

.REM %

IDENTIFICATION

PRODUCT CODE: AC-E493B-MC
PRODUCT TITLE: CZTUQB0 TM03/TU45 BSC FCTN
DATE CREATED: 25 MAY 1978
MAINTAINER: CSS - NASHUA
AUTHOR: J. G. ADAMS/R. J. COLLINS
UPDATE INFORMATION: DATE AUTHOR
29-FEB-80 VIJAY ANANDWALA

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

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

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

COPYRIGHT (C) 1975, 1980 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

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

1. ABSTRACT

THIS PROGRAM IS INTENDED TO TEST ALL OF THE BASIC FUNCTIONAL LEVEL OPERATIONS OF THE TM03/TU45 MAG TAPE SYSTEM. ALL FUNCTIONS; WRITE, READ, SPACE, ERASE, REWIND, ETC; WILL BE TESTED. IN ADDITION TO THE TM03/TU45 TESTS, THE RH WILL BE TESTED SEPARATELY IN SO FAR AS IT IS POSSIBLE TO SEPARATE THE RH FROM THE TM03/TU45 ITSELF.

2. REQUIREMENTS (HARDWARE)

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

3. LOADING PROCEDURE

USE STANDARD BINARY LOADING PROCEDURE

4. STARTING PROCEDURE

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

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

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

4.2 SAMPLE START AT 200

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

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

TM03-TU45 BASIC FUNCTIONS TEST (CZTUQB0)
TYPE ^C TO RESTART

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

5. CONSOLE SWITCH SETTING

CONTROL:

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

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

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

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

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

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

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

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

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

6. ERROR PRINTOUTS

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

EXAMPLES:

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

FT13: WRITE-READ TEST
WRITE ERROR NRZ

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

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

FT14: SPACE TEST
SPACE REVERSE ERROR NRZ

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

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

FT14: SPACE TEST
POSITION ERROR:
REVERSE ERROR EXPT:20 RCVD:22

▲

7. OPERATION

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

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

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

SINGLE TEST SELECTION: (SW0-SW4)

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

8. SUBTEST SUMMARIES

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

A. RH TESTS: THE FIRST TEN (10) TESTS WILL PERFORM BASIC RH
OPERATIONS AS FAR AS IS POSSIBLE WITHOUT REQUIRING
THE TMO3/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 TMO2 REGISTER ADDRESS
IN SEQUENCE STARTING AT THE ADDRESS
OF CS1 ENTERED BY THE OPERATOR.

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

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

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

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

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

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

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

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

B. TM03/TU45 BASIC FUNCTIONS: THE FOLLOWING FOURTEEN (14) TESTS WILL ASSURE OPERATION OF THE MAG TAPE BASIC FUNCTIONS.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

1825
1826
1827

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

1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851

```
:25 MAY 78  
:R. BARNES/R. J. COLLINS  
:MCALL .SACT11,.$EOP,$SCATCH,$SAVE,$RESTORE,$CHAIN,$SCHNMODE  
:NLIST MC  
:LIST ME  
:ENABLE ABS,AMA  
  
:CONSOLE SWITCHES*****  
:SW15(100000) 1=HALT ON ERROR  
: 0=CONTINUE  
:SW14(040000) 1=LOOP ON ERROR (SCOPE(040000) RH TESTS ONLY)  
: 0=CONTINUE  
:SW13(02000): 1=DO NOT PRINT ERRORS  
: 0=PRINT ERRORS  
:SW12(010000): 1=CONTINUOUS CYCLE  
: 0=HALT AT END OF PASS  
:SW11(40000): 1=INHIBIT ITERATIONS  
: 0=DO ITERATIONS  
:SW10(002000): 1=HALT AT END OF EACH TEST  
: 0=CONTINUE  
:SW0-4: SELECT TEST NUMBER :: 00=ALL TESTS  
:USE SOFTWARE SWR IF HARDWARE SWR <15::00> = 17777 OR NOT AVAIL.
```



```

1900 ;REGISTER EQUIVS*****
1901
1902 000000 R0=%0
1903 000001 R1=%1
1904 000002 R2=%2
1905 000003 R3=%3
1906 000004 R4=%4
1907 000005 R5=%5
1908 000006 SP=%6
1909 000007 PC=%7
1910
1911
1912
1913 ;ACT11 HOOK *****
(1) $SVPC= ;SAVE CURRENT LOCATION CTR
(1) 000764
(1) 000046 .=46 ;SET LOCATION 46
(1) 000046 .WORD SENDAD
(1) 000052 .=52 ;SET LOCATION 52 = 0
(1) 000052 .WORD 0 ;RESTORE LOCATION CTR
(1) 000764 .=$SVPC
(1)
1914 ;TTY INTERRUPT VECTOR*****
1915
1916 000060 .=60
1917 000060 012750 .WORD TTINT ;TTY INTERRUPT HEADER ADDRESS
1918 000062 000340 .WORD 340 ;PRIORITY LEVEL 7
1919
1920 ;SOFTWARE SWITCH REGISTER*****
1921 ;USED IF HARDWARE SWR <15::00> = 177777 OR NOT AVAIL.
1922
1923 000176 000176 .=176
1924 000176 000000 SWREG: 0 ;SOFTWARE SWITCH REGISTER
1925
1926 ;START ADDRESS*****
1927
1928
1929 000200 000200 .=200
1930 000200 000137 001600 JMP START ;PROGRAM START
1931
1932 ;RESTART ADDRESS*****
1933 000210 000210 .=210
1934 000210 000137 002472 JMP ST4
1935
1936 ;TM03 INTERRUPT VECTOR*****
1937
1938 000224 000224 .=224
1939 000224 012740 MTINT ;TAPE INTERRUPT HANDLER ADDRESS
1940 000226 000340 340
1941

```

| | | | | |
|------|--------|--------|--------------------------------|--------|
| 1943 | | | | |
| 1944 | 000510 | | .=510 | |
| 1945 | | | :MASS BUS REGISTER EQUIVS***** | |
| 1946 | | | | |
| 1947 | 000510 | 172440 | C1: | 172440 |
| 1948 | 000512 | 172442 | WC: | 172442 |
| 1949 | 000514 | 172444 | BA: | 172444 |
| 1950 | 000516 | 172446 | FC: | 172446 |
| 1951 | 000520 | 172450 | CS: | 172450 |
| 1952 | 000522 | 172452 | DS: | 172452 |
| 1953 | 000524 | 172454 | ER: | 172454 |
| 1954 | 000526 | 172456 | AS: | 172456 |
| 1955 | 000530 | 172460 | CC: | 172460 |
| 1956 | 000532 | 172462 | DB: | 172462 |
| 1957 | 000534 | 172464 | MR: | 172464 |
| 1958 | 000536 | 172466 | DT: | 172466 |
| 1959 | 000540 | 172470 | SN: | 172470 |
| 1960 | 000542 | 172472 | TC: | 172472 |
| 1961 | 000544 | 172474 | BAE: | 172474 |
| 1962 | | | | |
| 1963 | | | :CONSTANTS***** | |
| 1964 | | | | |
| 1965 | 000546 | 177776 | PSW: | 177776 |
| 1966 | 000550 | 177570 | SWR: | 177570 |
| 1967 | 000552 | 177560 | TKS: | 177560 |
| 1968 | 000554 | 177562 | TKB: | 177562 |
| 1969 | 000556 | 177564 | TPS: | 177564 |
| 1970 | 000560 | 177566 | TPB: | 177566 |
| 1971 | 000562 | 177777 | SERNUM: | 177777 |
| 1972 | 000564 | 000011 | DRVTP: | 011 |
| 1973 | 000566 | 000010 | ITAMT: | 10 |
| 1974 | 000570 | 000224 | VECT: | 224 |
| 1975 | 000572 | 172440 | REGS: | 172440 |
| 1976 | 000574 | 000004 | BTRP: | 4 |
| 1977 | 000576 | 000006 | BTRP2: | 6 |

:PROCESSOR STATUS
 :SWITCH REGISTER
 :TTY READER STATUS
 :TTY READ BUFFER
 :TTY PUNCH STATUS
 :TTY PUNCH BUFFER
 :SERIAL NUMBER
 :DRIVE TYPE
 :ITERATION AMOUNT
 :INTERRUPT VECTOR(RH)
 :STARTING REGISTER ADDRESS
 :BUS TRAP ADDRESS
 :BUS TRAP PRIORITY LEVEL 7


```
1979 ;FLAGS AND COUNTERS*****
1980
1981 000600 000000 TOB: 0
1982 000602 000000 TIB: 0
1983 000604 000000 RH17F: 0
1984 000606 000000 HDRFL: 0
1985 000610 000000 EMADDR: 0
1986 000612 000000 DRVN: 0
1987 000614 000000 SLVN: 0
1988 000616 000000 BADDR: 0
1989 000620 000000 FCNT: 0
1990 000622 000000 WCNT: 0
1991 000624 000000 RCNT: 0
1992 000626 000000 ERRP: 0
1993 000630 000000 ERRP1: 0
1994 000632 000000 RRD: 0
1995 000634 000000 RFD: 0
1996 000636 000000 RDYDX: 0
1997 000640 000000 OPDYX: 0
1998 000642 000000 SCNT: 0
1999 000644 000000 PFLG: 0
2000 000646 000000 RTRN: 0
2001 000650 000000 ERADD: 0
2002 000652 000000 TEMP1: 0
2003 000654 000000 TEMP2: 0
2004 000656 000000 TEMP3: 0
2005 000660 000000 STMSK: 0
2006 000662 000000 ITCNT: 0
2007 000664 000000 DSAV: 0
2008 000666 000000 SAV1: 0
2009 000670 000000 SAV2: 0
2010 000672 000000 SAV3: 0
2011 000674 000000 SCOLP: 0
2012 000676 000000 ITRLP: 0
2013 000700 000000 EXFL: 0
2014 000702 000000 PEXFL: 0
2015 000704 000000 STFLG: 0
2016 000706 000000 LTADD: 0
2017 000710 000000 FUN: 0
2018 000712 000000 SERFL: 0
2019 000714 000000 CRCNT: 0
2020 000716 000000 UDES: 0
2021 000720 000000 PATRN: 0
2022 000722 000000 RHTF: 0
2023 000724 000000 NRZOF: 0
2024 000726 000000 RHOF: 0
2025 000730 000000 PCNTR: 0
2026 000732 000000 TEMPST: 0
2027 000734 000000 COUNT: 0
2028 000736 000000 RDSW: 0
2029
```

2031
2032
2033

;DATA PATTERN GENERATORS*****

2034 000740 000000
2035 000742 012532
2036 000744 012552
2037 000746 012556
2038 000750 012564

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

| | | | |
|------|--------|--------|-------------------------------|
| 2040 | | | |
| 2041 | | | ;LOGIC TEST ENTRY TABLE***** |
| 2042 | | | |
| 2043 | 000752 | 000000 | TSTTBL: 0 |
| 2044 | 000754 | 000000 | 0 |
| 2045 | 000756 | 003310 | FT1 |
| 2046 | 000760 | 003310 | FT1 |
| 2047 | 000762 | 003410 | FT2 |
| 2048 | 000764 | 003410 | FT2 |
| 2049 | 000766 | 003732 | FT3 |
| 2050 | 000770 | 003732 | FT3 |
| 2051 | 000772 | 004152 | FT4 |
| 2052 | 000774 | 004152 | FT4 |
| 2053 | 000776 | 004300 | FT5 |
| 2054 | 001000 | 004300 | FT5 |
| 2055 | 001002 | 004472 | FT6 |
| 2056 | 001004 | 004472 | FT6 |
| 2057 | 001006 | 004744 | FT7 |
| 2058 | 001010 | 004744 | FT7 |
| 2059 | 001012 | 005040 | FT10 |
| 2060 | 001014 | 005040 | FT10 |
| 2061 | 001016 | 005174 | FT11 |
| 2062 | 001020 | 005174 | FT11 |
| 2063 | 001022 | 005312 | FT12 |
| 2064 | 001024 | 005312 | FT12 |
| 2065 | 001026 | 005424 | FT13 |
| 2066 | 001030 | 005424 | FT13 |
| 2067 | 001032 | 005736 | FT14 |
| 2068 | 001034 | 005736 | FT14 |
| 2069 | 001036 | 006610 | FT15 |
| 2070 | 001040 | 006610 | FT15 |
| 2071 | 001042 | 007010 | FT16 |
| 2072 | 001044 | 007010 | FT16 |
| 2073 | 001046 | 007236 | FT17 |
| 2074 | 001050 | 007236 | FT17 |
| 2075 | 001052 | 007640 | FT20 |
| 2076 | 001054 | 007640 | FT20 |
| 2077 | 001056 | 010064 | FT21 |
| 2078 | 001060 | 010064 | FT21 |
| 2079 | 001062 | 010416 | FT22 |
| 2080 | 001064 | 010416 | FT22 |
| 2081 | 001066 | 010622 | FT23 |
| 2082 | 001070 | 010622 | FT23 |
| 2083 | 001072 | 011042 | FT24 |
| 2084 | 001074 | 011042 | FT24 |
| 2085 | 001076 | 011234 | FT25 |
| 2086 | 001100 | 011234 | FT25 |
| 2087 | 001102 | 011426 | FT26 |
| 2088 | 001104 | 011426 | FT26 |
| 2089 | 001106 | 003202 | .WORD TEND |
| 2090 | 001110 | 000026 | .WORD 26 ;CONTAINS # OF TESTS |

```

2092          001600          . =1600
2093          :PROGRAM START AND HOUSEKEEPING*****
2094
2095 001600 012706 000500      START: MOV    #500,SP          ;SET STACK POINTER
2096 001604 013746 000006      MOV    @#6,-(SP)          ;SAVE VECTORS
2097 001610 013746 000004      MOV    @#4,-(SP)
2098 001614 012737 001640 000004      MOV    #1$,@#4          ;SET UP FOR TIMEOUT
2099 001622 005037 000006      CLR    @#6
2100 001626 022777 177777 176714      CMP    #-1,@SWR          ;REFERENCE HARDWARE SWITCH REGISTER
2101 001634 001402          BEQ    2$
2102 001636 000404          BR     3$
2103 001640 022626          1$:  CMP    (SP)+,(SP)+      ;ADJUST STACK
2104 001642 012737 000176 000550      2$:  MOV    #SWREG,SWR      ;POINT TO SOFTWARE SWITCH REG
2105 001650 012637 000004      3$:  MOV    (SP)+,@#4        ;RESTORE VECTORS
2106 001654 012637 000006      MOV    (SP)+,@#6
2107 001660 005027          CLR    (PC)+
(1) 001662 000000      CHNFLG: .WORD 0          ;;CLEAR CHAIN INDICATOR
(1)          ;;CHAIN MODE INDICATOR
(1)          ;;1/0 = CHAIN/NOT CHAIN MODE
(1) 001664 022737 003246 000042      CMP    #SENDAD,@#42     ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
(1) 001672 001404          BEQ    50$
(1) 001674 005737 000042          TST    @#42             ;;BRANCH IF IN DUMP MODE
(1) 001700 001413          BEQ    52$
(1) 001702 000406          BR     51$
(1) 001704 012737 000176 000550      50$:  MOV    #SWREG,SWR      ;;INVOKE SOFTWARE SWR
(1) 001712 012777 100000 176630      MOV    #100000,@SWR    ;;WITH HALT ON ERROR SET
(1) 001720 005237 001662      51$:  INC    CHNFLG          ;;SET CHNFLG = CHAIN MODE
(1) 001724 000137 002506      JMP    TSCD             ;;GO TO CHAIN ADDRESS
(1) 001730          52$:
2108 001730 122737 000006 000041      4$:  CMPB   #6,@#41         ;BRANCH IF LOADED VIA TMDP (DUMP MODE)
2109 001736 001004          BNE    5$
2110 001740 012704 017007          MOV    #MSG69,R4       ;ADVISE USER TO REMOVE TMDP FROM UUT
2111 001744 004737 013552          JSR    PC,TTOUT
2112 001750 012704 014600          5$:  MOV    #MSG3,R4
2113 001754 004737 013552          JSR    PC,TTOUT        ;PRINT TITLE
2114 001760 112737 000043 014600      MOVVB  #'#,MSG3        ;DO NOT PRINT TITLE ON RESTART
2115 001766 012704 014733      STOB: MOV    #MSG4,R4
2116 001772 004737 013552          JSR    PC,TTOUT        ;REQUEST REGISTER ADDRESS
2117 001776 013703 000572          MOV    REGS,R3
2118 002002 004737 013702          JSR    PC,OCTP         ;PRINT CURRENT ADDRESS
2119 002006 012705 000572          MOV    #REGS,R5        ;SET ADDRESS SAVE LOC
2120 002012 012701 000007          MOV    #7,R1           ;SET SIZE OF RESPONSE
2121 002016 012702 176400          MOV    #176400,R2      ;SET UPPER LIMIT
2122 002022 012703 172300          MOV    #172300,R3      ;SET LOWER LIMIT
2123 002026 004737 013212          JSR    PC,TTR          ;GO GET RESPONSE
2124 002032 012704 014756          MOV    #MSG5,R4
2125 002036 004737 013552          JSR    PC,TTOUT        ;REQUEST VECTOR
2126 002042 013703 000570          MOV    VECT,R3
2127 002046 004737 013702          JSR    PC,OCTP         ;PRINT CURRENT VECTOR
2128 002052 012705 000570          MOV    #VECT,R5        ;SET ADDRESS SAVE LOC
2129 002056 012701 000004          MOV    #4,R1           ;SET SIZE OF RESPONSE
2130 002062 012702 000224          MOV    #224,R2         ;SET UPPER LIMIT
2131 002066 012703 000150          MOV    #150,R3         ;SET LOWER LIMIT
2132 002072 004737 013212          JSR    PC,TTR          ;GO GET RESPONSE
2133 002076 013700 000570          MOV    VECT,R0         ;GET VECTOR
2134 002102 012720 012740          MOV    #MTINT,(R0)+    ;LOAD INTERRUPT ADDRESS IN VECTOR
2135 002106 012710 000340          MOV    #340,(R0)      ;LOAD PRIORITY
  
