS IN OCTAL

REGISTER START: 172440 (CR)
VECTOR: 224 (CR)
DRIVE NUMBER: 0 (3)
SLAVE NUMBER: 0 (6) SERIAL NO: 200
RH11 OR RH70: (0)
RH ONLY: (0)
NRZ ONLY: (1)

THIS EXAMPLE SLOWS THE PROGRAM START USING THE RH11 ADDRESS (CS1) OF 172440, AN INTERRUPT VECTOR OF 224,

TM02/TU45 BASIC FUNCTION TEST MACY11 30(1046) 13-JUN-78 13:49^{E 1} PAGE 4
CZTUJA.P11 07-JUN-78 16:58

SEQ 0004

DRIVE NUMBER 3, AND SLAVE NUMBER 6, NRZ ONLY.
NOTE THAT THE CURRENT VALUES FOR EACH PARAMETER IS
PRINTED AND MAY OR NOT BE CHANGED.

(PAGE 2)

5. CONSOLE SWITCH SETTING

ALL SWITCHES EXCEPT 5-9 ARE USED AND THE NORMAL, OR DEFAULT, RUN IS DONE WITH ALL SWITCHES SET TO ZERO (0). ALL SWITCHES ARE DYNAMIC, AND MAY BE CHANGED AT ANY TIME, ASSUMING A CONSOLE SWITCH REGISTER IS PRESENT. IF A CONSOLE SWITCH REGISTER IS NOT PRESENT, THE SWITCHES ARE NOT DYNAMIC AND MUST BE SET BEFORE STARTING, USING THE FOLLOWING PROCEDURE:

- A) LOAD ADDRESS 636(8) LABELLED "SWR"
- B) DEPOSIT THE VALUE 176(8)
- C) LOAD ADDRESS 176(8)
- D) DEPOSIT THE DESIRED SWITCH VALUE.

SW15: 1=HALT ON ERROR
0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
0=CONTINUE
SW13: 1=DO NOT PRINT ERRORS
0=PRINT ALL ERRORS
SW12: 1=INHIBIT ITERATION
0=DO ALL ITERATIONS PER TEST
SW11: 1=CONTINUOUS CYCLE
0=HALT AT END OF PASS
SW10: 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 CHANGING NUMBER.

(PAGE 3)

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

(PAGE 4)

7. OPERATION

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

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE
3. PRESS START
4. ENTER APPROPRIATE RESPONSES TO THE TTY REQUESTS

ALL CONSOLE 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.

SINGLE TEST SELECTION: (SW0-SW4)

WHEN SW0-4 ARE SET TO ZERO (00) THE SCHEDULAR WILL
EXECUTE ALL OF THE TESTS IN SEQUENCE (1-24). IF SW0-4 IS
SET TO SOME SPECIFIC TEST NUMBER (1-24) 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.

RH11 OR RH70 OPTION:

A ONE RESPONSE IS FOR THE RH70;
A ZERO RESPONSE IS FOR THE RH11.

RH ONLY OPTION:

BY RESPONDING TO THE REQUEST (RH ONLY:) WITH
A ONE (1), ONLY THE TESTS WHICH ARE POINTED TO
THE RH (TESTS 1 - 10) WILL BE EXECUTED IN
EACH PASS.

(PAGE 5)

8. TEST DESCRIPTION

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 TM02/TU45 ITSELF. (SEE RH ONLY OPTION; PAR 7)

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

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

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

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

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

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

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

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

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

(PAGE 6)

B. TM02/TU45 BASIC FUNCTIONS: THE FOLLOWING FOURTEEN (14)
TESTS WILL ASSURE OPERATION OF THE
MAG TAPE BASIC FUNCTIONS.

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

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

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

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

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

DATA RELATED ERRORS (PARITY ERROR, CRC ERROR, ETC) ARE IGNORED. THE ACTUAL
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

(PAGE 7)

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

TM02/TU45 BASIC FUNCTION TEST MACY11 30(1046) 13-JUN-78 13:49^{M 1} PAGE 12
CZTUJA.P11 07-JUN-78 16:58

SEQ 0012

10. END

(PAGE 8)

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 TM02 WILL ACCEPT AND EXECUTE ANOTHER COMMAND WHILE ITS SELECTED SLAVE IS REWINDING.

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

TM02/TU45 BASIC FUNCTION TEST
CZTUJA.P11 07-JUN-78 16:58

MACY11 30(1046) 13-JUN-78 13:49^{B 2} PAGE 14

SEQ 0014

(PAGE 9)

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: 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 11 = 1)

9. LISTING

-


```
485      .LIST  SEQ,LOC,BIN
486      .TITLE TM02/TU45 BASIC FUNCTION TEST
487      ;CZTUJA0
488      ;25 MAY 78
489      ;R. BARNES/R. J. COLLINS
490
491      .ENABL  ABS
492
493
494      ;CONSOLE SWITCHES*****
495      ;
496      ;SW15:  1=HALT ON ERROR
497      ;        0=CONTINUE
498      ;SW14:  1=LOOP ON ERROR (SCOPE)
499      ;        0=CONTINUE
500      ;SW13:  1=DO NOT PRINT ERRORS
501      ;        0=PRINT ERRORS
502      ;SW12:  1=INHIBIT ITERATIONS
503      ;        0=DO ITERATIONS
504      ;SW11:  1=CONTINUOUS CYCLE
505      ;        0=HALT AT END OF PASS
506      ;SW10:  1=HALT AT END OF EACH TEST
507      ;        0=CONTINUE
508      ;SW0-4: SELECT TEST NUMBER :: 00=ALL TESTS
```

[illegible]

```

555                                     ;REGISTER EQUIVS*****
556
557             000000                 R0=%0
558             000001                 R1=%1
559             000002                 R2=%2
560             000003                 R3=%3
561             000004                 R4=%4
562             000005                 R5=%5
563             000006                 SP=%6
564             000007                 PC=%7
565
566                                     ;TRAP CATCHERS*****
567
568             000000                 .=0
569             000200                 .REPT 200
570                                     .+2
571             HALT
572             .ENDR
573
574                                     ;TTY INTERRUPT VECTOR*****
575
576             000060                 .=60
577 000060 012004                 TTINT           ;TTY INTERRUPT HEADER ADDRESS
578 000062 000000                 0
579
580                                     ;START ADDRESS*****
581
582             000200                 .=200
583 000200 005000                 CLR           R0
584 000202 000167 001372         JMP           START ;PROGRAM START
585
586             000210                 .=210
587 000210 000240                 NOP
588 000212 012700 000001         MOV           #1,R0 ;SET NO HEADER FLAG
589 000216 000167 001356         JMP           START
590
591                                     ;TM02 INTERRUPT VECTOR*****
592
593             000224                 .=224
594 000224 011770                 MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
595 000226 000340                 340
596

```


597				
598	000600		.=600	
599			;MASS BUS REGISTER EQUIVS*****	
600				
601	000600	172440	C1:	172440
602	000602	172442	WC:	172442
603	000604	172444	BA:	172444
604	000606	172446	FC:	172446
605	000610	172450	CS:	172450
606	000612	172452	DS:	172452
607	000614	172454	ER:	172454
608	000616	172456	AS:	172456
609	000620	172460	CC:	172460
610	000622	172462	DB:	172462
611	000624	172464	MR:	172464
612	000626	172466	DT:	172466
613	000630	172470	SN:	172470
614	000632	172472	TC:	172472
615				
616			;CONSTANTS*****	
617				
618	000634	177776	PSW:	177776 ;PROCESSOR STATUS
619	000636	177570	SWR:	177570 ;SWITCH REGISTER
620	000640	177560	TKS:	177560 ;TTY READER STATUS
621	000642	177562	TKB:	177562 ;TTY READ BUFFER
622	000644	177564	TPS:	177564 ;TTY PUNCH STATUS
623	000646	177566	TPB:	177566 ;TTY PUNCH BUFFER
624	000650	177777	SERNUM:	177777 ;SERIAL NUMBER
625	000652	000011	DRVTP:	011 ;DRIVE TYPE
626	000654	000010	ITAMT:	10 ;ITERATION AMOUNT
627	000656	000224	VECT:	224 ;INTERRUPT VECTOR(RH)
628	000660	172440	REGS:	172440 ;STARTING REGISTER ADDRESS
629	000662	000004	BTRP:	4 ;BUS TRAP ADDRESS
630	000664	000006	BTRP2:	6 ;BUS TRAP PRIORITY LEVEL 7

631 ;FLAGS AND COUNTERS*****

632			
633	000666	000000	TOB: 0
634	000670	000000	TIB: 0
635	000672	000000	RH17F: 0
636	000674	000000	HDRFL: 0
637	000676	000000	EMADDR: 0
638	000700	000000	DRVN: 0
639	000702	000000	SLVN: 0
640	000704	000000	BADDR: 0
641	000706	000000	FCNT: 0
642	000710	000000	WCNT: 0
643	000712	000000	RCNT: 0
644	000714	000000	ERRP: 0
645	000716	000000	ERRP1: 0
646	000720	000000	RRD: 0
647	000722	000000	RFD: 0
648	000724	000000	RDYDX: 0
649	000726	000000	OPDYX: 0
650	000730	000000	SCNT: 0
651	000732	000000	PFLG: 0
652	000734	000000	RTRN: 0
653	000736	000000	ERADD: 0
654	000740	000000	TEMP1: 0
655	000742	000000	TEMP2: 0
656	000744	000000	TEMP3: 0
657	000746	000000	STMSK: 0
658	000750	000000	ITCNT: 0
659	000752	000000	DSAV: 0
660	000754	000000	SAV1: 0
661	000756	000000	SAV2: 0
662	000760	000000	SAV3: 0
663	000762	000000	SCOLP: 0
664	000764	000000	ITRLP: 0
665	000766	000000	EXFL: 0
666	000770	000000	PEXFL: 0
667	000772	000000	STFLG: 0
668	000774	000000	LTADD: 0
669	000776	000000	FUN: 0
670	001000	000000	SERFL: 0
671	001002	000000	CRCNT: 0
672	001004	000000	UDES: 0
673	001006	000000	PATRN: 0
674	001010	000000	RHTF: 0
675	001012	000000	NRZOF: 0
676	001014	000000	RHOF: 0
677	001016	000000	PCNTR: 0

678 ;DATA PATTERN GENERATORS*****

679			
680			
681	001020	000000	DATBL: 0
682	001022	011552	DATA0: DAT1 ;ALL ONE BITS
683	001024	011574	DATA1: DAT2 ;ALL ZERO BITS
684	001026	011602	DATA2: DAT3 ;ALTERNATING ONE/ZERO BITS
685	001030	011612	DATA3: DAT4 ;ALL BITS 0-377

```

686
687
688          ;LOGIC TEST ENTRY TABLE*****
689          TSTTBL: 0
690          0
691          FT1
692          FT1
693          FT2
694          FT2
695          FT3
696          FT3
697          FT4
698          FT4
699          FT5
700          FT5
701          FT6
702          FT6
703          FT7
704          FT7
705          FT10
706          FT10
707          FT11
708          FT11
709          FT12
710          FT12
711          FT13
712          FT13
713          FT14
714          FT14
715          FT15
716          FT15
717          FT16
718          FT16
719          FT17
720          FT17
721          FT20
722          FT20
723          FT21
724          FT21
725          FT22
726          FT22
727          FT23
728          FT23
729          FT24
730          FT24
731          0
732          0
733          0
734          0
  
```



```

735          001600          . =1600
736          ;PROGRAM START AND HOUSEKEEPING*****
737
738 001600 000240          START: NOP
739 001602 012777 000340 177024 MOV #340,@PSW          ;SET PRIORITY
740 001610 012706 000500      MOV #500,SP          ;SET STACK POINTER
741 001614 005700      TST R0          ;SEE IF PRINT HEADER
742 001616 001402      BEQ STOA          ;IF SO: BR
743 001620 000167 000562      JMP ST4          ;ELSE SKIP
744 001624 012704 013252      STOA: MOV #MSG3,R4
745 001630 004767 010532      JSR PC,TTOUT          ;PRINT TITLE
746 001634 012704 013357      STOB: MOV #MSG4,R4
747 001640 004767 010522      JSR PC,TTOUT          ;REQUEST REGISTER ADDRESS
748 001644 016703 177010      MOV REGS,R3
749 001650 004767 010642      JSR PC,OCTP          ;PRINT CURRENT ADDRESS
750 001654 012705 000660      MOV #REGS,R5          ;SET ADDRESS SAVE LOC
751 001660 012701 000006      MOV #6,R1          ;SET SIZE OF RESPONSE
752 001664 012702 177000      MOV #177000,R2          ;SET UPPER LIMIT
753 001670 012703 160000      MOV #160000,R3          ;SET LOWER LIMIT
754 001674 004767 010230      JSR PC,TTR          ;GO GET RESPONSE
755 001700 012704 013402      MOV #MSG5,R4
756 001704 004767 010456      JSR PC,TTOUT          ;REQUEST VECTOR
757 001710 016703 176742      MOV VECT,R3
758 001714 004767 010576      JSR PC,OCTP          ;PRINT CURRENT VECTOR
759 001720 012705 000656      MOV #VECT,R5          ;SET ADDRESS SAVE LOC
760 001724 012701 000003      MOV #3,R1          ;SET SIZE OF RESPONSE
761 001730 012702 000224      MOV #224,R2          ;SET UPPER LIMIT
762 001734 012703 000150      MOV #150,R3          ;SET LOWER LIMIT
763 001740 004767 010164      JSR PC,TTR          ;GO GET RESPONSE
764 001744 016700 176706      MOV VECT,R0          ;GET VECTOR
765 001750 012720 011770      MOV #MTINT,(R0)+          ;LOAD INTERRUPT ADDRESS IN VECTOR
766 001754 012710 000340      MOV #340,(R0)          ;LOAD PRIORITY
767 001760 016700 176674      MOV REGS,R0          ;GET START OF REGS
768 001764 012701 000016      MOV #16,R1          ;SET NUMBER OF REGS
769 001770 012702 000600      MOV #C1,R2          ;GET START OF TABLE
770 001774 010022          ST0: MOV R0,(R2)+          ;BUILD TABLE
771 001776 062700 000002      ADD #2,R0          ;BUMP ADDRESS
772 002002 005301      DEC R1          ;SEE IF DONE
773 002004 001373      BNE ST0          ;IF NOT: BR
774 002006 012702 000666      MOV #TOB,R2
775 002012 012700 000054      MOV #54,R0
776 002016 005022          ST1: CLR (R2)+          ;CLEAR FLAGS + COUNTERS
777 002020 005300      DEC R0
778 002022 001375      BNE ST1
779 002024 012767 000001 176756 MOV #1,RHTF          ;SET ADDRESS TEST FLAG
780 002032 000167 000376      JMP TSRH          ;GO DO INITIAL ADDRESS TEST PASS
781 002036 012704 013463      ST1A: MOV #MSG10,R4
782 002042 004767 010320      JSR PC,TTOUT          ;REQUEST DRIVE NUMBER
783 002046 012705 000700      MOV #DRVN,R5          ;SET ADDRESS OF DRIVE NUMBER SAVE
784 002052 012701 000001      MOV #1,R1          ;SET SIZE OF RESPONSE
785 002056 012702 000007      MOV #7,R2          ;SET UPPER LIMIT
786 002062 012703 000000      MOV #0,R3          ;SET LOWER LIMIT
787 002066 004767 010036      JSR PC,TTR          ;GO GET RESPONSE
788 002072 012777 000040 176510 MOV #40,@CS          ;SET INIT
789 002100 056777 176574 176502 BIS DRVN,@CS          ;SET DRIVE NUMBER
790 002106 005777 176466      TST @C1          ;ACCESS DRIVE

