TM03/TE16

CONTROL LOGIC TEST PART 1

MD-11-DZTEA-A

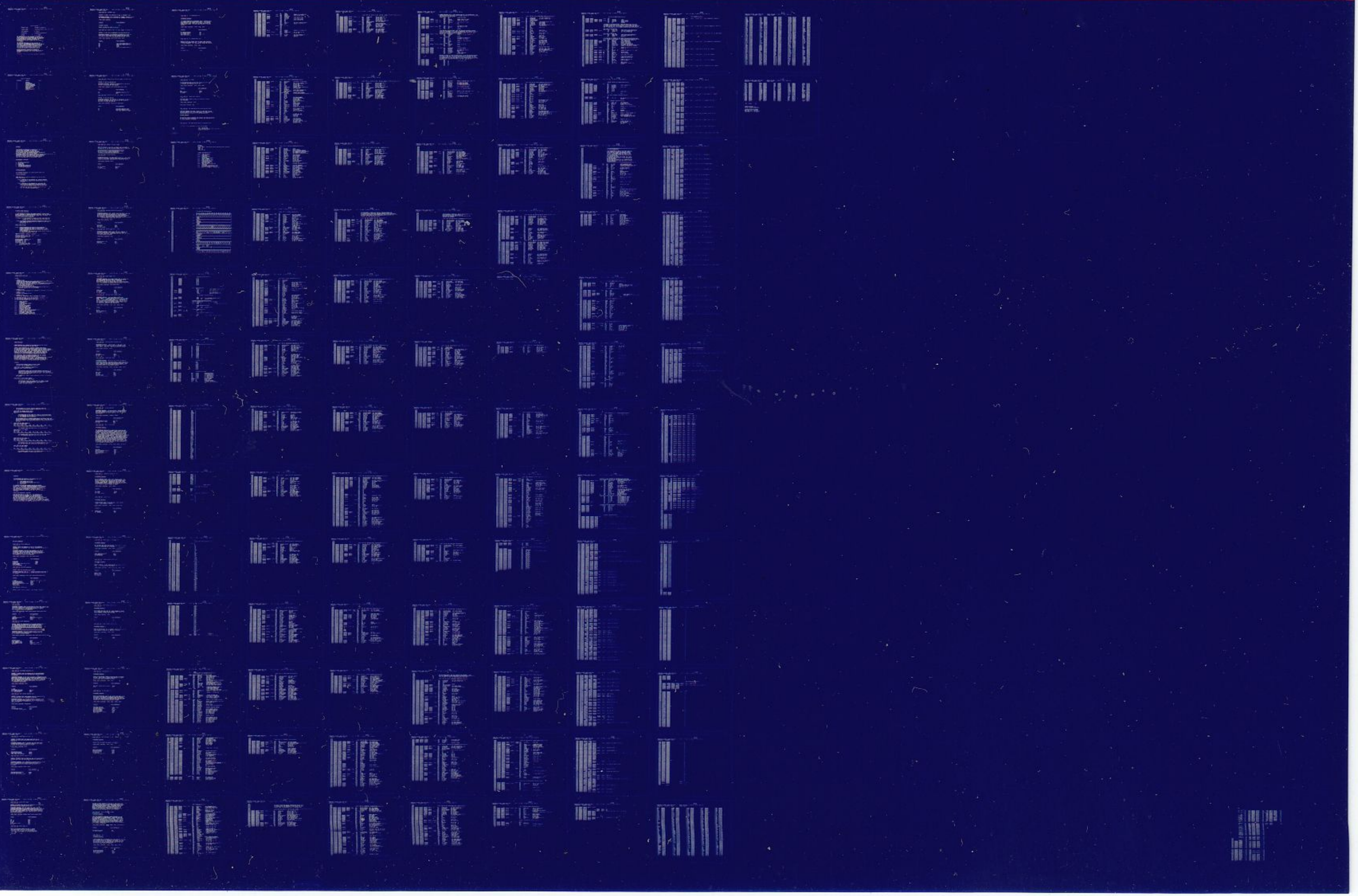
EP-DZTEA-A-DL-A

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



```
.REM    x
```

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZTEA-A-D
PRODUCT NAME:	TM03/TE16 CONTROL LOGIC TEST PART I
DATE CREATED:	21 FEB 77
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHOR:	J. G. ADAMS

TABLE OF CONTENTS

PARAGRAPH	SUBJECT	PAGE
1.	ABSTRACT	3
2.	REQUIREMENTS	3
3.	LOADING PROCEDURE	3
4.	STARTING PROCEDURE	3
5.	SWITCH SETTINGS	4
6.	ERROR PRINTOUTS	6
7.	OPERATION	8
8.	SUBTEST SUMMARIES	9
9.	LISTING	31

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO SEQUENTIALLY TEST ALL
CONTROL LOGIC FUNCTIONALY OF THE TM03.
EACH TEST WILL ATTEMPT TO ISOLATE FAILURES
TO THE MODULE LEVEL AND PROVIDE PRINTOUT INFORMATION
WHICH WILL IDENTIFY THE FAILING MODULE.
THE CONTROL LOGIC TESTS TEST ALL ERROR AND STATUS CONDITIONS AS WELL AS
ADDRESSING PROTOCOL AND OPERATIONAL LOGIC SEQUENCES.
THE LEVEL OF FAULT ISOLATION IS POSSIBLE BECAUSE OF
TM03 THE STRUCTURE AND ITS MAINTAINENCE MODES.

2. REQUIREMENTS (HARDWARE)

A. ANY PDP-11 PROCESSOR
B. 8K OF CORE
C. CONSOLE TTY
D. TM03 MAGTAPE CONTROLLER
E. MASSBUS CONTROLLER (RH)
F. TE16 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

THERE ARE TWO (2) STARTING ADDRESSES THAT MAY BE USED:
200(8) AND 210(8).

A. 200(8): STARTING AT THIS ADDRESS WILL CAUSE A PROGRAM
IDENTIFICATION HEADER TO BE PRINTED BEFORE TESTING
IS BEGUN.

B. 210(8): STARTING AT THIS ADDRESS WILL NOT PRINT THE
IDENTIFICATION HEADER AND IS THEREFORE GENERALLY
TO BE USED FOR RESTARTS RATHER THAN INITIAL START

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

4.1 AUTOMATIC MODE OPERATION

IF THIS PROGRAM IS LOADED & RUN UNDER AUTOMATIC (CHAIN) MODES DEFAULT RESPONSES TO OPERATOR REQUESTS ARE USED, AND THE SOFTWARE SWR INVOKED WITH A SWITCH SETTING OF 100000 (HALT ON ERROR). NO OPERATOR INTERVENTION IS REQUIRED.

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

**NOTE:THIS PROGRAM CONTAINS OPERATOR INTERVENTION TESTS.TO RUN THESE TESTS THE PROGRAM MUST BE LOADED IN 'DUMP' MODE AND SW09 SET TO 1.

4.2 SAMPLE START AT 200

**NOTE: DEFAULT RESPONSES ARE SHOWN IN ANGLE BRACKETS <>,
OPERATOR RESONSES ARE SHOWN IN PARENTHESES (), AND
MEMORY LOCATIONS CONTAINING THE DEFAULT ARE SHOWN IN
SQUARE BRACKETS [].
IN THIS EXAMPLE THE OPERATOR HAS CHOSEN DEFAULT RESPONSES.
TO INVOKE THE DEFAULT TYPE (CR).

```
PARAMETER REQUEST: <DEFAULT> (RESONSE)          [LOCATION:]

TM03-TE16 CONTROL LOGIC TEST- PART I (DZTEA-A)
***ASSURE TAPE IS AT BOT***
TYPE ↑C TO RESTART

REGISTER START: <172440> (CR)                   [REGS:]
VECTOR ADDRESS: <224> (CR)                      [VECT:]
TM03 DRIVE: <0> (CR)                            [DRVN:]
TE16 SLAVE: <0> (CR)                            [SLVN:]
STATIC TESTS ONLY: <0> (CR)                     [STATC:]
        IF THE SOFTWARE SWR IS INVOKED:
SWR = <000000> NEW = (CR)                       [SWREG:]
```

5. CONSOLE SWITCH SETTINGS

CONTROL:

1) CONTROL G (↑G):
   INVOKES THE SOFTWARE SWR AND ALLOWS USER TO ENTER SWITCH SETTING
   THE MACHINE WILL THEN TYPE: SWR=XXXXXX NEW=
   WHERE: XX'XXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.
   AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
   OF THE FOLLOWING AT THE TTY:
     A) TYPE THE 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 SWITCH REGISTER FROM HARDWARE TO SOFTWARE & VICE VERSA

3) CONTROL C (↑C):
   RESTARTS THE PROGRAM AT 200

4) CONTROL U (↑U):
   DELETES ALL CHARACTERS TYPED IN RESPONSE TO A REQUEST

ALL SWITCHES ARE USED (0-15) AND THE NORMAL, OR DEFAULT, RUN
IS DONE WITH ALL SWITCHES SET TO ZERO (0).
ALL SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME.

SW15: 1=HALT ON ERROR
       0=CONTINUE
SW14: 1=LOOP ON ERROR (SCOPE)
       0=CONTINUE
SW13: 1=DO NOT PRINT ERRORS
       0=PRINT ALL ERRORS
SW12: 1=DO CONTINUOUS CYCLE
       0=HALT AT END OF PASS
SW11: 1=INHIBIT ITERATIONS
       0=ITERATE EACH TEST ITS ASSIGNED AMOUNT
SW10: 1=HALT AT END OF CURRENT TEST
       0=CONTINUE TO NEXT TEST
SW9:   1=DO MANUAL ITERVENTION TESTS
       0=INHIBIT MANUAL INTERVENTION
SW5-0: SELECT INDIVIDUAL TEST ** 00=DO ALL TESTS

6. ERROR PRINTOUTS

ERROR PRINTOUTS WILL APPEAR IN TWO FORMS ONE FOR THE CONTROL LOGIC TESTS AND ANOTHER FOR THE DATA TESTS.