```

| | | | | | | | |
|------|--------|--------|--------|--------|-----|------------|-----------------------------------|
| 2136 | 002112 | 013700 | 000572 | | MOV | REGS,R0 | ;GET START OF REGS |
| 2137 | 002116 | 012701 | 000017 | | MOV | #17,R1 | ;SET NUMBER OF REGS |
| 2138 | 002122 | 012702 | 000510 | | MOV | #C1,R2 | ;GET START OF TABLE |
| 2139 | 002126 | 010022 | | ST0: | MOV | R0,(R2)+ | ;BUILD TABLE |
| 2140 | 002130 | 062700 | 000002 | | ADD | #2,R0 | ;BUMP ADDRESS |
| 2141 | 002134 | 005301 | | | DEC | R1 | ;SEE IF DONE |
| 2142 | 002136 | 001373 | | | BNE | ST0 | ;IF NOT: BR |
| 2143 | 002140 | 012702 | 000600 | | MOV | #TOB,R2 | |
| 2144 | 002144 | 012700 | 000054 | | MOV | #54,R0 | |
| 2145 | 002150 | 005022 | | ST1: | CLR | (R2)+ | ;CLEAR FLAGS + COUNTERS |
| 2146 | 002152 | 005300 | | | DEC | R0 | |
| 2147 | 002154 | 001375 | | | BNE | ST1 | |
| 2148 | 002156 | 012737 | 000001 | 000722 | MOV | #1,RHTF | ;SET ADDRESS TEST FLAG |
| 2149 | 002164 | 000137 | 003036 | | JMP | TSRH | ;GO DO INITIAL ADDRESS TEST PASS |
| 2150 | 002170 | 012704 | 015035 | ST1A: | MOV | #MSG10,R4 | |
| 2151 | 002174 | 004737 | 013552 | | JSR | PC,TTOUT | ;REQUEST DRIVE NUMBER |
| 2152 | 002200 | 013703 | 000612 | | MOV | DRVN,R3 | ;GET CURRENT DRIVE # |
| 2153 | 002204 | 004737 | 013702 | | JSR | PC,OCPT | ;AND TYPE IT |
| 2154 | 002210 | 012705 | 000612 | | MOV | #DRVN,R5 | ;SET ADDRESS OF DRIVE NUMBER SAVE |
| 2155 | 002214 | 012701 | 000002 | | MOV | #2,R1 | ;SET SIZE OF RESPONSE |
| 2156 | 002220 | 012702 | 000007 | | MOV | #7,R2 | ;SET UPPER LIMIT |
| 2157 | 002224 | 012703 | 000000 | | MOV | #0,R3 | ;SET LOWER LIMIT |
| 2158 | 002230 | 004737 | 013212 | | JSR | PC,TTR | ;GO GET RESPONSE |
| 2159 | 002234 | 012777 | 000040 | 176256 | MOV | #40,@CS | ;SET INIT |
| 2160 | 002242 | 053777 | 000612 | 176250 | BIS | DRVN,@CS | ;SET DRIVE NUMBER |
| 2161 | 002250 | 005777 | 176234 | | TST | @C1 | ;ACCESS DRIVE |
| 2162 | 002254 | 032777 | 010000 | 176236 | BIT | #10000,@CS | ;SEE IF NED |
| 2163 | 002262 | 001405 | | | BEQ | ST2 | ;IF NOT: BR |
| 2164 | 002264 | 012704 | 015767 | | MOV | #MSG41,R4 | |
| 2165 | 002270 | 004737 | 013552 | | JSR | PC,TTOUT | ;PRINT NOT AVAIL |
| 2166 | 002274 | 000735 | | | BR | ST1A | ;REDO DRIVE REQUEST |
| 2167 | 002276 | 012704 | 015055 | ST2: | MOV | #MSG11,R4 | |
| 2168 | 002302 | 004737 | 013552 | | JSR | PC,TTOUT | ;REQUEST SLAVE NUMBER |
| 2169 | 002306 | 013703 | 000614 | | MOV | SLVN,R3 | ;GET CURRENT SLAVE # |
| 2170 | 002312 | 004737 | 013702 | | JSR | PC,OCPT | ;AND TYPE IT |
| 2171 | 002316 | 012705 | 000614 | | MOV | #SLVN,R5 | ;SET ADDRESS OF SLAVE SAVE |
| 2172 | 002322 | 012701 | 000002 | | MOV | #2,R1 | ;SET SIZE OF RESPONSE |
| 2173 | 002326 | 012702 | 000007 | | MOV | #7,R2 | ;SET UPPER LIMIT |
| 2174 | 002332 | 012703 | 000000 | | MOV | #0,R3 | ;SET LOWER LIMIT |
| 2175 | 002336 | 004737 | 013212 | | JSR | PC,TTR | ;GO GET RESPONSE |
| 2176 | 002342 | 012777 | 000040 | 176150 | MOV | #40,@CS | ;INIT |
| 2177 | 002350 | 053777 | 000612 | 176142 | BIS | DRVN,@CS | ;SET DRIVE NUMBER |
| 2178 | 002356 | 013777 | 000614 | 176156 | MOV | SLVN,@C | ;LOAD SLAVE NUMBER |
| 2179 | 002364 | 032777 | 002000 | 176144 | BIT | #2000,@DT | ;SEE IF SLAVE PRESENT |
| 2180 | 002372 | 001005 | | | BNE | ST3 | ;IF SO: BR |
| 2181 | 002374 | 012704 | 016010 | | MOV | #MSG42,R4 | |
| 2182 | 002400 | 004737 | 013552 | | JSR | PC,TTOUT | ;PRINT NON-EXIST SLAVE |
| 2183 | 002404 | 000734 | | | BR | ST2 | ;REDO SLAVE REQUEST |
| 2184 | 002406 | 012704 | 016031 | ST3: | MOV | #MSG43,R4 | |
| 2185 | 002412 | 004737 | 013552 | | JSR | PC,TTOUT | ;PRINT SERIAL NUMBER TAG |
| 2186 | 002416 | 017703 | 176116 | | MOV | @SN,R3 | |
| 2187 | 002422 | 004737 | 014230 | | JSR | PC,SNPT | ;PRINT SERIAL NUMBER |
| 2188 | 002426 | 012704 | 016664 | | MOV | #MSG62,R4 | ;GET REQUEST |
| 2189 | 002432 | 004737 | 013552 | | JSR | PC,TTOUT | ;REQUEST RH ONLY RESPONSE |
| 2190 | 002436 | 013703 | 000726 | | MOV | RHOF,R3 | ;GET CURRENT FLAG SETTING |
| 2191 | 002442 | 004737 | 013702 | | JSR | PC,OCPT | ;AND TYPE IT |

```
2192 002446 012705 000726      MOV    #RHOF,R5      ;SET FLAG ADDRESS
2193 002452 012701 000002      MOV    #2,R1        ;SET SIZE OF RESPONSE
2194 002456 012702 000001      MOV    #1,R2        ;SET UPPER LIMIT
2195 002462 012703 000000      MOV    #0,R3        ;SET LOWER LIMIT
2196 002466 004737 013212      JSR    PC,TTR       ;GO GET RESPONSE
2197
2198                               ;START 210
2199 002472 012706 000500      ST4:  MOV    #500,SP ;SET STACK PTR
2200 002476 005037 000730      CLR    PCNTR       ;CLEAR PASS COUNTER
2201 002502 004737 014332      JSR    PC,GTSWR    ;GET SWITCHES
```

```

2203          :TEST SCHEDULAR*****
2204
2205 002506 005037 000604          TSCD: CLR    RH17F          :SET RH INDICATOR = RH11
2206 002512 013746 000004          MOV    @#4,-(SP)        :SAVE ERROR TRAP VECTOR
2207 002516 013746 000006          MOV    @#6,-(SP)        :AND PRIORITY
2208 002522 012737 002550 000004  MOV    #A1,@#4          :SET TIME OUT TRAP TO 1$ BELOW
2209 002530 005037 000006          CLR    @#6              :
2210 002534 005777 176004          TST    @BAE             :REFERENCE BAE REGISTER
2211 002540 012737 000001 000604  MOV    #1,RH17F        :SET FLAG = RH70
2212 002546 000401                    BR     A2                :
2213 002550 022626                    A1:  CMP    (SP)+, (SP)+  :RESTORE SP FROM TRAP
2214 002552 012637 000006          A2:  MOV    (SP)+, @#6   :
2215 002556 012637 000004          MOV    (SP)+, @#4      :
2216
2217 002562 052777 000100 175762  BIS    #100,@TKS        :SET KEYBOARD IE BIT
2218 002570 005037 000704          CLR    STFLG           :CLEAR SINGLE TEST FLAG
2219 002574 017700 175750          MOV    @SWR,R0
2220 002600 042700 177740          BIC    #177740,R0
2221 002604 001160                    BNE    STSCD            :GO SELECT SINGLE TEST
2222 002606 005737 001662          TST    CHNFLG          ;;BRANCH IF NOT IN CHAIN MODE
(1) 002612 001510                    BEQ    TSCDA
(1) 002614 012737 177777 000612  MOV    #-1,DRVN        ;;INITIALIZE DRIVE #
(1) 002622 012737 177777 000614  NXTDRV: MOV    #-1,SLVN  ;;INITIALIZE SLAVE #
(1) 002630 012777 000040 175662  1$:  MOV    #40,@CS        ;;INIT CONTROLLER
(1) 002636 005237 000612          INC    DRVN            ;;STEP DRIVE #
(1) 002642 022737 000010 000612  CMP    #10,DRVN        ;;EXIT IF ALL DRIVES TESTED
(1) 002650 001557                    BEQ    $DONE           ;;FOR AVAILABILITY
(1) 002652 013777 000612 175640  MOV    DRVN,@CS        ;;LOAD DRIVE #
(1) 002660 005777 175624          TST    @C1             ;;ACCESS DRIVE
(1) 002664 032777 010000 175626  BIT    #10000,@CS     ;;BRANCH IF DRIVE NON EXISTANT
(1) 002672 001356                    BNE    1$              ;;(NED = 1)
(1) 002674 005237 000614          NXTSLV: INC    SLVN     ;;STEP SLAVE # AND BRANCH
(1) 002700 001011                    BNE    1$              ;;IF NOT SLAVE 0
(1) 002702 005737 000612          TST    DRVN           ;;BRANCH IF NOT DRIVE # 0
(1) 002706 001006                    BNE    1$
(1) 002710 122737 000006 000041  CMPB   #6,@#41        ;;BRANCH IF NOT TMDP
(1) 002716 001002                    BNE    1$
(1) 002720 005237 000614          INC    SLVN           ;;STEP TO SLAVE # 1
(1) 002724 022737 000010 000614  1$:  CMP    #10,SLVN       ;;BRANCH IF ALL SLAVES TESTED
(1) 002732 001733                    BEQ    NXTDRV         ;;FOR AVAILABILITY
(1) 002734 013777 000614 175600  MOV    SLVN,@TC        ;;LOAD SLAVE UNIT #
(1) 002742 032777 002000 175566  BIT    #2000,@DT      ;;BRANCH IF SLAVE NOT
(1) 002750 001751                    BEQ    NXTSLV        ;;PRESENT (SPR = 0)
2223 002752 032777 010000 175542  BIT    #10000,@DS     :BRANCH IF MOL
2224 002760 001026                    BNE    TSRH           :IF THERE BRANCH
2225 002762 012704 017103          MOV    #MSG71,R4       :ADDRESS OF MESSAGE
2226 002766 004737 013552          JSR    PC,TTOUT        :PRINT IT
2227 002772 013703 000612          MOV    DRVN,R3         :DEVICE NUMBER
2228 002776 004737 013702          JSR    PC,OCTP        :PRINT NUMBER
2229 003002 012704 017114          MOV    #MSG72,R4       :
2230 003006 004737 013552          JSR    PC,TTOUT        :PRINT MESSAGE
2231 003012 013703 000614          MOV    SLVN,R3         :SLAVE NUMBER
2232 003016 004737 013702          JSR    PC,OCTP        :PRINT IT
2233 003022 012704 017124          MOV    #MSG73,R4       :
2234 003026 004737 013552          JSR    PC,TTOUT        :PRINT MESSAGE
2235 003032 000720                    BR     NXTSLV

```

| | | | | | | | | |
|------|--------|--------|--------|--------|----------|-------|---------------|-------------------------------|
| 2236 | 003034 | 000240 | | | TSCDA: | NOP | | |
| 2237 | 003036 | 012737 | 000752 | 000706 | TSRH: | MOV | #TSTTBL,LTADD | |
| 2238 | 003044 | 062737 | 000004 | 000706 | TSCD0: | ADD | #4,LTADD | |
| 2239 | 003052 | 013737 | 000706 | 000676 | TSCD1: | MOV | LTADD,IIRLP | |
| 2240 | 003060 | 062737 | 000002 | 000676 | | ADD | #2,IIRLP | ;SET ITERATION ADDRESS |
| 2241 | 003066 | 005037 | 000660 | | | CLR | STMSK | |
| 2242 | 003072 | 005037 | 000626 | | | CLR | ERRP | |
| 2243 | 003076 | 005037 | 000606 | | | CLR | HDRFL | ;CLEAR PRINT HEADER FLAG |
| 2244 | 003102 | 017700 | 175600 | | | MOV | @LTADD,RO | ;SET POINTER TO TEST |
| 2245 | 003106 | 000110 | | | | JMP | (RO) | ;GO TO TEST |
| 2246 | 003110 | 032777 | 002000 | 175432 | TSCD2: | BIT | #2000,@SWR | ;SEE IF HALT ON TEST |
| 2247 | 003116 | 001401 | | | | BEQ | TSCD3 | ;IF NOT: BR |
| 2248 | 003120 | 000000 | | | | HALT | | |
| 2249 | 003122 | 005737 | 000704 | | TSCD3: | TST | STFLG | ;SE IF SINGLE TEST |
| 2250 | 003126 | 001746 | | | | BEQ | TSCD0 | ;IF NOT: BR |
| 2251 | 003130 | 017700 | 175414 | | | MOV | @SWR,RO | |
| 2252 | 003134 | 042700 | 177740 | | | BIC | #177740,RO | ;BRANCH IF ALL TESTS SELECTED |
| 2253 | 003140 | 001002 | | | | BNE | +6 | |
| 2254 | 003142 | 000137 | 002506 | | | JMP | TSCD | |
| 2255 | 003146 | 012737 | 000001 | 000704 | STSCD: | MOV | #1,STFLG | ;SET SINGLE TEST FLAG |
| 2256 | 003154 | 023700 | 001110 | | | CMR | TLAST,RO | ;SEE IF EXCEEDED TESTS |
| 2257 | 003160 | 002410 | | | | BLT | TEND | ;IF SO: BR |
| 2258 | 003162 | 006300 | | | | ASL | RO | |
| 2259 | 003164 | 006100 | | | | ROL | RO | ;SET TABLE MODIFIER |
| 2260 | 003166 | 012737 | 000752 | 000706 | | MOV | #TSTTBL,LTADD | |
| 2261 | 003174 | 060037 | 000706 | | | ADD | RO,LTADD | ;SET TEST POINTER |
| 2262 | 003200 | 000724 | | | | BR | TSCD1 | |
| 2263 | 003202 | 005737 | 001662 | | TEND: | TST | CHNFLG | ;BRANCH IF IN CHAIN MODE |
| 2264 | 003206 | 001232 | | | | BNE | NXTSLV | |
| 2265 | 003210 | 012704 | 014771 | | \$DONE: | MOV | #MSG6,R4 | |
| 2266 | 003214 | 004737 | 013552 | | | JSR | PC,TTOUT | ;PRINT END OF PASS |
| 2267 | 003220 | 013703 | 000730 | | | MOV | PCNTR,R3 | |
| 2268 | 003224 | 004737 | 013702 | | | JSR | PC,OCTP | ;PRINT PASS NUMBER |
| 2269 | 003230 | 005000 | | | | CLR | RO | |
| 2270 | 003232 | 005300 | | | 1\$: | DEC | RO | |
| 2271 | 003234 | 001376 | | | | BNE | 1\$ | |
| 2272 | 003236 | 013700 | 000042 | | | MOV | @#42,RO | ;GET ACT11 RETURN ADDRESS |
| (1) | 003242 | 001405 | | | | BEQ | HERE | ;BRANCH IF NOT ACT11 |
| (1) | 003244 | 000005 | | | | RESET | | |
| (1) | 003246 | 004710 | | | \$ENDAD: | JSR | PC,(RO) | |
| (1) | 003250 | 000240 | | | | NOP | | |
| (1) | 003252 | 000240 | | | | NOP | | |
| (1) | 003254 | 000240 | | | | NOP | | |
| (1) | 003256 | 000240 | | | HERE: | NOP | | |
| 2273 | 003260 | 005737 | 001662 | | | TST | CHNFLG | ;BRANCH IF IN CHAIN MODE |
| 2274 | 003264 | 001005 | | | | BNE | TENDX | |
| 2275 | 003266 | 032777 | 010000 | 175254 | | BIT | #10000,@SWR | ;SEE IF HALT ON PASS |
| 2276 | 003274 | 001001 | | | | BNE | TENDX | ;IF NOT: BR |
| 2277 | 003276 | 000000 | | | | HALT | | |
| 2278 | 003300 | 005237 | 000730 | | TENDX: | INC | PCNTR | ;BUMP PASS COUNTER |
| 2279 | 003304 | 000137 | 002506 | | | JMP | TSCD | ;RESTART |


```
2281
2282
2283 ;RH ADDRESSING TEST*****
2284 003310 012737 017140 000610 FT1: MOV #MSFT1,EMADDR ;SET HEADER
2285 003316 012777 013076 175250 MOV #TRAP,@BTRP ;SET TRAP HANDLER ADDRESS
2286 003324 012777 000340 175244 MOV #340,@BTRP2
2287 003332 012700 000016 MOV #16,R0 ;SET NUMBER OF REGISTERS
2288 003336 013701 000510 MOV C1,R1 ;GET FIRST ADDRESS (CS1)
2289 003342 005711 FT1A: TST (R1) ;REFERENCE REGISTER
2290 003344 000240 NOP ;IF ADDRESS IS BAD, BUS TRAP WILL OCCUR
2291 003346 005300 FT1B: DEC R0 ;SEE IF DONE ALL
2292 003350 001403 BEQ FT1X ;IF SO: BR
2293 003352 062701 000002 ADD #2,R1 ;BUMP ADDRESS POINTER
2294 003356 000771 BR FT1A ;CONTINUE
2295 003360 012777 000006 175206 FT1X: MOV #6,@BTRP ;RESET TRAP CATCHER
2296 003366 005737 000722 TST RHTF ;SEE IF INITIAL ADDRESS TEST PASS
2297 003372 001404 BEQ FT1XX ;IF NOT: BR
2298 003374 005037 000722 CLR RHTF ;CLEAR FLAG
2299 003400 000137 002170 JMP ST1A ;RETURN
2300 003404 000137 003110 FT1XX: JMP TSCD2 ;RETURN TO SCHEDULAR
```

```

2302
2303                ;RH REGISTER BITS READ/WRITE*****
2304
2305 003410 012737 017165 000610 FT2:  MOV    #MSFT2,EMADDR  ;SET TEST HEADER
2306 003416 012701 177777          MOV    #-1,R1        ;SET ALL ONES PATTERN
2307 003422 004737 012712          FT2A: JSR    PC,INIT1   ;GO INIT
2308 003426 013700 000512          MOV    WC,R0        ;GET ADDRESS OF WORD COUNT
2309 003432 010102          MOV    R1,R2        ;SET EXPT REGISTER BIT PATTERN
2310 003434 010110          MOV    R1,(R0)      ;LOAD PATTERN
2311 003436 021002          CMP    (R0),R2      ;SEE IF EXPT=RCVD
2312 003440 001410          BEQ   FT2B          ;IF SO: BR
2313 003442 012737 015315 000650          MOV    #MSG25,ERADD ;SET CODE
2314 003450 012737 003422 000674          MOV    #FT2A,SCOLP  ;SET SCOPE
2315 003456 004737 003576          JSR    PC,FT2ER     ;GO DO ERROR
2316 003462 013700 000514          FT2B: MOV    BA,R0     ;GET ADDRESS OF BUS ADDRESS
2317 003466 010102          MOV    R1,R2
2318 003470 042702 000001          BIC   #1,R2        ;SET EXPT PATTERN
2319 003474 010110          MOV    R1,(R0)     ;LOAD PATTERN
2320 003476 020210          CMP    R2,(R0)     ;SEE IF EXPT=RCVD
2321 003500 001410          BEQ   FT2C          ;IF SO:BR
2322 003502 012737 015323 000650          MOV    #MSG26,ERADD ;SET ERROR CODE
2323 003510 012737 003462 000674          MOV    #FT2B,SCOLP  ;SET SCOPE ADDRESS
2324 003516 004737 003576          JSR    PC,FT2ER     ;GO DO ERROR
2325 003522 013700 000532          FT2C: MOV    DB,R0     ;GET ADDRESS OF DATA BUFFER
2326 003526 010102          MOV    R1,R2
2327 003530 010110          MOV    R1,(R0)     ;LOAD PATTERN
2328 003532 012703 004000          MOV    #4000,R3
2329 003536 005303          FT2D: DEC    R3      ;DELAY
2330 003540 001376          BNE   FT2D
2331 003542 020210          CMP    R2,(R0)     ;SEE IF EXPT=RCVD
2332 003544 001410          BEQ   FT2E          ;IF SO: BR
2333 003546 012737 015331 000650          MOV    #MSG27,EPADD ;SET ERROR CODE
2334 003554 012737 003522 000674          MOV    #FT2C,SCOLP  ;SET SCOPE ADDRESS
2335 003562 004737 003576          JSR    PC,FT2ER     ;GO DO ERROR
2336 003566 005701          FT2E: TST    R1      ;SEE IF DONE RESET
2337 003570 001453          BEQ   FT2X          ;IF SO: BR
2338 003572 005001          CLR   R1            ;SET ZERO PATTERN
2339 003574 000712          BR   FT2A           ;DO ZERO BITS
2340 003576 000240          FT2ER: NOP
2341 003600 032777 020000 174742          BIT   #20000,@SWR  ;SEE IF PRINT ERROR
2342 003606 001034          BNE   FT2ERB       ;IF NOT: BR
2343 003610 005737 000606          TST   HDRFL        ;SEE IF DONE HEADER
2344 003614 001004          BNE   FT2ERA       ;IF SO: BR
2345 003616 013704 000610          MOV   EMADDR,R4
2346 003622 004737 013552          JSR   PC,TTOUT     ;DO HEADER
2347 003626 012737 000001 000606          FT2ERA: MOV  #1,HDRFL  ;SET FLAG
2348 003634 013704 000650          MOV   ERADD,R4
2349 003640 004737 013552          JSR   PC,TTOUT     ;PRINT ERROR CODE
2350 003644 012704 015261          MOV   #MSG22,R4
2351 003650 004737 013552          JSR   PC,TTOUT     ;PRINT EXPT TAG
2352 003654 010103          MOV   R1,R3
2353 003656 004737 013670          JSR   PC,OCTPE     ;PRINT EXPT
2354 003662 012704 015271          MOV   #MSG23,R4
2355 003666 004737 013552          JSR   PC,TTOUT     ;PRINT RCVD TAG
2356 003672 011003          MOV   (R0),R3
2357 003674 004737 013670          JSR   PC,OCTPE     ;PRINT RCVD
  
```

| | | | | | | | | |
|------|--------|--------|--------|---------|------|----------|--|-----------------------------|
| 2358 | 003700 | 005777 | 174644 | FT2ERB: | TST | @SWR | | ;SEE IF HALT ON ERROR |
| 2359 | 003704 | 100001 | | | BPL | FT2ERC | | ;IF NOT: BR |
| 2360 | 003706 | 000000 | | | HALT | | | |
| 2361 | 003710 | 004737 | 012604 | FT2ERC: | JSR | PC,SCOPE | | ;GO SEE IF SCOPE ON ERROR |
| 2362 | 003714 | 000240 | | | NOP | | | |
| 2363 | 003716 | 000207 | | | RTS | PC | | ;IF NO SCOPE: CONTINUE TEST |
| 2364 | 003720 | 000240 | | FT2X: | NOP | | | |
| 2365 | 003722 | 004737 | 012640 | | JSR | PC,ITER | | ;GO SEE IF ITERATIONS |
| 2366 | 003726 | 000137 | 003110 | | JMP | TSCD2 | | ;RETURN TO SCHEDULAR |

```

2368
2369 ;RH INITIALIZE TEST*****
2370
2371 003732 012737 017222 000610 FT3: MOV #MSG3,EMADDR ;SET TEST HEADER
2372 003740 012737 003732 000674 MOV #FT3,SCOLP
2373 003746 004737 012712 JSR PC,INIT1 ;GO INIT
2374 003752 052777 020000 174540 BIS #20000,@CS ;FORCE UPE =1
2375 003760 000240 NOP
2376 003762 004737 012712 JSR PC,INIT1 ;GO INIT
2377 003766 005777 174516 TST @C1 ;SEE IF SC IS RESET
2378 003772 100005 BPL FT3A ;IF SO: BR
2379 003774 012737 015367 000650 MOV #MSG29,ERADD ;SET ERROR CODE
2380 004002 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2381 004006 032777 040000 174474 FT3A: BIT #40000,@C1 ;SEE IF TRE IS RESET
2382 004014 001405 BEQ FT3B ;IF SO: BR
2383 004016 012737 015416 000650 MOV #MSG30,ERADD ;SET ERROR CODE.
2384 004024 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2385 004030 017701 174464 FT3B: MOV @CS,R1 ;GET CS2
2386 004034 042701 000307 BIC #307,R1 ;MARK IR/OR
2387 004040 005701 TST R1 ;SEE IF RESET
2388 004042 001405 BEQ FT3X ;IF SO: BR
2389 004044 012737 015446 000650 MOV #MSG31,ERADD ;SET ERROR CODE
2390 004052 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2391 004056 004737 012640 FT3X: JSR PC,ITER ;GO SEE IF ITERATION
2392 004062 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR
2393
2394 ;ERROR REPORT SUBROUTINE
2395 004066 000240 FT3ER: NOP
2396 004070 032777 020000 174452 BIT #20000,@SWR ;SEE IF PRINT ERROR
2397 004076 001015 BNE 2$ ;IF NOT: BR
2398 004100 005737 000606 TST HDRFL ;SEE IF DONE HEADER
2399 004104 001006 BNE 1$ ;IF SO: BR
2400 004106 013704 000610 MOV EMADDR,R4
2401 004112 004737 013552 JSR PC,TTOUT ;PRINT HEADER
2402 004116 005237 000606 INC HDRFL
2403 004122 013704 000650 1$: MOV ERADD,R4
2404 004126 004737 013552 JSR PC,TTOUT ;PRINT ERROR CODE
2405 004132 005777 174412 2$: TST @SWR ;SEE IF HALT ON ERROR
2406 004136 100001 BPL 3$ ;IF NOT: BR
2407 004140 000000 HALT
2408 004142 000240 3$: NOP
2409 004144 004737 012604 JSR PC,SCOPE ;GO SEE IF SCOPE
2410 004150 000207 RTS ;IF NOT: BR
  