```

791	002112	032777	010000	176470	BIT	#10000,ACS	;SEE IF NED
792	002120	001405			BEQ	ST2	;IF NOT: BR
793	002122	012704	014415		MOV	#MSG41,R4	
794	002126	004767	010234		JSR	PC,TTOUT	;PRINT NOT AVAIL
795	002132	000741			BR	ST1A	;REDO DRIVE REQUEST
796	002134	012704	013503		ST2: MOV	#MSG11,R4	
797	002140	004767	010222		JSR	PC,TTOUT	;REQUEST SLAVE NUMBER
798	002144	012705	000702		MOV	#SLVN,R5	;SET ADDRESS OF SLAVE SAVE
799	002150	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
800	002154	012702	000007		MOV	#7,R2	;SET UPPER LIMIT
801	002160	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
802	002164	004767	007740		JSR	PC,TTR	;GO GET RESPONSE
803	002170	012777	000040	176412	MOV	#40,ACS	;INIT
804	002176	056777	176476	176404	BIS	DRVN,ACS	;SET DRIVE NUMBER
805	002204	016777	176472	176420	MOV	SLVN,ATC	;LOAD SLAVE NUMBER
806	002212	032777	002000	176406	BIT	#2000,ADT	;SEE IF SLAVE PRESENT
807	002220	001005			BNE	ST3	;IF SO: BR
808	002222	012704	014436		MOV	#MSG42,R4	
809	002226	004767	010134		JSR	PC,TTOUT	;PRINT NON-EXIST SLAVE
810	002232	000740			BR	ST2	;REDO SLAVE REQUEST
811	002234	012704	014457		ST3: MOV	#MSG43,R4	
812	002240	004767	010122		JSR	PC,TTOUT	;PRINT SERIAL NUMBER TAG
813	002244	017703	176360		MOV	ASN,R3	
814	002250	004767	010570		JSR	PC,SNPT	;PRINT SERIAL NUMBER
815	002254	012704	015241		MOV	#MSG61,R4	
816	002260	004767	010102		JSR	PC,TTOUT	;REQUEST RH11 OR RH70
817	002264	012705	000672		MOV	#RH17F,R5	;GET ADDRESS OF FLAG
818	002270	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
819	002274	012702	000001		MOV	#1,R2	;SET UPPER LIMIT
820	002300	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
821	002304	004767	007620		JSR	PC,TTR	;GET RESPONSE
822	002310	012704	015261		MOV	#MSG62,R4	
823	002314	004767	010046		JSR	PC,TTOUT	;REQUEST RH11 ONLY RESPONSE
824	002320	012705	001014		MOV	#RHOF,R5	;SET FLAG ADDRESS
825	002324	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
826	002330	012702	000001		MOV	#1,R2	;SET UPPER LIMIT
827	002334	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
828	002340	004767	007564		JSR	PC,TTR	;GO GET RESPONSE
829	002344	005767	176444		TST	RHOF	;SEE IF RH11 ONLY
830	002350	001016			BNE	ST4	;IF SO: BR
831	002352	012704	015132		MOV	#MSG55,R4	
832	002356	004767	010004		JSR	PC,TTOUT	;REQUEST NRZ ONLY RESPONSE
833	002362	012705	001012		MOV	#NRZOF,R5	;SET FLAG ADDRESS
834	002366	012701	000001		MOV	#1,R1	;SET SIZE OF RESPONSE
835	002372	012702	000001		MOV	#1,R2	;SET UPPER LIMIT
836	002376	012703	000000		MOV	#0,R3	;SET LOWER LIMIT
837	002402	004767	007522		JSR	PC,TTR	;GO GET RESPONSE
838	002406	005067	176404		ST4: CLR	PCNTR	;CLEAR PASS COUNTER


```

839                                     ;TEST SCHEDULAR*****
840
841 002412 000240                      TSCD:  NOP
842 002414 005067 176352                CLR      STFLG          ;CLEAR SINGLE TEST FLAG
843 002420 017700 176212                MOV      @SWR,R0
844 002424 042700 177740                BIC      #177740,R0
845 002430 005700                      TST      R0
846 002432 001053                      BNE      STSCD          ;GO SELECT SINGLE TEST
847 002434 012767 001032 176332  TSRH:  MOV      #TSTTBL,LTADD
848 002442 062767 000004 176324  TSCD0:  ADD      #4,LTADD
849 002450 016767 176320 176306        MOV      LTADD,ITRLP
850 002456 062767 000002 176300        ADD      #2,ITRLP          ;SET ITERATION ADDRESS
851 002464 005777 176304                TST      @LTADD
852 002470 001002                      BNE      TSCD1
853 002472 000167 000140                JMP      TEND          ;GO TO END ROUTINE
854 002476 000240                      TSCD1:  NOP
855 002500 005067 176242                CLR      STMSK
856 002504 005067 176204                CLR      ERRP
857 002510 005067 176160                CLR      HDRFL          ;CLEAR PRINT HEADER FLAG
858 002514 017700 176254                MOV      @LTADD,R0          ;SET POINTER TO TEST
859 002520 000110                      JMP      (R0)          ;GO TO TEST
860 002522 000240                      TSCD2:  NOP
861 002524 032777 002000 176104        BIT      #2000,@SWR          ;SEE IF HALT ON TEST
862 002532 001401                      BEQ      TSCD3          ;IF NOT: BR
863 002534 000000                      HALT
864 002536 000240                      TSCD3:  NOP
865 002540 005767 176226                TST      STFLG          ;SE IF SINGLE TEST
866 002544 001736                      BEQ      TSCD0          ;IF NOT: BR
867 002546 017700 176064                MOV      @SWR,R0
868 002552 042700 177740                BIC      #177740,R0          ;MASK TEST NUMBER
869 002556 005700                      TST      R0          ;SEE IF RETURN TO ALL
870 002560 001714                      BEQ      TSCD          ;IF SO: BR
871 002562 000240                      STSCD:  NOP
872 002564 012767 000001 176200        MOV      #1,STFLG          ;SET SINGLE TEST FLAG
873 002572 022700 000025                CMP      #25,R0          ;SEE IF EXCEEDED TESTS
874 002576 003417                      BLE      TEND          ;IF SO: BR
875 002600 000241                      CLC
876 002602 006100                      ROL      R0
877 002604 006100                      ROL      R0          ;SET TABLE MODIFIER
878 002606 012767 001032 176160        MOV      #TSTTBL,LTADD
879 002614 060067 176154                ADD      R0,LTADD          ;SET TEST POINTER
880 002620 016767 176150 176136        MOV      LTADD,ITRLP
881 002626 062767 000002 176130        ADD      #2,ITRLP          ;SET ITERATION POINTER
882 002634 000720                      BR      TSCD1
883 002636 012704 013415                TEND:  MOV      #MSG6,R4
884 002642 004767 007520                JSR      PC,TOUT          ;PRINT END OF PASS
885 002646 016703 176144                MOV      PCNTR,R3
886 002652 004767 007640                JSR      PC,OCTP          ;PRINT PASS NUMBER
887 002656 032777 004000 175752        BIT      #4000,@SWR          ;SEE IF HALT ON PASS
888 002664 001001                      BNE      TENDX          ;IF NOT: BR
889 002666 000000                      HALT
890 002670 005267 176122                TENDX:  INC      PCNTR          ;BUMP PASS COUNTER
891 002674 000167 177512                JMP      TSCD          ;RESTART

```



```

892
893
894
895 002700 012767 015274 175770 FT1: MOV #MSFT1,EMADDR ;SET HEADER
896 002706 012777 012014 175746 MOV #TRAP,@BTRP ;SET TRAP HANDLER ADDRESS
897 002714 012777 000340 175742 MOV #340,@BTRP2
898 002722 012700 000016 MOV #16,R0 ;SET NUMBER OF REGISTER
899 002726 016701 175646 MOV C1,R1 ;GET FIRST ADDRESS (CS1)
900 002732 005711 FT1A: TST (R1) ;REFERENCE REGISTER
901 002734 000240 NOP ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
902 002736 005300 FT1B: DEC R0 ;SEE IF DONE ALL
903 002740 001403 BEQ FT1X ;IF SO: BR
904 002742 062701 000002 ADD #2,R1 ;BUMP ADDRESS POINTER
905 002746 000771 BR FT1A ;CONTINUE
906 002750 012777 000006 175704 FT1X: MOV #6,@BTRP ;RESET TRAP CATCHER
907 002756 005767 176026 TST RHTF ;SEE IF INITIAL ADDRESS TEST PASS
908 002762 001404 BEQ FT1XX ;IF NOT: BR
909 002764 005067 176020 CLR RHTF ;CLEAR FLAG
910 002770 000167 177042 JMP ST1A ;RETURN
911 002774 000167 177522 FT1XX: JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

912
913
914 ;RH REGISTER BITS READ/WRITE*****
915 003000 012767 015321 175670 FT2: MOV #MSFT2,EMADDR ;SET TEST HEADER
916 003006 012701 177777 MOV #-1,R1 ;SET ALL ONES PATTERN
917 003012 004767 006724 FT2A: JSR PC,INIT1 ;GO INIT
918 003016 016700 175560 MOV WC,R0 ;GET ADDRESS OF WORD COUNT
919 003022 010102 MOV R1,R2 ;SET EXPT REGISTER BIT PATTERN
920 003024 010110 MOV R1,(R0) ;LOAD PATTERN
921 003026 021002 CMP (R0),R2 ;SEE IF EXPT=RCVD
922 003030 001410 BEQ FT2B ;IF SO: BR
923 003032 012767 013743 175676 MOV #MSG25,ERADD ;SET CODE
924 003040 012767 003012 175714 MOV #FT2A,SCOLP ;SET SCOPE
925 003046 004767 000116 JSR PC,FT2ER ;GO DO ERROR
926 003052 016700 175526 FT2B: MOV BA,R0 ;GET ADDRESS OF BUS ADDRESS
927 003056 010102 MOV R1,R2
928 003060 042702 000001 BIC #1,R2 ;SET EXPT PATTERN
929 003064 010110 MOV R1,(R0) ;LOAD PATTERN
930 003066 020210 CMP R2,(R0) ;SEE IF EXPT=RCVD
931 003070 001410 BEQ FT2C ;IF SO:BR
932 003072 012767 013751 175636 MOV #MSG26,ERADD ;SET ERROR CODE
933 003100 012767 003052 175654 MOV #FT2B,SCOLP ;SET SCOPE ADDRESS
934 003106 004767 000056 JSR PC,FT2ER ;GO DO ERROR
935 003112 016700 175504 FT2C: MOV DB,R0 ;GET ADDRESS OF DATA BUFFER
936 003116 010102 MOV R1,R2
937 003120 010110 MOV R1,(R0) ;LOAD PATTERN
938 003122 012703 004000 MOV #4000,R3
939 003126 005303 FT2D: DEC R3 ;DELAY
940 003130 001376 BNE FT2D
941 003132 020210 CMP R2,(R0) ;SEE IF EXPT=RCVD
942 003134 001410 BEQ FT2E ;IF SO: BR
943 003136 012767 013757 175572 MOV #MSG27,ERADD ;SET ERROR CODE
944 003144 012767 003112 175610 MOV #FT2C,SCOLP ;SET SCOPE ADDRESS
945 003152 004767 000012 JSR PC,FT2ER ;GO DO ERROR
946 003156 005701 FT2E: TST R1 ;SEE IF DONE RESET
947 003160 001454 BEQ FT2X ;IF SO: BR
948 003162 005001 CLR R1 ;SET ZERO PATTERN
949 003164 000167 177622 JMP FT2A ;DO ZERO BITS
950 003170 000240 FT2ER: NOP
951 003172 032777 020000 175436 BIT #20000,ASWR ;SEE IF PRINT ERROR
952 003200 001034 BNE FT2ERB ;IF NOT: BR
953 003202 005767 175466 TST HDRFL ;SEE ID DONE HEADER
954 003206 001004 BNE FT2ERA ;IF SO: BR
955 003210 016704 175462 MOV EMADDR,R4
956 003214 004767 007146 JSR PC,TTOUT ;DO HEADER
957 003220 012767 000001 175446 FT2ERA: MOV #1,HDRFL ;SET FLAG
958 003226 016704 175504 MOV ERADD,R4
959 003232 004767 007130 JSR PC,TTOUT ;PRINT ERROR CODE
960 003236 012704 013707 MOV #MSG22,R4 ;PRINT EXPT TAG
961 003242 004767 007120 JSR PC,TTOUT
962 003246 010103 MOV R1,R3 ;PRINT EXPT
963 003250 004767 007230 JSR PC,OCTPE
964 003254 012704 013717 MOV #MSG23,R4 ;PRINT RCVD TAG
965 003260 004767 007102 JSR PC,TTOUT
966 003264 011003 MOV (R0),R3
967 003266 004767 007212 JSR PC,OCTPE ;PRINT RCVD

```

968	003272	005777	175340	FT2ERB:	TST	@SWR	;SEE IF HALT ON ERROR
969	003276	100001			BPL	FT2ERC	;IF NOT: BR
970	003300	000000			HALT		
971	003302	004767	006326	FT2ERC:	JSR	PC,SCOPE	;GO SEE IF SCOPE ON ERROR
972	003306	000240			NOP		
973	003310	000207			RTS	PC	;IF NO SCOPE: CONTINUE TEST
974	003312	000240		FT2X:	NOP		
975	003314	004767	006356		JSR	PC,ITER	;GO SEE IF ITERATIONS
976	003320	000167	177176		JMP	TSCD2	;RETURN TO SCHEDULAR


```

977
978
979
980 003324 012767 015356 175344 FT3:  MOV    #MSFT3,EMADDR    ;SET TEST HEADER
981 003332 012767 003324 175422      MOV    #FT3,SCOLP
982 003340 004767 006376      JSR     PC,INIT1      ;GO INIT
983 003344 052777 020000 175236      BIS     #20000,@CS     ;FORCE UPE =1
984 003352 000240      NOP
985 003354 004767 006362      JSR     PC,INIT1      ;GO INIT
986 003360 005777 175214      TST     @C1           ;SEE IF SC IS RESET
987 003364 100005      BPL     FT3A          ;IF SO: BR
988 003366 012767 014015 175342      MOV    #MSG29,ERADD   ;SET ERROR CODE
989 003374 004767 000060      JSR     PC,FT3ER      ;GO DO ERROR
990 003400 032777 040000 175172 FT3A:  BIT     #40000,@C1   ;SEE IF TRE IS RESET
991 003406 001405      BEQ     FT3B          ;IF SO: BR
992 003410 012767 014044 175320      MOV    #MSG30,ERADD   ;SET ERROR CODE.
993 003416 004767 000036      JSR     PC,FT3ER      ;GO DO ERROR
994 003422 017701 175162      FT3B:  MOV    @CS,R1    ;GET CS2
995 003426 042701 000307      BIC     #307,R1       ;MARK IR/OR
996 003432 005701      TST     R1           ;SEE IF RESET
997 003434 001405      BEQ     FT3X          ;IF SO: BR
998 003436 012767 014074 175272      MOV    #MSG31,ERADD   ;SET ERROR CODE
999 003444 004767 000010      JSR     PC,FT3ER      ;GO DO ERROR
1000 003450 004767 006222      FT3X:  JSR     PC,ITER   ;GO SEE IF ITERATION
1001 003454 000167 177042      JMP     TSCD2          ;RETURN TO SCHEDULAR
1002 003460 032777 020000 175150 FT3ER:  BIT     #20000,@SWR ;SEE IF PRINT ERROR
1003 003466 001015      BNE     FT3ERB        ;IF NOT: BR
1004 003470 005767 175200      TST     HDRFL         ;SEE IF DONE HEADER
1005 003474 001006      BNE     FT3ERA        ;IF SO: BR
1006 003476 016704 175174      MOV    EMADDR,R4
1007 003502 004767 006660      JSR     PC,TTOUT      ;PRINT HEADER
1008 003506 005267 175162      INC     HDRFL
1009 003512 016704 175220      FT3ERA: MOV    ERADD,R4
1010 003516 004767 006644      JSR     PC,TTOUT      ;PRINT ERROR CODE
1011 003522 005777 175110      FT3ERB: TST     @SWR    ;SEE IF HALT ON ERROR
1012 003526 100001      BPL     FT3ERC        ;IF NOT: BR
1013 003530 000000      HALT
1014 003532 000240      FT3ERC: NOP
1015 003534 004767 006074      JSR     PC,SCOPE      ;GO SEE IF SCOPE
1016 003540 000207      RTS     PC           ;IF NOT: BR

```

```

1017
1018
1019
1020 003542 005767 175124          FT4:  TST      RH17F
1021 003546 001141                BNE      FT5X      ;IF RH70: BR
1022 003550 012767 015410 175120    MOV      #MSFT4,EMADDR ;SET TEST TEST HEADER
1023 003556 012777 000040 175024    MOV      #40,@CS      ;INIT
1024 003564 017700 175032            MOV      @DB,R0      ;READ DB
1025 003570 005777 175014            TST      @CS      ;SEE IF DLT IS SET
1026 003574 100013                BPL      FT4ER      ;IF NOT: BR
1027 003576 005777 174776            TST      @C1      ;SEE IF SC IS SET
1028 003602 100014                BPL      FT4ERA      ;IF NOT: BR
1029 003604 032777 040000 174766    BIT      #40000,@C1  ;SEE IF TRE IS SET
1030 003612 001414                BEQ      FT4ERB      ;IF NOT: BR
1031 003614 004767 006056          FT4X:  JSR      PC,ITER      ;GO SEE IF ITERATION
1032 003620 000167 176676            JMP      TSCD2      ;RETURN TO SCHEDULAR
1033 003624 012767 014124 175104    FT4ER:  MOV      #MSG32,ERADD ;SET ERROR CODE
1034 003632 000407                BR       FT4ERC
1035 003634 012767 014142 175074    FT4ERA: MOV      #MSG33,ERADD ;SET ERROR CODE
1036 003642 000403                BR       FT4ERC
1037 003644 012767 014157 175064    FT4ERB: MOV      #MSG34,ERADD ;SET ERROR CODE.
1038 003652 000240          FT4ERC:  NOP
1039 003654 012767 003542 175100    MOV      #FT4,SCOLP ;SET SCOPE ADDRESS
1040 003662 004767 177572            JSR      PC,FT3ER ;GO PRINT ERROR
1041 003666 000752                BR       FT4X