CONTROL LOGIC PRINTOUTS WILL CONTAIN A HEADER WHICH CALLS OUT THE TEST NUMBER, FUNCTION BEING TESTED, AND THE SUSPECT MODULE OR MODULES ON THE FIRST LINE. THE SECOND LINE WILL CONTAIN INFORMATION AS TO THE ACTUAL ERROR. BOTH THE EXPECTED RESULT AND THE ACTUAL RESULT OF THE TEST WILL BE GIVEN. LINE THREE WILL SHOW THE CONTENTS OF THE MAJOR REGISTERS AT THE TIME OF THE ERROR AND LINE FOUR WILL PRINT THE ITERATION NUMBER WHEN APPLICABLE.

DATA TESTS WILL PRINT A HEADER CONTAINING THE TEST NUMBER, AND A DESCRIPTION OF THE FUNCTION UNDER TEST. FOLLOWING THE HEADER WILL BE A LIST OF THE MAJOR REGISTERS WITH THE EXPECTED AND ACTUAL VALUES. ANY BAD DATA WILL BE PRINTED (PER CHARACTER) FOLLOWING THE REGISTER INFORMATION OR FOLLOWING THE HEADER IF NO STATUS ERRORS WERE ENCOUNTERED.

EXAMPLES:

1. THE FOLLOWING EXAMPLE SHOWS A TYPICAL ERROR PRINTOUT FOR THE ADDRESS TESTS (LT1-LT3).

```
LOGIC TEST 1: DRIVE ADDRESSING (MB909 OR RH)
NON-EXIST DRIVE 3 EXPT-NOT RECVD
ITER: 3
```

THIS PRINTOUT SHOWS THAT THE DRIVE ADDRESS (CS2 BITS 2,1,0) RESULTED IN THE DETECTION OF NED (BIT 12 OF CS2) FOR DRIVE THREE (3) WHEN THAT DRIVE SHOULD BE THERE. THIS ERROR OCCURRED ON ITERATION THREE (3).

2. THIS EXAMPLE WILL SHOW A TYPICAL PRINTOUT OF ONE OF THE REGISTER BIT TESTS.

```
LOGIC TEST 7: FC BIT TEST (MB705)
FC BITS 15-0 EXPT 177777 RECVD 177577
```

THIS PRINTOUT SHOWS THAT FRAME COUNT BIT SEVEN (7) WAS NOT SET WHEN IT SHOULD HAVE BEEN. NO ITERATION NUMBER IS DISPLAYED WHEN RUNNING WITH CONSOLE SWITCH TWELVE (12) SET TO A ONE (1).

3. THE FOLLOWING IS A TYPICAL PRINTOUT RESULTING FROM BAD STATUS DETECTION DURING A MANUAL INTERVENTION TEST (LT14-LT17)

```
LOGIC TEST 15: MANUAL STATUS TEST 2
BAD STATUS EXPT 100700 RCVD 000700
ITER: 0
```

THIS SHOWS THAT ON THE FIRST TRY (ITER: 0) THE ACTION TAKEN BY THE OPERATOR DID NOT RESULT IN THE PROPER STATUS DETECTION BY THE HARDWARE (ATA IS NOT SET).

4. THE FOLLOWING FOUR (4) EXAMPLES SHOW EACH OF THE ERROR TYPES THAT CAN BE DETECTED BY ANY OF THE ERROR FORCING TESTS. NOTE THAT ONE OR MORE OF THE ERROR TYPES COULD BE DETECTED ON A SINGLE EXECUTION OF THE TEST.

```
LOGIC TEST 24: DPAR (M8906 RH)
DPAR EXPT EXPT-NOT RCVD
CS1    WC     BA     FC     CS2    DS     ER     AS     MR     TC
004260 000000 033726 000000 000100 010600 000000 000000 177712 140300
```

THIS MESSAGE SHOWS THAT DPAR (BIT 5 OF ER) DID NOT SET.

```
LOGIC TEST 26: FCE (M8909)
ERR NOT SET
CS1    WC     BA     FC     CS2    DS     ER     AS     MR     TC
004260 000000 001376 000000 000100 110600 001000 000001 000000 100300
```

THIS MESSAGE SHOWS THAT WHILE FCE (BIT 9 OF ER) WAS INDEED SET, THE COMPOSITE ERROR BIT (BIT 14 OF DS) WAS NOT.

```
LOGIC TEST 30: DTE (M8906 RH)
UNEXPTECTED ERROR BITS
CS1    WC     BA     FC     CS2    DS     ER     AS     MR     TC
144260 002006 006600 000000 001300 150600 030000 000001 000017 100300
```

THIS MESSAGE SHOWS THAT WHILE THE PROPER ERROR BIT (DTE: BIT 12 OF ER) IS SET, OPI (BIT 13 OF ER) IS ALSO SET AND SHOULD NOT BE.

```
LOGIC TEST 32: UNS (M8909)
NOT RESET BY DRIVE CLEAR

CS1    WC     BA     FC     CS2    DS     ER     AS     MR     TC
144210 002006 006600 000000 001300 150000 040000 000001 000000 140307
```

THIS MESSAGE SHOWS THAT WHILE THE PROPER ERROR BITS WERE SET, THEY WERE NOT CLEARED BY A DRIVE CLEAR OPERATION.

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
3. PRESS START

ALL CONSOLE SWITCHES ARE DYNAMIC AND MAY BE CHANGED AT ANY TIME. THE NORMAL OPERATING SEQUENCE IS ALL SWITCHES DOWN (0). THE TEST WILL TAKE APPROXIMATELY 3 MINUTES TO RUN; HOWEVER, IF ITERATIONS ARE INHIBITED (SW11=1) THE TEST WILL RUN IN ABOUT 30 SECONDS. THE END OF PASS IS NOTED BY A PRINTOUT STATING END OF PASS, AND THE NUMBER OF THAT PASS.

SINGLE TEST SELECTION: (SW0-SW5)

WHEN SW0-SW5 ARE SET TO ZERO (00), THE SCHEDULAR WILL EXECUTE ALL TESTS IN SEQUENCE. IF SW0-SW5 ARE SET TO SOME SPECIFIC TEST NUMBER THEN THAT PARTICULAR TEST ONLY WILL BE EXECUTED UNTIL THE TEST SELECT NUMBER IS CHANGED. WHEN YOU WISH TO SELECT A PARTICULAR TEST, SET SW10 TO A ONE (1) IN ORDER TO STOP AT THE END OF THE CURRENT TEST BEFORE SELECTING A DIFFERENT TEST NUMBER. YOU MAY SELECT THAT NUMBER IN ANY DIRECTION (HIGHER OR LOWER) BECAUSE EACH TEST IS SELF CONTAINED.

8. SUB TEST SUMMARIES

LOGIC TEST #1: DRIVE ADDRESSING

PURPOSE: VERIFY THE PRESENCE OF TM03 AT THE ADDRESSES SPECIFIED BY THE OPERATOR. TEST OCCURS IMMEDIATELY AFTER DRIVE SELECTION.

PROGRAMMED SEQUENCE: FOR EACH TM03 ADDRESS (0-7) THE C1 REGISTER IS READ, AND THE NON-EXISTANT DRIVE (NED) BIT IS CHECKED. NED IS SET WHEN THE TM03 DOES NOT RESPOND TO DEM BY ISSUING TRA. IN THIS TEST, NED IS EXPECTED FOR EACH ADDRESS NOT TYPED BY THE OPERATOR.

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909

CIRCUITS	PRINT REFERENCES
RH-DS BITS	(CSR8)
RH-NED BIT	(CSR8)
MASSBUS CABLE C(DEM,TRA,DS BITS)	(MB3)
DRIVE ADDRESS	(MBI2)
DEM-TRA HANDSHAKE	

LOGIC TEST #2: REGISTER ADDRESSING

PURPOSE: CHECK THE REGISTER SELECT LINES

PROGRAMMED SEQUENCE: READ ALL 14 MASSBUS REGISTERS WHICH MAKE UP THE TAPE SYSTEM CHECKING FOR (1) CONTROL BUS PARITY ERROR AND (2) ILR BIT

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8909,M8905,M8903

CIRCUITS	PRINT REFERENCE
C-LINES	(MBI,2,3),(MBI3),(MBI4),(MBI5)
RH REGISTER SELECT	(BCTA)
TM03 REGISTER SELECT	(MBI2)
MASSBUS REGISTER SELECT LINES	(MBI,2)
PARITY TREE	(MBI4)
CPAR,ILR BITS	(MBI11)

LOGIC TEST #3: CONTROL BUS

PURPOSE: VERIFY THAT ALL CONTROL LINES PROPERLY TRANSMIT

ONES AND ZEROS.

PROGRAMMED SEQUENCE: WRITE FC REGISTER AND CHECK CPAR, READ FC AND CHECK MCPE, UPDATE DATA, REPEAT. DATA IS ALL 0'S, WALKING '1' BIT, ALL '0'S, 2 WALKING '1' BITS BEGINNING WITH BIT 0 AND 8
DATA IS CHECKED ALONG WITH ERROR BITS.

LIKELY FAULT LOCATIONS: M5904,CABLES,M5903YA,M8909,M8905,M8903

CIRCUITS	PRINT REFERENCE
C-LINES	(MBI,2,3)
C-BUS MULTIPLEXERS	(MBI3,4,5,8)(TCCM7)(MR)
ERROR BIT	(MBIII)
MCPE BIT	(PACA)

LOGIC TEST #4: SLAVE ADDRESSING

PURPOSE: VERIFY THE FUNCTIONING OF THE SLAVE ADDRESS BITS IN THE TAPE CONTROL REGISTER THE SLAVE ADDRESS BUS LINES, THE ADDRESS DECODE CIRCUIT IN THE TE16 AND THE SPR BIT.

```
**********************************************************
*IT IS REQUIRED THAT ONLY ONE SLAVE BE POWERED UP WHEN   *
*THIS TEST IS RUN.                                       *
**********************************************************
```

PROGRAMMED SEQUENCE: THE SLAVE ADDRESS BITS IN THE TAPE CONTROL REGISTER ARE LOADED WITH ALL 8 COMBINATIONS AND SPR IS CHECKED FOR EACH ADDRESS.