```

```
2412  
2413 ;RH11 SILO TEST 1: EPMTY SILO READ*****  
2414  
2415 004152 005737 000604 FT4: TST RH17F  
2416 004156 001022 BNE FT4X ;IF RH70: BR  
2417 004160 012737 017254 000610 MOV #MSFT4,EMADDR ;SET TEST TEST HEADER  
2418 004166 012777 000040 174324 MOV #40,@CS ;INIT  
2419 004174 017700 174332 MOV @DB,R0 ;READ DB  
2420 004200 005777 174314 TST @CS ;SEE IF DLT IS SET  
2421 004204 100013 BPL FT4ER ;IF NOT: BR  
2422 004206 005777 174276 TST @C1 ;SEE IF SC IS SET  
2423 004212 100014 BPL FT4ERA ;IF NOT: BR  
2424 004214 032777 040000 174266 B!T #40000,@C1 ;SEE IF TRE IS SET  
2425 004222 001414 BEQ FT4ERB ;IF NOT: BR  
2426 004224 004737 012640 FT4X: JSR PC,ITER ;GO SEE IF ITERATION  
2427 004230 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR  
2428 004234 012737 015476 000650 FT4ER: MOV #MSG32,ERADD ;SET ERROR CODE  
2429 004242 000407 BR FT4ERC  
2430 004244 012737 015514 000650 FT4ERA: MOV #MSG33,ERADD ;SET ERROR CODE  
2431 004252 000403 BR FT4ERC  
2432 004254 012737 015531 000650 FT4ERB: MOV #MSG34,ERADD ;SET ERROR CODE.  
2433 004262 000240 FT4ERC: NOP  
2434 004264 012737 004152 000674 MOV #FT4,SCOLP ;SET SCOPE ADDRESS  
2435 004272 004737 004066 JSR PC,FT3ER ;GO PRINT ERROR  
2436 004276 000752 BR FT4X
```

```

2438
2439
2440
2441 004300 005737 000604 FT5: TST RH17F ;SEE IF RH70
2442 004304 001066 BNE FT5X ;IF SO: BR
2443 004306 012737 017304 000610 MOV #MSG35,EMADDR ;SET TEST HEADER
2444 004314 012737 004322 000674 MOV #FT5A,SCOLP ;SET SCOPE ADDRESS
2445 004322 004737 012712 FT5A: JSR PC,INIT1 ;GO INIT
2446 004326 032777 000100 174164 BIT #100,@CS ;SEE IF IR IS SET
2447 004334 001005 BNE FT5B ;IF SO: BR
2448 004336 012737 015547 000650 MOV #MSG35,ERADD ;SET ERROR CODE
2449 004344 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2450 004350 032777 000200 174142 FT5B: BIT #200,@CS ;SEE IF OR IS RESET
2451 004356 001405 BEQ FT5C ;IF SO: BR
2452 004360 012737 015574 000650 MOV #MSG36,ERADD ;SET ERROR CODE
2453 004366 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2454 004372 012777 000000 174132 FT5C: MOV #0,@DB ;LOAD ZERO INTO SILO
2455 004400 032777 000200 174112 BIT #200,@CS ;SEE THAT OR RESET
2456 004406 001405 BEQ FT5D ;IF IT DOES: BR
2457 004410 012737 015623 000650 MOV #MSG37,ERADD ;SET ERROR CODE
2458 004416 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2459 004422 012777 177777 174102 FT5D: MOV #-1,@DB ;LOAD SILO WITH -1
2460 004430 012700 004000 MOV #4000,R0
2461 004434 032777 000200 174056 FT5E: BIT #200,@CS ;SEE IF OR IS SET
2462 004442 001007 BNE FT5X ;IF SO: BR
2463 004444 005300 DEC R0
2464 004446 001372 BNE FT5E ;AWAIT OR
2465 004450 012737 015623 000650 MOV #MSG37,ERADD ;SET ERROR CODE
2466 004456 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2467 004462 004737 012640 FT5X: JSR PC,ITER ;GO SEE IF ITERATION
2468 004466 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR
  
```

```

2470
2471 ;RH11 SILO TEST 3: SILO DATA TEST*****
2472
2473 004472 005737 000604 FT6: TST RH17F
2474 004476 001052 BNE FT6X ;IF RH70: BR
2475 004500 012737 017334 000610 MOV #MSFT6,EMADDR ;SET TEST HEADER
2476 004506 012737 004514 000674 MOV #FT6A,SCOLP ;SET SCOPE ADDRESS
2477 004514 012737 012712 FT6A: JSR PC,INIT1 ;GO INIT
2478 004520 000000 CLR R0 ;PRESET DATA
2479 004522 010077 174004 FT6B: MOV R0,@DB ;LOAD SILO
2480 004526 005200 INC R0 ;BUMP DATA
2481 004530 022700 000102 CMP #102,R0 ;SEE IF FILLED ALL
2482 004534 001372 BNE FT6B ;IF NOT: BR
2483 004536 032777 000100 173754 BIT #100,@CS ;SEE IF IR IS RESET.
2484 004544 001405 BEQ FT6C ;IF SO: BR
2485 004546 012737 015734 000650 MOV #MSG40,ERADD ;SET ERROR CODE
2486 004554 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2487 004560 032777 000200 173732 FT6C: BIT #200,@CS ;SEE IF OR IS SET
2488 004566 001005 BNE FT6D ;IF SO: BR
2489 004570 012737 015662 000650 MOV #MSG38,ERADD ;SET ERROR CODE
2490 004576 004737 004066 JSR PC,FT3ER ;GO DO ERROR
2491 004602 005000 FT6D: CLR R0 ;PRESET DATA
2492 004604 017701 173722 FT6E: MOV @DB,R1 ;READ SILO
2493 004610 020001 CMP R0,R1 ;SEE IF EXPT=RCVD
2494 004612 001010 BNE FT6DE ;IF NOT: BR
2495 004614 005200 INC R0 ;BUMP DATA
2496 004616 022700 000102 CMP #102,R0 ;SEE IF DONE ALL
2497 004622 001370 BNE FT6E ;IF NOT: BR
2498 004624 004737 012640 FT6X: JSR PC,ITER ;GO SEE IF ITERATION
2499 004630 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR
2500
2501 004634 000240 FT6DE: NOP
2502 004636 032777 020000 173704 BIT #20000,@SWR ;SEE IF PRINT ERROR
2503 004644 001032 BNE FT6DEB ;IF NOT: BR
2504 004646 005737 000606 TST HDRFL ;SEE IF DONE HEADER
2505 004652 013701 000610 MOV EMADDR,R1
2506 004656 004737 013552 JSR PC,TTOUT ;PRINT HEADER
2507 004662 005237 000606 INC HDRFL ;SET FLAG
2508 004666 012704 015714 FT6DEA: MOV #MSG39,R4
2509 004672 004737 013552 JSR PC,TTOUT ;PRINT SILO READ ERROR
2510 004676 012704 015261 MOV #MSG22,R4
2511 004702 004737 013552 JSR PC,TTOUT ;PRINT EXPT TAG
2512 004706 010003 MOV R0,R3
2513 004710 004737 013702 JSR PC,OCTP ;PRINT EXPT
2514 004714 012704 015271 MOV #MSG23,R4
2515 004720 004737 013552 JSR PC,TTOUT ;PRINT RCVD TAG
2516 004724 010103 MOV R1,R3
2517 004726 004737 013702 JSR PC,OCTP ;PRINT RCVD
2518 004732 005777 173612 FT6DEB: TST @SWR ;SEE IF HALT ON ERROR
2519 004736 100001 BPL FT6DEX ;IF NOT: BR
2520 004740 000000 HALT
2521 004742 000207 FT6DEX: RTS PC ;RETURN TO TEST

```

```
2523  
2524 ;RH11 SILO TEST 4: SILO OVERFLOW*****  
2525  
2526 004744 005737 000604 FT7: TST RH17F  
2527 004750 001021 BNE FT7X ;IF RH70: BR  
2528 004752 012737 017364 000610 MOV #MSFT7,EMADDR ;SET TEST HEADER  
2529 004760 012737 004744 000674 MOV #FT7,SCOLP ;SET SCOPE ADDRESS  
2530 004766 004737 012712 JSR PC,INIT1 ;GO INIT  
2531 004772 012700 000103 MOV #103,RO ;SET SIZE OF SILO +1  
2532 004776 010077 173530 FT7A: MOV RO,@DB ;LOAD SILO  
2533 005002 005300 DEC RO ;SEE IF DONE  
2534 005004 001374 BNE FT7A ;IF NOT: BR  
2535 005006 005777 173506 TST @CS ;SEE IF DLT IS SET  
2536 005012 100004 BPL FT7ER ;IF NOT: BR  
2537 005014 004737 012640 FT7X: JSR PC,ITER ;GO SEE IF ITERATION  
2538 005020 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR  
2539 005024 012737 015476 000650 FT7ER: MOV #MSG32,ERADD ;SET ERROR CODE  
2540 005032 004737 004066 JSR PC,FT3ER ;GO DO ERROR  
2541 005036 000766 BR FT7X
```



```

2543
2544
2545
2546 005040 005737 000604          FT10:  TST      RH17F
2547 005044 001034                BNE      FT10X          ;IF RH70: BR
2548 005046 012737 017414 000610    MOV      #MSFT10,EMADDR ;SET TEST HEADER
2549 005054 012737 005040 000674    MOV      #FT10,SCOLP    ;SET SCOPE ADDRESS
2550 005062 012777 000040 173430    MOV      #40,@CS        ;INITIALIZE
2551 005070 012700 000004          MOV      #4,R0          ;SET NUMBER OF SILO WRITER
2552 005074 010077 173432          FT10A:  MOV      R0,@DB      ;WRITE SILO
2553 005100 005300                DEC      R0              ;SEE IF DONE
2554 005102 001374                BNE      FT10A          ;IF NOT: BR
2555 005104 052777 000040 173406    BIS      #40,@CS        ;INITIALIZE
2556 005112 012777 177777 173412    MOV      #-1,@DB        ;WRITE SILO
2557 005120 017701 173406          MOV      @DB,R1         ;READ SILO 1
2558 005124 017701 173402          MOV      @DB,R1         ;READ SILO 2
2559 005130 005777 173364          TST      @CS            ;SEE IF DLT IS SET
2560 005134 100011                BPL      FT10ER         ;IF NOT: BR
2561 005136 004737 012640          FT10X:  JSR      PC,ITER    ;GO SEE IF ITERATION
2562 005142 005737 000726          TST      RHOF           ;SEE IF RH11 ONLY
2563 005146 001402                BEQ      FT10XX         ;IF NOT: BR
2564 005150 000137 003202          JMP      TEND           ;ELSE GO TO END
2565 005154 000137 003110          FT10XX: JMP      TSCD2         ;RETURN TO SCHEDULAR
2566 005160 012737 015476 000650    FT10ER: MOV      #MSG32,ERADD ;SET ERROR CODE
2567 005166 004737 004066          JSR      PC,FT3ER      ;GO DO ERROR
2568 005172 000761                BR       FT10X
  
```

```

2570                                     ;NOP TEST*****
2571
2572 005174 000240                                     FT11: NOP
2573 005176 012737 005174 000674 MOV #FT11,SCOLP ;SET SCOPE ADDRESS
2574 005204 004737 012712 JSR PC,INIT1
2575 005210 012737 000300 000716 MOV #300,UDES ;SET TC= ALL NRZ,NORM,ODD
2576 005216 012737 177777 000620 MOV #-1,FCNT ;SET FC= ALL OVER
2577 005224 012737 177777 000622 MOV #-1,WCNT ;SET WC= ALL OVER
2578 005232 012737 177777 000616 MOV #-1,BADDR ;SET BA= ALL OVER
2579 005240 012737 000001 000636 MOV #1,RDYDX ;SET DELAY
2580 005246 012737 000001 000640 MOV #1,OPDYX ;SET OP DELAY
2581 005254 012737 000001 000710 MOV #1,FUN ;SET NOP FUNCTIONS CODE
2582 005262 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
2583 005266 000240 NOP
2584 005270 012737 017445 000610 MOV #MSFT11,EMADDR
2585 005276 004737 012070 JSR PC,ERCHK ;GO CHECK REGISTER
2586 005302 004737 012640 JSR PC,ITER ;GO SEE IF ITERATIONS
2587 005306 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR
  
```

```
2589                                     ;REWIND TEST*****
2590
2591 005312 000240                         FT12: NOP
2592 005314 012737 005312 000674         MOV #FT12,SCOLP
2593 005322 004737 012712                 JSR PC,INIT1 ;GO INITIALIZE
2594 005326 052777 001700 173206         BIS #1700,@TC ;SET TO NRZ,NORMAL
2595 005334 012737 177760 000620         MOV #-20,FCNT ;SET FC=20
2596 005342 012737 177770 000622         MOV #-10,WCNT ;SET WC=10
2597 005350 012737 020260 000616         MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
2598 005356 012737 000007 000710         MOV #7,FUN ;SET REWIND OP CODE
2599 005364 004737 011640                 JSR PC,EXEC ;GO EXECUTE COMMAND
2600 005370 000240                         NOP
2601 005372 032777 020000 173122         FT12A: BIT #20000,@DS
2602 005400 001374                         BNE FT12A ;AWAIT PIP
2603 005402 012737 017465 000610         MOV #MSFT12,EMADDR
2604 005410 004737 012070                 JSR PC,ERCHK ;GO CHECK FOR ERROR
2605 005414 004737 012640                 JSR PC,ITER ;GO SEE IF ITERATION
2606 005420 000137 003110                 JMP TSCD2 ;RETURN TO SCHEDULAR
2607
```

```

2609                                     ;WRITE/READ TEST*****
2610
2611 005424 000240 FT13: NOP
2612 005426 012737 000001 000636 MOV #1,RDYDX
2613 005434 012737 000001 000640 MOV #1,OPDYX
2614 005442 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
2615 005450 012737 017510 000610 MOV #MSFT13,EMADDR ;SET TEST HEADER
2616 005456 012737 000001 000720 MOV #1,PATRN
2617 005464 004737 012472 JSR PC,DSUP ;SET UP ALL ONES DATA PATTERN
2618 005470 012737 001700 000716 MOV #1700,UDES ;SET TO 800 BPI NORMAL
2619 005476 004737 011772 FT13A: JSR PC,RWIND ;GO REWIND
2620 005502 012737 177600 000620 MOV #-200,FCNT ;SET FC
2621 005510 012737 177700 000622 MOV #-100,WCNT ;SET WC
2622 005516 012737 020260 000616 MOV #WDATA,BADDR ;SET BA
2623 005524 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
2624 005532 012737 015075 000626 FT13B: MOV #MSG12,ERRP
2625 005540 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
2626 005544 005037 000674 CLR SCOLP ;NO SCOPE LOOP
2627 005550 004737 012070 JSR PC,ERCHK ;GO CHECK ERROR
2628 005554 005337 000624 DEC RCNT ;SEE IF DONE ALL
2629 005560 001367 BNE FT13B ;IF NOT: BR
2630 005562 012737 000100 000624 MOV #100,RCNT ;SET RECORD COUNT
2631 005570 012737 021772 000616 MOV #RDATA,BADDR
2632 005576 062737 000200 000616 ADD #200,BADDR ;SET BA
2633 005604 012737 000077 000710 MOV #77,FUN ;SET READ REVERSE OP-CPDE
2634 005612 012737 015113 000626 FT13C: MOV #MSG13,ERRP
2635 005620 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
2636 005624 004737 012070 JSR PC,ERCHK ;GO CHECK ERROR
2637 005630 005337 000624 DEC RCNT ;SEE IF READ ALL
2638 005634 001371 BNE FT13C ;IF NOT:BR
2639 005636 162737 000200 000616 SUB #200,BADDR ;SET BA
2640 005644 012737 000071 000710 MOV #71,FUN ;SET READ FORWARD OP-CODE
2641 005652 012737 015140 000626 MOV #MSG14,ERRP
2642 005660 012737 000100 000624 FT13D: MOV #100,RCNT ;SET RECORD COUNT
2643 005666 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
2644 005672 004737 012070 JSR PC,ERCHK ;GO CHECK ERRORS
2645 005676 005337 000624 DEC RCNT ;SEE IF DONE ALL
2646 005702 001371 BNE FT13D ;IF NOT:BR
2647 005704 032737 002000 000716 BIT #2000,UDES ;SEE IF DONE PE
2648 005712 001007 BNE FT13X ;IF SO: BR
2649 005714 012737 002300 000716 MOV #2300,UDES ;SET PE MODE
2650 005722 012737 000100 000624 MOV #100,RCNT ;RESET RECORD COUNT
2651 005730 000662 BR FT13A ;GO DO NEXT DENSITY
2652 005732 000137 003110 FT13X: JMP TSCD2 ;RETURN TO SCHEDULAR

```

```

2654 ;SPACE TEST*****
2655
2656 005736 000240 FT14: NOP
2657 005740 012737 017537 000610 MOV #MSFT14,EMADDR ;SET TEST HEADER
2658 005746 012737 001700 000716 MOV #1700,UDES ;SET NRZ,NORMAL
2659 005754 004737 011772 FT14A1: JSR PC,RWIND ;GO INITIALIZE
2660 005760 012737 000100 000624 MOV #100,RCNT ;SET NUMBER OF RECORDER
2661 005766 012737 177777 020260 MOV #-1,WDATA ;SET DATA PATTERN
2662 005774 012737 177700 000620 MOV #-100,FCNT ;PRESET FRAME CNT
2663 006002 012737 177740 000622 MOV #-40,WCNT ;PRESET WORD CNT
2664 006010 004737 012712 FT14A: JSR PC,INIT1 ;GO REWIND
2665 006014 012737 001000 000640 MOV #1000,OPDYX
2666 006022 012737 040000 000636 MOV #40000,RDYDX
2667 006030 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
2668 006036 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
2669 006044 052777 000010 172446 BIS #10,ACS ;INHIBIT BUS ADDRESS INCREMENT
2670 006052 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
2671 006056 012737 016153 000626 MOV #MSG46,ERRP ;SET ERROR CODE
2672 006064 004737 012070 JSR PC,ERCHK ;GO CHECK ERRORS
2673 006070 005737 000712 TST SERFL ;SEE IF ERROR
2674 006074 001402 BEQ FT14A2 ;IF NOT: BR
2675 006076 000137 006562 JMP FT14X ;ELSE EXIT
2676 006102 005337 000620 FT14A2: DEC FCNT ;BUMP FC
2677 006106 032737 000001 000620 BIT #1,FCNT ;SEE IF SHOULD BUMP WC
2678 006114 001403 BEQ FT14A3 ;IF NOT: BR
2679 006116 162737 000001 000622 SUB #1,WCNT ;BUMP WC
2680 006124 005337 000624 FT14A3: DEC RCNT ;SEE IF DONE ALL
2681 006130 001327 BNE FT14A ;WRITE ALL RECORDS
2682 006132 012737 000100 000632 MOV #100,RRD ;PRESET RECORD POSITION
2683 006140 012737 000176 000634 MOV #176,RFD
2684 006146 012737 177701 000642 MOV #-77,SCNT ;SET SPACE AMOUNT
2685 006154 012737 000033 000710 FT14B: MOV #33,FUN ;SET OP-CODE SPACE REVERSE
2686 006162 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
2687 006166 012737 016224 000626 MOV #MSG48,ERRP ;SET ERROR CODE
2688 006174 004737 012070 JSR PC,ERCHK ;GO CHECK ERRORS
2689 006200 005737 000712 TST SERFL ;SEE IF ERROR
2690 006204 001166 BNE FT14X ;IF SO: BR
2691 006206 004737 006302 JSR PC,FT14RR ;GO READ REVERSE + CHECK DATA
2692 006212 000240 NOP
2693 006214 012737 000031 000710 MOV #31,FUN ;SET SPACE FORWARD OP-CODE
2694 006222 005237 000642 INC SCNT ;SET SPACE AMOUNT
2695 006226 001555 BEQ FT14X ;IF DONE: BR
2696 006230 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
2697 006234 012737 016177 000626 MOV #MSG47,ERRP ;SET ERROR CODE
2698 006242 004737 012070 JSR PC,ERCHK ;GO CHECK ERROR
2699 006246 005737 000712 TST SERFL ;SEE IF ERROR FLAG
2700 006252 001143 BNE FT14X ;IF NO: BR
2701 006254 004737 006344 JSR PC,FT14RF ;GO READ FORWARD FOR POSITION CHECK
2702 006260 000240 NOP
2703 006262 005237 000642 INC SCNT ;DECREMENT SPACE AMOUNT
2704 006266 001535 BEQ FT14X ;IF DONE: BR
2705 006270 005237 000632 INC RRD ;BUMP DATA EXPT
2706 006274 005337 000634 DEC RFD ;BUMP DATA EXPT
2707 006300 000725 BR FT14B
2708 006302 000240 FT14RR: NOP
2709 006304 012737 021772 000616 MOV #RDATA,BADDR ;SET BA