```

```

1042
1043
1044
1045 003670 005767 174776      FT5:  TST      RH17F      ;SEE IF RH70
1046 003674 001066              BNE      FT5X      ;IF SO: BR
1047 003676 012767 015440 174772  MOV      #MSFT5,EMADDR ;SET TEST HEADER
1048 003704 012767 003712 175050  MOV      #FT5A,SCOLP   ;SET SCOPE ADDRESS
1049 003712 004767 006024      FT5A: JSR      PC,INIT1    ;GO INIT
1050 003716 032777 000100 174664  BIT      #100,ACS      ;SEE IF IR IS SET
1051 003724 001005              BNE      FT5B      ;IF SO: BR
1052 003726 012767 014175 175002  MOV      #MSG35,ERADD  ;SET ERROR CODE
1053 003734 004767 000122      JSR      PC,FT5ER      ;GO DO ERROR
1054 003740 032777 000200 174642  FT5B: BIT      #200,ACS      ;SEE IF OR IS RESET
1055 003746 001405              BEQ      FT5C      ;IF SO: BR
1056 003750 012767 014222 174760  MOV      #MSG36,ERADD  ;SET ERROR CODE
1057 003756 004767 000100      JSR      PC,FT5ER      ;GO DO ERROR
1058 003762 012777 000000 174632  FT5C: MOV      #0,ADB      ;LOAD ZERO INTO SILO
1059 003770 032777 000200 174612  BIT      #200,ACS      ;SEE THAT OR RESET
1060 003776 001405              BEQ      FT5D      ;IF IT DOES: BR
1061 004000 012767 014251 174730  MOV      #MSG37,ERADD  ;SET ERROR CODE
1062 004006 004767 000050      JSR      PC,FT5ER      ;GO DO ERROR
1063 004012 012777 177777 174602  FT5D: MOV      #-1,ADB      ;LOAD SILO WITH -1
1064 004020 012700 004000      MOV      #4000,R0
1065 004024 032777 000200 174556  FT5E: BIT      #200,ACS      ;SEE IF OR IS SET
1066 004032 001007              BNE      FT5X      ;IF SO: BR
1067 004034 005300              DEC      R0
1068 004036 001372              BNE      FT5E      ;AWAIT OR
1069 004040 012767 014251 174670  MOV      #MSG37,ERADD  ;SET ERROR CODE
1070 004046 004767 000010      JSR      PC,FT5ER      ;GO DO ERROR
1071 004052 004767 005620      FT5X: JSR      PC,ITER      ;GO SEE IF ITERATION
1072 004056 000167 176440      JMP      TSCD2      ;RETURN TO SCHEDULAR
1073 004062 004767 177372      FT5ER: JSR      PC,FT3ER     ;GO PRINT ERROR
1074 004066 000240      NOP
1075 004070 000207      RTS      PC      ;CONTINUE TEST

```



```

1076
1077
1078
1079 004072 005767 174574 FT6: TST RH17F
1080 004076 001052 BNE FT6X ;IF RH70: BR
1081 004100 012767 015470 174570 MOV #MSGT6,EMADDR ;SET TEST HEADER
1082 004106 012767 004114 174646 MOV #FT6A,SCOLP ;SET SCOPE ADDRESS
1083 004114 004767 005622 FT6A: JSR PC,INIT1 ;GO INIT
1084 004120 005000 CLR R0 ;PRESET DATA
1085 004122 010077 174474 FT6B: MOV R0,@DB ;LOAD SILO
1086 004126 005200 INC R0 ;BUMP DATA
1087 004130 022700 000102 CMP #102,R0 ;SEE IF FILLED ALL
1088 004134 001372 BNE FT6B ;IF NOT: BR
1089 004136 032777 000100 174444 BIT #100,@CS ;SEE IF IR IS RESET.
1090 004144 001405 BEQ FT6C ;IF SO: BR
1091 004146 012767 014362 174562 MOV #MSG40,ERADD ;SET ERROR CODE
1092 004154 004767 000054 JSR PC,FT6ER ;GO DO ERROR
1093 004160 032777 000200 174422 FT6C: BIT #200,@CS ;SEE IF OR IS SET
1094 004166 001005 BNE FT6D ;IF SO: BR
1095 004170 012767 014310 174540 MOV #MSG38,ERADD ;SET ERROR CODE
1096 004176 004767 000032 JSR PC,FT6ER ;GO DO ERROR
1097 004202 005000 FT6D: CLR R0 ;PRESET DATA
1098 004204 017701 174412 FT6E: MOV @DB,R1 ;READ SILO
1099 004210 020001 CMP R0,R1 ;SEE IF EXPT=RCVD
1100 004212 001014 BNE FT6DE ;IF NOT: BR
1101 004214 005200 INC R0 ;BUMP DATA
1102 004216 022700 000102 CMP #102,R0 ;SEE IF DONE ALL
1103 004222 001370 BNE FT6E ;IF NOT: BR
1104 004224 004767 005446 FT6X: JSR PC,ITER ;GO SEE IF ITERATION
1105 004230 000167 176266 JMP TSCD2 ;RETURN TO SCHEDULAR
1106 004234 000240 FT6ER: NOP
1107 004236 004767 177216 JSR PC,FT3ER ;GO PRINT ERROR
1108 004242 000207 RTS PC ;RETURN
1109 004244 000240 FT6DE: NOP
1110 004246 032777 020000 174362 BIT #20000,@SWR ;SEE IF PRINT ERROR
1111 004254 001032 BNE FT6DEB ;IF NOT: BR
1112 004256 005767 174412 TST HDRFL ;SEE IF DONE HEADER
1113 004262 016701 174410 MOV EMADDR,R1
1114 004266 004767 006074 JSR PC,TTOUT ;PRINT HEADER
1115 004272 005267 174376 INC HDRFL ;SET FLAG
1116 004276 012704 014342 FT6DEA: MOV #MSG39,R4
1117 004302 004767 006060 JSR PC,TTOUT ;PRINT SILO READ ERROR
1118 004306 012704 013707 MOV #MSG22,R4
1119 004312 004767 006050 JSR PC,TTOUT ;PRINT EXPT TAG
1120 004316 010003 MOV R0,R3
1121 004320 004767 006172 JSR PC,OCTP ;PRINT EXPT
1122 004324 012704 013717 MOV #MSG23,R4
1123 004330 004767 006032 JSR PC,TTOUT ;PRINT RCVD TAG
1124 004334 010103 MOV R1,R3
1125 004336 004767 006154 JSR PC,OCTP ;PRINT RCVD
1126 004342 005777 174270 FT6DEB: TST @SWR ;SEE IF HALT ON ERROR
1127 004346 100001 BPL FT6DEX ;IF NOT: BR
1128 004350 000000 HALT
1129 004352 000207 FT6DEX: RTS PC ;RETURN TO TEST

```

```

1130
1131
1132
1133 004354 005767 174312          FT7:  TST      RH17F
1134 004360 001021                BNE      FT7X      ;IF RH70: BR
1135 004362 012767 015520 174306    MOV      #MSFT7,EMADDR ;SET TEST HEADER
1136 004370 012767 004354 174364    MOV      #FT7,SCOLP   ;SET SCOPE ADDRESS
1137 004376 004767 005340          JSR      PC,INIT1    ;GO INIT
1138 004402 012700 000103          MOV      #103,R0      ;SET SIZE OF SILO +1
1139 004406 010077 174210          FT7A:  MOV      R0,@DB    ;LOAD SILO
1140 004412 005300                DEC      R0          ;SEE IF DONE
1141 004414 001374                BNE      FT7A          ;IF NOT: BR
1142 004416 005777 174166          TST      @CS        ;SEE IF DLT IS SET
1143 004422 100004                BPL      FT7ER        ;IF NOT: BR
1144 004424 004767 005246          FT7X:  JSR      PC,ITER    ;GO SEE IF ITERATION
1145 004430 000167 176066          JMP      TSCD2      ;RETURN TO SCHEDULAR
1146 004434 012767 014124 174274    FT7ER: MOV      #MSG32,ERADD ;SET ERROR CODE
1147 004442 004767 177012          JSR      PC,FT3ER   ;GO DO ERROR
1148 004446 000766                BR       FT7X

```

```

1149
1150
1151
1152 004450 005767 174216 FT10: TST RH17F
1153 004454 001034 BNE FT10X ;IF RH70: BR
1154 004456 012767 015550 174212 MOV #MSFT10,EMADDR ;SET TEST HEADER
1155 004464 012767 004450 174270 MOV #FT10,SCOLP ;SET SCOPE ADDRESS
1156 004472 012777 000040 174110 MOV #40,ACS ;INITIALIZE
1157 004500 012700 000004 MOV #4,R0 ;SET NUMBER OF SILO WRITER
1158 004504 010077 174112 FT10A: MOV R0,ADB ;WRITE SILO
1159 004510 005300 DEC R0 ;SEE IF DONE
1160 004512 001374 BNE FT10A ;IF NOT: BR
1161 004514 052777 000040 174066 BIS #40,ACS ;INITIALIZE
1162 004522 012777 177777 174072 MOV #-1,ADB ;WRITE SILO
1163 004530 017701 174066 MOV ADB,R1 ;READ SILO 1
1164 004534 017701 174062 MOV ADB,R1 ;READ SILO 2
1165 004540 005777 174044 TST ACS ;SEE IF DLT IS SET
1166 004544 100011 BPL FT10ER ;IF NOT: BR
1167 004546 004767 005124 FT10X: JSR PC,ITER ;GO SEE IF ITERATION
1168 004552 005767 174236 TST RHOF ;SEE IF RH11 ONLY
1169 004556 001402 BEQ FT10XX ;IF NOT: BR
1170 004560 000167 176052 JMP TEND ;ELSE GO TO END
1171 004564 000167 175732 FT10XX: JMP TSCD2 ;RETURN TO SCHEDULAR
1172 004570 012767 014124 174140 FT10ER: MOV #MSG32,ERADD ;SET ERROR CODE
1173 004576 004767 176656 JSR PC,FT3ER ;GO DO ERROR
1174 004602 000761 BR FT10X

```



```

1175                                     :NOP TEST*****
1176
1177 004604 000240          FT11:  NOP
1178 004606 012767 004604 174146  MOV      #FT11,SCOLP      ;SET SCOPE ADDRESS
1179 004614 004767 005122          JSR      PC,INIT1
1180 004620 012767 000300 174156  MOV      #300,UDES      ;SET TC= ALL NRZ,NORM,ODD
1181 004626 012767 177777 174052  MOV      #-1,FCNT      ;SET FC= ALL OVER
1182 004634 012767 177777 174046  MOV      #-1,WCNT      ;SET WC= ALL OVER
1183 004642 012767 177777 174034  MOV      #-1,BADDR      ;SET BA= ALL OVER
1184 004650 012767 000001 174046  MOV      #1,RDYDX      ;SET DELAY
1185 004656 012767 000001 174042  MOV      #1,OPDYX      ;SET OP DELAY
1186 004664 012767 000001 174104  MOV      #1,FUN        ;SET NOP FUNCTIONS CODE
1187 004672 004767 003730          JSR      PC,EXEC        ;GO EXECUTE COMMAND
1188 004676 000240          NOP
1189 004700 012767 015601 173770  MOV      #MSFT11,EMADDR
1190 004706 004767 004144          JSR      PC,ERCHK      ;GO CHECK REGISTER
1191 004712 004767 004760          JSR      PC,ITER      ;GO SEE IF ITERATIONS
1192 004716 000167 175600          JMP      TSCD2        ;RETURN TO SCHEDULAR

```

```

1193                                     ;REWIND TEST*****
1194
1195 004722 000240          FT12:  NOP
1196 004724 012767 004722 174030  MOV  #FT12,SCOLP
1197 004732 004767 005004          JSR  PC,INIT1      ;GO INITIALIZE
1198 004736 052777 001700 173666  BIS  #1700,@TC      ;SET TO NRZ,NORMAL
1199 004744 012767 177760 173734  MOV  #-20,FCNT      ;SET FC=20
1200 004752 012767 177770 173730  MOV  #-10,WCNT      ;SET WC=10
1201 004760 012767 016232 173716  MOV  #WDATA,BADDR  ;SET BA=WRITE BUFFER
1202 004766 012767 000007 174002  MOV  #7,FUN        ;SET REWIND OP CODE
1203 004774 004767 003626          JSR  PC,EXEC        ;GO EXECUTE COMMAND
1204 005000 000240
1205 005002 032777 020000 173602  FT12A: BIT  #20000,@DS
1206 005010 001374          BNE  FT12A      ;AWAIT PIP
1207 005012 012767 015621 173656  MOV  #MSFT12,EMADDR
1208 005020 004767 004032          JSR  PC,ERCHK      ;GO CHECK FOR ERROR
1209 005024 004767 004646          JSR  PC,ITER      ;GO SEE IF ITERATION
1210 005030 000167 175466          JMP  TSCD2        ;RETURN TO SCHEDULAR
1211

```

```

1212                                     ;WRITE/READ TEST*****
1213
1214 005034 000240 FT13: NOP
1215 005036 012767 102300 173702 MOV #102300,STMSK ;IGNORE DATA RELATED ERRORS
1216 005044 012767 000001 173652 MOV #1,RDYDX
1217 005052 012767 000001 173646 MOV #1,OPDYX
1218 005060 012767 000100 173624 MOV #100,RCNT ;SET RECORD COUNT
1219 005066 012767 015644 173602 MOV #MSFT13,EMADDR ;SET TEST HEADER
1220 005074 012767 000001 173704 MOV #1,PATRN
1221 005102 004767 004352 JSR PC,DSUP ;SET UP ALL ONES DATA PATTERN
1222 005106 012767 000300 173670 MOV #300,UDES ;REWIND TO BOT
1223 005114 004767 003640 FT13A: JSR PC,RWND ;SET 200 BPI, NORMAL
1224 005120 012767 177600 173560 MOV #-200,FCNT ;SET FC
1225 005126 012767 177700 173554 MOV #-100,WCNT ;SET WC
1226 005134 012767 016232 173542 MOV #WDATA,BADDR ;SET BA
1227 005142 012767 000061 173626 MOV #61,FUN ;SET WRITE OP-CODE
1228 005150 012767 013523 173536 MOV #MSG12,ERRP
1229 005156 004767 003444 FT13B: JSR PC,EXEC ;GO EXECUTE COMMAND
1230 005162 005067 173574 CLR SCOLP ;NO SCOPE LOOP
1231 005166 004767 003664 JSR PC,ERCHK ;GO CHECK ERROR
1232 005172 005367 173514 DEC RCNT ;SEE IF DONE ALL
1233 005176 001367 BNE FT13B ;IF NOT: BR
1234 005200 012767 000100 173504 MOV #100,RCNT ;SET RECORD COUNT
1235 005206 012767 017744 173470 MOV #RDATA,BADDR
1236 005214 062767 000200 173462 ADD #200,BADDR ;SET BA
1237 005222 012767 000077 173546 MOV #77,FUN ;SET READ REVERSE OP-CPDE
1238 005230 012767 013541 173456 MOV #MSG13,ERRP
1239 005236 004767 003364 FT13C: JSR PC,EXEC ;GO EXECUTE COMMAND
1240 005242 004767 003610 JSR PC,ERCHK ;GO CHECK ERROR
1241 005246 005367 173440 DEC RCNT ;SEE IF READ ALL
1242 005252 001371 BNE FT13C ;IF NOT:BR
1243 005254 162767 000200 173422 SUB #200,BADDR ;SET BA
1244 005262 012767 000071 173506 MOV #71,FUN ;SET READ FORWARD OP-CODE
1245 005270 012767 013566 173416 MOV #MSG14,ERRP
1246 005276 012767 000100 173406 MOV #100,RCNT ;SET RECORD COUNT
1247 005304 004767 003316 FT13D: JSR PC,EXEC ;GO EXECUTE COMMAND
1248 005310 004767 003542 JSR PC,ERCHK ;GO CHECK ERRORS
1249 005314 005367 173372 DEC RCNT ;SEE IF DONE ALL
1250 005320 001371 BNE FT13D ;IF NOT:BR
1251 005322 032767 002000 173454 BIT #2000,UDES ;SEE IF DONE PE
1252 005330 001017 BNE FT13X ;IF SO: BR
1253 005332 062767 000400 173444 ADD #400,UDES ;SELECT NEXT DENSITY
1254 005340 032767 002000 173436 BIT #2000,UDES ;SEE IF PE
1255 005346 001403 BEQ FT13E ;IF NOT: BR
1256 005350 005767 173436 TST NRZOF ;SEE IF NRZ ONLY
1257 005354 001005 BNE FT13X ;IF SO: BR
1258 005356 012767 000100 173326 FT13E: MOV #100,RCNT ;RESET RECORD COUNT
1259 005364 000167 177524 JMP FT13A ;GO DO NEXT DENSITY
1260 005370 000167 175126 FT13X: JMP TSCD2 ;RETURN TO SCHEDULAR