LIKELY FAULTS LOCATIONS: M8905,M8907,CABLE,M9001,M8910,M9001YA,M8903

CIRCUITS	PRINT REFERENCE
REGISTER SELECT	(MBI2)
SLAVE ADDRESS BITS	(MR6)
SLAVE ADDRESS LINES	(M8907,2-2),(LAW6)
TE16 ADDRESS DECODE	(LAW6)
SPR BIT	(LAW6)(M9001YA)(TCCM7)

LOGIC TEST #5: MAINTENANCE REGISTER BITS

PURPOSE: TO VERIFY THAT THE VARIOUS BITS OF THE MAINTENANCE REGISTER CAN BE WRITTEN INTO AND READ AND OTHERWISE BEHAVE AS EXPECTED.

PROGRAMMED SEQUENCE: IN THE FIRST SEQUENCE AN INCREMENTING DATA WORD (0-37) IS WRITTEN INTO THE MR. WITH THE CONTENTS OF BITS 0-4 BEING CHECKED AFTER EACH OPERATION. THEN 15(OCTAL) IS WRITTEN INTO THE REGISTER WHICH SHOULD PERMIT BITS 7-15 TO BE WRITTEN FROM THE CONTROL BUS. THEN THE DATA WRITTEN INTO BITS 7-15 IS INCREMENTED AND CHECKED.

LIKELY FAULT LOCATIONS: M8905

CIRCUITS	PRINT REFERENCE
C-LINES	
MAINTENANCE REGISTER	(MR2,3,5)
M.R. FUNCTION DECODE	(MR5)
M.R. MULTIPLEXOR	(MR4)

LOGIC TEST #6: TAPE CONTROL REGISTER BITS

PURPOSE: TO VERIFY THAT TAPE CONTROL BITS 0-11 CAN BE WRITTEN INTO AND READ AND THAT TCW BEHAVES AS EXPECTED:

PROGRAMMED SEQUENCE: ALL 0'S DATA PATTERN IS WRITTEN TO AND READ FROM THE TAPE CONTROL REGISTER. TCW IS CHECKED FOR A "ONE". THIS SEQUENCE IS REPEATED WITH ALL "1" DATA AND AGAIN WITH ALL "0"'S.

LIKELY FAULT LOCATIONS: M8909,M8905

CURCUITS	PRINT REFERENCE
TM03 REGISTER SELECT	(MB12)
TC FLIP-FLOPS, MULTIPLEXERS	(MR6)

LOGIC TEST #7: FRAME COUNT BIT TEST

PURPOSE: TO VERIFY THAT THE FRAME COUNT BITS CAN BE WRITTEN INTO AND READ FROM AND ARE NEITHER STUCK AT 0 NOR STUCK AT 1.

PROGRAMMED SEQUENCE: DATA IS WRITTEN INTO THE FRAME COUNT REGISTER AND READ FROM IT. THE DATA PATTERN IS ALL ZEROS FOLLOWED BY ALL ONES FOLLOWED BY ALL ZEROS.

LIKELY FAULT LOCATIONS: M8909

CIRCUITS	PRINT REFERENCE
TM03 REGISTER SELECT	(M8I2)
FRAME COUNT REGISTER	(M8I8)
FRAME COUNT MULTIPLEXERS	(M8I10)

LOGIC TEST #10: FUNCTION CODE BIT TEST

PURPOSE: TO VERIFY THAT THE FUNCTION CODE BITS CAN BE WRITTEN INTO AND READ FROM AND ARE NEITHER STUCK AT 0 NOR STUCK AT 1.

PROGRAMMED SEQUENCE: THE C1 REGISTER IS WRITTEN WITH ALL ZEROS. DATA IS CHECKED ON THE 5 FUNCTION CODE BITS (BITS 1-5). BITS 1-5 ARE WRITTEN WITH ONES, CHECK AND REPEAT WITH ALL ZEROS.

LIKELY FAULT LOCATION: M8909, M8905

CIRCUITS	PRINT REFERENCE
TM03 REGISTER SELECTION	(M8I2)
FUNCTION CODE FLOPS	(M8I5)
FUNCTION CODE MULTIPLEXERS	(MR6)

LOGIC TEST #11: GO BIT SET, RESET

PURPOSE: TO VERIFY THAT THE GO BIT CAN BE SET IN A SIMULATED READ OPERATION AND CLEARED WITH AN INIT.

PROGRAMMED SEQUENCE: INIT AND CHECK THAT GO=0. SET UP A SIMULATED READ OPERATION BY LOADING A WAM3 15(OCTAL) INTO THE MAINTENANCE REGISTER, CLEARING THE FRAME COUNT REGISTER TO SET FCS, LOAD 1700 (FORMAT) INTO THE TAPE CONTROL REGISTER, SETTING READ COMMAND AND GO BIT. CHECK FOR GO=1. INIT AND CHECK THAT GO BIT=0.

LIKELY FAULT LOCATION: MASSBUS CABLE B(INIT),M8909,M8905

CIRCUIT	PRINT REFERENCE
FCS	M8I8
SET ILF	M8I7
SET NEF	M8I7
GO BIT	M8I5
GO BIT MULTIPLEXER	MR6
SET ILR	M8I2

LOGIC TEST #12: DRIVE READY BIT

TEST 12 IS AN EXACT REPEAT OF TEST 11 EXCEPT THAT DRIVE READY (DRY) IS CHECKED INSTEAD OF THE GO BIT. DRY IS SIMPLY GO L MULTIPLEXED ONTO THE C-LINES AS BIT SEVEN OF THE STATUS REGISTER.

PRINT REF TCCM7

LOGIC TEST #13: INTERRUPT TEST

PURPOSE: TO VERIFY THE OPERATION OF THE RH INTERRUPT LOGIC.

PROGRAMMED SEQUENCE: THE C1 REGISTER IS CLEARED, PRIORITY IS SET, THE INTERRUPT ENABLE BIT IS SET AND THE INTERRUPT IS AWAITED.

LIKELY FAULT LOCATION:

CIRCUITS	PRINT REFERENCE
INTERRUPT CONTROL	BCTF

MANUAL INTERVENTION TESTS 14,15,16,17

LOGIC TEST #14: STATUS AT BOT, ON LINE, LOADED, NO WRITE RING

PURPOSE: TO TEST FOR THE PRESENCE OF MOL,WRL,DPR,DRY,BOT.

PROGRAMMED SEQUENCE: THE OPERATOR IS INSTRUCTED TO LOAD THE DRIVE WITH A TAPE MINUS THE WRITE ENABLE RING AND PLACE THE DRIVE ON LINE AT BOT MOL,WRL,DPR,DRY,BOT ARE CHECKED.

LIKELY FAULT LOCATION: M8910,SLAVE CABLE, M8903

CIRCUIT	PRINT REFERENCE
MOL	LAM6,TCCM7,M8908,M9001YA,YC
WRL	LAM8,TCCM7,M8908,M9001YA,YC
DPR	TCCM7
DRY	TCCM7
BOT	LAM6,TCCM7,M8908YA,M8913,YA

LOGIC TEST #15: STATUS AT BOT,OFFLINE,LOADED, NO WRITE RING

PURPOSE: TO TEST ATA,DPR,DRY,SSC

PROGRAMMED SEQUENCE: OPERATOR IS INSTRUCTED TO TAKE DRIVE
OFFLINE: ATA,SSC,DPR,DRY ARE CHECKED.

LIKELY FAULT LOCATION: M8910,M8903,M8909,SLAVE CABLE

CIRCUIT	PRINT REFERENCE
SSC	LAW8,M8913,M8913YA,TCCM7
ATA	M8I3

LOGIC TEST #16: STATUS AT EOT,ON LINE, LOADED, NO WRITE RING

PURPOSE: TO TEST EOT,SSC,SLA

PROGRAMMED SEQUENCE: THE OPERATOR IS INSTRUCTED TO MOVE TO EOT
AND PLACE THE DRIVE ON LINE. EOT,SSC,SLA ARE CHECKED IN
ADDITION TO ATA,MOL,WEL,DPR,DRY

LIKELY FAULT LOCATION: M8910,SLAVE CABLE,M8903

CIRCUIT	PRINT REFERENCE
SSC	LAW8,M8913,M8913YA,TCCM7
EOT	LAW6,TCCM7,M8908YA,M8913YA
SLA	LAW8,TCCM7,M9001YA,YC,M8908

LOGIC TEST #17: STATUS AT ONLINE LOADED

TEST 17 IS EXACTLY LIKE TEST 16 EXCEPT THAT THE DRIVE IS REVERSED OFF OF EOT AND THE WRITE ENABLE RING IS INSTALLED.

****
EACH OF THE NEXT 11 TESTS ARE DESIGNED TO VERIFY
THE ABILITY TO SET SPECIFIC ERROR BITS.
****

LOGIC TEST #20: ILLEGAL FUNCTION

PROGRAMMED SEQUENCE: THE WORD COUNT IS SET TO -1, ALL CODES STORED IN THE ILLEGAL FUNCTION TABLE ARE LOADED AND ILF IS CHECKED FOR EACH ONE. THEN UNEXPECTED ERRORS ARE CHECKED.