```

| | | | | | | | |
|------|--------|--------|--------|--------|--------------|-------------|-----------------------------|
| 2710 | 006312 | 012737 | 000077 | 000710 | MOV | #77,FUN | :SET READ REVERSE OP-CODE |
| 2711 | 006320 | 004737 | 011640 | | JSR | PC,EXEC | :GO EXECUTE COMMAND |
| 2712 | 006324 | 000240 | | | NOP | | |
| 2713 | 006326 | 013705 | 000632 | | MOV | R5,R5 | |
| 2714 | 006332 | 020577 | 172160 | | CMP | R5,@FC | :SEE IF CORRECT RECORD |
| 2715 | 006336 | 001020 | | | BNE | FT14RER | :IF NOT: BR |
| 2716 | 006340 | 000137 | 006372 | | JMP | FT14EC | :GO CLEAR RH11 ERROR BIT |
| 2717 | 006344 | 000240 | | | FT14RF: NOP | | |
| 2718 | 006346 | 012737 | 000071 | 000710 | MOV | #71,FUN | :SET READ FORWARD OP-CODE |
| 2719 | 006354 | 004737 | 011640 | | JSR | PC,EXEC | :GO EXECUTE COMMAND |
| 2720 | 006360 | 013705 | 000634 | | MOV | R5,R5 | |
| 2721 | 006364 | 020577 | 172126 | | CMP | R5,@FC | :SEE IF CORRECT RECORD |
| 2722 | 006370 | 001003 | | | BNE | FT14RER | :IF NOT: BR |
| 2723 | 006372 | 004737 | 012712 | | FT14EC: JSR | PC,INIT1 | :CLEAR RH |
| 2724 | 006376 | 000207 | | | RTS | PC | :RETURN |
| 2725 | 006400 | 000240 | | | FT14RER: NOP | | |
| 2726 | 006402 | 032777 | 020000 | 172140 | BIT | #20000,@SWR | :SEE IF PRINT INHIBITED |
| 2727 | 006410 | 001060 | | | BNE | FT14R3 | :IF SO: BR |
| 2728 | 006412 | 012704 | 017537 | | MOV | #MSFT14,R4 | |
| 2729 | 006416 | 004737 | 013552 | | JSR | PC,TTOUT | :PRINT HEADER |
| 2730 | 006422 | 012704 | 015013 | | MOV | #MSG9,R4 | |
| 2731 | 006426 | 004737 | 013552 | | JSR | PC,TTOUT | :PRINT ERROR TYPE |
| 2732 | 006432 | 012704 | 015246 | | MOV | #MSG20,R4 | :SET NRZ TAG POINTER |
| 2733 | 006436 | 032737 | 002000 | 000716 | BIT | #2000,UDES | :SEE IF PE |
| 2734 | 006444 | 001402 | | | BEQ | FT14R0 | :IF NOT: BR |
| 2735 | 006446 | 012704 | 015254 | | MOV | #MSG21,R4 | :ELSE SET PE TAG POINTER |
| 2736 | 006452 | 004737 | 013552 | | FT14R0: JSR | PC,TTOUT | :PRINT TAG |
| 2737 | 006456 | 032737 | 000002 | 000710 | BIT | #2,FUN | :SEE IF READ REVERSE |
| 2738 | 006464 | 001003 | | | BNE | FT14R1 | :IF SO: BR |
| 2739 | 006466 | 012704 | 015226 | | MOV | #MSG17,R4 | |
| 2740 | 006472 | 000402 | | | BR | FT14R2 | :GO PRINT |
| 2741 | 006474 | 012704 | 015206 | | FT14R1: MOV | #MSG16,R4 | |
| 2742 | 006500 | 004737 | 013552 | | FT14R2: JSR | PC,TTOUT | :PRINT FRWD/REV |
| 2743 | 006504 | 012704 | 015261 | | MOV | #MSG22,R4 | |
| 2744 | 006510 | 004737 | 013552 | | JSR | PC,TTOUT | :PRINT EXPT TAG |
| 2745 | 006514 | 010503 | | | MOV | R5,R3 | |
| 2746 | 006516 | 042703 | 177700 | | BIC | #177700,R3 | :MASK RECORD NUMBER |
| 2747 | 006522 | 004737 | 013702 | | JSR | PC,OCPT | :PRINT EXPT RECORD NUMBER |
| 2748 | 006526 | 012704 | 015271 | | MOV | #MSG23,R4 | |
| 2749 | 006532 | 004737 | 013552 | | JSR | PC,TTOUT | :PRINT RCVD TAG |
| 2750 | 006536 | 017703 | 171754 | | MOV | @FC,R3 | |
| 2751 | 006542 | 042703 | 177700 | | BIC | #177700,R3 | :MASK RECORD NUMBER |
| 2752 | 006546 | 004737 | 013702 | | JSR | PC,OCPT | :PRINT ACTUAL RECORD NUMBER |
| 2753 | 006552 | 005777 | 171772 | | FT14R3: TST | @SWR | :SEE IF HALT ON ERROR |
| 2754 | 006556 | 100001 | | | BPL | FT14X | :IF NOT: BR |
| 2755 | 006560 | 000000 | | | HALT | | |
| 2756 | 006562 | 032737 | 002000 | 000716 | FT14X: BIT | #2000,UDES | :SEE IF DONE PE |
| 2757 | 006570 | 001005 | | | BNE | FT14XX | :IF SO: BR |
| 2758 | 006572 | 012737 | 002300 | 000716 | MOV | #2300,UDES | :SET TO PE |
| 2759 | 006600 | 000137 | 005754 | | JMP | FT14A1 | :DO IN PE |
| 2760 | 006604 | 000137 | 003110 | | FT14XX: JMP | TSCD2 | :RETURN TO SCHEDULAR |

```

2762                                     ;ERASE TEST*****
2763
2764 006610 000240          FT15:  NOP
2765 006612 005037          CLR      STMSK
2766 006616 012737 000660  MOV     #100,RDYDX
2767 006624 012737 000010 000640  MOV     #10,OPDYX
2768 006632 012737 017561 000610  MOV     #MSFT15,EMADDR ;SET TEST HEADER
2769 006640 004737 011772          JSR     PC,RWIND      ;REWIND
2770 006644 012737 021772 000616  MOV     #RDATA,BADDR ;SET BA
2771 006652 012737 001700 000716  MOV     #1700,UDES    ;SET NRZ, NORMAL
2772 006660 012737 000025 000710  FT15A: MOV     #25,FUN    ;SET ERASE OP-CODE
2773 006666 012737 000400 000624  MOV     #400,RCNT    ;SET TO ERASE 256 TIMES
2774 006674 004737 011640          FT15B: JSR     PC,EXEC    ;GO EXECUTE COMMAND
2775 006700 012737 016153 000626  MOV     #MSG46,ERRP  ;SET ERROR CODE
2776 006706 004737 012070          JSR     PC,ERCHK    ;GO CHECK ERRORS
2777 006712 005737 000712          TST     SERFL      ;SEE IF ANY ERRORS
2778 006716 001032          BNE     FT15X      ;IF SO EXIT
2779 006720 005337 000624          DEC     RCNT       ;SEE IF DONE ERASING
2780 006724 001363          BNE     FT15B      ;IF NOT: BR
2781 006726 000240          NOP
2782 006730 004737 011772          JSR     PC,RWIND    ;REWIND
2783 006734 012737 177600 000622  MOV     #-200,WCNT   ;SET WC
2784 006742 012737 000071 000710  MOV     #71,FUN     ;SET READ FORWARD OP-CODE
2785 006750 012737 000040 000636  MOV     #40,RDYDX   ;SET DELAY
2786 006756 004737 011640          JSR     PC,EXEC    ;GO EXECUTE COMMAND
2787 006762 000240          NOP
2788 006764 012737 016615 000626  MOV     #MSG60,ERRP  ;SET ERROR CODE
2789 006772 012737 020000 000660  MOV     #20000,STMSK
2790 007000 004737 012070          JSR     PC,ERCHK    ;GO CHECK ERRORS
2791 007004 000137 003110          FT15X: JMP     TSCD2    ;RETURN TO SCHEDULAR

```

i

```

2793                                     ;TAPE MARK WRITE/READ TEST*****
2794
2795 007010 000240                      FT16:  NOP
2796 007012 012737 000001 000636      MOV    #1,RDYDX
2797 007020 012737 001000 000640      MOV    #1000,OPDYX
2798 007026 012737 017603 000610      MOV    #MSGFT16,EMADDR ;SET HEADER
2799 007034 012737 001700 000716      MOV    #1700,UDES      ;SET TO NRZ,NORMAL,ODD
2800 007042 004737 011772              FT16A: JSR    PC,RWIND   ;INIT AND REWIND SLAVE
2801 007046 012737 177760 000620      FT16B: MOV    #-20,FCNT  ;FC=20
2802 007054 012737 177770 000622      MOV    #-10,WCNT      ;WC=10
2803 007062 012737 000027 000710      MOV    #27,FUN        ;SET WRITE TAPE MARK OP-CODE
2804 007070 004737 011640              JSR    PC,EXEC         ;GO EXECUTE COMMAND
2805 007074 012737 001000 000660      MOV    #1000,STMSK    ;SET FOR FCE MASK
2806 007102 012737 015075 000626      MOV    #MSG12,ERRP    ;SET ERROR CODE
2807 007110 004737 012070              JSR    PC,ERCHK        ;GO CHECK ERROR
2808 007114 004737 012432              JSR    PC,TMCHK       ;GO SEE IF TM SET
2809 007120 012737 000077 000710      MOV    #77,FUN        ;SET USED REVERSE OP-CODE
2810 007126 004737 011640              JSR    PC,EXEC         ;GO EXECUTE COMMAND
2811 007132 012737 001000 000660      MOV    #1000,STMSK    ;SET FCE ERROR MASK
2812 007140 012737 015113 000626      MOV    #MSG13,ERRP    ;SET ERROR CODE
2813 007146 004737 012070              JSR    PC,ERCHK        ;GO CHECK ERRORS
2814 007152 004737 012432              JSR    PC,TMCHK       ;GO SEE IF TM SET
2815 007156 012737 000071 000710      MOV    #71,FUN        ;SET READ FORWARD OP-CODE
2816 007164 004737 011640              JSR    PC,EXEC         ;GO EXECUTE COMMAND
2817 007170 012737 015140 000626      MOV    #MSG14,ERRP    ;SET ERROR CODE
2818 007176 004737 012070              JSR    PC,ERCHK        ;TO CHECK ERRORS
2819 007202 004737 012432              JSR    PC,TMCHK       ;GO SEE IF TM SET
2820 007206 032737 002000 000716      BIT    #2000,UDES     ;SEE IF DONE PE
2821 007214 001004                      BNE    FT16X          ;IF SO: BR
2822 007216 012737 002300 000716      MOV    #2300,UDES     ;SET PE, NORMAL
2823 007224 000706                      BR     FT16A          ;DO IN PE
2824 007226 004737 012640              FT16X: JSR    PC,ITER   ;DO ITERATIONS
2825 007232 000137 003110              JMP    TSCD2          ;RETURN TO SCHEDULAR
2826

```



```
2828
2829
2830 ;TAPE MARK SPACE TEST*****
2831 007236 005037 000624 FT17: CLR RCNT
2832 007242 012737 017644 000610 MOV #MSFT17,EMADDR ;SET HEADER
2833 007250 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ
2834 007256 004737 011772 FT17A: JSR PC,RWIND ;REWIND TAPE
2835 007262 012737 000027 000710 FT17B: MOV #27,FUN
2836 007270 012737 040000 000636 MOV #40000,RDYDX ;SET DRY DELAY
2837 007276 012737 040000 000640 MOV #40000,OPDYX ;SET OP DELAY
2838 007304 004737 011640 JSR PC,EXEC ;GO WRITE TM
2839 007310 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
2840 007316 012737 015165 000626 MOV #MSG15,ERRP ;SET ERROR TYPE
2841 007324 004737 012070 JSR PC,ERCHK ;GO CHECK ERROR
2842 007330 005737 000712 TST SERFL ;SEE IF ERROR
2843 007334 001137 BNE FT17X ;IF SO: BR
2844 007336 004737 012432 JSR PC,TMCHK ;GO SEE IF TM SET
2845 007342 000240 NOP
2846 007344 000240 NOP
2847 007346 032737 000100 000624 BIT #100,RCNT ;SEE IF DONE PATTERN
2848 007354 001045 BNE FT17D ;IF SO: BR
2849 007356 062737 000020 000624 ADD #20,RCNT ;ADD 20 TO RECORD COUNT
2850 007364 013737 000624 000652 MOV RCNT,TEMP1 ;SAVE RECORD COUNT
2851 007372 012737 177600 000622 MOV #-200,WCNT ;WC=128
2852 007400 012737 177400 000620 MOV #-400,FCNT ;FC=256
2853 007406 012737 020260 000616 MOV #WDATA,BADDR ;BA=WRITE BUFFER
2854 007414 012737 000061 000710 MOV #61,FUN ;SET WRITE OP CODE
2855 007422 000240 FT17C: NOP
2856 007424 000240 NOP
2857 007426 004737 011640 JSR PC,EXEC ;GO WRITE
2858 007432 012737 015075 000626 MOV #MSG12,ERRP ;SET ERROR CODE
2859 007440 012737 102300 000660 MOV #102300,STMSK ;MASK DATA RELATED ERRORS
2860 007446 004737 012070 JSR PC,ERCHK ;GO CHECK ERROR
2861 007452 005737 000712 TST SERFL ;SEE IF ERROR
2862 007456 001066 BNE FT17X ;IF SO: BR
2863 007460 005337 000652 DEC TEMP1 ;SEE IF DONE ALL
2864 007464 001356 BNE FT17C ;IF NOT: BR
2865 007466 000675 BR FT17B ;ELSE GO DO TM
2866 007470 000240 FT17D: NOP
2867 007472 012737 000033 000710 MOV #33,FUN ;SET SPACE REVERSE
2868 007500 012737 015206 000626 MOV #MSG16,ERRP ;SET ERROR CODE
2869 007506 012737 177600 000642 FT17D1: MOV #-200,SCNT ;SET TO 200 RECORDS
2870 007514 012737 000005 000624 MOV #5,RCNT ;SET NUMBER OF OPS TO DO
2871 007522 004737 012712 FT17E: JSR PC,INIT1 ;GO INIT
2872 007526 004737 011640 JSR PC,EXEC ;GO SPACE
2873 007532 012737 001000 000660 MOV #1000,STMSK ;SET ERROR MASK
2874 007540 004737 012070 JSR PC,ERCHK ;GO CHECK ERROR
2875 007544 005737 000712 TST SERFL ;SEE IF ERROR
2876 007550 001031 BNE FT17X ;IF SO: BR
2877 007552 004737 012432 JSR PC,TMCHK ;GO SEE IF TM SET
2878 007556 005337 000624 DEC RCNT ;SEE IF DONE SPACES
2879 007562 001357 BNE FT17E ;IF NOT: BR
2880 007564 022737 000031 000710 CMP #31,FUN ;SEE IF DONE FORWARD
2881 007572 001407 BEQ FT17F ;IF SO: BR
2882 007574 012737 015226 000626 MOV #MSG17,ERRP ;SET ERROR CODE
2883 007602 012737 000031 000710 MOV #31,FUN ;SET TO SPACE FORWARD
```

| | | | | | | | | | |
|------|--------|--------|--------|--------|--------|-----|-------------|--|----------------------|
| 2884 | 007610 | 000736 | | | | BR | FT17D1 | | :DO FORWARD |
| 2885 | 007612 | 032737 | 002000 | 000716 | FT17F: | BIT | #2000, UDES | | :SEE IF DONE PE |
| 2886 | 007620 | 001005 | | | | BNE | FT17X | | :IF SO: BR |
| 2887 | 007622 | 012737 | 002300 | 000716 | | MOV | #2300, UDES | | :SET TO PE |
| 2888 | 007630 | 000137 | 007256 | | | JMP | FT17A | | :GO PE |
| 2889 | 007634 | 000137 | 003110 | | FT17X: | JMP | TSCD2 | | :RETURN TO SCHEDULAR |

```

2891
2892
2893
2894 007640 000240
2895 007642 012737 017672 000610
2896 007650 012737 001700 000716
2897 007656 004737 011772
2898 007662 012737 000003 000720
2899 007670 004737 012472
2900 007674 012737 020260 000616
2901 007702 012737 177400 000620
2902 007710 012737 177600 000622
2903 007716 012737 000061 000710
2904 007724 004737 011640
2905 007730 012737 016153 000626
2906 007736 004737 012070
2907 007742 005737 000712
2908 007746 001042
2909 007750 012737 015206 000626
2910 007756 012737 000057 000710
2911 007764 062737 000376 000616
2912 007772 004737 011640
2913 007776 004737 012070
2914 010002 012737 015226 000626
2915 010010 012737 000051 000710
2916 010016 162737 000376 000616
2917 010024 004737 011640
2918 010030 004737 02070
2919 010034 032737 002000 000716
2920 010042 001004
2921 010044 012737 002300 000716
2922 010052 000701
2923 010054 004737 012640
2924 010060 000157 003110

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

```

2926
2927
2928 ;ERASE, HEAD TEST*****
2929 010064 012737 017723 000610 FT21: MOV #MSFT21,EMADDR ;SET TEST HEADER
2930 010072 004737 011772 FT21A: JSR PC,RWIND ;GO REWIND
2931 010076 012737 000003 000720 MOV #3,PATRN
2932 C10104 004737 012472 JSR PC,DSUP ;GO SET PATTERN 3
2933 010110 012737 020260 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
2934 010116 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
2935 010124 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
2936 010132 012737 001700 000716 MOV #1700,UDES ;SET NRZ, NORMAL
2937 010140 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
2938 010146 004737 011640 JSR PC,EXEC ;GO DO WRITE 1
2939 010152 012737 015075 000626 MOV #MSG12,ERRP ;SET ERROR CODE
2940 010160 004737 012070 JSR PC,ERCHK ;GO CHECK FOR ERROR
2941 010164 004737 011640 JSR PC,EXEC ;YES DO WRITE 2
2942 010170 004737 012070 JSR PC,ERCHK ;YES CHECK FOR ERROR
2943 010174 000240 NOP
2944 010176 004737 011772 JSR PC,RWIND ;GO REWIND
2945 010202 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400(10)
2946 010210 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200(10)
2947 010216 004737 011640 JSR PC,EXEC ;GO REWRITE RECORD 1-WH TO EH
2948 010222 000240 FT21SCP:NOP
2949 010224 004737 011772 JSR PC,RWIND ;REWIND
2950 010230 012737 021772 000616 MOV #RDATA,BADDR ;SET BA=READ BUFFER
2951 010236 012737 177160 000620 MOV #-400.,FCNT ;SET FC=400
2952 010244 012737 177470 000622 MOV #-200.,WCNT ;SET WC=200
2953 010252 012737 000071 000710 MOV #71,FUN ;SET READ OP-CODE
2954 010260 004737 011640 JSR PC,EXEC ;GO READ RECORD 1
2955 010264 012737 015140 000626 MOV #MSG14,ERRP ;SET ERROR CODE
2956 010272 004737 012070 JSR PC,ERCHK ;GO CHECK FOR ERROR
2957 010276 000240 NOP
2958 010300 052777 000010 170212 BIS #10,@CS ;INHIBIT BA INCREMENT
2959 010306 012737 176340 000620 MOV #-800.,FCNT ;SET FC=800(10)
2960 010314 012737 177160 000622 MOV #-400.,WCNT ;SET WC=400(10)
2961 010322 004737 011640 JSR PC,EXEC ;GO READ RECORD 2
2962 010326 022777 001440 170162 CMP #800.,@FC ;SEE IF READ RECORD 2 OK
2963 010334 001424 BEQ FT21X ;IF SO: BR
2964 010336 022777 001441 170152 CMP #801.,@FC ;BRANCH IF IN GREY AREA
2965 010344 001420 BEQ FT21X
2966 010346 022777 001440 170142 1$: CMP #800.,@FC ;BRANCH IF ERASE HEAD REVERSED
2967 010354 010140 BLOS FT21B ;IF SO: BR
2968 010356 012737 016046 000650 MOV #MSG44,ERADD ;SET ERASE HEAD INOPERATIVE ERROR CODE
2969 010364 000403 BR FT21C
2970 010366 012737 016076 000650 FT21B: MOV #MSG45,ERADD ;SET ERASE HEAD REVERSED ERROR CODE
2971 010374 012737 010222 000674 FT21C: MOV #FT21SCP,SCOLP ;SET SCOPE ADDRESS
2972 010402 004737 004066 JSR PC,FT3ER ;GO PRINT ERROR
2973 010406 004737 012640 FT21X: JSR PC,ITER ;GO SEE IF ITERATION
2974 010412 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR
2975
2976

```

```

2978                                     ;BUFFERED COMMAND TEST*****
2979
2980 010416 012737 017752 000610 FT22: MOV #MSFT22,EMADDR ;SET TEST HEADER
2981 010424 004737 011772 JSR PC,RWIND ;GO REWIND
2982 010430 012700 000003 MOV #3,RO ;SET NUMBER OF WRITES
2983 010434 012737 001700 000716 MOV #1700,UDES ;SET TO NRZ NORMAL
2984 010442 012737 C20260 000616 MOV #WDATA,BADDR ;SET BA=WRITE BUFFER
2985 010450 012737 177000 000620 MOV #-1000,FCNT ;SET FC=1000
2986 010456 012737 177400 000622 MOV #-400,WCNT ;SET WC=400
2987 010464 012737 000061 000710 MOV #61,FUN ;SET WRITE OP-CODE
2988 010472 004737 011640 FT22A: JSR PC,EXEC ;GO DO WRITE
2989 010476 005300 DEC RO ;SEE IF DONE ALL
2990 010500 001374 BNE FT22A ;IF NOT: BR
2991 010502 000240 NOP
2992 010504 012777 000007 167776 MOV #7,@C1 ;START REWIND
2993 010512 032777 000200 170002 FT22B: BIT #200,@DS
2994 010520 001774 BEQ FT22B
2995 010522 004737 012712 JSR PC,INIT1 ;INITIALIZE
2996 010526 012737 000010 000636 MOV #10,RDYDX ;SET LONG READY DELAY
2997 010534 004737 011640 JSR PC,EXEC ;ISSUE BUFFERED WRITE
2998 010540 000240 NOP
2999 010542 012737 016251 000626 MOV #MSG49,ERRP ;SET ERROR CODE
3000 010550 012737 102300 000660 MOV #102300,STMSK ;MARK DATA ERROR
3001 010556 004737 012070 JSR PC,ERCHK ;GO CHECK ERROR
3002 010562 032777 000002 167732 BIT #2,@DS ;SEE IF BOT IS SET
3003 010570 001410 BEQ FT22X ;IF NOT: BR
3004 010572 012737 016277 000650 MOV #MSG50,ERADD ;SET ERROR CODE
3005 010600 012737 010416 000674 MOV #FT22,SCOLP
3006 010606 004737 004066 JSR PC,FT3ER ;GO DO ERROR
3007 010612 004737 012640 FT22X: JSR PC,ITER ;GO SEE IF ITERATION
3008 010616 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULAR
3009
3010
  
```