```



```

1261                                     ;SPACE TEST*****
1262
1263 005374 000240 FT14: NOP
1264 005376 012767 015673 173272 MOV #MSG46,EMADDR ;SET TEST HEADER
1265 005404 012767 001700 173372 MOV #1700, UDES ;SET NRZ,NORMAL
1266 005412 004767 003342 FT14A1: JSR PC,RWND ;GO INITIALIZE
1267 005416 012767 000100 173266 MOV #100,RCNT ;SET NUMBER OF RECORDER
1268 005424 012767 177777 010600 MOV #-1,WDATA ;SET DATA PATTERN
1269 005432 012767 177700 173246 MOV #-100,FCNT ;PRESET FRAME CNT
1270 005440 012767 177740 173242 MOV #-40,WCNT ;PRESET WORD CNT
1271 005446 004767 004270 FT14A: JSR PC,INIT1 ;GO REWIND
1272 005452 012767 001000 173246 MOV #1000,OPDYX
1273 005460 012767 040000 173236 MOV #40000,RDYDX
1274 005466 012767 000061 173302 MOV #61,FUN ;SET WRITE OP-CODE
1275 005474 012767 102300 173244 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1276 005502 052777 000010 173100 BIS #10,ACS ;INHIBIT BUS ADDRESS INCREMENT
1277 005510 004767 003112 JSR PC,EXEC ;GO EXECUTE COMMAND
1278 005514 000240 NOP
1279 005516 012767 014576 173170 MOV #MSG46,ERRP ;SET ERROR CODE
1280 005524 004767 003326 JSR PC,ERCHK ;GO CHECK ERRORS
1281 005530 005767 173244 TST SERFL ;SEE IF ERROR
1282 005534 001402 BEQ FT14A2 ;IF NOT: BR
1283 005536 000167 000466 JMP FT14X ;ELSE EXIT
1284 005542 162767 000001 173136 FT14A2: SUB #1,FCNT ;BUMP FC
1285 005550 032767 000001 173130 BIT #1,FCNT ;SEE IF SHOULD BUMP WC
1286 005556 001403 BEQ FT14A3 ;IF NOT: BR
1287 005560 162767 000001 173122 SUB #1,WCNT ;BUMP WC
1288 005566 005367 173120 FT14A3: DEC RCNT ;SEE IF DONE ALL
1289 005572 001325 BNE FT14A ;WRITE ALL RECORDS
1290 005574 000240 NOP
1291 005576 012767 000100 173114 MOV #100,RRD ;PRESET RECORD POSITION
1292 005604 012767 000176 173110 MOV #176,RFD
1293 005612 000240 NOP
1294 005614 012767 177701 173106 MOV #-77,SCNT ;SET SPACE AMOUNT
1295 005622 012767 000033 173146 FT14B: MOV #33,FUN ;SET OP-CODE SPACE REVERSE
1296 005630 004767 002772 JSR PC,EXEC ;GO EXECUTE COMMAND
1297 005634 012767 014647 173052 MOV #MSG48,ERRP ;SET ERROR CODE
1298 005642 004767 003210 JSR PC,ERCHK ;GO CHECK ERRORS
1299 005646 005767 173126 TST SERFL ;SEE IF ERROR
1300 005652 001166 BNE FT14X ;IF SO: BR
1301 005654 004767 000070 JSR PC,FT14RR ;GO READ REVERSE + CHECK DATA
1302 005660 000240 NOP
1303 005662 012767 000031 173106 MOV #31,FUN ;SET SPACE FORWARD OP-CODE
1304 005670 005267 173034 INC SCNT ;SET SPACE AMOUNT
1305 005674 001555 BEQ FT14X ;IF DONE: BR
1306 005676 004767 002724 JSR PC,EXEC ;GO EXECUTE COMMAND
1307 005702 012767 014622 173004 MOV #MSG47,ERRP ;SET ERROR CODE
1308 005710 004767 003142 JSR PC,ERCHK ;GO CHECK ERROR
1309 005714 005767 173060 TST SERFL ;SEE IF ERROR FLAG
1310 005720 001143 BNE FT14X ;IF NO: BR
1311 005722 004767 000064 JSR PC,FT14RF ;GO READ FORWARD FOR POSITION CHECK
1312 005726 000240 NOP
1313 005730 005267 172774 INC SCNT ;DECREMENT SPACE AMOUNT
1314 005734 001535 BEQ FT14X ;IF DONE: BR
1315 005736 005267 172756 INC RRD ;BUMP DATA EXPT
1316 005742 005367 172754 DEC RFD ;BUMP DATA EXPT

```

1317	005746	000725			BR	FT14B	
1318	005750	000240			FT14RR: NOP		
1319	005752	012767	017744	172724	MOV	#RDATA,BADDR	;SET BA
1320	005760	012767	000077	173010	MOV	#77,FUN	;SET READ REVERSE OP-CODE
1321	005766	004767	002634		JSR	PC,EXEC	;GO EXECUTE COMMAND
1322	005772	000240			NOP		
1323	005774	016705	172720		MOV	RRD,R5	
1324	006000	020577	172602		CMP	R5,@FC	;SEE IF CORRECT RECORD
1325	006004	001020			BNE	FT14RER	;IF NOT: BR
1326	006006	000167	000026		JMP	FT14EC	;GO CLEAR RH11 ERROR BIT
1327	006012	000240			FT14RF: NOP		
1328	006014	012767	000071	172754	MOV	#71,FUN	;SET READ FORWARD OP-CODE
1329	006022	004767	002600		JSR	PC,EXEC	;GO EXECUTE COMMAND
1330	006026	016705	172670		MOV	RFD,R5	
1331	006032	020577	172550		CMP	R5,@FC	;SEE IF CORRECT RECORD
1332	006036	001003			BNE	FT14RER	;IF NOT: BR
1333	006040	004767	003676		FT14EC: JSR	PC,INIT1	;CLEAR RH
1334	006044	000207			RTS	PC	;RETURN
1335	006046	000240			FT14RER: NOP		
1336	006050	032777	020000	172560	BIT	#20000,@SWR	;SEE IF PRINT INHIBITED
1337	006056	001060			BNE	FT14R3	;IF SO: BR
1338	006060	012704	015673		MOV	#MSFT14,R4	
1339	006064	004767	004276		JSR	PC,TTOUT	;PRINT HEADER
1340	006070	012704	013441		MOV	#MSG9,R4	
1341	006074	004767	004266		JSR	PC,TTOUT	;PRINT ERROR TYPE
1342	006100	012704	013674		MOV	#MSG20,R4	;SET NRZ TAG POINTER
1343	006104	032767	002000	172672	BIT	#2000,UDES	;SEE IF PE
1344	006112	001402			BEQ	FT14R0	;IF NOT: BR
1345	006114	012704	013702		MOV	#MSG21,R4	;ELSE SET PE TAG POINTER
1346	006120	004767	004242		FT14R0: JSR	PC,TTOUT	;PRINT TAG
1347	006124	032767	000002	172644	BIT	#2,FUN	;SEE IF READ REVERSE
1348	006132	001003			BNE	FT14R1	;IF SO: BR
1349	006134	012704	013654		MOV	#MSG17,R4	
1350	006140	000402			BR	FT14R2	;GO PRINT
1351	006142	012704	013634		FT14R1: MOV	#MSG16,R4	
1352	006146	004767	004214		FT14R2: JSR	PC,TTOUT	;PRINT FRWD/REV
1353	006152	012704	013707		MOV	#MSG22,R4	
1354	006156	004767	004204		JSR	PC,TTOUT	;PRINT EXPT TAG
1355	006162	010503			MOV	R5,R3	
1356	006164	042703	177700		BIC	#177700,R3	;MASK RECORD NUMBER
1357	006170	004767	004322		JSR	PC,OCTP	;PRINT EXPT RECORD NUMBER
1358	006174	012704	013717		MOV	#MSG23,R4	
1359	006200	004767	004162		JSR	PC,TTOUT	;PRINT RCVD TAG
1360	006204	017703	172376		MOV	@FC,R3	
1361	006210	042703	177700		BIC	#177700,R3	;MASK RECORD NUMBER
1362	006214	004767	004276		JSR	PC,OCTP	;PRINT ACTUAL RECORD NUMBER
1363	006220	005777	172412		FT14R3: TST	@SWR	;SEE IF HALT ON ERROR
1364	006224	100001			BPL	FT14X	;IF NOT: BR
1365	006226	000000			HALT		
1366	006230	005767	172556		FT14X: TST	NRZOF	;SEE IF NRZ ONLY
1367	006234	001011			BNE	FT14XX	;IF SO: BR
1368	006236	032767	002000	172540	BIT	#2000,UDES	;SEE IF DONE PE
1369	006244	001005			BNE	FT14XX	;IF SO: BR
1370	006246	012767	002300	172530	MOV	#2300,UDES	;SET TO PE
1371	006254	000167	177132		JMP	FT14A1	;DO IN PE
1372	006260	000167	174236		FT14XX: JMP	TSCD2	;RETURN TO SCHEDULAR


```

1373                                     ;ERASE TEST*****
1374
1375 006264 000240                      FT15: NOP
1376 006266 005067 172454              CLR      STMSK
1377 006272 012767 000100 172424      MOV      #100,RDYDX
1378 006300 012767 000010 172420      MOV      #10,OPDYX
1379 006306 012767 015715 172362      MOV      #MSFT15,EMADDR ;SET TEST HEADER
1380 006314 004767 002440              JSR      PC,RWND ;REWIND
1381 006320 012767 017744 172356      MOV      #RDATA,BADDR ;SET BA
1382 006326 012767 001700 172450      MOV      #1700,UDS ;SET NRZ, NORMAL
1383 006334 012767 000025 172434      FT15A: MOV      #25,FUN ;SET ERASE OP-CODE
1384 006342 012767 000200 172342      MOV      #200,RCNT ;SET TO ERASE 128 TIMES
1385 006350 004767 002252              FT15B: JSR      PC,EXEC ;GO EXECUTE COMMAND
1386 006354 012767 014576 172332      MOV      #MSG46,ERRP ;SET ERROR CODE
1387 006362 004767 002470              JSR      PC,ERCHK ;GO CHECK ERRORS
1388 006366 005767 172406              TST      SERFL ;SEE IF ANY ERRORS
1389 006372 001032                      BNE      FT15X ;IF SO EXIT
1390 006374 005367 172312              DEC      RCNT ;SEE IF DONE ERASING
1391 006400 001363                      BNE      FT15B ;IF NOT: BR
1392 006402 000240                      NOP
1393 006404 004767 002350              JSR      PC,RWND ;REWIND
1394 006410 012767 177600 172272      MOV      #-200,WCNT ;SET WC
1395 006416 012767 000071 172352      MOV      #71,FUN ;SET READ FORWARD OP-CODE
1396 006424 012767 000040 172272      MOV      #40,RDYDX ;SET DELAY
1397 006432 004767 002170              JSR      PC,EXEC ;GO EXECUTE COMMAND
1398 006436 000240                      NOP
1399 006440 012767 015172 172246      MOV      #MSG60,ERRP ;SET ERROR CODE
1400 006446 012767 020000 172272      MOV      #20000,STMSK
1401 006454 004767 002376              JSR      PC,ERCHK ;GO CHECK ERRORS
1402 006460 000167 174036              FT15X: JMP      TSCD2 ;RETURN TO SCHEDULAR

```


;TAPE MARK WRITE/READ TEST*****

1403						
1404						
1405	006464	000240			FT16:	NOP
1406	006466	012767	000001	172230		MOV #1,RDYDX
1407	006474	012767	001000	172224		MOV #1000,OPDYX
1408	006502	012767	015737	172166		MOV #MSFT16,EMADDR ;SET HEADER
1409	006510	012767	001700	172266		MOV #1700,UDES ;SET TO NRZ,NORMAL,ODD
1410	006516	004767	002236		FT16A:	JSR PC,RWND ;REWIND
1411	006522	012767	177760	172156	FT16B:	MOV #-20,FCNT ;FC=20
1412	006530	012767	177770	172152		MOV #-10,WCNT ;WC=10
1413	006536	012767	000027	172232		MOV #27,FUN ;SET WRITE TAPE MARK OP-CODE
1414	006544	004767	002056			JSR PC,EXEC ;GO EXECUTE COMMAND
1415	006550	012767	001000	172170		MOV #1000,STMSK ;SET FOR FCE MASK
1416	006556	012767	013523	172130		MOV #MSG12,ERRP ;SET ERROR CODE
1417	006564	004767	002266			JSR PC,ERCHK ;GO CHECK ERROR
1418	006570	004767	002624			JSR PC,TMCHK ;GO SEE IF TM SET
1419	006574	012767	000077	172174		MOV #77,FUN ;SET USED REVERSE OP-CODE
1420	006602	004767	002020			JSR PC,EXEC ;GO EXECUTE COMMAND
1421	006606	012767	001000	172132		MOV #1000,STMSK ;SET FCE ERROR MASK
1422	006614	012767	013541	172072		MOV #MSG13,ERRP ;SET ERROR CODE
1423	006622	004767	002230			JSR PC,ERCHK ;GO CHECK ERRORS
1424	006626	004767	002566			JSR PC,TMCHK ;GO SEE IF TM SET
1425	006632	012767	000071	172136		MOV #71,FUN ;SET READ FORWARD OP-CODE
1426	006640	004767	001762			JSR PC,EXEC ;GO EXECUTE COMMAND
1427	006644	012767	013566	172042		MOV #MSG14,ERRP ;SET ERROR CODE
1428	006652	004767	002200			JSR PC,ERCHK ;TO CHECK ERRORS
1429	006656	004767	002536			JSR PC,TMCHK ;GO SEE IF TM SET
1430	006662	032767	002000	172114		BIT #2000,UDES ;SEE IF DONE PE
1431	006670	001012				BNE FT16X ;IF SO: BR
1432	006672	005767	172114			TST NRZOF ;SEE IF NRZ ONLY
1433	006676	001007				BNE FT16X ;IF SO: BR
1434	006700	012767	002300	172076		MOV #2300,UDES ;SET PE, NORMAL
1435	006706	004767	003030			JSR PC,INIT1 ;INITIALIZE
1436	006712	000167	177604			JMP FT16B ;DO IN PE
1437	006716	004767	002754		FT16X:	JSR PC,ITER ;DO ITERATIONS
1438	006722	000167	173574			JMP TSCD2 ;RETURN TO SCHEDULAR
1439						

```

1440
1441
1442
1443 006726 005067 171760 FT17: CLR RCNT
1444 006732 012767 016000 171736 MOV #MSFT17,EMADDR ;SET HEADER
1445 006740 012767 001700 172036 MOV #1700,UDES ;SET TO NRZ
1446 006746 004767 002006 FT17A: JSR PC,RWND ;REWIND TAPE
1447 006752 012767 000027 172016 FT17B: MOV #27,FUN
1448 006760 012767 040000 171736 MOV #40000,RDIDX ;SET DRY DELAY
1449 006766 012767 040000 171732 MOV #40000,OPDYX ;SET OP DELAY
1450 006774 004767 001626 JSR PC,EXEC ;GO WRITE TM
1451 007000 012767 102300 171740 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1452 007006 012767 013613 171700 MOV #MSG15,ERRP ;SET ERROR TYPE
1453 007014 004767 002036 JSR PC,ERCHK ;GO CHECK ERROR
1454 007020 005767 171754 TST SERFL ;SEE IF ERROR
1455 007024 001144 BNE FT17X ;IF SO: BR
1456 007026 004767 002366 JSR PC,TMCHK ;GO SEE IF TM SET
1457 007032 000240 NOP
1458 007034 000240 NOP
1459 007036 032767 000100 171646 BIT #100,RCNT ;SEE IF DONE PATTERN
1460 007044 001046 BNE FT17D ;IF SO: BR
1461 007046 062767 000020 171636 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
1462 007054 016767 171632 171656 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
1463 007062 012767 177600 171620 MOV #-200,WCNT ;WC=128
1464 007070 012767 177400 171610 MOV #-400,FCNT ;FC=256
1465 007076 012767 016232 171600 MOV #WDATA,BADDR ;BA=WRITE BUFFER
1466 007104 012767 000061 171664 MOV #61,FUN ;SET WRITE OP CODE
1467 007112 000240 FT17C: NOP
1468 007114 000240 NOP
1469 007116 004767 001504 JSR PC,EXEC ;GO WRITE
1470 007122 012767 013523 171564 MOV #MSG12,ERRP ;SET ERROR CODE
1471 007130 012767 102300 171610 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
1472 007136 004767 001714 JSR PC,ERCHK ;GO CHECK ERROR
1473 007142 005767 171632 TST SERFL ;SEE IF ERROR
1474 007146 001073 BNE FT17X ;IF SO: BR
1475 007150 005367 171564 DEC TEMP1 ;SEE IF DONE ALL
1476 007154 001356 BNE FT17C ;IF NOT: BR
1477 007156 000167 177570 JMP FT17B ;ELSE GO DO TM
1478 007162 000240 FT17D: NOP
1479 007164 012767 000033 171604 MOV #33,FUN ;SET SPACE REVERSE
1480 007172 012767 013634 171514 MOV #MSG16,ERRP ;SET ERROR CODE
1481 007200 012767 177600 171522 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
1482 007206 012767 000005 171476 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
1483 007214 004767 002522 FT17E: JSR PC,INIT1 ;GO INIT
1484 007220 004767 001402 JSR PC,EXEC ;GO SPACE
1485 007224 012767 001000 171514 MOV #1000,STMSK ;SET ERROR MASK
1486 007232 004767 001620 JSR PC,ERCHK ;GO CHECK ERROR
1487 007236 005767 171536 TST SERFL ;SEE IF ERROR
1488 007242 001035 BNE FT17X ;IF SO: BR
1489 007244 004767 002150 JSR PC,TMCHK ;GO SEE IF TM SET
1490 007250 005367 171436 DEC RCNT ;SEE IF DONE SPACES
1491 007254 001357 BNE FT17E ;IF NOT: BR
1492 007256 022767 000031 171512 CMP #31,FUN ;SEE IF DONE FORWARD
1493 007264 001410 BEQ FT17F ;IF SO: BR
1494 007266 012767 013654 171420 MOV #MSG17,ERRP ;SET ERROR CODE
1495 007274 012767 000031 171474 MOV #31,FUN ;SET TO SPACE FORWARD