LIKELY FAULT LOCATION: ME909

CIRCUIT	PRINT REFERENCE
SET ILF DECODE	MBI5,MBI7
ILF FLOP	MBI11
ILF MULTIPLEXER	MBI10

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

PROGRAMMED SEQUENCE: INIT, SELECT SLAVE AND DRIVE. LOAD 300
@ TAPE CONTROL REGISTER LOAD WAM3 IN THE MAINTENANCE
REGISTER. LOAD THE C1 REGISTER WITH A READ COMMAND AND GO
BIT. ATTEMPT TO WRITE THE FRAME COUNT REGISTER. READ
ERROR REGISTER. CHECKING FOR RMR. CHECK FOR UNEXPECTED ERRORS
WAIT FOR ACCL. DELAY. DO EOP CLEAR.

LIKELY FAULT LOCATION: MB909

CIRCUIT	PRINT REFERENCE
RMR DECODE	MBI2
RMR FLOP	MBI11
RMR MULTIPLEXER	MBI10

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)

PROGRAMMED SEQUENCE: WRITE 20(8) INTO CS2. ENABLING THE
WRITING OF EVEN PARITY ON MASSBUS. WRITE ALL ONES TO
FRAME COUNT. RESET PAT. CHECK ERROR REGISTER FOR CPAR CHECK
FOR OTHER UNEXPECTED ERRORS.

LIKELY FAULT LOCATIONS: MB909

CIRCUIT	PRINT REFERENCE
MASSBUS PARITY TREE	MBI4
CPAR FLOP	MBI11
CPAR MULTIPLEXER	MBI10

LOGIC TEST #23: FORMAT ERROR (FMT)

PROGRAMMED SEQUENCE: AN ILLEGAL FORMAT CODE IS LOADED INTO THE TAPE CONTROL REGISTER. WAM3 IS LOADED INTO THE MR READ COMMAND AND THE GO BIT IS SET. THE ERROR REGISTER IS CHECKED FOR FORMAT ERROR AND UNEXPECTED ERROR BITS. THIS SEQUENCE IS REPEATED FOR ALL ILLEGAL FORMAT CODES

LIKELY FAULT LOCATIONS: M8905,M8906,M8909

CIRCUIT	PRINT REFERENCE
FORMAT BITS	MR6
ILF DECODE	BF3
ILF FLOP	M8I11
ILF MULTIPLEXERS	M8I10

LOGIC TEST #24: DATA BUS PARITY ERROR (DPAR)

PROGRAMMED SEQUENCE: SET UP A WRAP 2 AS FOLLOWS: NORMAL FORMAT ----> TAPE CONTROL REGISTER, -10 ----> WORD COUNT, -20 ----> FRAME COUNT, WAM2 ----> MAINTENANCE REGISTER. LOAD WRITE COMMAND AND GO BIT. SET PAT BIT IN CS2. AFTER A DELAY MR IS LOADED 4 TIMES CAUSING 2 DATA BUS TRANSFERS. DPAR AND CPAR ARE CHECKED. THEN A CHECK FOR UNEXPECTED ERRORS IS MADE MASKING OPI.

LIKELY FAULT LOCATIONS: DBUS LINES, M8905, M8906

CIRCUIT	PRINT REFERENCE
MM CLK	MR5
WRT CLK GENERATION	TCCM4
DPAR FLOP	M8I11
DATA BUS PARITY TREE	BF3

LOGIC TEST #25: NON-EXECUTABLE FUNCTION (NEF)

PROGRAMMED SEQUENCE: LOAD FC WITH -1. SET WAM 2. SET WRITE AND GO. ILF SHOULD SET DUE TO TOO SMALL INITIAL FRAME COUNT. CHECK ILF. CHECK FOR UNEXPECTED ERRORS.

LIKELY FAULT LOCATION: MB909

CIRCUIT	PRINT REFERENCE
NEF FLOP	MBI11
NEF MULTIPLEXER	MBI10
SET NEF	MBI7

LOGIC TEST #26: FRAME COUNT ERROR

PROGRAMMED SEQUENCE: SET WC TO -10, FC TO -20 WAM3 IN MAINTENANCE REGISTER, LOAD WRITE AND GO, DELAY ISSUE MM OR CLEAR. CHECK FCE AND CHECK FOR UNEXPECTED ERRORS. FRAME COUNT ERROR SHOULD BE SET BECAUSE A WRITE OPERATION WAS TERMINATED PRIOR TO A WORD COUNT OVERFLOW.

LIKELY FAULT LOCATIONS: MB909, MB CABLE, MB903, MB905

CIRCUITS	PRINT REFERENCE
RUN LINE	MB1
EBL PLS	MBI9
FCE FLOP	MBI11
SHUTDOWN LOGIC	TCCM5
MAINT. FUNCTION DECODE	MR5

LOGIC TEST #27: ILLEGAL REGISTER

PROGRAMMED SEQUENCE: IF THE RH HAS ALL MASSBUS REGISTER OPEN (MOST SYSTEM IN THE FIELD DON'T), ALL THE ILLEGAL REGISTER ADDRESSES ARE READ, CHECKING THE ILR BIT AFTER EACH ATTEMPT.

LIKELY FAULT LOCATIONS: MASSBUSS, M8909

CIRCUITS	PRINT REFERENCE
REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MB12
ILR FLOP	MB111

LOGIC TEST #30: DRIVE TIMING ERROR

PROGRAMMED SEQUENCE:

THE MAINTENANCE REGISTER IS LOADED WITH A FUNCTION THAT IS DESIGNED TO CRIPPLE OCCUPIED. FRAME COUNT REGISTER IS CLEARED TO SET FCS LOAD WRITE COMMAND AND GO BIT. CHECK FOR DTE. THEN DRIVE IS INITIALIZED. FCS IS SET AND WRP 3 CODE IS LOADED INTO MR. WRITE COMMAND AND GO BIT ARE SET. AFTER DELAY FOR ACCELERATION, THE MR CLOCK IS GENERATED AND ANOTHER CHECK IS MADE FOR DTE. FINAL CHECK IS MADE FOR ERRORS OTHER THAN OPI. THE FIRST MAINTENANCE REGISTER CODE WHICH CRIPPLES THE OCCUPIED RECEIVER CAUSES OCCUPIED TO BE ASSERTED AND TESTS THE CIRCUITRY WHICH CHECKS FOR OCCUPIED WHEN A DATA TRANSFER COMMAND IS INITIATED. THE SECOND TEST UTILIZES THE FACT THAT THE WRP 3 CODE INHIBITS THE MASSBUS WCLK RECEIVER CREATING A SITUATION WHERE SCLK IS NOT FOLLOWED BY A WRITE CLOCK.

LIKELY FAULT LOCATIONS: M8909, M8905, M8906, M8 CABLES

CIRCUITS	PRINT REFERENCES
DTE FLOP	MB111
CRIPPLE OCCUPIED FUNCTION	MR5
WRP 3 FUNCTION	MR5
PREVIOUS OCCUPIED CHECK	MB17
CHECK FOR WCLK	BF2
MM CLK	MR5

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

SET UP INCLUDES FORMAT, WRP 2 (BIT FIDDLER WRITE), FCS. WRITE COMMAND AND GO BIT ARE SET AND THE PROGRAM DELAYS FOR OPI. A SECOND TEST INVOLVES SETTING UP WRP 3 AND ISSUING A READ COMMAND. ESSENTIALLY THIS TEST UTILIZES THE WRAPAROUND CODES TO PREVENT ANY RECORDS BEING DETECTED AFTER A READ OR A WRITE COMMAND IS ISSUED.

LIKELY FAULT LOCATIONS: M8903, M8909

CIRCUITS	PRINT REFERENCES
OPI TIMER	TCCM5
OPI FLOP	M8I11
OPI TIMER CONTROL	M8I7

LOGIC TEST 32: UNSAFE (UNS)

PROGRAMMED SEQUENCE:

A NON-EXISTANT SLAVE IS SELECTED AND A READ COMMAND IS ISSUED. UNSAFE ERROR IS CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8910, SLAVE CABLE

CIRCUITS	PRINT REFERENCES
UNSAFE FLOP	M8I11
SET UNSAFE	M8I7
MOL GENERATION	LAW6

LOGIC TEST 33: POSITIONING IN PROGRESS (PIP)

PROGRAMMED SEQUENCE:

SET UP DRIVE AND SLAVE ARE SELECTED, FCS IS SET. A SPACE COMMAND IS ISSUED AND PIP IS CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8903

CIRCUITS	PRINT REFERENCES
SPACE FUNCTION DECODE	M8IS
PIP GENERATION	TCCM7
STATUS REGISTER	TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

DENSITY CODES 0 - 4 ARE LOADED AND PES IS CHECKED FOR EACH CODE. IT IS EXPECTED ONLY FOR DENSITY 4.

LIKELY FAULT LOCATIONS: M8905, SLAVE BUS, M8911, M8903

CIRCUITS	PRINT REFERENCES
DENSITY BITS	MR6
DENSITY LINES	SBC
PES CIRCUIT	SC3
PES STATUS BIT	TCCM7

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

SETUP FORMAT AND WRP-3 ARE SET, READ COMMAND IS ISSUED. TCW IS CHECKED. DRIVE IS INITIALIZED, TAPE CONTROL REGISTER IS WRITTEN TO AND TCW IS CHECKED.

LIKELY FAULT LOCATION: M8905

CIRCUIT	PRINT REFERENCES
TCW	MR6

LOGIC TEST 36: FRAME COUNTER STATUS (FCS)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, FCS IS CHECKED, DRIVE IS INITIALIZED, FRAME COUNTER IS WRITTEN TO, AND FCS IS CHECKED.

LIKELY FAULT LOCATIONS: M8909, M8903

CIRCUITS	PRINT REFERENCES
FCS BIT	M8I8
FCS MULTIPLEXER	TCCM7

LOGIC TEST 37: ACCELERATION (ACCL)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, FORMAT IS SET AND ACCL IS CHECKED
FOR ONE. WAM 3 CODE IS LOADED, READ COMMAND IS ISSUED.
AFTER A DELAY ACCL IS CHECKED FOR ZERO.