```

3012                                     ;READ-IN PRESET TEST*****
3013
3014 010622 005737 000614          FT23: TST      SLVN          ;SEE IF SLAVE SELECT=0
3015 010626 001103                    BNE      FT23X        ;IF NOT:BR
3016 010630 012737 020007 000610    MOV      #MSFT23,EMADDR ;SET TEST HEADER
3017 010636 004737 012712                    JSR      PC,INIT1     ;GO INIT
3018 010642 012737 001700 000716    MOV      #1700,UDES    ;SET TO NRZ NORMAL
3019 010650 012737 020260 000616    MOV      #WDATA,BADDR ;SET BA=WRITE BUFFER
3020 010656 012737 177400 000620    MOV      #-400,FCNT   ;SET FC=400
3021 010664 012737 177600 000622    MOV      #-200,WCNT   ;SET WC=200
3022 010672 012737 000061 000710    MOV      #61,FUN      ;SET WRITE OP-CODE
3023 010700 004737 011640                    JSR      PC,EXEC      ;GO DO WRITE
3024 010704 000240                    NOP
3025 010706 004737 012712                    JSR      PC,INIT1     ;INITIALIZE
3026 010712 012737 000021 000710    MOV      #21,FUN      ;SET READ-IN PRESET OP CODE
3027 010720 004737 011640                    JSR      PC,EXEC      ;GO DO COMMAND
3028 010724 005000                    CLR      RO
3029 010726 012703 000004                    MOV      #4,R3        ;SET MULT
3030 010732 032777 020000 167562    FT23A: BIT      #20000,ADS ;SEE IF PIP RESET
3031 010740 001404                    BEQ      FT23B        ;IF SO: BR
3032 010742 005300                    DEC      RO
3033 010744 001372                    BNE      FT23A        ;AWAIT PIP RESET
3034 010746 005303                    DEC      R3
3035 010750 001370                    BNE      FT23A        ;DELAY
3036 010752 032777 000002 167542    FT23B: BIT      #2,ADS  ;SEE IF BOT
3037 010760 001010                    BNE      FT23C        ;IF SO: BR
3038 010762 012737 016335 000650    MOV      #MSG51,ERADD ;SET ERROR CODE
3039 010770 012737 010622 000674    MOV      #FT23,SCOLP
3040 010776 004737 004066                    JSR      PC,FT3ER     ;GO DO ERROR
3041 011002 012701 141000          FT23C: MOV      #141000,R1 ;SET EXPT TC
3042 011006 013700 000542                    MOV      TC,RO        ;SET TC ADDRESS
3043 011012 020110                    CMP      R1,(RO)      ;SEE IF EXPT=RCVD
3044 011014 001410                    BEQ      FT23X        ;IF SO: BR
3045 011016 012737 016371 000650    MOV      #MSG52,ERADD ;SET ERROR CODE
3046 011024 012737 010622 000674    MOV      #FT23,SCOLP ;CLEAR SCOPE ADDRESS
3047 011032 004737 003576                    JSR      PC,FT2ER     ;GO DO ERROR
3048 011036 000137 003110          FT23X: JMP      TSCD2   ;RETURN TO SCHEDULAR
3049
3050

```

```

3052
3053
3054
3055 011042 012737 020076 000610 FT24: MOV #MSFT24,EMADDR ;SET ERROR MSG HEADER
3056 011050 004737 011772 JSR PC,RWIND ;REWIND SLAVE
3057 011054 012737 000001 000720 MOV #1,PATRN ;SELECT PATTERN
3058 011062 004737 012472 JSR PC,DSUP ;GO DO DATA SETUP
3059 011066 012737 020260 000616 MOV #WDATA,BADDR ;SET BUS ADDRESS,
3060 011074 012737 177400 000620 MOV #-400,FCNT ;FRAME COUNT,
3061 011102 012737 177600 000622 MOV #-200,WCNT ;WORD COUNT,
3062 011110 012737 001700 000716 MOV #1700,UDES ;& SLAVE DESC = NRZ NORMAL
3063 011116 012737 000061 000710 MOV #61,FUN ;LOAD OP CODE WRITE FWD
3064 011124 004737 011640 JSR PC,EXEC ;GO EXECUTE COMMAND
3065 011130 012737 016153 000626 MOV #MSG46,ERRP ;SET ERROR MSG ADDRESS
3066 011136 004737 012070 JSR PC,ERCHK ;GO CHECK ERRORS
3067 011142 005737 000712 TST SERFL ;BRANCH IF AN ERROR OCCURRED
3068 011146 001026 BNE FT24X
3069 011150 004737 011772 JSR PC,RWIND ;REWIND SLAVE
3070 011154 012737 021772 000616 MOV #RDATA,BADDR ;SET BUS ADDRESS FOR READ
3071 011162 012737 002300 000716 MOV #2300,UDES ;SET SLAVE DESC = PE,NORMAL
3072 011170 012737 000071 000710 MOV #71,FUN ;SET OP CODE = READ FWD
3073 011176 004737 011640 JSR PC,EXEC ;GO READ RECORD
3074 011202 032777 000040 167312 BIT #40,@DS ;BRANCH ID PES BIT CLEARED
3075 011210 001405 BEQ FT24X
3076 011212 012737 016714 000650 MOV #MSG63,ERADD
3077 011220 004737 004066 JSR PC,FT3ER ;GO PROCESS ERROR
3078 011224 004737 012640 FT24X: JSR PC,ITER
3079 011230 000137 003110 JMP TSCD2 ;RETURN TO SCHEDULER
3080
  
```

```

3082
3083
3084 011234 012737 020154 000610 :AUTO-DENSITY SELECT TEST: WRITE-PE,READ-NRZ
3085 011242 004737 011772 FT25: MOV #MSFT25,EMADDR :SET ERROR MESSAGE ADDRESS
3086 011246 012737 000001 000720 JSR PC,RWIND :REWIND SLAVE
3087 011254 004737 012472 MOV #1,PATRN :SELECT PATTERN
3088 011260 012737 020260 000616 JSR PC,DSUP :GO DO DATA SETUP
3089 011266 012737 177400 000620 MOV #WDATA,BADDR :SET BUS ADDRESS
3090 011274 012737 177600 000622 MOV #-400,FCNT :FRAME COUNT,
3091 011302 012737 002300 000716 MOV #-200,WCNT :WORD COUNT,
3092 011310 012737 000061 000710 MOV #2300,UDES :& SLAVE DESC = PE,NORMAL
3093 011316 004737 011640 JSR #61,FUN :LOAD WRITE OP CODE
3094 011322 012737 016153 000626 JSR PC,EXEC :GO EXECUTE WRITE
3095 011330 004737 012070 MOV #MSG46,ERRP :SET ERROR MSG HDR
3096 011334 005737 000712 JSR PC,ERCHK :GO CHECK FOR ERRORS
3097 011340 001026 TST SERFL :BRANCH IF ERROR OCCURED
3098 011342 004737 011772 BNE FT25X
3099 011346 012737 021772 000616 JSR PC,RWIND :REWIND SLAVE
3100 011354 012737 001700 000716 MOV #RDATA,BADDR :SET BUS ADDRESS FOR READ
3101 011362 012737 000071 000710 MOV #1700,UDES :SET SLAVE DESC = NRZ,NORMAL
3102 011370 004737 011640 JSR #71,FUN :SET READ FWD OP CODE
3103 011374 032777 000040 167120 JSR PC,EXEC :GO EXECUTE
3104 011402 001005 BIT #40,ADS :BRANCH ID PES BIT GOT SET
3105 011404 012737 016745 000650 BNE FT25X
3106 011412 004737 004066 MOV #MSG64,ERADD
3107 011416 004737 012640 FT25X: JSR PC,FT3ER :GO PROCESS ERROR
3108 011422 000137 003110 JSR PC,ITER :ITERATION LOOP
3109 JMP TSCD2 :RETURN TO SCHEDULER
3110 :REWIND: OFF LINE TEST*****
3111
3112 011426 032777 010000 167114 FT26: BIT #10000,@SWR :SEE IF IN CONTINUOUS MODE
3113 011434 001077 BNE FT26XX :IF SO: BR
3114 011436 005737 001662 TST CHNFLG :BRANCH IF CHAIN MODE
3115 011442 001074 BNE FT26XX
3116 011444 012737 020042 000610 MOV #MSFT26,EMADDR :SET TEST HEADER
3117 011452 004737 011772 JSR PC,RWIND :REWIND & SELECT SLAVE
3118 011456 012737 000001 000720 MOV #1,PATRN :SELECT PATTERN (ALL 1'S)
3119 011464 004737 012472 JSR PC,DSUP :FILL WRITE BUFFER
3120 011470 012737 020260 000616 MOV #WDATA,BADDR :SET WRITE BUFFER BUS ADDRESS
3121 011476 012737 177400 000620 MOV #-400,FCNT :SET FRAME COUNT
3122 011504 012737 177600 000622 MOV #-200,WCNT :SET WORD COUNT
3123 011512 012737 001700 000716 MOV #1700,UDES :SET UNIT DESCRIPTION - NRZ
3124 011520 012737 000061 000710 MOV #61,FUN :SET WRITE COMMAND
3125 011526 004737 011640 JSR PC,EXEC :GO WRITE A RECORD
3126 011532 012777 000003 166750 MOV #3,@C1 :ISSUE REWIND: OFF LINE COMMAND
3127 011540 005037 000674 CLR SCOLP :CLEAR SCOPE LOOP
3128 011544 012700 004000 MOV #4000,RO
3129 011550 005300 1$: DEC RO :DELAY
3130 011552 001376 BNE 1$
3131 011554 032777 010000 166740 BIT #10000,ADS :SEE IF MOL IS RESET
3132 011562 001406 BEQ 2$ :IF SO: BR
3133 011564 012737 016410 000650 MOV #MSG53,ERADD :SET ERROR CODE
3134 011572 004737 004066 JSR PC,FT3ER :GO DO ERROR
3135 011576 000412 BR FT26X
3136 011600 013700 000524 2$: MOV ER,RO :GET ADDRESS OF ERROR REG
3137 011604 005001 CLR R1 :RESULT SHOULD BE 0

```


| | | | | | | | |
|------|--------|--------|--------|--------|-------------|--------------|--------------------------|
| 3138 | 011606 | 020110 | | | CMP | R1,(R0) | ;BRANCH IF ERROR REG = 0 |
| 3139 | 011610 | 001405 | | | BEQ | FT26X | |
| 3140 | 011612 | 012737 | 017001 | 000650 | MOV | #MSG67,ERADD | ;SET ERROR MSG HEADER |
| 3141 | 011620 | 004737 | 003576 | | JSR | PC,FT2ER | ;GO TYPE ERROR |
| 3142 | 011624 | 012704 | 016435 | | FT26X: MOV | #MSG54,R4 | |
| 3143 | 011630 | 004737 | 013552 | | JSR | PC,TIOUT | ;PRINT ON LINE REQUEST |
| 3144 | 011634 | 000137 | 003110 | | FT26XX: JMP | TSCD2 | ;RETURN TO SCHEDULER |
| 3145 | | | | | | | |

```

3147                                     ;COMMAND EXECUTE SUBROUTINE*****
3148
3149 011640 000240                               EXEC:  NOP
3150 011642 053777 000716 166672             BIS     UDES,@TC      ;LOAD TAPE CONT
3151 011650 013777 000622 166634             MOV     WCNT,@WC     ;LOAD WC
3152 011656 013777 000620 166632             MOV     FCNT,@FC     ;LOAD FC
3153 011664 013777 000616 166622             MOV     BADDR,@BA   ;LOAD BA
3154 011672 022737 000031 000710             CMP     #31,FUN     ;SEE IF SPACE FORWARD
3155 011700 001404                               BEQ     EXECA        ;IF SO: BR
3156 011702 022737 000033 000710             CMP     #33,FUN     ;SEE IF SPACE REVERSE
3157 011710 001003                               BNE     EXECB        ;IF NOT: BR
3158 011712 013777 000642 166576             EXECA:  MOV     SCNT,@FC ;SET SPACE COUNT
3159 011720 000240                               EXECB:  NOP
3160 011722 013777 000710 166560             MOV     FUN,@C1     ;LOAD OP-CODE + GO
3161 011730 000240                               NOP
3162 011732 013703 000636                               MOV     RDYDX,R3    ;SET DELAY
3163 011736 005004                               CLR     R4
3164 011740 032777 000200 166554             EXECC:  BIT     #200,@DS ;SEE IF DRY
3165 011746 001004                               BNE     EXECX        ;IF SO: BR
3166 011750 005304                               DEC     R4
3167 011752 001372                               BNE     EXECC
3168 011754 005303                               DEC     R3           ;DELAY FOR DRY
3169 011756 001370                               BNE     EXECC
3170 011760 013703 000640                               EXECX:  MOV     OPDYX,R3
3171 011764 005303                               EXECXA: DEC     R3           ;DELAY
3172 011766 001376                               BNE     EXECXA
3173 011770 000207                               EXECXX: RTS          ;RETURN TO CALLER
3174

```

```
3176                                     ;REWIND SUBROUTINE*****
3177
3178 011772 000240                RWND:  NOP
3179 011774 004737 012712        JSR   PC,INIT1           ;INIT
3180 012000 012777 000007 166502 MOV   #7,@C1           ;START REWIND
3181 012006 012700 040000        MOV   #40000,R0
3182 012012 005300                RWNDA: DEC   R0
3183 012014 001376                BNE   RWNDA           ;DELAY
3184 012016 032777 020000 166476 RWNDB: BIT   #20000,@DS
3185 012024 001374                BNE   RWNDB           ;AWAIT PIP
3186 012026 032777 000002 166466 BIT   #2,@DS           ;SEE IF BOT
3187 012034 001012                BNE   RWNDX           ;IF SO: BR
3188 012036 013704 000610        MOV   EMADDR,R4
3189 012042 004737 013552        JSR   PC,TTOUT        ;PRINT HEADER
3190 012046 012704 014562        MOV   #MSG2,R4
3191 012052 004737 013552        JSR   PC,TTOUT        ;PRINT REWIND ERROR
3192 012056 000137 003110        JMP   TSCD2           ;RETURN TO SECHEDULAR
3193 012062 004737 012712        RWNDX: JSR  PC,INIT1   ;INIT
3194 012066 000207                RTS   PC              ;RETURN TO CALLER
3195
```

```

3197                                     :ERROR CHECK SUBROUTINE*****
3198
3199 012070 005037 000712          ERCHK: CLR      SERFL      :CLEAR FLAG
3200 012074 017737 166422 000664  MOV      @DS,DSAV    :SAVE DRIVE STATUS REGISTER
3201 012102 032777 040000 166412  BIT      #40000,@DS  :SEE IF ERROR
3202 012110 001001                BNE      ERPT        :IF SO: BR
3203 012112 000207                RTS      PC           :RETURN
3204 012114 017704 166404          ERPT:  MOV      @ER,R4    :GET ERROR REGISTER
3205 012120 032737 002000 000716  BIT      #2000,UDES  :SEE IF PE
3206 012126 001403                BEQ      ERPTA1      :IF SO: BR
3207 012130 042737 000200 000660  BIC      #200,STMSK  :RESET PEF MASK
3208 012136 043704 000660          ERPTA1: BIC      STMSK,R4   :MASK DONT CARE BITS
3209 012142 001530                BEQ      ERPTX        :IF NO UNEXPECTED ERRORS: BR
3210 012144 012737 000001 000712  ERPTG: MOV      #1,SERFL  :SET FLAG
3211 012152 032777 020000 166370  BIT      #20000,@SWR :SEE IF SHOULD PRINT ERRORS
3212 012160 001115                BNE      ERPTD        :IF NOT: BR
3213 012162 005737 000606          TST      HDRFL       :SEE IF DONE HEADER
3214 012166 001006                BNE      ERPTA        :IF SO: BR
3215 012170 005237 000606          INC      HDRFL       :SET HEADER FLAG
3216 012174 013704 000610          MOV      EMADDR,R4
3217 012200 004737 013552          JSR      PC,TTOUT    :PRINT HEADER
3218 012204 013704 000626          ERPTA: MOV      ERRP,R4  :GET ERROR CODE
3219 012210 001414                BEQ      ERPTB        :IF NONE: BR
3220 012212 004737 013552          JSR      PC,TTOUT    :PRINT ERROR CODE
3221 012216 012704 015246          MOV      #MSG20,R4   :SET NRZ TAG
3222 012222 032777 002000 166312  BIT      #2000,@TC   :SEE IF PE
3223 012230 001402                BEQ      ERPT1A       :IF NOT: BR
3224 012232 012704 015254          MOV      #MSG21,R4   :ELSE SET PE TAG
3225 012236 004737 013552          ERPT1A: JSR      PC,TTOUT  :PRINT TAG
3226 012242 013704 000630          ERPTB: MOV      ERRP1,R4 :SEE IF CODE 2
3227 012246 001402                BEQ      ERPTB1       :IF NOT: BR
3228 012250 004737 013552          JSR      PC,TTOUT    :PRINT CODE 2
3229 012254 032777 004000 166266  ERPTB1: BIT      #4000,@SWR  :SEE IF ITERATION
3230 012262 001010                BNE      ERPTC        :IF NOT: BR
3231 012264 012704 016571          MOV      #MSG56,R4
3232 012270 004737 013552          JSR      PC,TTOUT    :PRINT ITER TAG
3233 012274 013703 000662          MOV      ITCNT,R3
3234 012300 004737 013702          JSR      PC,OCTP     :PRINT ITERATION
3235 012304 012704 014474          ERPTC: MOV      #MSG1,R4
3236 012310 004737 013552          JSR      PC,TTOUT    :PRINT REGISTER TAG
3237 012314 017703 166170          MOV      @C1,R3
3238 012320 004737 013670          JSR      PC,OCTPE    :PRINT CS1
3239 012324 017703 166162          MOV      @WC,R3
3240 012330 004737 013670          JSR      PC,OCTPE    :PRINT WC
3241 012334 017703 166154          MOV      @BA,R3
3242 012340 004737 013670          JSR      PC,OCTPE    :PRINT BA
3243 012344 017703 166146          MOV      @FC,R3
3244 012350 004737 013670          JSR      PC,OCTPE    :PRINT FC
3245 012354 017703 166140          MOV      @CS,R3
3246 012360 004737 013670          JSR      PC,OCTPE    :PRINT CS2
3247 012364 017703 166132          MOV      @DS,R3
3248 012370 004737 013670          JSR      PC,OCTPE    :PRINT DS
3249 012374 017703 166124          MOV      @ER,R3
3250 012400 004737 013670          JSR      PC,OCTPE    :PRINT ER
3251 012404 017703 166132          MOV      @TC,R3
3252 012410 004737 013670          JSR      PC,OCTPE    :PRINT TC

```

| | | | | | | | |
|------|--------|--------|--------|---------|------|----------|-----------------------|
| 3253 | 012414 | 005777 | 166130 | ERPTD: | TST | @SWR | :SEE IF HALT ON ERROR |
| 3254 | 012420 | 100001 | | | BPL | ERPTX | :IF NOT: BR |
| 3255 | 012422 | 000000 | | | HALT | | |
| 3256 | 012424 | 004737 | 012712 | ERPTX: | JSR | PC,INIT1 | :INIT |
| 3257 | 012430 | 000207 | | ERPTXX: | RTS | PC | :RETURN |
| 3258 | | | | | | | |
| 3259 | | | | | | | |

```

3261                                     ;TAPE MARK STATUS CHECK*****
3262
3263 012432 032737 000004 000664 TMCHK: BIT #4, DSAV ;SEE IF TM SET
3264 012440 001401 BEQ TMCHK1 ;IF NOT: BR
3265 012442 000207 TMCHK0: RTS PC ;ELSE RETURN
3266 012444 005737 000712 TMCHK1: TST SERFL ;SEE IF HAD ERROR
3267 012450 001374 BNE TMCHK0 ;IF SO: BR
3268 012452 012737 016601 000630 MOV #MSG57, ERRP1 ;SET ERPOP CODE 2
3269 012460 004737 012144 JSR PC, ERFTG ;GO PRINT TM ERROR
3270 012464 005037 000630 CLR ERRP1 ;CLEAR CODE 2 FLAG
3271 012470 000207 RTS PC ;RETURN
3272
3273                                     ;DATA SETUP ROUTINE*****
3274
3275 012472 000240 DSUP: NOP
3276 012474 012703 DSO: MOV #WDATA, R3 ;R3 = ADDRS OF WRITE BUFFER
3277 012500 013701 MOV PATRN, R1 ;R1 = PATTERN SELECTOR
3278 012504 006301 ASL R1 ;MAKE PATTERN SELECTOR EVEN
3279 012506 004771 000740 JSR PC, @DATBL(R1) ;GO GENERATE PATTERN
3280 012512 012702 000640 MOV #640, R2 ;R2=BUFFER SIZE +2
3281 012516 012701 021772 MOV #RDATA, R1 ;R1=READ DATA START
3282 012522 005021 I$: CLR (R1)+ ;CLEAR BUFFER
3283 012524 005302 DEC R2 ;SEE IF DONE ALL
3284 012526 001375 BNE I$ ;IF NOT: BR
3285 012530 000207 RTS PC ;EXIT
3286
3287                                     ;ALL ONES*****
3288
3289 012532 012701 177777 DAT1: MOV #-1, R1 ;R1=DATA
3290 012536 012702 000640 DAT1A: MOV #640, R2 ;R2=WORD COUNT +2
3291 012542 010123 I$: MOV R1, (R3)+ ;LOAD BUFFER
3292 012544 005302 DEC R2 ;SEE IF DONE
3293 012546 001375 BNE I$ ;IF NOT: BR
3294 012550 000207 RTS PC
3295
3296                                     ;ALL ZEROS*****
3297
3298 012552 005001 DAT2: CLR R1 ;R1=DATA
3299 012554 000770 BR DAT1A ;LOAD BUFFER
3300
3301                                     ;ONE/ZERO IN ALTERNATING CHARACTERS*****
3302
3303 012556 012701 125125 DAT3: MOV #125125, R1 ;R1=DATA
3304 012562 000765 BR DAT1A ;LOAD BUFFER
3305
3306                                     ;ALL BITS 0-377*****
3307
3308 012564 005001 DAT4: CLR R1 ;R1=STARTING DATA
3309 012566 012702 001500 MOV #1500, R2 ;R2=CHARACTER COUNT
3310 012572 110123 I$: MOV R1, (R3)+ ;LOAD BUFFER
3311 012574 105201 INCB R1 ;BUMP DATA
3312 012576 005302 DEC R2 ;SEE IF DONE
3313 012600 001374 BNE I$ ;IF NOT: DR
3314 012602 000207 RTS PC
3315

```