```

1496	007302	000167	177672		JMP	FT17D1	:DO FORWARD	
1497	007306	032767	002000	171470	FT17F:	BIT	#2000,UDES	:SEE IF DONE PE
1498	007314	001010				BNE	FT17X	:IF SO: BR
1499	007316	005767	171470			TST	NRZOF	:SEE IF NRZ ONLT
1500	007322	001005				BNE	FT17X	:IF SO: BR
1501	007324	012767	002300	171452		MOV	#2300,UDES	:SET TO PE
1502	007332	000167	177410			JMP	FT17A	:GO PE
1503	007336	000167	173160		FT17X:	JMP	TSCD2	:RETURN TO SCHEDULAR


```

1504
1505
1506
1507 007342 000240 FT20: NOP
1508 007344 012767 016026 171324 MOV #MSFT20,EMADDR ;SET HEADER
1509 007352 004767 001402 JSR PC,RWND ;REWIND
1510 007356 012767 000003 171422 MOV #3,PATRN
1511 007364 004767 002070 JSR PC,DSUP ;GO SET PATTERN 3
1512 007370 012767 016232 171306 MOV #WDATA,BADDR ;SET BA
1513 007376 012767 177400 171302 MOV #-400,FCNT ;SET FC
1514 007404 012767 177600 171276 MOV #-200,WCNT ;SET WC
1515 007412 012767 001700 171364 MOV #1700,UDES ;SET NRZ NORMAL
1516 007420 012767 000061 171350 FT20A: MOV #61,FUN ;SET WRITE OP CODE
1517 007426 004767 001174 JSR PC,EXEC ;GO WRITE RECORD
1518 007432 012767 014576 171254 MOV #MSG46,ERRP ;SET ERROR CODE
1519 007440 004767 001412 JSR PC,ERCHK ;GO CHECK ERROR
1520 007444 005767 171330 TST SERFL ;SEE IF ERORR
1521 007450 001050 BNE FT20X ;IF SO: BR
1522 007452 012767 013634 171234 MOV #MSG16,ERRP ;SET REVERSE ERROR TAG
1523 007460 012767 000057 171310 MOV #57,FUN ;SET REVERSE WRITE CHECK OP-CODE
1524 007466 062767 000376 171210 ADD #376,BADDR ;SET BA FOR REVERSE CHECK
1525 007474 004767 001126 JSR PC,EXEC ;GO DO REVERSE CHECK
1526 007500 004767 001352 JSR PC,ERCHK ;GO CHECK ERROR
1527 007504 012767 013654 171202 FT20B: MOV #MSG17,ERRP ;SET FORWARD TAG
1528 007512 012767 000051 171256 MOV #51,FUN ;SET FORWARD CHECK OP CODE
1529 007520 162767 000376 171156 SUB #376,BADDR ;SET BA FOR FORWARD CHECK
1530 007526 004767 001074 JSR PC,EXEC ;GO DO FORWARD CHECK
1531 007532 004767 001320 JSR PC,ERCHK ;GO CHECK ERROR
1532 007536 032767 002000 171240 FT20C: BIT #2000,UDES ;SEE IF DONE PE
1533 007544 001012 BNE FT20X ;IF SO: BR
1534 007546 005767 171240 TST NRZOF ;SEE IF NRZ ONLY
1535 007552 001007 BNE FT20X ;IF SO: BR
1536 007554 012767 002300 171222 MOV #2300,UDES ;ELSE SET PE
1537 007562 004767 002154 JSR PC,INIT1 ;GO INIT
1538 007566 000167 177626 JMP FT20A ;DO IN PE
1539 007572 004767 002100 FT20X: JSR PC,ITER ;DO ITERATIONS
1540 007576 000167 172720 JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

1541
1542
1543
1544 007602 012767 016057 171066 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
1545 007610 004767 001144 JSR PC,RWIND ;GO REWIND
1546 007614 012767 000003 171164 MOV #3,PATRN
1547 007622 004767 001632 JSR PC,DSUP ;GO SET PATTERN 3
1548 007626 012767 016232 171050 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1549 007634 012767 176340 171044 MOV #-1440,FCNT ;SET FC=800
1550 007642 012767 177160 171040 MOV #-620,WCNT ;SET WC=400
1551 007650 012767 001700 171126 MOV #1700,UDS ;SET NRZ, NORMAL
1552 007656 012767 000061 171112 MOV #61,FUN ;SET WRITE OP-CODE
1553 007664 004767 000736 JSR PC,EXEC ;GO DO WRITE 1
1554 007670 012767 013523 171016 MOV #MSG12,ERRP ;SET ERROR CODE
1555 007676 004767 001154 JSR PC,ERCHK ;GO CHECK FOR ERROR
1556 007702 004767 000720 JSR PC,EXEC ;YES DO WRITE 2
1557 007706 004767 001144 JSR PC,ERCHK ;YES CHECK FOR ERROR
1558 007712 000240 NOP
1559 007714 004767 001040 JSR PC,RWIND ;GO REWIND
1560 007720 012767 177160 170760 MOV #-620,FCNT ;SET FC=400
1561 007726 012767 177470 170754 MOV #-310,WCNT ;SET WC=200
1562 007734 004767 000666 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
1563 007740 000240 FT21A: NOP
1564 007742 004767 001012 JSR PC,RWIND ;REWIND
1565 007746 012767 017744 170730 MOV #RDATA,BADDR ;SET BA=READ BUFFER
1566 007754 012767 177160 170724 MOV #-620,FCNT ;SET FC=400
1567 007762 012767 177470 170720 MOV #-310,WCNT ;SET WC=200
1568 007770 012767 000071 171000 MOV #71,FUN ;SET READ OP-CODE
1569 007776 004767 000624 JSR PC,EXEC ;GO READ RECORD 1
1570 010002 012767 013566 170704 MOV #MSG14,ERRP ;SET ERROR CODE
1571 010010 004767 001042 JSR PC,ERCHK ;GO CHECK FOR ERROR
1572 010014 000240 NOP
1573 010016 052777 000010 170564 BIS #10,ACS ;INHIBIT BA INCREMENT
1574 010024 012767 176340 170654 MOV #-1440,FCNT ;SET FC=800
1575 010032 012767 177160 170650 MOV #-620,WCNT ;SET WC=400
1576 010040 004767 000562 JSR PC,EXEC ;GO READ RECORD 2
1577 010044 022777 001440 170534 CMP #1440,AFc ;SEE IF READ RECORD 2
1578 010052 001410 BEQ FT21X ;IF SO: BR
1579 010054 012767 014474 170654 MOV #MSG44,ERADD ;SET ERROR CODE
1580 010062 012767 007740 170672 MOV #FT21A,SCOLP ;SET SCOPE ADDRESS
1581 010070 004767 173364 JSR PC,FT3ER ;GO PRINT ERROR
1582 010074 004767 001576 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
1583 010100 000167 172416 JMP TSCD2 ;RETURN TO SCHEDULAR
1584
1585

```

```

1586                                     ;BUFFERED COMMAND TEST*****
1587
1588 010104 012767 016106 170564 FT22: MOV #MSFT22,EMADDR ;SET TEST HEADER
1589 010112 004767 000642 JSR PC,RWND ;GO REWIND
1590 010116 012700 000003 MOV #3,R0 ;SET NUMBER OF WRITES
1591 010122 012767 001700 170654 MOV #1700,UDES ;SET TO NRZ NORMAL
1592 010130 012767 016232 170546 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
1593 010136 012767 177000 170542 MOV #-1000,FCNT ;SET FC=1000
1594 010144 012767 177400 170536 MOV #-400,WCNT ;SET WC=400
1595 010152 012767 000061 170616 MOV #61,FUN ;SET WRITE OP-CODE
1596 010160 004767 000442 FT22A: JSR PC,EXEC ;GO DO WRITE
1597 010164 005300 DEC R0 ;SEE IF DONE ALL
1598 010166 001374 BNE FT22A ;IF NOT: BR
1599 010170 000240 NOP
1600 010172 012777 000007 170400 MOV #7,AC1 ;START REWIND
1601 010200 032777 000200 170404 FT22B: BIT #200,ADS
1602 010206 001774 BEQ FT22B
1603 010210 012767 000010 170506 MOV #10,RDYDX ;SET LONG READY DELAY
1604 010216 004767 000404 JSR PC,EXEC ;ISSUE BUFFERED WRITE
1605 010222 000240 NOP
1606 010224 012767 014674 170462 MOV #MSG49,ERRP ;SET ERROR CODE
1607 010232 012767 102300 170506 MOV #102300,STMSK ;MARK DATA ERROR
1608 010240 004767 000612 JSR PC,ERCHK ;GO CHECK ERROR
1609 010244 032777 000002 170340 BIT #2,ADS ;SEE IF BOT IS SET
1610 010252 001410 BEQ FT22X ;IF NOT: BR
1611 010254 012767 014722 170454 MOV #MSG50,ERADD ;SET ERROR CODE
1612 010262 012767 010104 170472 MOV #FT22,SCOLP
1613 010270 004767 173164 JSR PC,FT3ER ;GO DO ERROR
1614 010274 004767 001376 FT22X: JSR PC,ITER ;GO SEE IF ITERATION
1615 010300 000167 172216 JMP TSCD2 ;RETURN TO SCHEDULAR
1616
1617

```



```

1618                                     ;READ-IN PRESET TEST*****
1619
1620 010304 005767 170372          FT23: TST      SLVN          ;SEE IF SLAVE SELECT=0
1621 010310 001103                BNE      FT23X          ;IF NOT:BR
1622 010312 012767 016143 170356    MOV      #MSFT23,EMADDR ;SET TEST HEADER
1623 010320 004767 001416                JSR      PC,INIT1    ;GO INIT
1624 010324 012767 001700 170452    MOV      #1700,UDES      ;SET TO NRZ NORMAL
1625 010332 012767 016232 170344    MOV      #WDATA,BADDR   ;SET BA=WRITE BUFFER
1626 010340 012767 177400 170340    MOV      #-400,FCNT     ;SET FC=400
1627 010346 012767 177600 170334    MOV      #-200,WCNT     ;SET WC=200
1628 010354 012767 000061 170414    MOV      #61,FUN       ;SET WRITE OP-CODE
1629 010362 004767 000240                JSR      PC,EXEC      ;GO DO WRITE
1630 010366 000240                NOP
1631 010370 004767 001346                JSR      PC,INIT1    ;INITIALIZE
1632 010374 012767 000021 170374    MOV      #21,FUN       ;SET READ-IN PRESET OP CODE
1633 010402 004767 000220                JSR      PC,EXEC      ;GO DO COMMAND
1634 010406 005000                CLR      R0
1635 010410 012703 000004                MOV      #4,R3        ;SET MULT
1636 010414 032777 020000 170170    FT23A: BIT      #20000,ADS ;SEE IF PIP RESET
1637 010422 001404                BEQ      FT23B          ;IF SO: BR
1638 010424 005300                DEC      R0
1639 010426 001372                BNE      FT23A          ;AWAIT PIP RESET
1640 010430 005303                DEC      R3
1641 010432 001370                BNE      FT23A          ;DELAY
1642 010434 032777 000002 170150    FT23B: BIT      #2,ADS    ;SEE IF BOT
1643 010442 001010                BNE      FT23C          ;IF SO: BR
1644 010444 012767 014760 170264    MOV      #MSG51,ERADD  ;SET ERROR CODE
1645 010452 012767 010304 170302    MOV      #FT23,SCOLP
1646 010460 004767 172774                JSR      PC,FT3ER     ;GO DO ERROR
1647 010464 012701 141000          FT23C: MOV      #141000,R1 ;SET EXPT TC
1648 010470 016700 170136                MOV      TC,R0       ;SET TC ADDRESS
1649 010474 020110                CMP      R1,(R0)     ;SEE IF EXPT=RCVD
1650 010476 001410                BEQ      FT23X          ;IF SO: BR
1651 010500 012767 015014 170230    MOV      #MSG52,ERADD  ;SET ERROR CODE
1652 010506 012767 010304 170246    MOV      #FT23,SCOLP  ;CLEAR SCOPE ADDRESS
1653 010514 004767 172450                JSR      PC,FT2ER     ;GO DO ERROR
1654 010520 000167 171776          FT23X: JMP      TSCD2    ;RETURN TO SCHEDULAR
1655
1656

```

```

1657                                     ;REWIND: OFF LINE TEST*****
1658
1659 010524 032777 004000 170104 FT24: BIT #4000,@SWR ;SEE IF IN CONTINUOUS MODE
1660 010532 001033 BNE FT24XX ;IF SO: BR
1661 010534 012767 016176 170134 MOV #MSGFT24,EMADDR ;SET TEST HEADER
1662 010542 004767 001174 JSR PC,INIT1 ;GO INITIAIZE
1663 010546 012777 000003 170024 MOV #3,@C1 ;ISSUE REWIND: OFF LINE COMMAND
1664 010554 012700 004000 MOV #4000,R0
1665 010560 005300 FT24A: DEC R0 ;DELAY
1666 010562 001376 BNE FT24A
1667 010564 032777 010000 170020 BIT #10000,@DS ;SEE IF MOL IS RESET
1668 010572 001407 BEQ FT24X ;IF SO: BR
1669 010574 005067 170162 CLR SCOLP ;ASSURE NO SCOPE
1670 010600 012767 015033 170130 MOV #MSG53,ERADD ;SET ERROR CODE
1671 010606 004767 172646 JSR PC,FT3ER ;GO DO ERROR
1672 010612 012704 015057 FT24X: MOV #MSG54,R4
1673 010616 004767 001544 JSR PC,TTOUT ;PRINT ON LINE REQUEST
1674 010622 000167 171674 FT24XX: JMP TSCD2 ;RETURN TO SCHEDULAR
1675
1676
  
```

```

1677                                     ;COMMAND EXECUTE SUBROUTINE*****
1678
1679 010626 000240                               EXEC:  NOP
1680 010630 056777 170150 167774                BIS    UDES,@TC          ;LOAD TAPE CONT
1681 010636 016777 170046 167736                MOV    WCNT,@WC          ;LOAD WC
1682 010644 016777 170036 167734                MOV    FCNT,@FC          ;LOAD FC
1683 010652 016777 170026 167724                MOV    BADDR,@BA        ;LOAD BA
1684 010660 022767 000031 170110                CMP    #31,FUN          ;SEE IF SPACE FORWARD
1685 010666 001404                               BEQ    EXECA              ;IF SO: BR
1686 010670 022767 000033 170100                CMP    #33,FUN          ;SEE IF SPACE REVERSE
1687 010676 001003                               BNE    EXECB              ;IF NOT: BR
1688 010700 016777 170024 167700                EXECA:  MOV    SCNT,@FC        ;SET SPACE COUNT
1689 010706 000240                               EXECB:  NOP
1690 010710 016777 170062 167662                MOV    FUN,@C1          ;LOAD OP-CODE + GO
1691 010716 000240                               NOP
1692 010720 016703 170000                MOV    RDYDX,R3          ;SET DELAY
1693 010724 005004                               CLR    R4
1694 010726 032777 000200 167656                EXECC:  BIT    #200,@DS        ;SEE IF DRY
1695 010734 001004                               BNE    EXECX              ;IF SO: BR
1696 010736 005304                               DEC    R4
1697 010740 001372                               BNE    EXECC
1698 010742 005303                               DEC    R3                ;DELAY FOR DRY
1699 010744 001370                               BNE    EXECC
1700 010746 016703 167754                EXECX:  MOV    OPDYX,R3
1701 010752 005303                EXECXA:  DEC    R3                ;DELAY
1702 010754 001376                               BNE    EXECXA
1703 010756 000207                EXECXX:  RTS    PC                ;RETURN TO CALLER
1704