LIKELY FAULT LOCATIONS: M8903, M8911

CIRCUITS	PRINT REFERENCES
ACCL BIT, MOTION DELAY COUNTER	TCCM3
CLOCK	SC2

LOGIC TEST 40: PE TAPE MARK (TM)

PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, WAMO IS SET, WRITE TAPE MARK IS SET.
AFTER DELAY TAPE MARK BIT IS CHECKED. WAMO MULTIPLEXES
THE OUTPUT OF THE WRITE DATA GENERATOR ONTO THE RDA LINES.
THE DATA SYNC MODULES SYNC ON THE DATA AND SEND ENVELOPE
INFORMATION TO THE TAPE MARK DETECTOR ON M8902.

LIKELY FAULT LOCATIONS: M8902, M8901, M8903, M8905

CIRCUITS	PRINT REFERENCES
TAPE MARK DETECTOR	TCPE4, TCPE5
TAPE MARK MULTIPLEXER	TCCM7
ENVELOPE SIGNALS	DS 3, 5, 7
WRITE DATA BUFFER	TCCM2
RDA MULTIPLEXERS	TCCM6
WRITE TAPE MARK FUNCTION	M8I5
WAMO SIGNAL	MR5

LOGIC TEST 41: NRZ TAPE MARK (TM VPE, ITM)

PROGRAMMED SEQUENCE:

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8903, M8904

CIRCUITS	PRINT REFERENCES
WRITE DATA BUFFER	TCCM2
RSDO MULTIPLEXER	TCCM6
RDA MULTIPLEXERS	TCCM6
TM DETECTOR	CNRZ4
ILLEGAL TAPE MARK FLOP	CNRZ4

THE NEXT 5 TESTS CONSISTS OF WRITING ON TAPE USING MAINTENANCE MODE FUNCTIONS TO FORCE ERROR CONDITIONS TO CHECK THE ERROR CHECKING CAPABILITIES. OCCASIONAL ERRORS MAY RESULT FROM TAPE DEFECTS. CONSTANT ERROR MAY BE THE RESULT OF PROBLEMS WITH ERROR CHECKING CIRCUITRY OR PROBLEMS WITH THE DRIVE. DEBUG OF THE PROBLEMS MAY BE EASIER USING DATA RELIABILITY OF UTILITY DRIVER.

LOGIC TEST 42: CYCLIC REDUNDANCY ERROR

PROGRAMMED SEQUENCE:

FIRST THE DIAGNOSTIC PERFORMS A WRAP0 DESIGNED TO LOAD THE CRC CHECKER IN A KNOWN MANNER. CHECK ARE MADE FOR LRC ERROR AND THE CONTENT OF CRC REGISTER. THEN A WRITE OPERATION IS PERFORMED USING A MAINT. MODE (IICC) WHICH INHIBITS THE INITIALIZATION OF THE CRC CHECKER. THE CRC CHECKER LOGIC WHICH HAS NOT BEEN CLEARED SHOULD DETECT A CRC ERROR. UNEXPECTED ERROR BITS MAY INDICATE PROBLEMS WITH THE WRITE OPERATION.

LIKELY FAULT LOCATIONS: M8905, M8904, G056, SLAVE CABLE, M8910

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
CRC CHECK CIRCUIT	CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL) WHICH ASSERTS WD(SB) SL THROUGHOUT THE RECORD. ALL ONES DATA IS USED SO THAT THE FUNCTION ONLY INTERFERES WITH THE WRITING OF THE LRC CHARACTER WHEN NONE OF THE TM03 WRITE DATA LINES SHOULD BE ASSERTED.

LIKELY FAULT LOCATIONS: M8505, M8903, M8910, M8904

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
WRITE LINE DRIVERS	TCCM2
WRITE HEAD DRIVERS	LAW3, 4
LRC CHECKING	CNRZ3

LOGIC TEST 44: PE CORRECTABLE DATA

PROGRAMMED SEQUENCE:

A PE WRITE OPERATION IS PERFORMED USING A FUNCTION WHICH WILL GROUND THE BIT STROBE LINE ON BIT 1. THIS SHOULD CAUSE THE BIT1 DEAD TRACK FLOP TO ASSERT AND CAUSE CORECTABLE DATA ERROR. THE DEAD TRACK REGISTER IS CHECKED FOR BIT 1.

LIKELY FAULT LOCATIONS: M8905, M8901, M8902

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
BIT STROBE CIRCUIT	DS4
DEAD TRACK FLOP	DS5, TCPE2
DEAD TRACK REGISTER	MR4

LOGIC TEST 45: PE INCORRECTABLE DATA

REPEAT OF TEST 44, EXCEPT THAT THE MAINT. MODE FUNCTION GROUNDS BITS STROBE FOR BITS 1, 2 AND THE WD LINE FOR BIT 5 IN HELD ASSERTED. INC. DATA AND PCF ERRORS ARE EXPECTED.

LIKELY FAULT LOCATIONS: M8902, M8901

CIRCUIT	PRINT REFERENCE
INC ERROR, PEF,	TCPE2

LOGIC TEST 46: PE FORMAT

THE MM FUNCTION USED IN THIS TEST INVERTS THE DATA USED
IN PREAMBLE AND POSTAMBLE OF BIT ONE.

LIKELY FAULT LOCATIONS: M8902, M8903, M8905

CIRCUITS	PRINT REFERENCES
PEF.	TCPE2
WRITE BUFFER	TCCM2
MM DECODE	MRS

LOGIC TEST 47: FRAME COUNT OVERFLOW

THIS TEST USES A WRAP2 TO CHECK THE OVERFLOW OF FRAME
COUNT REGISTER.

LIKELY FAULT LOCATION: M8909

FRAME COUNT REGISTER MBI8

LOGIC TEST 50: NEF WHEN WRITING PE ON NRZ SELECTED SLAVE

THIS TEST ENSURES THAT WHEN A SLAVE IS IN NRZ MODE A WRITE
OPERATION WHEN OFF BOT IN PE MODE RESULTS IN A NON-EXECUTABLE
FUNCTION AND SETS THE NEF BIT IN THE ERROR REGISTER.

PROGRAM SEQUENCE:

THE SELECTED SLAVE IS REWOUND AND PLACED IN NRZ MODE AND SPACED
OFF BOT. A PE WRITE OPERATION IS INITIATED, AND THE NEF BIT IN
THE ERROR REGISTER IS CHECKED.