```

3317
3318
3319
3320           ;SCOPE LOOP ON ERROR SUBROUTINE*****
3321
3322 012604 000240          SCOPE:  NOP
3323 012606 032777 040000 165734  BIT      #40000,@SWR      ;SEE IF LOOP ON ERROR
3324 012614 001001          BNE      1$           ;IF SO: BR
3325 012616 000207          RTS      PC           ;ELSE EXIT
3326 012620 000240          1$:    NOP
3327 012622 005737 000674  TST      SCOLP        ;SEE IF SCOPE ADDRESS
3328 012626 001001          BNE      2$           ;IF NOT: BR
3329 012630 000207          RTS      PC           ;ELSE EXIT
3330 012632 022626          2$:    CMP      (SP)+,(SP)+  ;RESET STACK
3331 012634 000177 166034  JMP      @SCOLP      ;LOOP ON ERROR
3332
3333           ;TEST ITERATION SUBROUTINE*****
3334
3335 012640 000240          ITER:  NOP
3336 012642 032777 004000 165700  BIT      #4000,@SWR      ;SEE IF ITERATIONS
3337 012650 001403          BEQ     2$           ;IF SO: BR
3338 012652 005037 000662  1$:    CLR      ITCNT        ;CLEAR ITERATION COUNTER
3339 012656 000207          RTS      PC           ;ELSE EXIT
3340 012660 005737 000730  2$:    TST      PCNTR      ;DO SINGLE SUBTEST ITERATION
3341 012664 001772          BEQ     1$           ;ON FIRST PASS
3342 012666 005237 000662  INC      ITCNT        ;BUMP COUNTER
3343 012672 023737 000662 000566  CMP      ITCNT,ITAMT   ;SEE IF DONE ALL
3344 012700 001764          BEQ     1$           ;IF SO: BR
3345 012702 005726          TST      (SP)+       ;RESET STACK
3346 012704 017700 165766  MOV     @ITRLP,R0     ;SET ITERATION POINTER
3347 012710 000110          JMP     (R0)         ;GO ITERATE
3348
3349           ;INITIALIZE SUBROUTINE*****
3350
3351 012712 000240          INIT1: NOP
3352 012714 012777 000040 165576  MOV     #40,@CS       ;INIT
3353 012722 013777 000612 165570  INIT2: MOV     DRVN,@CS    ;SELECT DRIVE
3354 012730 013777 000614 165604  MOV     SLVN,@TC     ;SELECT SLAVE
3355 012736 000207          RTS     PC           ;RETURN
3356
  
```

```

3358                                     ;MAG TAPE INTERRUPT HANDLER*****
3359
3360 012740 000240                               MTINT: NOP
3361 012742 013716 000646                     MOV     RTRN,(SP)      ;RETURN TO (RTRN)
3362 012746 000002                               RTI                    ;RETURN
3363
3364                                     ;TTY INTERRUPT HANDLER*****
3365
3366 012750 017746 165600                       TTINT: MOV     @TKB,-(SP)  ;GET CHARACTER
3367 012754 042716 000200                       BIC     #200,(SP)     ;CLEAR PARITY BIT
3368 012760 122716 000003                       CMPB    #3,(SP)      ;BRANCH IF NOT CONTROL C
3369 012764 001010                               BNE     1$
3370 012766 005737 001662                       TST     CHNFLG        ;INHIBIT ^C IF CHAIN MODE
3371 012772 001005                               BNE     1$
3372 012774 005077 165546                       CLR     @PSW
3373 013000 000005                               RESET
3374 013002 000137 000200                       JMP     @#200         ;RESTART PROGRAM
3375 013006 122716 000001                       1$:  CMPB    #1,(SP)     ;BRANCH IF NOT ^A
3376 013012 001017                               BNE     2$
3377 013014 022737 000176 000550                CMP     #SWREG,SWR    ;BRANCH IF HARDWARE SWR IS INVOKED
3378 013022 001016                               BNE     3$
3379 013024 012737 177570 000550                MOV     #177570,SWR   ;INVOKE HARDWARE SWR
3380 013032 004737 014430                       JSR     PC,SAVE        ;SAVE REGISTERS ON THE STACK
3381 013036 012704 017055                       MOV     #MSG70,R4     ;TYPE 'HARDWARE SWR IN USE'
3382 013042 004737 013552                       JSR     PC,TTOUT
3383 013046 004737 014452                       JSR     PC,RESTORE
3384 013052 122716 000007                       2$:  CMPB    #7,(SP)     ;BRANCH IF NOT ^G
3385 013056 001005                               BNE     4$
3386 013060 012737 000176 000550                3$:  MOV     #SWREG,SWR  ;INVOKE SOFTWARE SWR
3387 013066 004737 014332                       JSR     PC,GTSWR      ;GET SOFTWARE SWITCHES
3388 013072 005726                               4$:  TST     (SP)+      ;POP CHARACTER OFF THE STACK
3389 013074 000002                               RTI
3390
3391                                     ;BUS ADDRESS TRAP HANDLER*****
3392
3393 013076 000240                               TRAP:  NOP
3394 013100 052777 020000 165442                BIT     #20000,@SWR   ;SEE IF SHOULD PRINT ERRORS
3395 013106 001020                               BNE     TRAP2         ;IF NOT: BR
3396 013110 005737 000606                       TST     HDRFL        ;SEE IF DONE HEADER
3397 013114 001006                               BNE     TRAP1         ;IF SO: BR
3398 013116 005237 000606                       INC     HDRFL        ;ELSE SET HEADER FLAG
3399 013122 013704 000610                       MOV     EMADDR,R4
3400 013126 004737 013552                       JSR     PC,TTOUT     ;PRINT HEADER
3401 013132 012704 015301                       TRAP1: MOV     #MSG24,R4
3402 013136 004737 013552                       JSR     PC,TTOUT     ;PRINT ERROR
3403 013142 010103                               MOV     R1,R3        ;GET ADDRESS THAT CAUSED THE TRAP
3404 013144 004737 013702                       JSR     PC,OCTP      ;PRINT ADDRESS OF TRAP
3405 013150 005777 165374                       TRAP2: TST     @SWR   ;SEE IF HALT ON ERROR
3406 013154 100001                               BPL     TRAPX        ;IF NOT: BR
3407 013156 000000                               HALT
3408 013160 022626                               TRAPX: CMP     (SP)+,(SP)+ ;RESET STACK
3409 013162 012737 003342 000674                MOV     #FT1A,SCOLP  ;SET SCOPE ADDRESS
3410 013170 004737 012604                       JSR     PC,SCOPE     ;GO SEE IF SCOPE LOOP
3411 013174 005737 000722                       TST     RHTF        ;SEE IF INITIAL ADDRESS TEST
3412 013200 001402                               BEQ     TRAPXX       ;IF NOT: BR
3413 013202 000137 001766                       JMP     STOB        ;ELSE REDO ADDRESS REQUEST

```


TMO3/TU45 BASIC FUNCTION TEST MACY11 30A(1052) 13-MAY-80 15:25 F 5
CZTUQG.P11 13-MAY-80 15:20 PAGE 48-1

SEQ 0057

3414 013206 000137 003346 TRAPXX: JMP FT1B ;RETURN TO TEST 1
3415

```

3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434 013212 010146 TTR: MOV R1, -(SP) ;SAVE CHAR COUNT ON STACK
3435 013214 011601 10$: MOV (SP), R1 ;RESTORE CHAR COUNT (FOR ^U)
3436 013216 005037 000652 CLR TEMP1 ;CLEAR FIRST CHARACTER FLAG
3437 013222 005000 CLR R0
3438 013224 004737 013472 1$: JSR PC, TTIN ;GO READ CHARACTER
3439 013230 122737 000903 000602 CMPB #3, TIB ;BRANCH IF NOT ^C
3440 013236 001003 BNE 11$
3441 013240 000005 RESET ;RESET
3442 013242 000137 000200 JMP @#200 ;RESTART
3443 013246 122737 000015 000602 11$: CMPB #15, TIB ;SEE IF CR
3444 013254 001004 BNE 2$ ;IF NOT: BR
3445 013256 005737 000652 TST TEMP1 ;SEE IF FIRST CHARACTER
3446 013262 001471 BEQ 9$ ;IF SO: BR
3447 013264 000457 BR 6$ ;ELSE GO LOAD VALUE
3448 013266 122737 000025 000602 2$: CMPB #25, TIB ;BRANCH IF NOT CONTROL U
3449 013274 001005 BNE 21$
3450 013276 012704 016775 MOV #MSG65, R4 ;TYPE <CR><LF>
3451 013302 004737 013552 JSR PC, TTOUT
3452 013306 000742 BR 10$ ;RESTART
3453 013310 122737 000177 000602 21$: CMPB #177, TIB ;BRANCH IF NOT 'RUBOUT'
3454 013316 001012 BNE 3$
3455 013320 000241 CLC ;REMOVE LAST CHARACTER
3456 013322 006000 ROR R0
3457 013324 006200 ASR R0
3458 013326 006200 ASR R0
3459 013330 012704 016777 MOV #MSG66, R4 ;TYPE '\ '
3460 013334 004737 013552 JSR PC, TTOUT
3461 013340 005201 INC R1 ;DECREMENT CHAR RECEIVED COUNT
3462 013342 000730 BR 1$ ;GET NEXT CHARACTER
3463 013344 122737 000060 000602 3$: CMPB #60, TIB ;SEE IF CHAR IS LESS THAN 0
3464 013352 101402 BLOS 4$ ;IF NOT: BR
3465 013354 000137 013452 JMP TIB ;ELSE GO TO ERROR
3466 013360 122737 000070 000602 4$: CMPB #70, TIB ;SEE IF CHAR IS GREATER THAN 7
3467 013366 101002 BHI 5$ ;IF NOT: BR
3468 013370 000137 013452 JMP TIB ;ELSE GO TO ERROR
3469 013374 005237 000652 5$: INC TEMP1 ;SET FIRST CHARACTER FLAG
3470 013400 006300 ASL R0
3471 013402 006300 ASL R0 ;SHIFT 3 LEFT
3472 013404 006300 ASL R0

```

```

3473 013406 042737 177770 000602      BIC      #177770,TIB      ;STRIP ASCII
3474 013414 053700 000602      BIS      TIB,R0          ;LOAD CHARACTER
3475 013420 005301              DEC      R1              ;SEE IF DONE
3476 013422 001300              BNE     1$              ;IF NOT: BR
3477 013424 020002              CMP     R0,R2          ;SEE IF EXCEEDED MAXIMUM LIMIT
3478 013426 101402              BLOS   7$              ;IF NOT: BR
3479 013430 000137 013452      JMP     T1NER          ;ELSE GO TO ERROR
3480 013434 020300              CMP     R3,R0          ;SEE IF BELOW MINIMUM LIMIT
3481 013436 101402              BLOS   8$              ;IF NOT: BR
3482 013440 000137 013452      JMP     T1NER          ;ELSE GO TO ERROR
3483 013444 010015              MOV     R0,(R5)        ;LOAD VALUE
3484 013446 005726              TST    (SP)+           ;POP CHAR COUNT OFF STACK
3485 013450 000207              RTS     PC              ;EXIT
3486
3487                          ;TTY ENTRY ERROR SUBROUTINE*****
3488
3489 013452 012704 015007      T1NER:  MOV     #MSG7,R4
3490 013456 004737 013552      JSR    PC,TTOUT        ;PRINT?
3491 013462 005726              TST    (SP)+           ;POP CHAR COUNT OFF STACK
3492 013464 162716 000020      SUB    #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
3493 013470 000207              RTS     PC              ;REDO VALUE ENTRY
3494
3495                          ;TTY READ SUBROUTINE*****
3496
3497 013472 017746 165050      TTIN:   MOV     @PSW-(SP) ;SAVE CURRENT PSW
3498 013476 052777 000340 165042      BIS    #340,@PSW      ;SET BR7 TO PREVENT INTRPT
3499 013504 005277 165042      INC    @TKS
3500 013510 105777 165036      1$:    TSTB   @TKS
3501 013514 100375              BPL    1$
3502 013516 012677 165024      MOV    (SP)+,@PSW      ;RESTORE PSW,OK TO INTRPT NOW
3503 013522 117737 165026 000602      MOVB   @TKB,TIB
3504 013530 042737 000200 000602      BIC    #200,TIB        ;STRIP PARITY BIT
3505 013536 013737 000602 000600      MOV    TIB,TOB         ;MOVE CHAR TO OUTPUT BFR
3506 013544 004737 013652      JSR    PC,TOG          ;AND TYPE IT
3507 013550 000207              RTS     PC
3508
3509                          ;TTY OUTPUT SUBROUTINE*****
3510
3511 013552 112437 000600      TTOUT:  MOVB   (R4)+,TOB
3512 013556 122737 000043 000600      CMPB   #43,TOB
3513 013564 001440              BEQ    TEX
3514 013566 122737 000045 000600      CMPB   #45,TOB
3515 013574 001403              BEQ    1$
3516 013576 004737 013652      JSR    PC,TOG
3517 013602 000763              BR     TTOUT
3518 013604 112737 000015 000600      1$:    MOVB   #15,TOB
3519 013612 004737 013652      JSR    PC,TOG
3520 013616 012703 000004      MOV    #4,R3
3521 013622 005037 000600      2$:    CLR    TOB
3522 013626 004737 013652      JSR    PC,TOG
3523 013632 005303              DEC    R3
3524 013634 001372              BNE    2$              ;DO FILLERS
3525 013636 112737 000012 000600      MOVB   #12,TOB
3526 013644 004737 013652      JSR    PC,TOG
3527 013650 000740              BR     TTOUT
3528 013652 105777 164700      TOG:   TSTB   @TPS

```

| | | | | | | | |
|------|--------|--------|--------|--------|------|------|----------|
| 3529 | 013656 | 100375 | | | | BPL | TOG |
| 3530 | 013660 | 113777 | 000600 | 164672 | | MOVB | TOB,@TPB |
| 3531 | 013666 | 000207 | | | TEX: | RTS | PC |
| 3532 | | | | | | | |
| 3533 | | | | | | | |

```

3535                                     :OCTAL OUTPUT SUBROUTINE*****
3536
3537 013670 012737 000001 014120 OCTPE: MOV #1,OFL
3538 013676 010304          MOV R3,R4
3539 013700 000410          BR OCTP0
3540 013702 005037 014120 OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
3541 013706 010304          OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
3542 013710 001004          BNE OCTP0 ;IF NOT ZERO: BR
3543 013712 004737 014100 JSR PC,OCTPG1 ;ELSE PRINT ZERO
3544 013716 000137 014042 JMP OCTP3 ;SPACE AND EXIT
3545 013722 032704 100000 OCTP0: BIT #100000,R4 ;SEE IF MSD = 1
3546 013726 001406          BEQ OCTP1 ;IF NOT: BR
3547 013730 012704 000001 MOV #1,R4
3548 013734 004737 014056 JSR PC,OCTPG ;PRINT 1
3549 013740 000137 013752 JMP OCTP2
3550 013744 005004          OCTP1: CLR R4
3551 013746 004737 014056 JSR PC,OCTPG ;PRINT 0
3552 013752 010304          OCTP2: MOV R3,R4
3553 013754 006004          ROR R4
3554 013756 006004          ROR R4
3555 013760 006004          ROR R4 ;POSITION DIGIT
3556 013762 006004          ROR R4
3557 013764 000304          SWAB R4
3558 013766 004737 014056 JSR PC,OCTPG ;PRINT DIGIT 2
3559 013772 010304          MOV R3,R4
3560 013774 006004          ROR R4
3561 013776 000304          SWAB R4
3562 014000 004737 014056 JSR PC,OCTPG ;PRINT DIGIT 3
3563 014004 010304          MOV R3,R4
3564 014006 006104          ROL R4
3565 014010 006104          ROL R4
3566 014012 000304          SWAB R4
3567 014014 004737 014056 JSR PC,OCTPG ;PRINT DIGIT 4
3568 014020 010304          MOV R3,R4
3569 014022 006004          ROR R4
3570 014024 006004          ROR R4
3571 014026 006004          ROR R4
3572 014030 004737 014056 JSR PC,OCTPG
3573 014034 010304          MOV R3,R4
3574 014036 004737 014056 JSR PC,OCTPG ;PRINT DIGIT 5
3575 014042 012737 000240 000600 OCTP3: MOV #240,TOB
3576 014050 004737 013652 JSR PC,TOB ;PRINT SPACE
3577 014054 000207          RTS PC ;EXIT
3578 014056 042704 177770 OCTPG: BIC #177770,R4
3579 014062 001004          BNE OCTPG0
3580 014064 005737 014120 TST OFL
3581 014070 001001          BNE OCTPG0
3582 014072 000207          RTS PC
3583
3584 014074 005237 014120 OCTPG0: INC OFL
3585 014100 052704 000260 OCTPG1: BIS #260,R4
3586 014104 010437 000600 MOV R4,TOB
3587 014110 004737 013652 JSR PC,TOB
3588 014114 010304          MOV R3,R4
3589 014116 000207          RTS PC
3590 014120 000000          OFL: 0 ;FIRST CHAR FLAG

```

```

3591
3592                ;DATA CHARACTER OUTPUT SUBROUTINE*****
3593
3594 014122 005037 000600      DOUT: CLR      TOB
3595 014126 012704 000010      MOV      #10,R4                ;SET NUMBER TO PRINT
3596 014132 110337 000600      MOVB    R3,TOB
3597 014136 105777 164414      1$:    TSTB   @TPS
3598 014142 100375              BPL     1$
3599 014144 132737 000200 000600 BITB    #200,TOB
3600 014152 001404              BEQ     2$
3601 014154 012777 000061 164376 MOV     #061,@TPB
3602 014162 000403              BR      3$
3603 014164 012777 000060 164366 2$:    MOV     #060,@TPB
3604 014172 006137 000600      3$:    ROL     TOB
3605 014176 005304              DEC     R4
3606 014200 001356              BNE    1$
3607 014202 000207              RTS     PC
3608
3609 014204 013703 000656      DOUTD: MOV     TEMP3,R3
3610 014210 000303              SWAB   R3
3611 014212 004737 014122      JSR    PC,DOUT
3612 014216 013703 000656      MOV     TEMP3,R3
3613 014222 004737 014122      JSR    PC,DOUT
3614 014226 000207              RTS     PC
3615
3616                ;TU45 SERIAL NUMBER PRINT SUBROUTINE*****
3617
3618 014230 010304      SNPT: MOV     R3,R4
3619 014232 000304      SWAB   R4
3620 014234 006004      ROR    R4
3621 014236 006004      ROR    R4
3622 014240 006004      ROR    R4
3623 014242 006004      ROR    R4                ;GET FIRST DIGIT
3624 014244 004737 014306      JSR    PC,SNPG          ;GO PRINT
3625 014250 010304      MOV     R3,R4
3626 014252 000304      SWAB   R4                ;GET SECOND DIGIT
3627 014254 004737 014306      JSR    PC,SNPG          ;GO PRINT
3628 014260 010304      MOV     R3,R4
3629 014262 006004      ROR    R4
3630 014264 006004      ROR    R4
3631 014266 006004      ROR    R4
3632 014270 006004      ROR    R4                ;GET THIRD DIGIT
3633 014272 004737 014306      JSR    PC,SNPG          ;GO PRINT
3634 014276 010304      MOV     R3,R4                ;GET FOURTH DIGIT
3635 014300 004737 014306      JSR    PC,SNPG          ;GO PRINT
3636 014304 000207      RTS     PC                ;EXIT
3637 014306 012737 000260 000600 SNPG: MOV     #260,TOB          ;SET BASE = 0
3638 014314 042704 177760      BIC    #177760,R4        ;MASK DIGIT
3639 014320 050437 000600      BIS    R4,TOB            ;SET ASCII
3640 014324 004737 013652      JSR    PC,TOG            ;TYPE DIGIT
3641 014330 000207      RTS     PC                ;RETURN
3642

```

```

3644
3645
3646 014332 022737 000176 000550 :ROUTINE TO LOAD NEW VALUE INTO SWITCHES
3647 014340 001032 GTSWR: CMP #SWREG,SWR :BRANCH IF SOFTWARE SWR
3648 014342 004737 014430 BNE 1$ :NOT INVOKED
3649 014346 012704 020236 JSR PC,SAVE :SAVE REGISTERS ON THE STACK
3650 014352 004737 013552 MOV #SMSWR,R4
3651 014356 017703 164166 JSR PC,TTOUT
3652 014362 004737 013670 MOV @SWR,R3
3653 014366 012704 020245 JSR PC,OCTPE
3654 014372 004737 013552 MOV #SMNEW,R4
3655 014376 013705 000550 JSR PC,TTOUT
3656 014402 012701 000007 MOV SWR,R5 :TTR ROUTINE RETURNS NEW VALUE TO (R5)
3657 014406 012702 177777 MOV #7,R1 :LIMIT RESPONSE TO 7 CHARS
3658 014412 012703 000000 MOV #177777,R2 :BETWEEN 0 AND 177777
3659 014416 004737 013212 MOV #0,R3
3660 014422 004737 014452 JSR PC,TTR
3661 014426 000207 1$: RTS PC :RESTORE REGISTERS
3662
3663 :ROUTINE TO SAVE REGISTERS ON THE STACK
(1) 014430 010546 .SAVE: MOV %5,-(SP) ;;R5 IS SAVED AT 12(SP)
(1) 014432 010446 MOV %4,-(SP) ;;R4 IS SAVED AT 10(SP)
(1) 014434 010346 MOV %3,-(SP) ;;R3 IS SAVED AT 6(SP)
(1) 014436 010246 MOV %2,-(SP) ;;R2 IS SAVED AT 4(SP)
(1) 014440 010146 MOV %1,-(SP) ;;R1 IS SAVED AT 2(SP)
(1) 014442 010046 MOV %0,-(SP) ;;R0 IS SAVED AT (SP)
(1) 014444 016646 000014 MOV 14(SP),-(SP) ;;PUSH RETURN PC ON THE STACK
(1) 014450 000207 RTS PC ;;RETURN TO CALLER
3664
(1) 014452 012666 000014 :ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
(1) 014456 012600 .RESTORE:MOV (SP)+,14(SP) ;;STORE RETURN PC ON STACK
(1) 014460 012601 MOV (SP)+,%0
(1) 014462 012602 MOV (SP)+,%1
(1) 014464 012603 MOV (SP)+,%2
(1) 014466 012604 MOV (SP)+,%3
(1) 014470 012605 MOV (SP)+,%4
(1) 014472 000207 RTS PC ;;RETURN
3665

```