```



```

1705                                     ;REWIND SUBROUTINE*****
1706
1707 010760 000240          RWND:  NOP
1708 010762 004767 000754    JSR    PC,INIT1          ;INIT
1709 010766 012777 000007 167604  MOV    #7,@C1          ;START REWIND
1710 010774 012700 040000    MOV    #40000,R0
1711 011000 005300          RWNDA: DEC    R0
1712 011002 001376          BNE    RWNDA          ;DELAY
1713 011004 032777 020000 167600  RWNDB: BIT    #20000,@DS
1714 011012 001374          BNE    RWNDB          ;AWAIT PIP
1715 011014 032777 000002 167570  BIT    #2,@DS        ;SEE IF BOT
1716 011022 001012          BNE    RWNDX          ;IF SO: BR
1717 011024 016704 167646    MOV    EMADDR,R4
1718 011030 004767 001332    JSR    PC,TTOUT          ;PRINT HEADER
1719 011034 012704 013234    MOV    #MSG2,R4
1720 011040 004767 001322    JSR    PC,TTOUT          ;PRINT REWIND ERROR
1721 011044 000167 171452    JMP     TSCD2          ;RETURN TO SECHEDULAR
1722 011050 004767 000666    RWNDX: JSR    PC,INIT1      ;INIT
1723 011054 000207          RTS     PC                ;RETURN TO CALLER
1724

```

```

1725                                     ;ERROR CHECK SUBROUTINE*****
1726
1727 011056 005067 167716 ERCHK: CLR SERFL ;CLEAR FLAG
1728 011062 017767 167524 167662 MOV @DS,DSAV ;SAVE DRIVE STATUS REGISTER
1729 011070 032777 040000 167514 BIT #40000,@DS ;SEE IF ERROR
1730 011076 001001 BNE ERPT ;IF SO: BR
1731 011100 000207 RTS PC ;RETURN
1732 011102 017704 167506 ERPT: MOV @ER,R4 ;GET ERROR REGISTER
1733 011106 032767 002000 167670 BIT #2000,UDES ;SEE IF PE
1734 011114 001403 BEQ ERPTA1 ;IF SO: BR
1735 011116 042767 000200 167622 BIC #200,STMSK ;RESET PEF MASK
1736 011124 046704 167616 ERPTA1: BIC STMSK,R4 ;MASK DONT CARE BITS
1737 011130 001530 BEQ ERPTX ;IF NO UNEXPECTED ERRORS: BR
1738 011132 012767 000001 167640 ERPTG: MOV #1,SERFL ;SET FLAG
1739 011140 032777 020000 167470 BIT #20000,@SWR ;SEE IF SHOULD PRINT ERRORS
1740 011146 001115 BNE ERPTD ;IF NOT: BR
1741 011150 005767 167520 TST HDRFL ;SEE IF DONE HEADER
1742 011154 001006 BNE ERPTA ;IF SO: BR
1743 011156 005267 167512 INC HDRFL ;SET HEADER FLAG
1744 011162 016704 167510 MOV EMADDR,R4
1745 011166 004767 001174 JSR PC,TTOUT ;PRINT HEADER
1746 011172 016704 167516 ERPTA: MOV ERRP,R4 ;GET ERROR CODE
1747 011176 001414 BEQ ERPTB ;IF NONE: BR
1748 011200 004767 001162 JSR PC,TTOUT ;PRINT ERROR CODE
1749 011204 012704 013674 MOV #MSG20,R4 ;SET NRZ TAG
1750 011210 032777 002000 167414 BIT #2000,@TC ;SEE IF PE
1751 011216 001402 BEQ ERPT1A ;IF NOT: BR
1752 011220 012704 013702 MOV #MSG21,R4 ;ELSE SET PE TAG
1753 011224 004767 001136 ERPT1A: JSR PC,TTOUT ;PRINT TAG
1754 011230 016704 167462 ERPTB: MOV ERRP1,R4 ;SEE IF CODE 2
1755 011234 001402 BEQ ERPTB1 ;IF NOT: BR
1756 011236 004767 001124 JSR PC,TTOUT ;PRINT CODE 2
1757 011242 032777 010000 167366 ERPTB1: BIT #10000,@SWR ;SEE IF ITERATION
1758 011250 001010 BNE ERPTC ;IF NOT: BR
1759 011252 012704 015146 MOV #MSG56,R4
1760 011256 004767 001104 JSR PC,TTOUT ;PRINT ITER TAG
1761 011262 016703 167462 MOV ITCNT,R3
1762 011266 004767 001224 JSR PC,OCTP ;PRINT ITERATION
1763 011272 012704 013146 ERPTC: MOV #MSG1,R4 ;PRINT REGISTER TAG
1764 011276 004767 001064 JSR PC,TTOUT
1765 011302 017703 167272 MOV @C1,R3 ;PRINT CS1
1766 011306 004767 001172 JSR PC,OCTPE
1767 011312 017703 167264 MOV @WC,R3 ;PRINT WC
1768 011316 004767 001162 JSR PC,OCTPE
1769 011322 017703 167256 MOV @BA,R3 ;PRINT BA
1770 011326 004767 001152 JSR PC,OCTPE
1771 011332 017703 167250 MOV @FC,R3 ;PRINT FC
1772 011336 004767 001142 JSR PC,OCTPE
1773 011342 017703 167242 MOV @CS,R3 ;PRINT CS2
1774 011346 004767 001132 JSR PC,OCTPE
1775 011352 017703 167234 MOV @DS,R3 ;PRINT DS
1776 011356 004767 001122 JSR PC,OCTPE
1777 011362 017703 167226 MOV @ER,R3 ;PRINT ER
1778 011366 004767 001112 JSR PC,OCTPE
1779 011372 017703 167234 MOV @TC,R3 ;PRINT TC
1780 011376 004767 001102 JSR PC,OCTPE

```

1781 011402 005777 167230
1782 011406 100001
1783 011410 000000
1784 011412 004767 000324
1785 011416 000207
1786
1787

ERPTD: TST @SWR ;SEE IF HALT ON ERROR
BPL ERPTX ;IF NOT: BR
HALT
ERPTX: JSR PC,INIT1 ;INIT
ERPTXX: RTS PC ;RETURN


```

1788                                     ;TAPE MARK STATUS CHECK*****
1789
1790 011420 032767 000004 167324 TMCHK: BIT #4,DSAV ;SEE IF TM SET
1791 011426 001401 BEQ TMCHK1 ;IF NOT: BR
1792 011430 000207 TMCHK0: RTS PC ;ELSE RETURN
1793 011432 005767 167342 TMCHK1: TST SERFL ;SEE IF HAD ERROR
1794 011436 001374 BNE TMCHK0 ;IF SO: BR
1795 011440 012767 015156 167250 MOV #MSG57,ERRP1 ;SET ERROR CODE 2
1796 011446 004767 177460 JSR PC,ERPTG ;GO PRINT TM ERROR
1797 011452 005067 167240 CLR ERRP1 ;CLEAR CODE 2 FLAG
1798 011456 000207 RTS PC ;RETURN
1799
1800                                     ;DATA SETUP ROUTINE*****
1801
1802 011460 000240 DSUP: NOP
1803 011462 012703 016232 DS0: MOV #WDATA,R3 ;R3 = ADDRS OF WRITE BUFFER
1804 011466 016701 167314 MOV PATRN,R1 ;R1 = PATTERN SELECTOR
1805 011472 000241 CLC
1806 011474 006101 ROL R1 ;MAKE PATTERN SELECTOR EVEN
1807 011476 000171 001020 JMP @DATBL(R1) ;GO GENERATE PATTERN
1808 011502 032777 010000 167116 DS1: BIT #10000,@DT ;SEE IF SEVEN TRACK
1809 011510 001410 BEQ DS3 ;IF NOT: BR
1810 011512 012702 000640 MOV #640,R2 ;SET BUFFER SIZE
1811 011516 012701 016232 MOV #WDATA,R1 ;SET START OF BUFFER
1812 011522 042721 140300 DS2: BIC #140300,(R1)+ ;MASK FOR 7 CH
1813 011526 005302 DEC R2 ;SEE IF DONE
1814 011530 001374 BNE DS2 ;IF NOT: BR
1815 011532 012702 000640 DS3: MOV #640,R2 ;R2=BUFFER SIZE +2
1816 011536 012701 017744 MOV #RDATA,R1 ;R1=READ DATA START
1817 011542 005021 DS4: CLR (R1)+ ;CLEAR BUFFER
1818 011544 005302 DEC R2 ;SEE IF DONE ALL
1819 011546 001375 BNE DS4 ;IF NOT: BR
1820 011550 000207 RTS PC ;EXIT
1821
1822                                     ;ALL ONES*****
1823
1824 011552 012701 177777 DAT1: MOV #-1,R1 ;R1=DATA
1825 011556 012702 000640 DAT1A: MOV #640,R2 ;R2=WORD COUNT +2
1826 011562 010123 DAT1B: MOV R1,(R3)+ ;LOAD BUFFER
1827 011564 005302 DEC R2 ;SEE IF DONE
1828 011566 001375 BNE DAT1B ;IF NOT: BR
1829 011570 000167 177706 JMP DS1 ;RETURN
1830
1831                                     ;ALL ZEROS*****
1832
1833 011574 005001 DAT2: CLR R1 ;R1=DATA
1834 011576 000167 177754 JMP DAT1A ;LOAD BUFFER
1835

```

Address	Hex	Hex	Hex	Hex	Label	Instruction	Comment
1836						;ONE/ZERO IN ALTERNATING CHARACTERS*****	
1837							
1838	011602	012701	125125		DATA3:	MOV #125125,R1	;R1=DATA
1839	011606	000167	177744			JMP DATA1A	;LOAD BUFFER
1840							
1841						;ALL BITS 0-377*****	
1842							
1843	011612	005001			DATA4:	CLR R1	;R1=STARTING DATA
1844	011614	012702	001500			MOV #1500,R2	;R2=CHARACTER COUNT
1845	011620	110123			DATA4A:	MOVB R1,(R3)+	;LOAD BUFFER
1846	011622	105201				INCB R1	;BUMP DATA
1847	011624	005302				DEC R2	;SEE IF DONE
1848	011626	001374				BNE DATA4A	;IF NOT: BR
1849	011630	000167	177646			JMP DS1	;RETURN
1850							
1851						;SCOPE LOOP ON ERROR SUBROUTINE*****	
1852							
1853							
1854	011634	000240			SCOPE:	NOP	
1855	011636	032777	040000	166772		BIT #40000,@SWR	;SEE IF LOOP ON ERROR
1856	011644	001001				BNE SCOPE1	;IF SO: BR
1857	011646	000207				RTS PC	;ELSE EXIT
1858	011650	000240			SCOPE1:	NOP	
1859	011652	005767	167104			TST SCOLP	;SEE IF SCOPE ADDRESS
1860	011656	001001				BNE SCOPE2	;IF NOT: BR
1861	011660	000207				RTS PC	;ELSE EXIT
1862	011662	005726			SCOPE2:	TST (SP)+	;RESET STACK
1863	011664	005726				TST (SP)+	;BUMP STACK
1864	011666	000240				NOP	
1865	011670	000240				NOP	
1866	011672	000177	167064			JMP @SCOLP	;LOOP ON ERROR
1867							
1868						;TEST ITERATION SUBROUTINE*****	
1869							
1870	011676	000240			ITER:	NOP	
1871	011700	032777	010000	166730		BIT #10000,@SWR	;SEE IF ITERATIONS
1872	011706	001403				BEQ ITER1	;IF SO: BR
1873	011710	005067	167034		ITER0:	CLR ITCNT	;CLEAR ITERATION COUNTER
1874	011714	000207				RTS PC	;ELSE EXIT
1875	011716	005267	167026		ITER1:	INC ITCNT	;BUMP COUNTER
1876	011722	026767	167022	166724		CMP ITCNT,ITAMT	;SEE IF DONE ALL
1877	011730	001767				BEQ ITER0	;IF SO: BR
1878	011732	005726				TST (SP)+	;RESET STACK
1879	011734	017700	167024			MOV @ITRLP,R0	;SET ITERATION POINTER
1880	011740	000110				JMP (R0)	;GO ITERATE
1881							
1882						;INITIALIZE SUBROUTINE*****	
1883							
1884	011742	000240			INIT1:	NOP	
1885	011744	012777	000040	166636		MOV #40,@CS	;INIT
1886	011752	016777	166722	166630	INIT2:	MOV DRVN,@CS	;SELECT DRIVE
1887	011760	016777	166716	166644		MOV SLVN,@TC	;SELECT SLAVE
1888	011766	000207				RTS PC	;RETURN
1889							

```

1890                                     ;MAG TAPE INTERRUPT HANDLER*****
1891
1892 011770 000240      MTINT: NOP
1893 011772 022626      CMP      (SP)+,(SP)+      ;RESET STACK POINTER
1894 011774 000240      NOP
1895 011776 000240      NOP
1896 012000 000177 166730      JMP      @RTN      ;RETURN TO CALLER
1897
1898                                     ;TTY INTERRUPT HANDLER*****
1899
1900 012004 000240      TTINT: NOP
1901 012006 000240      NOP
1902 012010 000240      NOP
1903 012012 000002      RTI
1904
1905                                     ;BUS ADDRESS TRAP HANDLER*****
1906
1907 012014 000240      TRAP:  NOP
1908 012016 032777 020000 166612      BIT      #20000,@SWR      ;SEE IF SHOULD PRINT ERRORS
1909 012024 001020      BNE      TRAP2      ;IF NOT: BR
1910 012026 005767 166642      TST      HDRFL      ;SEE IF DONE HEADER
1911 012032 001006      BNE      TRAP1      ;IF SO: BR
1912 012034 005267 166634      INC      HDRFL      ;ELSE SET HEADER FLAG
1913 012040 016704 166632      MOV      EMADDR,R4
1914 012044 004767 000316      JSR      PC,TTOUT      ;PRINT HEADER
1915 012050 012704 013727      TRAP1:  MOV      #MSG24,R4
1916 012054 004767 000306      JSR      PC,TTOUT      ;PRINT ERROR
1917 012060 010103      MOV      R1,R3
1918 012062 004767 000430      JSR      PC,OCTP      ;PRINT ADDRESS OF TRAP
1919 012066 005777 166544      TRAP2:  TST      @SWR      ;SEE IF HALT ON ERROR
1920 012072 100001      BPL      TRAPX      ;IF NOT: BR
1921 012074 000000      HALT
1922 012076 022626      TRAPX:  CMP      (SP)+,(SP)+      ;RESET STACK
1923 012100 012767 002732 166654      MOV      #FT1A,SCOLP      ;SET SCOPE ADDRESS
1924 012106 004767 177522      JSR      PC,SCOPE      ;GO SEE IF SCOPE LOOP
1925 012112 005767 166672      TST      RHTF      ;SEE IF INITIAL ADDRESS TEST
1926 012116 001402      BEQ      TRAPXX      ;IF NOT: BR
1927 012120 000167 167510      JMP      STOB      ;ELSE REDO ADDRESS REQUEST
1928 012124 000167 170606      TRAPXX: JMP      FT1B      ;RETURN TO TEST 1
1929

```



```

1930 ;*****
1931 ;TTY ENTRY SUBROUTINE:
1932 ;
1933 ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
1934 ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
1935 ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
1936 ;LIMITS. ALL RESPONSE MUST BE TYPED IN OCTAL
1937 ; (0-7) AND MUST FALL WITHIN THE LIMITS SET BY
1938 ;THE CALLING ROUTINE.
1939 ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
1940 ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
1941 ;MAY BE REENTERED.
1942 ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
1943 ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
1944 ;CARRIAGE RETURN
1945 ;*****
1946
1947 012130 005067 166604 TTR: CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
1948 012134 005000 CLR RO
1949 012136 004767 000152 TTR0: JSR PC,TTIN ;GO READ CHARACTER
1950 012142 122767 000215 166520 CMPB #215,TIB ;SEE IF CR
1951 012150 001005 BNE TTR1 ;IF NOT: BR
1952 012152 005767 166562 TST TEMP1 ;SEE IF FIRST CHARACTER
1953 012156 001446 BEQ TTR5 ;IF SO: BR
1954 012160 000167 000066 JMP TTR2 ;ELSE GO LOAD VALUE
1955 012164 122767 000260 166476 TTR1: CMPB #260,TIB ;SEE IF CHAR IS LESS THAN 0
1956 012172 101402 BLOS TTR1A ;IF NOT: BR
1957 012174 000167 000076 JMP TTR1A ;ELSE GO TO ERROR
1958 012200 122767 000270 166462 TTR1A: CMPB #270,TIB ;SEE IF CHAR IS GREATER THAN 7
1959 012206 101002 BHI TTR1B ;IF NOT: BR
1960 012210 000167 000062 JMP TTR1B ;ELSE GO TO ERROR
1961 012214 005267 166520 TTR1B: INC TEMP1 ;SET FIRST CHARACTER FLAG
1962 012220 000241 CLC
1963 012222 006100 ROL RO
1964 012224 000241 CLC
1965 012226 006100 ROL RO ;SHIFT 3 LEFT
1966 012230 000241 CLC
1967 012232 006100 ROL RO
1968 012234 042767 177770 166426 BIC #177770,TIB ;STRIP ASCII
1969 012242 056700 166422 BIS TIB,RO ;LOAD CHARACTER
1970 012246 005301 DEC R1 ;SEE IF DONE
1971 012250 001332 BNE TTR0 ;IF NOT: BR
1972 012252 020002 TTR2: CMP RO,R2 ;SEE IF EXCEEDED MAXIMUM LIMIT
1973 012254 101402 BLOS TTR3 ;IF NOT: BR
1974 012256 000167 000014 JMP TTR3 ;ELSE GO TO ERROR
1975 012262 020300 TTR3: CMP R3,RO ;SEE IF BELOW MINIMUM LIMIT
1976 012264 101402 BLOS TTR4 ;IF NOT: BR
1977 012266 000167 000004 JMP TTR4 ;ELSE GO TO ERROR
1978 012272 010015 TTR4: MOV RO,(R5) ;LOAD VALUE
1979 012274 000207 TTR5: RTS PC ;EXIT
1980