LOGIC TEST 51: NEF WHEN WRITING NRZ ON PE SELECTED SLAVE

THIS TEST IS THE COMPLEMENT OF LOGIC TEST 50 ABOVE.

```
%
1171                                    .LIST   BIN,LOC,SEQ
1172                                    .TITLE TM03/TE16 CONTROL LOGIC TEST PART I
1173                                    ;MAINDEC-11-DZTEA-A-D
```

```
                                ;FEB 77
                                ;R. BARNES
                                .MCALL  .$ACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE
                                .NLIST  MC
                                .LIST   ME
                                .ENABLE ABS,AMA


                                ;CONSOLE SWITCHES*********
                                ;
                                ;SW15:  1=HALT ON ERROR
                                ;       0=CONTINUE
                                ;SW14:  1=LOOP ON ERROR
                                ;       0=CONTINUE
                                ;SW13:  1=DO NOT PRINT ERRORS
                                ;       0=PRINT ERRORS
                                ;SW12:  1=HALT AT END OF PASS
                                ;       0=CONTINUOUS CYCLE
                                ;SW11:  1=INHIBIT ITERATIONS
                                ;       0=DO ITERATIONS
                                ;SW10:  1=HALT AT END OF EACH TEST
                                ;       0=CONTINUE
                                ;SW9:   1=DO MANUAL INTERVENTION TESTS
                                ;       0=INHIBIT MANUAL INTERVENTION
                                ;SW0-5: SELECT TEST NUMBER :: 00=ALL TESTS
```

```
1199                                    ;TE16 REGISTER BITS**********
1200
1201                                    ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
1202                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1203                                ;CS1:;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;FUN;GO ;
1204                                    ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
1205                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1206                                ;WC: ; WORD                                                         ;
1207                                    ; COUNT                                                         ;
1208                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1209                                ;BA: ; BUS                                                          ;
1210                                    ; ADDRESS                                                       ;
1211                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1212                                ;FC: ; FRAME                                                        ;
1213                                    ; COUNT                                                         ;
1214                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1215                                ;CS2:;DLT;WCE;UPE;NED;NEM;PGE;MDF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
1216                                    ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;
1217                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1218                                ;DS: ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;TMK;BOT;SLA;
1219                                    ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
1220                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1221                                ;ER: ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;MSG;PEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
1222                                    ;CRC;   ;   ;   ;ITM;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
1223                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1224                                ;ATS: ; ATTENTION                                                   ;
1225                                    ; SUMMARY                                                       ;
1226                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1227                                ;CC: ; CHECK                                                        ;
1228                                    ; CHARACTER                                                     ;
1229                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1230                                ;DB: ; DATA                                                         ;
1231                                    ; BUFFER                                                        ;
1232                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1233                                ;MR: ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
1234                                    ; 7 ; 6 ; 5 ; 4 ; 3 ; 2 ; 1 ; 0 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
1235                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1236                                ;DT: ; DRIVE     ; 7 ;   ;SLV;                                     ;
1237                                    ; TYPE      ;CH ;   ;PR ;                                     ;
1238                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1239                                ;SN: ; SERIAL                                                       ;
1240                                    ; NUMBER                                                        ;
1241                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
1242                                ;TC: ;ACL;FCS;SAC;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
1243                                    ;   ;   ;DT ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
1244                                    ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
                                        ;REGISTER EQUIVS**********

                    000000              R0=%0
                    000001              R1=%1
                    000002              R2=%2
                    000003              R3=%3
                    000004              R4=%4
                    000005              R5=%5
                    000006              SP=%6
                    000007              PC=%7


                                        ;ACT11 HOOK ***********
                    000764              $SVPC=.                 ;SAVE CURRENT LOCATION CTR
                    000046              .=46
000046  002502                          .WORD   $ENDAD          ;SET LOCATION 46
                    000052              .=52
000052  000000                          .WORD   0               ;SET LOCATION 52 = 0
                    000764              .=$SVPC                 ;RESTORE LOCATION CTR

                                        ;TTY INTERRUPT VECTOR**********

                    000060              .=60
000060  016650                          .WORD   TTINT   ;TTY INTERRUPT HEADER ADDRESS
000062  000340                          .WORD   340     ;PRIORITY LEVEL 7

                                        ;SOFTWARE SWITCH REGISTER**********
                                        ;USED IF HARDWARE SWR = 177777 OR NOT AVAILABLE
                    000176              .=176
000176  000000                  SWREG:  .WORD   0                       ;SOFTWARE SWITCH REGISTER

                                        ;START ADDRESS**********
                    000200              .=200
000200  000137  001330                  JMP     START   ;PROGRAM START

                                        ;RESTART ADDRESS**********
                    000210              .=210
000210  000137  002066                  JMP     ST2

                                        ;TM03 INTERRUPT VECTOR**********

                    000224              .=224
000224  016640                          MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
000226  000340                          340
```

```
1290
1291            000510                          .=510
1292                                            ;MASS BUS REGISTER EQUIVS**********
1293
1294    000510  172440                 C1:     172440
1295    000512  172442                 WC:     172442
1296    000514  172444                 BA:     172444
1297    000516  172446                 FC:     172446
1298    000520  172450                 CS:     172450
1299    000522  172452                 DS:     172452
1300    000524  172454                 ER:     172454
1301    000526  172456                 AS:     172456
1302    000530  172460                 CC:     172460
1303    000532  172462                 DB:     172462
1304    000534  172464                 MR:     172464
1305    000536  172466                 DT:     172466
1306    000540  172470                 SN:     172470
1307    000542  172472                 TC:     172472
1308
1309                                            ;ILLEGAL FUNCTION CODES
1310
1311    000544  005405                 ILFT:   5405
1312    000546  007415                         7415
1313    000550  016423                         16423
1314    000552  020437                         20437
1315    000554  022443                         22443
1316    000556  025447                         25447
1317    000560  031455                         31455
1318    000562  033465                         33465
1319    000564  036473                         36473
1320
1321                                            ;CONSTANTS**********
1322
1323    000566  177776                 PSW:    177776          ;PROCESSOR STATUS
1324    000570  177570                 SWR:    177570          ;SWITCH REGISTER
1325    000572  177560                 TKS:    177560          ;TTY READER STATUS
1326    000574  177562                 TKB:    177562          ;TTY READ BUFFER
1327    000576  177564                 TPS:    177564          ;TTY PUNCH STATUS
1328    000600  177566                 TPB:    177566          ;TTY PUNCH BUFFER
1329    000602  177777                 SERNUM: 177777          ;SERIAL NUMBER
1330    000604  000011                 DRVTP:  011             ;DRIVE TYPE
1331    000606  000020                 ITAMT:  20              ;ITERATION AMOUNT
1332    000610  000224                 VECT:   224             ;INTERRUPT VECTOR(RH)
1333    000612  172440                 REGS:   172440          ;STARTING REGISTER ADDRESS
```

```
1334                                            ;FLAGS AND COUNTERS**********
1335
1336  000614  000000                            TOB:    0
1337  000616  000000                            TIB:    0
1338  000620  000000                            HDRFL:  0
1339  000622  000000                            EMADDR: 0
1340  000624  000000                            DRVN:   0
1341  000626  000000                            TR00:   0
1342  000630  000000                            TR01:   0
1343  000632  000000                            TR02:   0
1344  000634  000000                            TR03:   0
1345  000636  000000                            TR04:   0
1346  000640  000000                            TR05:   0
1347  000642  000000                            TR06:   0
1348  000644  000000                            TR07:   0
1349  000646  000000                            TR10:   0
1350  000650  000000                            TR11:   0
1351  000652  000000                            TR12:   0
1352  000654  000000                            TR13:   0
1353  000656  000000                            TR14:   0
1354  000660  000000                            TR15:   0
1355  000662  000000                            NRZOF:  0
1356  000664  000000                            SLVN:   0
1357  000666  000000                            PFLG:   0
1358  000670  000000                            RTRN:   0
1359  000672  000000                            ERADD:  0
1360  000674  000000                            TEMP1:  0
1361  000676  000000                            TEMP2:  0
1362  000700  000000                            TEMP3:  0
1363  000702  000000                            ITCNT:  0
1364  000704  000000                            SAV1:   0
1365  000706  000000                            SAV2:   0
1366  000710  000000                            SAV3:   0
1367  000712  000000                            SCOLP:  0
1368  000714  000000                            ITRLP:  0
1369  000716  000000                            EXFL:   0
1370  000720  000000                            ATAF:   0
1371  000722  000000                            SLAF:   0
1372  000724  000000                            SSCF:   0