```

3667                                     ;MESSAGE TABLE*****
3668
3669 014474 041445 030523 020040 MSG1: .ASCII /%CS1 WC BA FC CS2 /
      014502 020040 041527 020040
      014510 020040 041040 020101
      014516 020040 020040 041506
      014524 020040 020040 041440
      014532 031123 020040 020040
3670 014540 051504 020040 020040 .ASCII /DS ER TC%/
      014546 042440 020122 020040
      014554 020040 041524 021445
3671 014562 051045 053505 047111 MSG2: .ASCII /%REWIND ERROR#/
      014570 020104 051105 047522
      014576 021522
3672 014600 022445 046524 031460 MSG3: .ASCII /%TM03-TU45 BASIC FUNCTION TEST (CZTUQB0)%/
      014606 052055 032125 020065
      014614 040502 044523 020103
      014622 052506 041516 044524
      014630 047117 052040 051505
      014636 020124 041450 052132
      014644 050525 030102 022451
3673 014652 054524 042520 036040 .ASCII /TYPE <CR> TO TERMINATE RESPONSE & ^C TO RESTART%/
      014660 051103 020076 047524
      014666 052040 051105 044515
      014674 040516 042524 051040
      014702 051505 047520 051516
      014710 020105 020046 041536
      014716 052040 020117 042522
      014724 052123 051101 022524
      014732 043
3674 014733 045 042522 044507 MSG4: .ASCII /%REGISTER START = #/
      014740 052123 051105 051440
      014746 040524 052122 036440
      014754 021440
3675 014756 053045 041505 047524 MSG5: .ASCII /%VECTOR = #/
      014764 020122 020075 043
3676 014771 045 047105 020104 MSG6: .ASCII /%END OF PASS #/
      014776 043117 050040 051501
      015004 020123 043
3677 015007 040 020077 043 MSG7: .ASCII / ? #/
3678 015013 045 C, 520 044523 MSG9: .ASCII /%POSITION ERROR: #/
      015020 044524 047117 042440
      015026 051122 051117 020072
      015034 043
3679 015035 045 051104 053111 MSG10: .ASCII /%DRIVE NUMBER: #/
      015042 020105 052516 041115
      015050 051105 020072 043
3680 015055 045 046123 053101 MSG11: .ASCII /%SLAVE NUMBER: #/
      015062 020105 052516 041115
      015070 051105 020072 043
3681 015075 045 051127 052111 MSG12: .ASCII /%WRITE ERROR #/
      015102 020105 051105 047522
      015110 020122 043
3682 015113 045 042522 042101 MSG13: .ASCII /%READ REVERSE ERROR #/
      015120 051040 053105 051105
      015126 042523 042440 051122

```


| | | | | | | | | | |
|------|--------|--------|--------|--------|--------|--------|----------------------------|--|--|
| 3683 | 015134 | 051117 | 021440 | | | | | | |
| | 015140 | 051045 | 040505 | 020104 | MSG14: | .ASCII | /%READ FORWARD ERROR #/ | | |
| | 015146 | 047506 | 053522 | 051101 | | | | | |
| | 015154 | 020104 | 051105 | 047522 | | | | | |
| 3684 | 015162 | 020122 | 043 | | | | | | |
| | 015165 | 045 | 051127 | 052111 | MSG15: | .ASCII | /%WRITE TM ERROR #/ | | |
| | 015172 | 020105 | 046524 | 042440 | | | | | |
| | 015200 | 051122 | 051117 | 021440 | | | | | |
| 3685 | 015206 | 051045 | 053105 | 051105 | MSG16: | .ASCII | /%REVERSE ERROR #/ | | |
| | 015214 | 042523 | 042440 | 051122 | | | | | |
| | 015222 | 051117 | 021440 | | | | | | |
| 3686 | 015226 | 043045 | 051117 | 040527 | MSG17: | .ASCII | /%FORWARD ERROR #/ | | |
| | 015234 | 042122 | 042440 | 051122 | | | | | |
| | 015242 | 051117 | 021440 | | | | | | |
| 3687 | 015246 | 047040 | 055122 | 021440 | MSG20: | .ASCII | / NRZ #/ | | |
| 3688 | 015254 | 050040 | 020105 | 043 | MSG21: | .ASCII | / PE #/ | | |
| 3689 | 015261 | 040 | 054105 | 052120 | MSG22: | .ASCII | / EXPT: #/ | | |
| | 015266 | 020072 | 043 | | | | | | |
| 3690 | 015271 | 040 | 041522 | 042126 | MSG23: | .ASCII | / RCVD: #/ | | |
| | 015276 | 020072 | 043 | | | | | | |
| 3691 | 015301 | 045 | 052502 | 020123 | MSG24: | .ASCII | /%BUS TRAP: #/ | | |
| | 015306 | 051124 | 050101 | 020072 | | | | | |
| | 015314 | 043 | | | | | | | |
| 3692 | 015315 | 045 | 041527 | 020072 | MSG25: | .ASCII | /%WC: #/ | | |
| | 015322 | 043 | | | | | | | |
| 3693 | 015323 | 045 | 040502 | 020072 | MSG26: | .ASCII | /%BA: #/ | | |
| | 015330 | 043 | | | | | | | |
| 3694 | 015331 | 045 | 041104 | 020072 | MSG27: | .ASCII | /%DB: #/ | | |
| | 015336 | 043 | | | | | | | |
| 3695 | 015337 | 045 | 047111 | 052111 | MSG28: | .ASCII | /%INIT DID NOT CLEAR RH #/ | | |
| | 015344 | 042040 | 042111 | 047040 | | | | | |
| | 015352 | 052117 | 041440 | 042514 | | | | | |
| | 015360 | 051101 | 051040 | 020110 | | | | | |
| | 015366 | 043 | | | | | | | |
| 3696 | 015367 | 045 | 041523 | 047040 | MSG29: | .ASCII | /%SC NOT RESET BY INIT #/ | | |
| | 015374 | 052117 | 051040 | 051505 | | | | | |
| | 015402 | 052105 | 041040 | 020131 | | | | | |
| | 015410 | 047111 | 052111 | 021440 | | | | | |
| 3697 | 015416 | 052045 | 042522 | 047040 | MSG30: | .ASCII | /%TRE NOT RESET BY INIT #/ | | |
| | 015424 | 052117 | 051040 | 051505 | | | | | |
| | 015432 | 052105 | 041040 | 020131 | | | | | |
| | 015440 | 047111 | 052111 | 021440 | | | | | |
| 3698 | 015446 | 041445 | 031123 | 047040 | MSG31: | .ASCII | /%CS2 NOT RESET BY INIT #/ | | |
| | 015454 | 052117 | 051040 | 051505 | | | | | |
| | 015462 | 052105 | 041040 | 020131 | | | | | |
| | 015470 | 047111 | 052111 | 021440 | | | | | |
| 3699 | 015476 | 042045 | 052114 | 047040 | MSG32: | .ASCII | /%DLT NOT SET #/ | | |
| | 015504 | 052117 | 051440 | 052105 | | | | | |
| | 015512 | 021440 | | | | | | | |
| 3700 | 015514 | 051445 | 020103 | 047516 | MSG33: | .ASCII | /%SC NOT SET #/ | | |
| | 015522 | 020124 | 042523 | 020124 | | | | | |
| | 015530 | 043 | | | | | | | |
| 3701 | 015531 | 045 | 051124 | 020105 | MSG34: | .ASCII | /%TRE NOT SET #/ | | |
| | 015536 | 047516 | 020124 | 042523 | | | | | |
| | 015544 | 020124 | 043 | | | | | | |
| 3702 | 015547 | 045 | 051111 | 047040 | MSG35: | .ASCII | /%IR NOT SET BY INIT #/ | | |

| | | | | | |
|------|--------|--------|--------|--------|---|
| | 015554 | 052117 | 051440 | 052105 | |
| | 015562 | 041040 | 020131 | 047111 | |
| | 015570 | 052111 | 021440 | | |
| 3703 | 015574 | 047445 | 020122 | 047516 | MSG36: .ASCII /%OR NOT RESET BY INIT #/ |
| | 015602 | 020124 | 042522 | 042523 | |
| | 015610 | 020124 | 054502 | 044440 | |
| | 015616 | 044516 | 020124 | 043 | |
| 3704 | 015623 | 045 | 051117 | 047040 | MSG37: .ASCII /%OR NOT RESET BY 1 SILO ENTRY #/ |
| | 015630 | 052117 | 051040 | 051505 | |
| | 015636 | 052105 | 041040 | 020131 | |
| | 015644 | 020061 | 044523 | 047514 | |
| | 015652 | 02440 | 052116 | 054522 | |
| | 015660 | 021440 | | | |
| 3705 | 01566 | 0474 | 020122 | 047516 | MSG38: .ASCII /%OR NOT SET BY SILO FULL #/ |
| | 015670 | 020124 | 042523 | 020124 | |
| | 015676 | 054502 | 051440 | 046111 | |
| | 015704 | 020117 | 052506 | 046114 | |
| | 015712 | 021440 | | | |
| 3706 | 015714 | 041045 | 042101 | 051440 | MSG39: .ASCII /%BAD SILO READ #/ |
| | 015722 | 046111 | 020117 | 042522 | |
| | 015730 | 042101 | 021440 | | |
| 3707 | 015734 | 044445 | 020122 | 047516 | MSG40: .ASCII /%IR NOT RESET BY SILO FULL#/ |
| | 015742 | 020124 | 042522 | 042523 | |
| | 015750 | 020124 | 054502 | 051440 | |
| | 015756 | 046111 | 020117 | 052506 | |
| | 015764 | 046114 | 043 | | |
| 3708 | 015767 | 045 | 047516 | 026516 | MSG41: .ASCII /%NON-EXIST DRIVE#/ |
| | 015774 | 054105 | 051511 | 020124 | |
| | 016002 | 051104 | 053111 | 021505 | |
| 3709 | 016010 | 047045 | 047117 | 042455 | MSG42: .ASCII /%NON-EXIST SLAVE#/ |
| | 016016 | 044530 | 052123 | 051440 | |
| | 016024 | 040514 | 042526 | 043 | |
| 3710 | 016031 | 045 | 042523 | 044522 | MSG43: .ASCII /%SERIAL NO: #/ |
| | 016036 | 046101 | 047040 | 035117 | |
| | 016044 | 021440 | | | |
| 3711 | 016046 | 042445 | 040522 | 042523 | MSG44: .ASCII /%ERASE HEAD INOPERATIVE#/ |
| | 016054 | 044040 | 040505 | 020104 | |
| | 016062 | 047111 | 050117 | 051105 | |
| | 016070 | 052101 | 053111 | 021505 | |
| 3712 | 016076 | 050045 | 051517 | 044523 | MSG45: .ASCII /%POSSIBLE ERASE HEAD PROBLEM: / |
| | 016104 | 046102 | 020105 | 051105 | |
| | 016112 | 051501 | 020105 | 042510 | |
| | 016120 | 042101 | 050040 | 047522 | |
| | 016126 | 046102 | 046505 | 020072 | |
| 3713 | 016134 | 044103 | 041505 | 020113 | .ASCII /CHECK POLARITY#/ |
| | 016142 | 047520 | 040514 | 044522 | |
| | 016150 | 054524 | 043 | | |
| 3714 | 016153 | 045 | 042523 | 026524 | MSG46: .ASCII /%SET-UP WRITE ERROR#/ |
| | 016160 | 050125 | 053440 | 044522 | |
| | 016166 | 042524 | 042440 | 051122 | |
| | 016174 | 051117 | 043 | | |
| 3715 | 016177 | 045 | 050123 | 041501 | MSG47: .ASCII /%SPACE FORWARD ERROR#/ |
| | 016204 | 020105 | 047506 | 053522 | |
| | 016212 | 051101 | 020104 | 051105 | |
| | 016220 | 047522 | 021522 | | |
| 3716 | 016224 | 051445 | 040520 | 042503 | MSG48: .ASCII /%SPACE REVERSE ERROR#/ |

| | | | | | |
|------|--------|--------|--------|--------|---|
| | 016232 | 051040 | 053105 | 051105 | |
| | 016240 | 042523 | 042440 | 051122 | |
| | 016246 | 051117 | 043 | | |
| 3717 | 016251 | 045 | 052502 | 043106 | MSG49: .ASCII /%BUFFERED WRITE ERROR#/ |
| | 016256 | 051105 | 042105 | 053440 | |
| | 016264 | 044522 | 042524 | 042440 | |
| | 016272 | 051122 | 051117 | 043 | |
| 3718 | 016277 | 045 | 047502 | 020124 | MSG50: .ASCII /%BOT SET AFTER BUFFERED WRITE#/ |
| | 016304 | 042523 | 020124 | 043101 | |
| | 016312 | 042524 | 020122 | 052502 | |
| | 016320 | 043106 | 051105 | 042105 | |
| | 016326 | 053440 | 044522 | 042524 | |
| | 016334 | 043 | | | |
| 3719 | 016335 | 045 | 047516 | 041040 | MSG51: .ASCII /%NO BOT FROM READ IN PRESET#/ |
| | 016342 | 052117 | 043040 | 047522 | |
| | 016350 | 020115 | 042522 | 042101 | |
| | 016356 | 044440 | 020116 | 051120 | |
| | 016364 | 051505 | 052105 | 043 | |
| 3720 | 016371 | 045 | 041524 | 044440 | MSG52: .ASCII /%TC INCORRECT #/ |
| | 016376 | 041516 | 051117 | 042522 | |
| | 016404 | 052103 | 021440 | | |
| 3721 | 016410 | 046445 | 046117 | 043040 | MSG53: .ASCII /%MOL FAILED TO CLEAR#/ |
| | 016416 | 044501 | 042514 | 020104 | |
| | 016424 | 047524 | 041440 | 042514 | |
| | 016432 | 051101 | 043 | | |
| 3722 | 016435 | 045 | 051045 | 051505 | MSG54: .ASCII /%%RESET SLAVE TO ON LINE BEFORE CONTINUING/ |
| | 016442 | 052105 | 051440 | 040514 | |
| | 016450 | 042526 | 052040 | 020117 | |
| | 016456 | 047117 | 046040 | 047111 | |
| | 016464 | 020105 | 042502 | 047506 | |
| | 016472 | 042522 | 041440 | 047117 | |
| | 016500 | 044524 | 052516 | 047111 | |
| | 016506 | 107 | | | |
| 3723 | 016507 | 045 | 042523 | 020124 | .ASCII /%SET SW12=1 IF YOU DON'T WISH TO REPEAT THIS TEST#/ |
| | 016514 | 053523 | 031061 | 030475 | |
| | 016522 | 044440 | 020106 | 047531 | |
| | 016530 | 020125 | 047504 | 023516 | |
| | 016536 | 020124 | 044527 | 044123 | |
| | 016544 | 052040 | 020117 | 042522 | |
| | 016552 | 042520 | 052101 | 052040 | |
| | 016560 | 044510 | 020123 | 042524 | |
| | 016566 | 052123 | 043 | | |
| 3724 | 016571 | 040 | 052111 | 051105 | MSG56: .ASCII / ITER: #/ |
| | 016576 | 020072 | 043 | | |
| 3725 | 016601 | 045 | 046524 | 047040 | MSG57: .ASCII /%TM NOT SET#/ |
| | 016606 | 052117 | 051440 | 052105 | |
| | 016614 | 043 | | | |
| 3726 | 016615 | 045 | 044505 | 044124 | MSG60: .ASCII /%EITHER TAPE NOT ERASED OR OPI PROBLEM#/ |
| | 016622 | 051105 | 052040 | 050101 | |
| | 016630 | 020105 | 047516 | 020124 | |
| | 016636 | 051105 | 051501 | 042105 | |
| | 016644 | 047440 | 020122 | 050117 | |
| | 016652 | 020111 | 051120 | 041117 | |
| | 016660 | 042514 | 021515 | | |
| 3727 | 016664 | 051045 | 020110 | 047117 | MSG62: .ASCII /%RH ONLY (NO=0,YES=1): #/ |
| | 016672 | 054514 | 024040 | 047516 | |

| | | | | | |
|------|--------|--------|--------|--------|--|
| | 016700 | 030075 | 054454 | 051505 | |
| | 016706 | 030475 | 035051 | 021440 | |
| 3728 | 016714 | 042045 | 042111 | 047040 | MSG63: .ASCII /%DID NOT AUTO SELECT NRZ#/ |
| | 016722 | 052117 | 040440 | 052125 | |
| | 016730 | 020117 | 042523 | 042514 | |
| | 016736 | 052103 | 047040 | 055122 | |
| | 016744 | 043 | | | |
| 3729 | 016745 | 045 | 044504 | 020104 | MSG64: .ASCII /%DID NOT AUTO SELECT PE#/ |
| | 016752 | 047516 | 020124 | 052501 | |
| | 016760 | 047524 | 051440 | 046105 | |
| | 016766 | 041505 | 020124 | 042520 | |
| | 016774 | 043 | | | |
| 3730 | 016775 | 045 | 043 | | MSG65: .ASCII /%#/ |
| 3731 | 016777 | 134 | 043 | | MSG66: .ASCII /\#/ |
| 3732 | 017001 | 045 | 051105 | 020072 | MSG67: .ASCII /%ER: #/ |
| | 017006 | 043 | | | |
| 3733 | 017007 | 045 | 042522 | 047515 | MSG69: .ASCII /%REMOVE TMDP FROM SLAVE TO BE TESTED%#/ |
| | 017014 | 042526 | 052040 | 042115 | |
| | 017022 | 020120 | 051106 | 046517 | |
| | 017030 | 051440 | 040514 | 042526 | |
| | 017036 | 052040 | 020117 | 042502 | |
| | 017044 | 052040 | 051505 | 042524 | |
| | 017052 | 022504 | 043 | | |
| 3734 | 017055 | 045 | 040510 | 042122 | MSG70: .ASCII /%HARDWARE SWR IN USE%#/ |
| | 017062 | 040527 | 042522 | 051440 | |
| | 017070 | 051127 | 044440 | 020116 | |
| | 017076 | 051525 | 022505 | 043 | |
| 3735 | 017103 | 045 | 042045 | 044522 | MSG71: .ASCII /%%DRIVE #/ |
| | 017110 | 042526 | 021440 | | |
| 3736 | 017114 | 051454 | 040514 | 042526 | MSG72: .ASCII /,SLAVE #/ |
| | 017122 | 021440 | | | |
| 3737 | 017124 | 047516 | 020124 | 047117 | MSG73: .ASCII /NOT ON LINE#/ |
| | 017132 | 046040 | 047111 | 021505 | |
| 3738 | | | | | |