```

```

1981                                     ;TTY ENTRY ERROR SUBROUTINE*****
1982
1983 012276 012704 013433          T1NER: MOV    #MSG7,R4
1984 012302 004767 000060          JSR     PC,TTOUT          ;PRINT?
1985 012306 162716 000020          SUB     #20,(SP)          ;RESET SP TO START OF VALUE ROUTINE
1986 012312 000207                  RTS      PC              ;REDO VALUE ENTRY
1987
1988                                     ;TTY READ SUBROUTINE*****
1989
1990 012314 005077 166320          TTIN:  CLR     @TKS
1991 012320 005077 166316          CLR     @TKB
1992 012324 005067 166340          CLR     T1B
1993 012330 005277 166304          INC      @TKS
1994 012334 105777 166300          TTIN1: TSTB   @TKS
1995 012340 100375                  BPL      TTIN1
1996 012342 017767 166274 166320          MOV    @TKB,T1B
1997 012350 105777 166270          TTIN2: TSTB   @TPS
1998 012354 100375                  BPL      TTIN2
1999 012356 116777 166306 166262          MOVB   T1B,@TPB
2000 012364 000207                  RTS      PC
2001
2002                                     ;TTY OUTPUT SUBROUTINE*****
2003
2004 012366 112467 166274          TTOUT: MOVB   (R4)+,TOB
2005 012372 122767 000043 166266          CMPB   #43,TOB
2006 012400 001440                  BEQ      TEX
2007 012402 122767 000045 166256          CMPB   #45,TOB
2008 012410 001403                  BEQ      TCRLF
2009 012412 004767 000050          JSR      PC,TOG
2010 012416 000763                  BR       TTOUT
2011 012420 112767 000015 166240          TCRLF: MOVB   #15,TOB
2012 012426 004767 000034          JSR      PC,TOG
2013 012432 012703 000004          MOV      #4,R3
2014 012436 005067 166224          TCRLFA: CLR     TOB
2015 012442 004767 000020          JSR      PC,TOG
2016 012446 005303                  DEC      R3
2017 012450 001372                  BNE      TCRLFA
2018 012452 112767 000012 166206          MOVB   #12,TOB
2019 012460 004767 000002          JSR      PC,TOG
2020 012464 000740                  BR       TTOUT
2021 012466 105777 166152          TOG:   TSTB   @TPS
2022 012472 100375                  BPL      TOG
2023 012474 116777 166166 166144          MOVB   TOB,@TPB
2024 012502 000207                  TEX:   RTS      PC
2025
2026

```

;DO FILLERS

```

2027                                     ;OCTAL OUTPUT SUBROUTINE*****
2028
2029 012504 012767 000001 000222 OCTPE: MOV #1,OFL
2030 012512 010304          MOV R3,R4
2031 012514 000410          BR OCTP0
2032 012516 005067 000212 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
2033 012522 010304          OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
2034 012524 001004          BNE OCTP0 ;IF NOT ZERO: BR
2035 012526 004767 000162          JSR PC,OCTPG1 ;ELSE PRINT ZERO
2036 012532 000167 000120          JMP OCTP3 ;SPACE AND EXIT
2037 012536 032704 100000          OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
2038 012542 001406          BEQ OCTP1 ;IF NOT: BR
2039 012544 012704 000001          MOV #1,R4
2040 012550 004767 000116          JSR PC,OCTPG ;PRINT 1
2041 012554 000167 000006          JMP OCTP2
2042 012560 005004          OCTP1: CLR R4
2043 012562 004767 000104          JSR PC,OCTPG ;PRINT 0
2044 012566 010304          OCTP2: MOV R3,R4
2045 012570 006004          ROR R4
2046 012572 006004          ROR R4
2047 012574 006004          ROR R4 ;POSITION DIGIT
2048 012576 006004          ROR R4
2049 012600 000304          SWAB R4
2050 012602 004767 000064          JSR PC,OCTPG ;PRINT DIGIT 2
2051 012606 010304          MOV R3,R4
2052 012610 006004          ROR R4
2053 012612 000304          SWAB R4
2054 012614 004767 000052          JSR PC,OCTPG ;PRINT DIGIT 3
2055 012620 010304          MOV R3,R4
2056 012622 006104          ROL R4
2057 012624 006104          ROL R4
2058 012626 000304          SWAB R4
2059 012630 004767 000036          JSR PC,OCTPG ;PRINT DIGIT 4
2060 012634 010304          MOV R3,R4
2061 012636 006004          ROR R4
2062 012640 006004          ROR R4
2063 012642 006004          ROR R4
2064 012644 004767 000022          JSR PC,OCTPG
2065 012650 010304          MOV R3,R4
2066 012652 004767 000014          JSR PC,OCTPG ;PRINT DIGIT 5
2067 012656 012767 000240 166002 OCTP3: MOV #240,TOB
2068 012664 004767 177576          JSR PC,TOG ;PRINT SPACE
2069 012670 000207          RTS PC ;EXIT
2070 012672 042704 177770          OCTPG: BIC #177770,R4
2071 012676 001004          BNE OCTPG0
2072 012700 005767 000030          TST OFL
2073 012704 001001          BNE OCTPG0
2074 012706 000207          RTS PC
2075 012710 005267 000020          OCTPG0: INC OFL
2076 012714 052704 000260          OCTPG1: BIS #260,R4
2077 012720 010467 165742          MOV R4,TOB
2078 012724 004767 177536          JSR PC,TOG
2079 012730 010304          MOV R3,R4
2080 012732 000207          RTS PC
2081 012734 000000          OFL: 0 ;FIRST CHAR FLAG
2082

```



```

2083                                     ;DATA CHARACTER OUTPUT SUBROUTINE*****
2084
2085 012736 005067 165724          DOUT: CLR      TOB
2086 012742 012704 000010          MOV      #10,R4          ;SET NUMBER TO PRINT
2087 012746 110367 165714          MOV      R3,TOB
2088 012752 105777 165666          DOUT1: TSTB     @TPS
2089 012756 100375                BPL      DOUT1
2090 012760 132767 000200 165700  BITB     #200,TOB
2091 012766 001404                BEQ      DOUT2
2092 012770 012777 000061 165650  MOV      #061,@TPB
2093 012776 000403                BR       DOUT3
2094 013000 012777 000060 165640  DOUT2: MOV      #060,@TPB
2095 013006 006167 165654          DOUT3: ROL      TOB
2096 013012 005304                DEC      R4
2097 013014 001356                BNE      DOUT1
2098 013016 000207                RTS      PC
2099 013020 016703 165720          DOUTD: MOV      TEMP3,R3
2100 013024 000303                SWAB     R3
2101 013026 004767 177704          JSR      PC,DOUT
2102 013032 016703 165706          MOV      TEMP3,R3
2103 013036 004767 177674          JSR      PC,DOUT
2104 013042 000207                RTS      PC
2105
2106                                     ;TU45 SERIAL NUMBER PRINT SUBROUTINE*****
2107
2108 013044 010304          SNPT:  MOV      R3,R4
2109 013046 000304          SWAB     R4
2110 013050 006004          ROR      R4
2111 013052 006004          ROR      R4
2112 013054 006004          ROR      R4
2113 013056 006004          ROR      R4
2114 013060 004767 000036          JSR      PC,SNPG          ;GET FIRST DIGIT
2115 013064 010304          MOV      R3,R4          ;GO PRINT
2116 013066 000304          SWAB     R4
2117 013070 004767 000026          JSR      PC,SNPG          ;GET SECOND DIGIT
2118 013074 010304          MOV      R3,R4          ;GO PRINT
2119 013076 006004          ROR      R4
2120 013100 006004          ROR      R4
2121 013102 006004          ROR      R4
2122 013104 006004          ROR      R4
2123 013106 004767 000010          JSR      PC,SNPG          ;GET THIRD DIGIT
2124 013112 010304          MOV      R3,R4          ;GO PRINT
2125 013114 004767 000002          JSR      PC,SNPG          ;GET FOURTH DIGIT
2126 013120 000207          RTS      PC          ;GO PRINT
2127 013122 012767 000260 165536  SNPG:  MOV      #260,TOB          ;EXIT
2128 013130 042704 177760          BIC      #177760,R4          ;SET BASE = 0
2129 013134 050467 165526          BIS      R4,TOB          ;MASK DIGIT
2130 013140 004767 177322          JSR      PC,TOG          ;SET ASCII
2131 013144 000207          RTS      PC          ;TYPE DIGIT
2132                                     ;RETURN

```

```

2133                                     ;MESSAGE TABLE*****
2134
2135 013146 041445 030523 020040 MSG1: .ASCII /%CS1 WC BA FC CS2 /
2136 013154 020040 041527 020040
2137 013162 020040 041040 020101
2138 013170 020040 020040 041506
2139 013176 020040 020040 041440
2140 013204 031123 020040 020040
2141 013212 051504 020040 020040 .ASCII /DS ER TC%/
2142 013220 042440 020122 020040
2143 013226 020040 041524 021445
2144 013234 051045 053505 047111 MSG2: .ASCII /%REWIND ERROR#/
2145 013242 020104 051105 047522
2146 013250 021522
2147 013252 022445 046524 031060 MSG3: .ASCII /%TM02-TU45 BASIC FUNCTION TEST (CZTUJA0)%/
2148 013260 052055 032125 020065
2149 013266 040502 044523 020103
2150 013274 052506 041516 044524
2151 013302 047117 052040 051505
2152 013310 020124 041450 052132
2153 013316 045125 030101 022451
2154 013324 047105 042524 020122 .ASCII /ENTER CONDITIONS IN OCTAL%/
2155 013332 047503 042116 052111
2156 013340 047511 051516 044440
2157 013346 020116 041517 040524
2158 013354 022514 043
2159 013357 045 042522 044507 MSG4: .ASCII /%REGISTER START = #/
2160 013364 052123 051105 051440
2161 013372 040524 052122 036440
2162 013400 021440
2163 013402 053045 041505 047524 MSG5: .ASCII /%VECTOR = #/
2164 013410 020122 020075 043
2165 013415 045 047105 020104 MSG6: .ASCII /%END OF PASS #/
2166 013422 043117 050040 051501
2167 013430 020123 043
2168 013433 040 020077 043 MSG7: .ASCII / ? #/
2169 013437 045 043 MSG8: .ASCII /%#/
2170 013441 045 047520 044523 MSG9: .ASCII /%POSITION ERROR: #/
2171 013446 044524 047117 042440
2172 013454 051122 051117 020072
2173 013462 043
2174 013463 045 051104 053111 MSG10: .ASCII /%DRIVE NUMBER: #/
2175 013470 020105 052516 041115
2176 013476 051105 020072 043
2177 013503 045 046123 053101 MSG11: .ASCII /%SLAVE NUMBER: #/
2178 013510 020105 052516 041115
2179 013516 051105 020072 043
2180 013523 045 051127 052111 MSG12: .ASCII /%WRITE ERROR #/
2181 013530 020105 051105 047522
2182 013536 020122 043
2183 013541 045 042522 042101 MSG13: .ASCII /%READ REVERSE ERROR #/
2184 013546 051040 053105 051105
2185 013554 042523 042440 051122
2186 013562 051117 021440
2187 013566 051045 040505 020104 MSG14: .ASCII /%READ FORWARD ERROR #/
2188 013574 047506 053522 051101

```

2189	013602	020104	051105	047522		
2190	013610	020122	043			
2191	013613	045	051127	052111	MSG15:	.ASCII /%WRITE TM ERROR #/
2192	013620	020105	046524	042440		
2193	013626	051122	051117	021440		
2194	013634	051045	053105	051105	MSG16:	.ASCII /%REVERSE ERROR #/
2195	013642	042523	042440	051122		
2196	013650	051117	021440			
2197	013654	043045	051117	040527	MSG17:	.ASCII /%FORWARD ERROR #/
2198	013662	042122	042440	051122		
2199	013670	051117	021440			
2200	013674	047040	055122	021440	MSG20:	.ASCII / NRZ #/
2201	013702	050040	020105	043	MSG21:	.ASCII / PE #/
2202	013707	040	054105	052120	MSG22:	.ASCII / EXPT: #/
2203	013714	020072	043			
2204	013717	040	041522	042126	MSG23:	.ASCII / RCVD: #/
2205	013724	020072	043			
2206	013727	045	052502	020123	MSG24:	.ASCII /%BUS TRAP: #/
2207	013734	051124	050101	020072		
2208	013742	043				
2209	013743	045	041527	020072	MSG25:	.ASCII /%WC: #/
2210	013750	043				
2211	013751	045	040502	020072	MSG26:	.ASCII /%BA: #/
2212	013756	043				
2213	013757	045	041104	020072	MSG27:	.ASCII /%DB: #/
2214	013764	043				
2215	013765	045	047111	052111	MSG28:	.ASCII /%INIT DID NOT CLEAR RH #/
2216	013772	042040	042111	047040		
2217	014000	052117	041440	042514		
2218	014006	051101	051040	020110		
2219	014014	043				
2220	014015	045	041523	047040	MSG29:	.ASCII /%SC NOT RESET BY INIT #/
2221	014022	052117	051040	051505		
2222	014030	052105	041040	020131		
2223	014036	047111	052111	021440		
2224	014044	052045	042522	047040	MSG30:	.ASCII /%TRE NOT RESET BY INIT #/
2225	014052	052117	051040	051505		
2226	014060	052105	041040	020131		
2227	014066	047111	052111	021440		
2228	014074	041445	031123	047040	MSG31:	.ASCII /%CS2 NOT RESET BY INIT #/
2229	014102	052117	051040	051505		
2230	014110	052105	041040	020131		
2231	014116	047111	052111	021440		
2232	014124	042045	052114	047040	MSG32:	.ASCII /%DLT NOT SET #/
2233	014132	052117	051440	052105		
2234	014140	021440				
2235	014142	051445	020103	047516	MSG33:	.ASCII /%SC NOT SET #/
2236	014150	020124	042523	020124		
2237	014156	043				
2238	014157	045	051124	020105	MSG34:	.ASCII /%TRE NOT SET #/
2239	014164	047516	020124	042523		
2240	014172	020124	043			
2241	014175	045	051111	047040	MSG35:	.ASCII /%IR NOT SET BY INIT #/
2242	014202	052117	051440	052105		
2243	014210	041040	020131	047111		
2244	014216	052111	021440			

2245	014222	047445	020122	047516	MSG36: .ASCII /%OR NOT RESET BY INIT #/
2246	014230	020124	042522	042523	
2247	014236	020124	054502	044440	
2248	014244	044516	020124	043	
2249	014251	045	051117	047040	MSG37: .ASCII /%OR NOT RESET BY 1 SILO ENTRY #/
2250	014256	052117	051040	051505	
2251	014264	052105	041040	020131	
2252	014272	020061	044523	047514	
2253	014300	042440	052116	054522	
2254	014306	021440			
2255	014310	047445	020122	047516	MSG38: .ASCII /%OR NOT SET BY SILO FULL #/
2256	014316	020124	042523	020124	
2257	014324	054502	051440	046111	
2258	014332	020117	052506	046114	
2259	014340	021440			
2260	014342	041045	042101	051440	MSG39: .ASCII /%BAD SILO READ #/
2261	014350	046111	020117	042522	
2262	014356	042101	021440		
2263	014362	044445	020122	047516	MSG40: .ASCII /%IR NOT RESET BY SILO FULL#/
2264	014370	020124	042522	042523	
2265	014376	020124	054502	051440	
2266	014404	046111	020117	052506	
2267	014412	046114	043		
2268	014415	040	047516	026516	MSG41: .ASCII / NON-EXIST DRIVE#/
2269	014422	054105	051511	020124	
2270	014430	051104	053111	021505	
2271	014436	047040	047117	042455	MSG42: .ASCII / NON-EXIST SLAVE#/
2272	014444	044530	052123	051440	
2273	014452	040514	042526	043	
2274	014457	040	042523	044522	MSG43: .ASCII / SERIAL NO: #/
2275	014464	046101	047040	035117	
2276	014472	021440			
2277	014474	042445	040522	042523	MSG44: .ASCII /%ERASE HEAD INOPERATIVE/
2278	014502	044040	040505	020104	
2279	014510	047111	050117	051105	
2280	014516	052101	053111	105	
2281	014523	045	044103	041505	.ASCII /%CHECK POLARITY#/
2282	014530	020113	047520	040514	
2283	014536	044522	054524	043	
2284	014543	045	051105	051501	MSG45: .ASCII /%ERASE HEAD POLARITY WRONG#/
2285	014550	020105	042510	042101	
2286	014556	050040	046117	051101	
2287	014564	052111	020131	051127	
2288	014572	047117	021507		
2289	014576	051445	052105	052455	MSG46: .ASCII /%SET-UP WRITE ERROR#/
2290	014604	020120	051127	052111	
2291	014612	020105	051105	047522	
2292	014620	021522			
2293	014622	051445	040520	042503	MSG47: .ASCII /%SPACE FORWARD ERROR#/
2294	014630	043040	051117	040527	
2295	014636	042122	042440	051122	
2296	014644	051117	043		
2297	014647	045	050123	041501	MSG48: .ASCII /%SPACE REVERSE ERROR#/
2298	014654	020105	042522	042526	
2299	014662	051522	020105	051105	
2300	014670	047522	021522		