1373  000726  000000                            ERRF:   0
1374  000730  000000                            ASF:    0
1375  000732  000000                            SCF:    0
1376  000734  000000                            TREF:   0
1377  000736  000000                            PEXFL:  0
1378  000740  000000                            STFLG:  0
1379  000742  000000                            LTADD:  0
1380  000744  000000                            T24FL:  0
1381  000746  000000                            ADDFL:  0
1382  000750  000000                            WAM:    0
1383  000752  000000                            FUN:    0
1384  000754  000000                            DATC:   0
1385  000756  000000                            WTAD:   0
1386  000760  000000                            DATAD:  0
1387  000762  000000                            RDAD:   0
1388  000764  000000                            W2FLG:  0
1389  000766  000000                            DERFL:  0
```

```
1390  000770  000000                PREFL:  0
1391  000772  000000                SERFL:  0
1392  000774  000000                CRCNT:  0
1393  000776  000000                UDES:   0
1394  001000  000000                WPGFL:  0
1395  001002  000000                PATRN:  0
1396  001004  000000                STATF:  0
1397  001006  000000                RDRVF:  0
1398  001010  000000                RCDP:   0
1399  001012  000000                STATC:  0
1400  001014  000000                SKAT:   0
1401  001016  000000                PCNTR:  0               ;PASS COUNTER
1402
1403                                  ;EXPT WRAP STATUS*******
1404
1405  001020  000000                WCS1:   0
1406  001022  000000                WCS2:   0
1407  001024  000000                WDS:    0
1408  001026  000000                WER:    0
1409
1410                                  ;CORE DUMP PATTERNS*******
1411
1412  001030  000005                WCDP2:  5
1413  001032  000005                        5
1414  001034  000012                        12
1415  001036  000012                        12
1416  001040  000000                        0
1417  001042  000017                WCDP0:  17
1418  001044  000017                        17
1419  001046  000017                        17
1420  001050  000017                        17
1421  001052  000000                        0
```

```
1422
1423                                    ;LOGIC TEST ENTRY TABLE**********
1424
1425  001054  000000                    TSTTBL: 0
1426  001056  000000                            0
1427  001060  002552                            LT1
1428  001062  002552                            LT1
1429  001064  003026                            LT2
1430  001066  003026                            LT2
1431  001070  003232                            LT3
1432  001072  003234                            LT3IT
1433  001074  003414                            LT4
1434  001076  003414                            LT4
1435  001100  003714                            LT5
1436  001102  003722                            LT5IT
1437  001104  004104                            LT6
1438  001106  004106                            LT6IT
1439  001110  004220                            LT7
1440  001112  004222                            LT7IT
1441  001114  004334                            LT10
1442  001116  004336                            LT10IT
1443  001120  004460                            LT11
1444  001122  004462                            LT11IT
1445  001124  004704                            LT12
1446  001126  004706                            LT12IT
1447  001130  005100                            LT13
1448  001132  005110                            LT13IT
1449  001134  005202                            LT14
1450  001136  005242                            LT14IT
1451  001140  005312                            LT15
1452  001142  005352                            LT15IT
1453  001144  005422                            LT16
1454  001146  005462                            LT16IT
1455  001150  005532                            LT17
1456  001152  005572                            LT17IT
1457  001154  005642                            LT20
1458  001156  005656                            LT20IT
1459  001160  006004                            LT21
1460  001162  006020                            LT21IT
1461  001164  006150                            LT22
1462  001166  006164                            LT22IT
1463  001170  006274                            LT23
1464  001172  006310                            LT23IT
1465  001174  006424                            LT24
1466  001176  006440                            LT24IT
1467  001200  006750                            LT25
1468  001202  006756                            LT25IT
1469  001204  007102                            LT26
1470  001206  007110                            LT26IT
1471  001210  007316                            LT27
1472  001212  007342                            LT27IT
1473  001214  007434                            LT30
1474  001216  007456                            LT30IT
1475  001220  007752                            LT31
1476  001222  007760                            LT31IT
1477  001224  010564                            LT32
```

```
1478  001226  010600                        LT32IT
1479  001230  010736                        LT33
1480  001232  010742                        LT33IT
1481  001234  011030                        LT34
1482  001236  011044                        LT34IT
1483  001240  011164                        LT35
1484  001242  011200                        LT35IT
1485  001244  011336                        LT36
1486  001246  011352                        LT36IT
1487  001250  011456                        LT37
1488  001252  011472                        LT37IT
1489  001254  011626                        LT40
1490  001256  011642                        LT40IT
1491  001260  011746                        LT41
1492  001262  011762                        LT41IT
1493  001264  012210                        LT42
1494  001266  012246                        LT42IT
1495  001270  012540                        LT43
1496  001272  012566                        LT43IT
1497  001274  012774                        LT44
1498  001276  013022                        LT44IT
1499  001300  013242                        LT45
1500  001302  013270                        LT45IT
1501  001304  013506                        LT46
1502  001306  013534                        LT46IT
1503  001310  013744                        LT47
1504  001312  013760                        LT47IT
1505  001314  014134                        LT50
1506  001316  014150                        LT50IT
1507  001320  014310                        LT51
1508  001322  014324                        LT51IT
1509  001324  002436          TADX:   .WORD   TEND
1510  001326  000051          TLAST:  .WORD   51          ;CONTAINS # OF TESTS
```

```
1511                                                    .EVEN
1512                                                    ;PROGRAM START AND HOUSEKEEPING**********
1513
1514                                                    ;NOTE: PROGRAM STARTS HERE ON START AT 200
1515  001330  012706  000500          START:  MOV     #500,SP         ;SET STACK POINTER
1516  001334  013746  000004                  MOV     @#4,-(SP)       ;SAVE ERROR TRAP VECTOR
1517  001340  013746  000006                  MOV     @#6,-(SP)       ;AND VECTOR +2
1518  001344  012737  001370  000004          MOV     #1$,@#4         ;SET NEW VECTOR
1519  001352  005037  000006                  CLR     @#6             ;AND PSW
1520  001356  012777  177777  177204          CMP     #-1,@SWR        ;USE SOFTWARE SWITCH IF HARDWARE
1521  001364  001402                          BEQ     2$              ;IS = 177777
1522  001366  000404                          BR      3$              ;OTHERWISE USE HARDWARE SWR
1523  001370  022626                  1$:     CMP     (SP)+,(SP)+     ;RESET STACK PTR
1524  001372  012737  000176  000570  2$:     MOV     #SWREG,SWR      ;SET SOFTWARE SWITCH REGISTER
1525  001400  012637  000006          3$:     MOV     (SP)+,@#6       ;RESTORE ERROR TRAP VECTORS
1526  001404  012637  000004                  MOV     (SP)+,@#4
1527  001410  005037  001014                  CLR     SKAT            ;CLEAR SKIP ADDRESS TEST FLAG
1528  001414  005027                          CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
1529  001416  000000          CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
1530                                                          ;;1/0 = CHAIN/NOT CHAIN MODE
1531  001420  022737  002502  000042          CMP     #SENDAD,@#42    ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
1532  001426  001404                          BEQ     50$
1533  001430  005737  000042                  TST     @#42            ;;BRANCH IF IN DUMP MODE
1534  001434  001413                          BEQ     52$
1535  001436  000406                          BR      51$
1536  001440  012737  000176  000570  50$:    MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
1537  001446  012777  100000  177114          MOV     #100000,@SWR    ;;WITH HALT ON ERROR SET
1538  001454  005237  001416          51$:    INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
1539  001460  000137  002106                  JMP     TSCD            ;;GO TO CHAIN ADDRESS
1540  001464                          52$:
1541  001464  122737  000006  000041  4$:     CMPB    #6,@#41         ;BRANCH IF NOT LOADED VIA TMDP
1542  001472  001004                          BNE     5$
1543  001474  012704  022622                  MOV     #MSG62,R4       ;ADVISE USER TO REMOVE TMDP FROM
1544  001500  004737  017320                  JSR     PC,TTOUT        ;UNIT UNDER TEST
1545  001504  012704  020236          5$:     MOV     #MSG1,R4
1546  001510  004737  017320                  JSR     PC,TTOUT        ;PRINT TITLE
1547  001514  112737  000043  020236          MOVB    #'#,MSG1        ;DO NOT PRINT TITLE ON RESTART
1548  001522  012704  022172                  MOV     #MSG44,R4
1549  001526  004737  017320                  JSR     PC,TTOUT        ;REQUEST REGISTER ADDRESS
1550  001532  013703  000612                  MOV     REGS,R3
1551  001536  004737  017446                  JSR     PC,OCTP         ;PRINT CURRENT ADDRESS
1552  001542  012705  000612                  MOV     #REGS,R5        ;SET ADDRESS SAVE LOC
1553  001546  012701  000007                  MOV     #7,R1           ;SET SIZE OF RESPONSE
1554  001552  012702  176400                  MOV     #176400,R2      ;SET UPPER LIMIT
1555  001556  012703  172300                  MOV     #172300,R3      ;SET LOWER LIMIT
1556  001562  004737  016776                  JSR     PC,TTR          ;GO GET RESPONSE
1557  001566  012704  022214                  MOV     #MSG45,R4
1558  001572  004737  017320                  JSR     PC,TTOUT        ;REQUEST VECTOR
1559  001576  013703  000610                  MOV     VECT,R3
1560  001602  004737  017446                  JSR     PC,OCTP         ;PRINT CURRENT VECTOR
1561  001606  012705  000610                  MOV     #VECT,R5        ;SET ADDRESS SAVE LOC
1562  001612  012701  000004                  MOV     #4,R1           ;SET SIZE OF RESPONSE
1563  001616  012702  000224                  MOV     #224,R2         ;SET UPPER LIMIT
1564  001622  012703  000150                  MOV     #150,R3         ;SET LOWER LIMIT
1565  001626  004737  016776                  JSR     PC,TTR          ;GO GET RESPONSE
1566  001632  013700  000610                  MOV     VECT,R0         ;GET VECTOR
```

```
1567  001636  012720  016640          MOV     #MTINT,(R0)+    ;LOAD INTERRUPT ADDRESS IN VECTOR
1568  001642  012710  000340          MOV     #340,(R0)       ;LOAD PRIORITY
1569  001646  013700  000612          MOV     REGS,R0         ;GET START OF REGS
1570  001652  012701  000016          MOV     #16,R1          ;SET NUMBER OF REGS
1571  001656  012702  000510          MOV     #C1,R2          ;GET START OF TABLE
1572  001662  010022          STO:    MOV     R0,(R2)+        ;BUILD TABLE
1573  001664  062700  000002          ADD     #2,R0           ;BUMP ADDRESS
1574  001670  005301                  DEC     R1              ;SEE IF DONE
1575  001672  001373                  BNE     STO             ;IF NOT: BR
1576  001674  012702  000614          MOV     #T08,R2
1577  001700  012700  000077          MOV     #77,R0
1578  001704  005022          ST1:    CLR     (R2)+           ;CLEAR FLAGS + COUNTERS
1579  001706  005300                  DEC     R0
1580  001710  001375                  BNE     ST1