```

3740                                     ;TEST HEADERS*****
3741
3742 017140 022445 052106 035061 MSFT1: .ASCII /%FT1:RH ADDRESSING #/
      017146 044122 040440 042104
      017154 042522 051523 047111
      017162 020107 043
3743 017165 045 043045 031124 MSFT2: .ASCII /%FT2:RH REGISTER BITS TEST #/
      017172 051072 020110 042522
      017200 044507 052123 051105
      017206 041040 052111 020123
      017214 042524 052123 021440
3744 017222 022445 052106 035063 MSFT3: .ASCII /%FT3:RH INITIALIZE TEST #/
      017230 044122 044440 044516
      017236 044524 046101 055111
      017244 020105 042524 052123
      017252 021440
3745 017254 022445 052106 035064 MSFT4: .ASCII /%FT4:RH11 SILO TEST 1 #/
      017262 044122 030461 051440
      017270 046111 020117 042524
      017276 052123 030440 021440
3746 017304 022445 052106 035065 MSFT5: .ASCII /%FT5:RH11 SILO TEST 2 #/
      017312 044122 030461 051440
      017320 046111 020117 042524
      017326 052123 031040 021440
3747 017334 022445 052106 035066 MSFT6: .ASCII /%FT6:RH11 SILO TEST 3 #/
      017342 044122 030461 051440
      017350 046111 020117 042524
      017356 052123 031440 021440
3748 017364 022445 052106 035067 MSFT7: .ASCII /%FT7:RH11 SILO TEST 4 #/
      017372 044122 030461 051440
      017400 046111 020117 042524
      017406 052123 032040 021440
3749 017414 022445 052106 030061 MSFT10: .ASCII /%FT10:RH11 SILO TEST 5 #/
      017422 051072 030510 020061
      017430 044523 047514 052040
      017436 051505 020124 020065
      017444 043
3750 017445 045 043045 030524 MSFT11: .ASCII /%FT11:NOP TEST#/
      017452 035061 047516 020120
      017460 042524 052123 043
3751 017465 045 043045 030524 MSFT12: .ASCII /%FT12:REWIND TEST#/
      017472 035062 042522 044527
      017500 042116 052040 051505
      017506 021524
3752 017510 022445 052106 031461 MSFT13: .ASCII /%FT13:WRITE-READ TEST#/
      017516 053472 044522 042524
      017524 051055 040505 020104
      017532 042524 052123 043
3753 017537 045 043045 030524 MSFT14: .ASCII /%FT14:SPACE TEST#/
      017544 035064 050123 041501
      017552 020105 042524 052123
      017560 043
3754 017561 045 043045 030524 MSFT15: .ASCII /%FT15:ERASE TEST#/
      017566 035065 051105 051501
      017574 020105 042524 052123
      017602 043
  
```

| | | | | | | |
|------|--------|--------|--------|--------|-----------------|---|
| 3755 | 017603 | 045 | 043045 | 030524 | MSFT16: .ASCII | /%FT16:TAPE MARK WRITE-READ TEST#/ |
| | 017610 | 035066 | 040524 | 042520 | | |
| | 017616 | 046440 | 051101 | 020113 | | |
| | 017624 | 051127 | 052111 | 026505 | | |
| | 017632 | 042522 | 042101 | 052040 | | |
| | 017640 | 051505 | 021524 | | | |
| 3756 | 017644 | 022445 | 052106 | 033461 | MSFT17: .ASCII | /%FT17:TM SPACE TEST #/ |
| | 017652 | 052072 | 020115 | 050123 | | |
| | 017660 | 041501 | 020105 | 042524 | | |
| | 017666 | 052123 | 021440 | | | |
| 3757 | 017672 | 022445 | 052106 | 030062 | MSFT20: .ASCII | /%FT20:WRITE CHECK TEST #/ |
| | 017700 | 053472 | 044522 | 042524 | | |
| | 017706 | 041440 | 042510 | 045503 | | |
| | 017714 | 052040 | 051505 | 020124 | | |
| | 017722 | 043 | | | | |
| 3758 | 017723 | 045 | 043045 | 031124 | MSFT21: .ASCII | /%FT21:ERASE HEAD TEST#/ |
| | 017730 | 035061 | 051105 | 051501 | | |
| | 017736 | 020105 | 042510 | 042101 | | |
| | 017744 | 052040 | 051505 | 021524 | | |
| 3759 | 017752 | 022445 | 052106 | 031062 | MSFT22: .ASCII | /%FT22:BUFFERED COMMAND TEST#/ |
| | 017760 | 041072 | 043125 | 042506 | | |
| | 017766 | 042522 | 020104 | 047503 | | |
| | 017774 | 046515 | 047101 | 020104 | | |
| | 020002 | 042524 | 052123 | 043 | | |
| 3760 | 020007 | 045 | 043045 | 031124 | MSFT23: .ASCII | /%FT23:READ IN PRESET TEST#/ |
| | 020014 | 035063 | 042522 | 042101 | | |
| | 020022 | 044440 | 020116 | 051120 | | |
| | 020030 | 051505 | 052105 | 052040 | | |
| | 020036 | 051505 | 021524 | | | |
| 3761 | 020042 | 022445 | 052106 | 033062 | MSFT26: .ASCII | /%FT26:REWIND-OFF LINE TEST#/ |
| | 020050 | 051072 | 053505 | 047111 | | |
| | 020056 | 026504 | 043117 | 020106 | | |
| | 020064 | 044514 | 042516 | 052040 | | |
| | 020072 | 051505 | 021524 | | | |
| 3762 | 020076 | 022445 | 052106 | 032062 | MSFT24: .ASCII | /%FT24:AUTO DENSITY SELECT: WRITE-NRZ,READ-PE#/ |
| | 020104 | 040472 | 052125 | 020117 | | |
| | 020112 | 042504 | 051516 | 052111 | | |
| | 020120 | 020131 | 042523 | 042514 | | |
| | 020126 | 052103 | 020072 | 051127 | | |
| | 020134 | 052111 | 026505 | 051116 | | |
| | 020142 | 026132 | 042522 | 042101 | | |
| | 020150 | 050055 | 021505 | | | |
| 3763 | 020154 | 022445 | 052106 | 032462 | MSFT25: .ASCII | /%FT25:AUTO DENSITY SELECT: WRITE-PE,READ-NRZ#/ |
| | 020162 | 040472 | 052125 | 020117 | | |
| | 020170 | 042504 | 051516 | 052111 | | |
| | 020176 | 020131 | 042523 | 042514 | | |
| | 020204 | 052103 | 020072 | 051127 | | |
| | 020212 | 052111 | 026505 | 042520 | | |
| | 020220 | 051054 | 040505 | 026504 | | |
| | 020226 | 051116 | 021532 | | | |
| 3764 | 020232 | 057045 | 021507 | | \$CNTG: .ASCII | /%^G#/ |
| 3765 | 020236 | 051445 | 051127 | 020075 | \$MSWR: .ASCII | /%SWR= #/ |
| | 020244 | 043 | | | | |
| 3766 | 020245 | 040 | 047040 | 053505 | \$MNEW: .ASCII | / NEW= #/ |
| | 020252 | 020075 | 043 | | | |
| 3767 | 020255 | 077 | 021445 | | \$QUEST: .ASCII | /?%#/ |

3768
3769
3770
3771 020260 000000
3772 021772 021772
3773 021772 000000
3774
3775 000001

WDATA: 0 .EVEN
RDATA: 0 .-.+1510
0
.END

| | | | | | | | | | | | | | | | | | | | |
|--------|--------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|------|--|--|--|--|
| FT15X | 007004 | 2778 | 2791# | | | | | | | | | | | | | | | | |
| FT16 | 007010 | 2071 | 2072 | 2795# | | | | | | | | | | | | | | | |
| FT16A | 007042 | 2800# | 2823 | | | | | | | | | | | | | | | | |
| FT16B | 007046 | 2801# | | | | | | | | | | | | | | | | | |
| FT16X | 007226 | 2821 | 2824# | | | | | | | | | | | | | | | | |
| FT17 | 007236 | 2073 | 2074 | 2831# | | | | | | | | | | | | | | | |
| FT17A | 007256 | 2834# | 2888 | | | | | | | | | | | | | | | | |
| FT17B | 007262 | 2835# | 2865 | | | | | | | | | | | | | | | | |
| FT17C | 007422 | 2855# | 2864 | | | | | | | | | | | | | | | | |
| FT17D | 007470 | 2848 | 2866# | | | | | | | | | | | | | | | | |
| FT17D1 | 007506 | 2869# | 2884 | | | | | | | | | | | | | | | | |
| FT17E | 007522 | 2871# | 2879 | | | | | | | | | | | | | | | | |
| FT17F | 007612 | 2881 | 2885# | | | | | | | | | | | | | | | | |
| FT17X | 007634 | 2843 | 2862 | 2876 | 2886 | 2889# | | | | | | | | | | | | | |
| FT2 | 003410 | 2047 | 2048 | 2305# | | | | | | | | | | | | | | | |
| FT2A | 003422 | 2307# | 2314 | 2339 | | | | | | | | | | | | | | | |
| FT2B | 003462 | 2312 | 2316# | 2323 | | | | | | | | | | | | | | | |
| FT2C | 003522 | 2321 | 2325# | 2334 | | | | | | | | | | | | | | | |
| FT2D | 003536 | 2329# | 2330 | | | | | | | | | | | | | | | | |
| FT2E | 003566 | 2332 | 2336# | | | | | | | | | | | | | | | | |
| FT2ER | 003576 | 2315 | 2324 | 2335 | 2340# | 3047 | 3141 | | | | | | | | | | | | |
| FT2ERA | 003626 | 2344 | 2347# | | | | | | | | | | | | | | | | |
| FT2ERB | 003700 | 2342 | 2358# | | | | | | | | | | | | | | | | |
| FT2ERC | 003710 | 2359 | 2361# | | | | | | | | | | | | | | | | |
| FT2X | 003720 | 2337 | 2364# | | | | | | | | | | | | | | | | |
| FT20 | 007640 | 2075 | 2076 | 2894# | | | | | | | | | | | | | | | |
| FT20A | 007656 | 2897# | 2922 | | | | | | | | | | | | | | | | |
| FT20B | 010002 | 2914# | | | | | | | | | | | | | | | | | |
| FT20C | 010034 | 2919# | | | | | | | | | | | | | | | | | |
| FT20X | 010054 | 2908 | 2920 | 2923# | | | | | | | | | | | | | | | |
| FT21 | 010064 | 2077 | 2078 | 2929# | | | | | | | | | | | | | | | |
| FT21A | 010072 | 2930# | | | | | | | | | | | | | | | | | |
| FT21B | 010366 | 2967 | 2970# | | | | | | | | | | | | | | | | |
| FT21C | 010374 | 2969 | 2971# | | | | | | | | | | | | | | | | |
| FT21SC | 010222 | 2948# | 2971 | | | | | | | | | | | | | | | | |
| FT21X | 010406 | 2963 | 2965 | 2973# | | | | | | | | | | | | | | | |
| FT22 | 010416 | 2079 | 2080 | 2980# | 3005 | | | | | | | | | | | | | | |
| FT22A | 010472 | 2988# | 2990 | | | | | | | | | | | | | | | | |
| FT22B | 010512 | 2993# | 2994 | | | | | | | | | | | | | | | | |
| FT22X | 010612 | 3003 | 3007# | | | | | | | | | | | | | | | | |
| FT23 | 010622 | 2081 | 2082 | 3014# | 3039 | 3046 | | | | | | | | | | | | | |
| FT23A | 010732 | 3030# | 3033 | 3035 | | | | | | | | | | | | | | | |
| FT23B | 010752 | 3031 | 3036# | | | | | | | | | | | | | | | | |
| FT23C | 011002 | 3037 | 3041# | | | | | | | | | | | | | | | | |
| FT23X | 011036 | 3015 | 3044 | 3048# | | | | | | | | | | | | | | | |
| FT24 | 011042 | 2083 | 2084 | 3055# | | | | | | | | | | | | | | | |
| FT24X | 011224 | 3068 | 3075 | 3078# | | | | | | | | | | | | | | | |
| FT25 | 011234 | 2085 | 2086 | 3084# | | | | | | | | | | | | | | | |
| FT25X | 011416 | 3097 | 3104 | 3107# | | | | | | | | | | | | | | | |
| FT26 | 011426 | 2087 | 2088 | 3112# | | | | | | | | | | | | | | | |
| FT26X | 011624 | 3135 | 3139 | 3142# | | | | | | | | | | | | | | | |
| FT26XX | 011634 | 3113 | 3115 | 3144# | | | | | | | | | | | | | | | |
| FT3 | 003732 | 2049 | 2050 | 2371# | 2372 | | | | | | | | | | | | | | |
| FT3A | 004006 | 2378 | 2381# | | | | | | | | | | | | | | | | |
| FT3B | 004030 | 2382 | 2385# | | | | | | | | | | | | | | | | |
| FT3ER | 004066 | 2380 | 2384 | 2390 | 2395# | 2435 | 2449 | 2453 | 2458 | 2466 | 2486 | 2490 | 2490 | 2490 | 2567 | | | | |

TM03/TU45 BASIC FUNCTION TEST
CZTU08.P11 13-MAY-80 15:20

MACY11 30A(1052) 13-MAY-80 15:25 PAGE 54-4
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0076

| | | | | | | | |
|--------|--------|-------|-------|-------|-------|-------|-------|
| MSFT20 | 017672 | 2895 | 3757# | | | | |
| MSFT21 | 017723 | 2929 | 3758# | | | | |
| MSFT22 | 017752 | 2980 | 3759# | | | | |
| MSFT23 | 020007 | 3016 | 3760# | | | | |
| MSFT24 | 020076 | 3055 | 3762# | | | | |
| MSFT25 | 020154 | 3084 | 3763# | | | | |
| MSFT26 | 020042 | 3116 | 3761# | | | | |
| MSFT3 | 017222 | 2771 | 3744# | | | | |
| MSFT4 | 017254 | 2417 | 3745# | | | | |
| MSFT5 | 017304 | 2443 | 3746# | | | | |
| MSFT6 | 017334 | 2475 | 3747# | | | | |
| MSFT7 | 017364 | 2528 | 3748# | | | | |
| MSG1 | 014474 | 3235 | 3669# | | | | |
| MSG10 | 015035 | 2150 | 3679# | | | | |
| MSG11 | 015055 | 2167 | 3680# | | | | |
| MSG12 | 015075 | 2624 | 2806 | 2858 | 2939 | 3681# | |
| MSG13 | 015113 | 2634 | 2812 | 3682# | | | |
| MSG14 | 015140 | 2641 | 2817 | 2955 | 3683# | | |
| MSG15 | 015165 | 2840 | 3684# | | | | |
| MSG16 | 015206 | 2741 | 2868 | 2909 | 3685# | | |
| MSG17 | 015226 | 2739 | 2882 | 2914 | 3686# | | |
| MSG2 | 014562 | 3190 | 3671# | | | | |
| MSG20 | 015246 | 2732 | 3221 | 3687# | | | |
| MSG21 | 015254 | 2735 | 3224 | 3688# | | | |
| MSG22 | 015261 | 2350 | 2510 | 2743 | 3689# | | |
| MSG23 | 015271 | 2354 | 2514 | 2748 | 3690# | | |
| MSG24 | 015301 | 3401 | 3691# | | | | |
| MSG25 | 015315 | 2313 | 3692# | | | | |
| MSG26 | 015323 | 2322 | 3693# | | | | |
| MSG27 | 015331 | 2333 | 3694# | | | | |
| MSG28 | 015337 | 3695# | | | | | |
| MSG29 | 015367 | 2379 | 3696# | | | | |
| MSG3 | 014600 | 2112 | 2114* | 3672# | | | |
| MSG30 | 015416 | 2383 | 3697# | | | | |
| MSG31 | 015446 | 2389 | 3698# | | | | |
| MSG32 | 015476 | 2428 | 2539 | 2566 | 3699# | | |
| MSG33 | 015514 | 2430 | 3700# | | | | |
| MSG34 | 015531 | 2432 | 3701# | | | | |
| MSG35 | 015547 | 2448 | 3702# | | | | |
| MSG36 | 015574 | 2452 | 3703# | | | | |
| MSG37 | 015623 | 2457 | 2465 | 3704# | | | |
| MSG38 | 015662 | 2489 | 3705# | | | | |
| MSG39 | 015714 | 2508 | 3706# | | | | |
| MSG4 | 014733 | 2115 | 3674# | | | | |
| MSG40 | 015734 | 2485 | 3707# | | | | |
| MSG41 | 015767 | 2164 | 3708# | | | | |
| MSG42 | 016010 | 2181 | 3709# | | | | |
| MSG43 | 016031 | 2184 | 3710# | | | | |
| MSG44 | 016046 | 2968 | 3711# | | | | |
| MSG45 | 016076 | 2970 | 3712# | | | | |
| MSG46 | 016153 | 2671 | 2775 | 2905 | 3065 | 3094 | 3714# |
| MSG47 | 016177 | 2697 | 3715# | | | | |
| MSG48 | 016224 | 2687 | 3716# | | | | |
| MSG49 | 016251 | 2999 | 3717# | | | | |
| MSG5 | 014756 | 2124 | 3675# | | | | |
| MSG50 | 016277 | 3004 | 3718# | | | | |

TM03/TU45 BASIC FUNCTION TEST
CZTUQB.P11 13-MAY-80 15:20

MACY11 30A(1052) 13-MAY-80 15:25 PAGE 54-5
CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0077

| | | | | | | | | | | | | | | |
|--------|--------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| MSG51 | 016335 | 3038 | 3719# | | | | | | | | | | | |
| MSG52 | 016371 | 3045 | 3720# | | | | | | | | | | | |
| MSG53 | 016410 | 3133 | 3721# | | | | | | | | | | | |
| MSG54 | 016435 | 3142 | 3722# | | | | | | | | | | | |
| MSG56 | 016571 | 3231 | 3724# | | | | | | | | | | | |
| MSG57 | 016601 | 3268 | 3725# | | | | | | | | | | | |
| MSG6 | 014771 | 2265 | 3676# | | | | | | | | | | | |
| MSG60 | 016615 | 2788 | 3726# | | | | | | | | | | | |
| MSG62 | 016664 | 2188 | 3727# | | | | | | | | | | | |
| MSG63 | 016714 | 3076 | 3728# | | | | | | | | | | | |
| MSG64 | 016745 | 3105 | 3729# | | | | | | | | | | | |
| MSG65 | 016775 | 3450 | 3730# | | | | | | | | | | | |
| MSG66 | 016777 | 3459 | 3731# | | | | | | | | | | | |
| MSG67 | 017001 | 3140 | 3732# | | | | | | | | | | | |
| MSG69 | 017007 | 2110 | 3733# | | | | | | | | | | | |
| MSG7 | 015007 | 3489 | 3677# | | | | | | | | | | | |
| MSG70 | 017055 | 3381 | 3734# | | | | | | | | | | | |
| MSG71 | 017103 | 2225 | 3735# | | | | | | | | | | | |
| MSG72 | 017114 | 2229 | 3736# | | | | | | | | | | | |
| MSG73 | 017124 | 2233 | 3737# | | | | | | | | | | | |
| MSG9 | 015013 | 2730 | 3678# | | | | | | | | | | | |
| MTINT | 012740 | 1939 | 2134 | 3360# | | | | | | | | | | |
| NRZOF | 000724 | 2023# | | | | | | | | | | | | |
| NXTDRV | 002622 | 2222# | | | | | | | | | | | | |
| NXTSLV | 002674 | 2222# | 2235 | 2264 | | | | | | | | | | |
| OCTP | 013702 | 2118 | 2127 | 2153 | 2170 | 2191 | 2228 | 2232 | 2268 | 2513 | 2517 | 2747 | 2752 | 3234 |
| | | 3404 | 3540# | | | | | | | | | | | |
| OCTPE | 013670 | 2353 | 2357 | 3238 | 3240 | 3242 | 3244 | 3246 | 3248 | 3250 | 3252 | 3537# | 3652 | |
| OCTPE1 | 013706 | 3541# | | | | | | | | | | | | |
| OCTPG | 014056 | 3548 | 3551 | 3558 | 3562 | 3567 | 3572 | 3574 | 3578# | | | | | |
| OCTPG0 | 014074 | 3579 | 3581 | 3584# | | | | | | | | | | |
| OCTPG1 | 014100 | 3543 | 3585# | | | | | | | | | | | |
| OCTP0 | 013722 | 3539 | 3542 | 3545# | | | | | | | | | | |
| OCTP1 | 013744 | 3546 | 3550# | | | | | | | | | | | |
| OCTP2 | 013752 | 3549 | 3552# | | | | | | | | | | | |
| OCTP3 | 014042 | 3544 | 3575# | | | | | | | | | | | |
| OFL | 014120 | 3537* | 3540* | 3580 | 3584* | 3590# | | | | | | | | |
| OPDYX | 000640 | 1997# | 2580* | 2613* | 2665* | 2767* | 2797* | 2837* | 3170 | | | | | |
| PATRN | 000720 | 2021# | 2616* | 2898* | 2931* | 3057* | 3086* | 3118* | 3277 | | | | | |
| PCNTR | 000730 | 2025# | 2200* | 2267 | 2278* | 3340 | | | | | | | | |
| PEXFL | 000702 | 2014# | | | | | | | | | | | | |
| PFLG | 000644 | 1999# | | | | | | | | | | | | |
| PSW | 000546 | 1965# | 3372* | 3497 | 3498* | 3502* | | | | | | | | |
| RCNT | 000624 | 1991# | 2614* | 2628* | 2630* | 2637* | 2642* | 2645* | 2650* | 2660* | 2680* | 2773* | 2779* | 2831* |
| | | 2847 | 2849* | 2850 | 2870* | 2878* | | | | | | | | |
| RDATA | 021772 | 2631 | 2709 | 2770 | 2950 | 3070 | 3099 | 3281 | 3773# | | | | | |
| RDSW | 000736 | 2028# | | | | | | | | | | | | |
| RDYDX | 000636 | 1996# | 2579* | 2612* | 2666* | 2766* | 2785* | 2796* | 2836* | 2996* | 3162 | | | |
| REGS | 000572 | 1975# | 2117 | 2119 | 2136 | | | | | | | | | |
| RFD | 000634 | 1995# | 2683* | 2706* | 2720 | | | | | | | | | |
| RHOF | 000726 | 2024# | 2190 | 2192 | 2562 | | | | | | | | | |
| RHTF | 000722 | 2022# | 2148* | 2296 | 2298* | 3411 | | | | | | | | |
| RH17F | 000604 | 1983# | 2205* | 2211* | 2415 | 2441 | 2473 | 2526 | 2546 | | | | | |
| RRD | 000632 | 1994# | 2682* | 2705* | 2713 | | | | | | | | | |
| RTRN | 000646 | 2000# | 3361 | | | | | | | | | | | |
| RWLD | 011772 | 2619 | 2659 | 2769 | 2782 | 2800 | 2834 | 2897 | 2930 | 2944 | 2949 | 2981 | 3056 | 3069 |

| | | | |
|---------|-------|-------|------|
| DTBOOT | 1243# | | |
| GETANS | 767# | | |
| LDPDR | 515# | | |
| LPDP11 | 1268# | | |
| PSPTAG | 746# | | |
| RFGBOX | 132# | | |
| RESLDR | 873# | | |
| SAVLDR | 855# | | |
| SVTK\$ | 1141# | | |
| \$CATCH | 1124# | 1830# | 1911 |
| \$CHAIN | 89# | 1830# | 2107 |
| \$CHMO | 105# | 1830# | 2222 |
| \$CNV16 | 606# | | |
| \$CNV18 | 635# | | |
| \$CNV48 | 704# | | |
| \$CPCHK | 897# | | |
| \$CPREG | 17# | | |
| \$CPVEC | 167# | | |
| \$FPREG | 46# | | |
| \$GETAN | 771# | | |
| \$KMPHE | 347# | | |
| \$KWDR | 998# | | |
| \$KW11 | 929# | | |
| \$LCTRL | 2# | | |
| \$LPREG | 186# | | |
| \$MAMFO | 1176# | | |
| \$MIBIT | 207# | | |
| \$MPREG | 264# | | |
| \$PDREI | 385# | | |
| \$POWER | 439# | | |
| \$PSWEI | 147# | | |
| \$RECO | 795# | | |
| \$RESLD | 876# | | |
| \$RESTO | 476# | 1830# | 3664 |
| \$SAVE | 464# | 1830# | 3663 |
| \$SAVLD | 858# | | |
| \$SETTB | 508# | | |
| \$SHIFT | 489# | | |
| \$SMRE | 309# | | |
| \$STINS | 8# | | |
| \$STKPT | 202# | | |
| \$ST200 | 1136# | | |
| \$SVTK | 1146# | | |
| \$SWOPT | 56# | | |
| \$TCDRV | 1029# | | |
| \$TCREG | 192# | | |
| \$TRAPS | 402# | | |
| \$TYPE | 518# | | |
| \$TYPEF | 592# | | |
| \$UMRE | 272# | | |
| \$VECTA | 1163# | | |
| .\$ACT1 | 67# | 1830# | 1913 |
| .\$EOP | 78# | 1830# | 2272 |

. ABS. 021774 000

TM03/TU45 BASIC FUNCTION TEST
CZTUQB.P11 13-MAY-80 15:20

MACY11 30A(1052) 13-MAY-80 15:25 ^{D 7} PAGE 55-1
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0081

ERRORS DETECTED: 0

CZTUQB.CZTUQB/CRF=CZTUQB.P11
RUN-TIME: 13 22 2 SECONDS
RUN-TIME RATIO: 66/38=1.7
CORE USED: 14K (27 PAGES)