2301	014674	041045	043125	042506	MSG49: .ASCII /%BUFFERED WRITE ERROR#/
2302	014702	042522	020104	051127	
2303	014710	052111	020105	051105	
2304	014716	047522	021522		
2305	014722	041045	052117	051440	MSG50: .ASCII /%BOT SET AFTER BUFFERED WRITE#/
2306	014730	052105	040440	052106	
2307	014736	051105	041040	043125	
2308	014744	042506	042522	020104	
2309	014752	051127	052111	021505	
2310	014760	047045	020117	047502	MSG51: .ASCII /%NO BOT FROM READ IN PRESET#/
2311	014766	020124	051106	046517	
2312	014774	051040	040505	020104	
2313	015002	047111	050040	042522	
2314	015010	042523	021524		
2315	015014	052045	020103	047111	MSG52: .ASCII /%TC INCORRECT #/
2316	015022	047503	051122	041505	
2317	015030	020124	043		
2318	015033	045	046123	053101	MSG53: .ASCII /%SLAVE NOT OFF LINE#/
2319	015040	020105	047516	020124	
2320	015046	043117	020106	044514	
2321	015054	042516	043		
2322	015057	045	051045	051505	MSG54: .ASCII /%%RESET SLAVE TO ON LINE BEFORE CONTINUING#/
2323	015064	052105	051440	040514	
2324	015072	042526	052040	020117	
2325	015100	047117	046040	047111	
2326	015106	020105	042502	047506	
2327	015114	042522	041440	047117	
2328	015122	044524	052516	047111	
2329	015130	021507			
2330	015132	047045	055122	047440	MSG55: .ASCII /%NRZ ONLY: #/
2331	015140	046116	035131	021440	
2332	015146	044440	042524	035122	MSG56: .ASCII / ITER: #/
2333	015154	021440			
2334	015156	052045	020115	047516	MSG57: .ASCII /%TM NOT SET#/
2335	015164	020124	042523	021524	
2336	015172	042445	052111	042510	MSG60: .ASCII /%EITHER TAPE NOT ERASED OR OPI PROBLEM#/
2337	015200	020122	040524	042520	
2338	015206	047040	052117	042440	
2339	015214	040522	042523	020104	
2340	015222	051117	047440	044520	
2341	015230	050040	047522	046102	
2342	015236	046505	043		
2343	015241	045	044122	030461	MSG61: .ASCII /%RH11 OR RH70: #/
2344	015246	047440	020122	044122	
2345	015254	030067	020072	043	
2346	015261	045	044122	047440	MSG62: .ASCII /%RH ONLY: #/
2347	015266	046116	035131	021440	
2348					

```

2349                                     ;TEST HEADERS*****
2350
2351 015274 022445 052106 035061 MSFT1: .ASCII /%%FT1:RH ADDRESSING #/
2352 015302 044122 040440 042104
2353 015310 042522 051523 047111
2354 015316 020107 043
2355 015321 045 043045 031124 MSFT2: .ASCII /%%FT2:RH REGISTER BITS TEST #/
2356 015326 051072 020110 042522
2357 015334 044507 052123 051105
2358 015342 041040 052111 020123
2359 015350 042524 052123 021440
2360 015356 022445 052106 035063 MSFT3: .ASCII /%%FT3:RH INITIALIZE TEST #/
2361 015364 044122 044440 044516
2362 015372 044524 046101 055111
2363 015400 020105 042524 052123
2364 015406 021440
2365 015410 022445 052106 035064 MSFT4: .ASCII /%%FT4:RH11 SILO TEST 1 #/
2366 015416 044122 030461 051440
2367 015424 046111 020117 042524
2368 015432 052123 030440 021440
2369 015440 022445 052106 035065 MSFT5: .ASCII /%%FT5:RH11 SILO TEST 2 #/
2370 015446 044122 030461 051440
2371 015454 046111 020117 042524
2372 015462 052123 031040 021440
2373 015470 022445 052106 035066 MSFT6: .ASCII /%%FT6:RH11 SILO TEST 3 #/
2374 015476 044122 030461 051440
2375 015504 046111 020117 042524
2376 015512 052123 031440 021440
2377 015520 022445 052106 035067 MSFT7: .ASCII /%%FT7:RH11 SILO TEST 4 #/
2378 015526 044122 030461 051440
2379 015534 046111 020117 042524
2380 015542 052123 032040 021440
2381 015550 022445 052106 030061 MSFT10: .ASCII /%%FT10:RH11 SILO TEST 5 #/
2382 015556 051072 030510 020061
2383 015564 044523 047514 052040
2384 015572 051505 020124 020065
2385 015600 043
2386 015601 045 043045 030524 MSFT11: .ASCII /%%FT11:NOP TEST#/
2387 015606 035061 047516 020120
2388 015614 042524 052123 043
2389 015621 045 043045 030524 MSFT12: .ASCII /%%FT12:REWIND TEST#/
2390 015626 035062 042522 044527
2391 015634 042116 052040 051505
2392 015642 021524
2393 015644 022445 052106 031461 MSFT13: .ASCII /%%FT13:WRITE-READ TEST#/
2394 015652 053472 044522 042524
2395 015660 051055 040505 020104
2396 015666 042524 052123 043
2397 015673 045 043045 030524 MSFT14: .ASCII /%%FT14:SPACE TEST#/
2398 015700 035064 050123 041501
2399 015706 020105 042524 052123
2400 015714 043
2401 015715 045 043045 030524 MSFT15: .ASCII /%%FT15:ERASE TEST#/
2402 015722 035065 051105 051501
2403 015730 020105 042524 052123
2404 015736 043

```


2405	015737	045	043045	030524	MSFT16: .ASCII /%%FT16:TAPE MARK WRITE-READ TEST#/ /
2406	015744	035066	040524	042520	
2407	015752	046440	051101	020113	
2408	015760	051127	052111	026505	
2409	015766	042522	042101	052040	
2410	015774	051505	021524		
2411	016000	022445	052106	033461	MSFT17: .ASCII /%%FT17:TM SPACE TEST #/ /
2412	016006	052072	020115	050123	
2413	016014	041501	020105	042524	
2414	016022	052123	021440		
2415	016026	022445	052106	030062	MSFT20: .ASCII /%%FT20:WRITE CHECK TEST #/ /
2416	016034	053472	044522	042524	
2417	016042	041440	042510	045503	
2418	016050	052040	051505	020124	
2419	016056	043			
2420	016057	045	043045	031124	MSFT21: .ASCII /%%FT21:ERASE HEAD TEST#/ /
2421	016064	035061	051105	051501	
2422	016072	020105	042510	042101	
2423	016100	052040	051505	021524	
2424	016106	022445	052106	031062	MSFT22: .ASCII /%%FT22:BUFFERED COMMAND TEST#/ /
2425	016114	041072	043125	042506	
2426	016122	042522	020104	047503	
2427	016130	046515	047101	020104	
2428	016136	042524	052123	043	
2429	016143	045	043045	031124	MSFT23: .ASCII /%%FT23:READ IN PRESET TEST#/ /
2430	016150	035063	042522	042101	
2431	016156	044440	020116	051120	
2432	016164	051505	052105	052040	
2433	016172	051505	021524		
2434	016176	022445	052106	032062	MSFT24: .ASCII /%%FT24:REWIND-OFF LINE TEST#/ /
2435	016204	051072	053505	047111	
2436	016212	026504	043117	020106	
2437	016220	044514	042516	052040	
2438	016226	051505	021524		
2439					
2440					
2441					
2442	016232	000000			WDATA: .EVEN 0
2443		017744			.=.+1510
2444	017744	000000			RDATA: 0
2445					
2446		000001			.END

AS	000616	ERRP	000714	FT16A	006516	FT4ER	003624	MSFT22	016106
BA	000604	ERRP1	000716	FT16B	006522	FT4ERA	003634	MSFT23	016143
BADDR	000704	EXEC	010626	FT16X	006716	FT4ERB	003644	MSFT24	016176
BTRP	000662	EXECA	010700	FT17	006726	FT4ERC	003652	MSFT3	015356
BTRP2	000664	EXECB	010706	FT17A	006746	FT4X	003614	MSFT4	015410
CC	000620	EXECC	010726	FT17B	006752	FT5	003670	MSFT5	015440
CRCNT	001002	EXECX	010746	FT17C	007112	FT5A	003712	MSFT6	015470
CS	000610	EXECXA	010752	FT17D	007162	FT5B	003740	MSFT7	015520
C1	000600	EXECXX	010756	FT17D1	007200	FT5C	003762	MSG1	013146
DATA0	001022	EXFL	000766	FT17E	007214	FT5D	004012	MSG10	013463
DATA1	001024	FC	000606	FT17F	007306	FT5E	004024	MSG11	013503
DATA2	001026	FCNT	000706	FT17X	007336	FT5ER	004062	MSG12	013523
DATA3	001030	FT1	002700	FT2	003000	FT5X	004052	MSG13	013541
DATBL	001020	FT1A	002732	FT2A	003012	FT6	004072	MSG14	013566
DAT1	011552	FT1B	002736	FT2B	003052	FT6A	004114	MSG15	013613
DAT1A	011556	FT1X	002750	FT2C	003112	FT6B	004122	MSG16	013634
DAT1B	011562	FT1XX	002774	FT2D	003126	FT6C	004160	MSG17	013654
DAT2	011574	FT10	004450	FT2E	003156	FT6D	004202	MSG2	013234
DAT3	011602	FT10A	004504	FT2ER	003170	FT6DE	004244	MSG20	013674
DAT4	011612	FT10ER	004570	FT2ERA	003220	FT6DEA	004276	MSG21	013702
DAT4A	011620	FT10X	004546	FT2ERB	003272	FT6DEB	004342	MSG22	013707
DB	000622	FT10XX	004564	FT2ERC	003302	FT6DEX	004352	MSG23	013717
DOUT	012736	FT11	004604	FT2X	003312	FT6E	004204	MSG24	013727
DOUTD	013020	FT12	004722	FT20	007342	FT6ER	004234	MSG25	013743
DOUT1	012752	FT12A	005002	FT20A	007420	FT6X	004224	MSG26	013751
DOUT2	013000	FT13	005034	FT20B	007504	FT7	004354	MSG27	013757
DOUT3	013006	FT13A	005114	FT20C	007536	FT7A	004406	MSG28	013765
DRVN	000700	FT13B	005156	FT20X	007572	FT7ER	004434	MSG29	014015
DRVTP	000652	FT13C	005236	FT21	007602	FT7X	004424	MSG3	013252
DS	000612	FT13D	005304	FT21A	007740	FUN	000776	MSG30	014044
DSAV	000752	FT13E	005356	FT21X	010074	HDRFL	000674	MSG31	014074
DSUP	011460	FT13X	005370	FT22	010104	INIT1	011742	MSG32	014124
DS0	011462	FT14	005374	FT22A	010160	INIT2	011752	MSG33	014142
DS1	011502	FT14A	005446	FT22B	010200	ITAMT	000654	MSG34	014157
DS2	011522	FT14A1	005412	FT22X	010274	ITCNT	000750	MSG35	014175
DS3	011532	FT14A2	005542	FT23	010304	ITER	011676	MSG36	014222
DS4	011542	FT14A3	005566	FT23A	010414	ITER0	011710	MSG37	014251
DT	000626	FT14B	005622	FT23B	010434	ITER1	011716	MSG38	014310
EMADDR	000676	FT14C	006040	FT23C	010464	ITRLP	000764	MSG39	014342
ER	000614	FT14RE	006046	FT23X	010520	LTADD	000774	MSG4	013357
ERADD	000736	FT14RF	006012	FT24	010524	MR	000624	MSG40	014362
ERCHK	011056	FT14RR	005750	FT24A	010560	MSFT1	015274	MSG41	014415
ERPT	011102	FT14R0	006120	FT24X	010612	MSFT10	015550	MSG42	014436
ERPTA	011172	FT14R1	006142	FT24XX	010622	MSFT11	015601	MSG43	014457
ERPTA1	011124	FT14R2	006146	FT3	003324	MSFT12	015621	MSG44	014474
ERPTB	011230	FT14R3	006220	FT3A	003400	MSFT13	015644	MSG45	014543
ERPTB1	011242	FT14X	006230	FT3B	003422	MSFT14	015673	MSG46	014576
ERPTC	011272	FT14XX	006260	FT3ER	003460	MSFT15	015715	MSG47	014622
ERPTD	011402	FT15	006264	FT3ERA	003512	MSFT16	015737	MSG48	014647
ERPTG	011132	FT15A	006334	FT3ERB	003522	MSFT17	016000	MSG49	014674
ERPTX	011412	FT15B	006350	FT3ERC	003532	MSFT2	015321	MSG5	013402
ERPTXX	011416	FT15X	006460	FT3X	003450	MSFT20	016026	MSG50	014722
ERPT1A	011224	FT16	006464	FT4	003542	MSFT21	016057	MSG51	014760

MSG52	015014	OFL	012734	SCOLP	000762	TCRLFA	012436	TSCD1	002476
MSG53	015033	OPDYX	000726	SCOPE	011634	TEMP1	000740	TSCD2	002522
MSG54	015057	PATRN	001006	SCOPE1	011650	TEMP2	000742	TSCD3	002536
MSG55	015132	PCNTR	001016	SCOPE2	011662	TEMP3	000744	TSRH	002434
MSG56	015146	PEXFL	000770	SERFL	001000	TEND	002636	TSTTBL	001032
MSG57	015156	PFLG	000732	SERNUM	000650	TENDX	002670	TTIN	012314
MSG6	013415	PSW	000634	SLVN	000702	TEX	012502	TTINT	012004
MSG60	015172	RCNT	000712	SN	000630	TIB	000670	TTIN1	012334
MSG61	015241	RDATA	017744	SNPG	013122	TINER	012276	TTIN2	012350
MSG62	015261	RDYDX	000724	SNPT	013044	TKB	000642	TTOUT	012366
MSG7	013433	REGS	000660	START	001600	TKS	000640	TTR	012130
MSG8	013437	RFD	000722	STFLG	000772	TMCHK	011420	TTR0	012136
MSG9	013441	RHOF	001014	STMSK	000746	TMCHK0	011430	TTR1	012164
MTINT	011770	RHTF	001010	STSCD	002562	TMCHK1	011432	TTR1A	012200
NRZOF	001012	RH17F	000672	STO	001774	TOB	000666	TTR1B	012214
OCTP	012516	RRD	000720	STOA	001624	TOG	012466	TTR2	012252
OCTPE	012504	RTRN	000734	STOB	001634	TPB	000646	TTR3	012262
OCTPE1	012522	RWND	010760	ST1	002016	TPS	000644	TTR4	012272
OCTPG	012672	RWDA	011000	ST1A	002036	TRAP	012014	TTR5	012274
OCTPG0	012710	RWDB	011004	ST2	002134	TRAPX	012076	UDES	001004
OCTPG1	012714	RWDX	011050	ST3	002234	TRAPXX	012124	VECT	000656
OCTP0	012536	SAV1	000754	ST4	002406	TRAP1	012050	WC	000602
OCTP1	012560	SAV2	000756	SWR	000636	TRAP2	012066	WCNT	000710
OCTP2	012566	SAV3	000760	TC	000632	TSCD	002412	WDATA	016232
OCTP3	012656	SCNT	000730	TCRLF	012420	TSCD0	002442	.	= 017746

. ABS. 017746 000

ERRORS DETECTED: 0

.CZTUJA.SEQ/SOL_CZTUJA.P11
 RUN-TIME: 14 28 2 SECONDS
 RUN-TIME RATIO: 167/45=3.6
 CORE USED: 5K (10 PAGES)