1581  001712  012704  022562          MOV     #MSG57,R4       ;REQUEST TM03 DRIVE #
1582  001716  004737  017320          JSR     PC,TTOUT
1583  001722  013703  000624          MOV     DRVN,R3         ;GET CURRENT DRIVE #
1584  001726  004737  017446          JSR     PC,OCTP         ;PRINT IT
1585  001732  012705  000624          MOV     #DRVN,R5        ;TTR ROUTINE RETURNS USER VALUE TO (R5)
1586  001736  012701  000002          MOV     #2,R1           ;LIMIT RESPONSE
1587  001742  012702  000007          MOV     #7,R2           ;LIMIT RANGE TO 0-7
1588  001746  012703  000000          MOV     #0,R3
1589  001752  004737  016776          JSR     PC,TTR          ;GET USER RESPONSE
1590  001756  012704  022600          MOV     #MSG58,R4       ;REQUEST TE16 SLAVE #
1591  001762  004737  017320          JSR     PC,TTOUT
1592  001766  013703  000664          MOV     SLVN,R3         ;GET CURRENT SLAVE #
1593  001772  004737  017446          JSR     PC,OCTP         ;AND PRINT IT
1594  001776  012705  000664          MOV     #SLVN,R5        ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1595  002002  012701  000002          MOV     #2,R1           ;LIMIT RESONSE TO 1 CHARACTER
1596  002006  012702  000007          MOV     #7,R2           ;BETWEEN 0 AND 7
1597  002012  012703  000000          MOV     #0,R3
1598  002016  004737  016776          JSR     PC,TTR          ;GET USER RESPONSE
1599  002022  012704  022535          MOV     #MSG56,R4
1600  002026  004737  017320          JSR     PC,TTOUT        ;REQUEST STATIC ONLY
1601  002032  013703  001012          MOV     STATC,R3        ;GET CURRENT VALUE
1602  002036  004737  017446          JSR     PC,OCTP         ;AND TYPE IT
1603  002042  012705  001012          MOV     #STATC,R5       ;SET ADDRESS OF STATIC FLAG
1604  002046  012701  000002          MOV     #2,R1           ;SET SIZE OF RESPONSE
1605  002052  012702  000001          MOV     #1,R2           ;SET UPPER LIMIT
1606  002056  012703  000000          MOV     #0,R3           ;SET LOWER LIMIT
1607  002062  004737  016776          JSR     PC,TTR          ;GET RESPONSE
1608
1609                          ;START 210
1610  002066  012706  000500  ST2:    MOV     #500,SP         ;SET STACK PTR
1611  002072  005037  001006          CLR     RDRVF           ;CLEAR REVERSE FLAG
1612  002076  005037  001016          CLR     PCNTR           ;CLEAR PASS COUNTER
1613  002102  004737  020074          JSR     PC,GTSWR        ;GET SWITCHES
```

```
1614
1615                                                    ;TEST SCHEDULAR**********
1616
1617  002106  052777  000100  176456  TSCD:   BIS     #100,@TKS       ;SET KEYBOARD IE BIT
1618  002114  005037  001000          CLR     WRPGFL                  ;CLEAR WRAP PATRN FLAG
1619  002120  005037  000740          CLR     STFLG                   ;CLEAR SINGLE TEST FLAG
1620  002124  017700  176440          MOV     @SWR,R0
1621  002130  042700  177700          BIC     #177700,R0              ;BRANCH IF SINGLE
1622  002134  001122                  BNE     STSCD                   ;TEST SELECTED
1623  002136  005737  001416          TST     CHNFLG                  ;;BRANCH IF NOT IN CHAIN MODE
1624  002142  001457                  BEQ     TSCDA
1625  002144  012737  177777  000624  MOV     #-1,DRVN                ;;INITIALIZE DRIVE #
1626  002152  012737  177777  000664  NXTDRV: MOV     #-1,SLVN        ;;INITIALIZE SLAVE #
1627  002160  012777  000040  176332  1$:     MOV     #40,@CS         ;;INIT CONTROLLER
1628  002166  005237  000624          INC     DRVN                    ;;STEP DRIVE #
1629  002172  022737  000010  000624  CMP     #10,DRVN                ;;EXIT IF ALL DRIVES TESTED
1630  002200  001521                  BEQ     SLONE                   ;;FOR AVAILABILITY
1631  002202  013777  000624  176310  MOV     DRVN,@CS                ;;LOAD DRIVE #
1632  002210  005777  176274          TST     @CS                     ;;ACCESS DRIVE
1633  002214  032777  010000  176276  BIT     #10000,@CS              ;;BRANCH IF DRIVE NON EXISTANT
1634  002222  001356                  BNE     1$                      ;;(NED = 1)
1635  002224  005237  000664          INC     SLVN                    ;;STEP SLAVE # AND BRANCH
1636  002230  001011                  BNE     1$                      ;;IF NOT SLAVE 0
1637  002232  005737  000624          TST     DRVN                    ;;BRANCH IF NOT DRIVE # 0
1638  002236  001006                  BNE     1$
1639  002240  122737  000006  000041  CMPB    #6,@#41                 ;;BRANCH IF NOT TMDP
1640  002246  001002                  BNE     1$
1641  002250  005237  000664          INC     SLVN                    ;;STEP TO SLAVE # 1
1642  002254  022737  000010  000664  1$:     CMP     #10,SLVN        ;;BRANCH IF ALL SLAVES TESTED
1643  002262  001733                  BEQ     NXTDRV                  ;;FOR AVAILABILITY
1644  002264  013777  000664  176250  MOV     SLVN,@TC                ;;LOAD SLAVE UNIT #
1645  002272  032777  002000  176236  BIT     #2000,@DT               ;;BRANCH IF SLAVE NOT
1646  002300  001751                  BEQ     NXTSLV                  ;;PRESENT (SPR = 0)
1647  002302  012737  001054  000742  TSCDA:  MOV     #TSTTBL,LTADD
1648  002310  062737  000004  000742  TSCDO:  ADD     #4,LTADD
1649  002316  013737  000742  000714  TSCD1:  MOV     LTADD,ITRLP
1650  002324  062737  000002  000714          ADD     #2,ITRLP        ;SET ITERATION ADDRESS
1651  002332  005037  000620          CLR     HDRFL                   ;CLEAR PRINT HEADER FLAG
1652  002336  017700  176400          MOV     @LTADD,R0               ;SET POINTER TO TEST
1653  002342  000110                  JMP     (R0)                    ;GO TO TEST
1654  002344  032777  002000  176216  TSCD2:  BIT     #2000,@SWR      ;SEE IF HALT ON TEST
1655  002352  001403                  BEQ     TSCD3                   ;IF NOT: BR
1656  002354  000000                  HALT
1657  002356  005037  001000          CLR     WRPGFL                  ;CLEAR WRAP DATA GENERATOR FLAG
1658  002362  005737  000740          TST     STFLG                   ;SE IF SINGLE TEST
1659  002366  001750                  BEQ     TSCDO                   ;IF NOT: BR
1660  002370  017700  176174          MOV     @SWR,R0
1661  002374  042700  177700          BIC     #177700,R0              ;BRANCH IF ALL TESTS DESIRED
1662  002400  001642                  BEQ     TSCD                    ;IF SO: BR
1663  002402  012737  000001  000740  STSCD:  MOV     #1,STFLG        ;SET SINGLE TEST FLAG
1664  002410  023700  001326          CMP     TLAST,R0                ;SEE IF EXCEEDED TESTS
1665  002414  002410                  BLT     TEND                    ;IF SO: BR
1666  002416  006300                  ASL     R0
1667  002420  006100                  ROL     R0                      ;SET TABLE MODIFIER
1668  002422  012737  001054  000742          MOV     #TSTTBL,LTADD
1669  002430  060037  000742          ADD     R0,LTADD                ;SET TEST POINTER
```

```
1670  002434  000730                        BR      TSC01
1671  002436  005737  001416          TEN0: TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
1672  002442  001770                        BNE     NXTSLV          ;STEP TO NEXT SLAVE
1673  002444  012704  022032          S00NE: MOV    #MSG41,R4
1674  002450  004737  017320                JSR     PC,TTOUT        ;PRINT END OF PASS
1675  002454  013703  001016                MOV     PCNTR,R3
1676  002460  004737  017446                JSR     PC,OCTP         ;PRINT PASS NUMBER
1677  002464  005000                        CLR     R0              ;DELAY WAITING FOR
1678  002466  005300                  1$:   DEC     R0
1679  002470  001376                        BNE     1$
1680  002472  013700  000042                MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
1681  002476  001405                        BEQ     HERE            ;BRANCH IF NOT ACT11
1682  002500  000005                        RESET
1683  002502  004710                  SENDAD: JSR     PC,(R0)
1684  002504  000240                        NOP
1685  002506  000240                        NOP
1686  002510  000240                        NOP
1687  002512  000240                  HERE: NOP
1688  002514  005737  001416                TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
1689  002520  001005                        BNE     TENDX
1690  002522  032777  010000  176040        BIT     #10000,@SHR     ;SEE IF HALT ON PASS
1691  002530  001001                        BNE     TENDX           ;IF NOT: BR
1692  002532  000000                        HALT
1693  002534  012737  000001  001014  TENDX: MOV    #1,SKAT         ;SET SKIP ADDRESS TEST FLAG
1694  002542  005237  001016                INC     PCNTR           ;BUMP PASS COUNTER
1695  002546  000137  002106                JMP     TSC0            ;RESTART
```

```
1696                                                    ;LOGIC TEST 1: DRIVE ADDRESSING**********
1697
1698  002552  013737  000624  000700  LT1:    MOV     DRVN,TEMP3      ;GET DRIVE # TO BE TESTED
1699  002560  013701  000624                  MOV     DRVN,R1
1700  002564  005737  001014                  TST     SKAT            ;SEE IF SKIP ADDRESS TESTS
1701  002570  001403                          BEQ     1$              ;IF NOT: BR
1702  002572  005737  000740                  TST     STFLG           ;SEE IF SINGLE TEST
1703  002576  001511                          BEQ     LT1X            ;IF NOT: BR
1704  002600  032777  001000  175762  1$:     BIT     #1000,@SWR      ;BRANCH IF MAN INTERVENTION
1705  002606  001433                          BEQ     LT1A            ;NOT SELECTED
1706  002610  012704  020475          LT1GO:  MOV     #MSG2A,R4
1707  002614  004737  016572                  JSR     PC,INST         ;PRINT TEST INSTRUCTIONS
1708  002620  012737  022716  000622  LT1G:   MOV     #MSLT1,EMADDR   ;SET HEADER ADDRESS
1709  002626  012704  020434                  MOV     #MSG2,R4
1710  002632  004737  017320                  JSR     PC,TTOUT        ;REQUEST DRIVE NUMBER
1711  002636  012705  000700                  MOV     #TEMP3,R5       ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1712  002642  012701  000002                  MOV     #2,R1
1713  002646  012702  000007                  MOV     #7,R2
1714  002652  012703  000000                  MOV     #0,R3
1715  002656  004737  016776                  JSR     PC,TTR          ;GET DRIVE NUMBER
1716  002662  005737  000674                  TST     TEMP1           ;SEE IF ANOTHER DRIVE
1717  002666  001455                          BEQ     LT1X            ;IF NOT: BR
1718  002670  005001                          CLR     R1              ;SELECT DRIVE 0
1719  002672  012700  000010                  MOV     #10,R0          ;SET NUMBER OF DRIVES
1720  002676  012777  000040  175614  LT1A:   MOV     #40,@CS         ;INIT
1721  002704  010177  175610                  MOV     R1,@CS          ;SELECT DRIVE
1722  002710  005777  175574                  TST     @CS             ;ACCESS DRIVE
1723  002714  032777  010000  175576          BIT     #10000,@CS      ;SEE IF NED
1724  002722  001010                          BNE     LT1B            ;IF SO: BR
1725  002724  032777  001000  175636          BIT     #1000,@SWR      ;BRANCH IF NOT MANUAL INTERVENTION
1726  002732  001433                          BEQ     LT1X
1727  002734  023701  000700                  CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1728  002740  001404                          BEQ     LT1C            ;IF NOT: BR
1729  002742  000407                          BR      LT1ER           ;ELSE GO TO ERROR
1730  002744  023701  000700          LT1B:   CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1731  002750  001410                          BEQ     LT1ER1
1732  002752  005300                  LT1C:   DEC     R0
1733  002754  001721                          BEQ     LT1G            ;IF DONE ALL: BR
1734  002756  005201                          INC     R1              ;SELECT NEXT DRIVE
1735  002760  000746                          BR      LT1A            ;CONTINUE
1736  002762  012737  000001  000716  LT1ER:  MOV     #1,EXFL         ;FLAG EXPT
1737  002770  000403                          BR      LT1ER2
1738  002772  012737  000002  000716  LT1ER1: MOV     #2,EXFL         ;FLAG NOT EXPT
1739  003000  012737  020624  000672  LT1ER2: MOV     #MSG3,ERRDD     ;FLAG CONDITION
1740  003006  012737  002676  000712          MOV     #LT1A,SCOLP     ;SET SCOPE ADDRESS
1741  003014  004737  014670                  JSR     PC,LTGER        ;GO PRINT LOGIC TEST ERROR
1742  003020  000754                          BR      LT1C            ;CONTINUE TEST
1743  003022  000137  002344          LT1X:   JMP     TSCD2           ;RETURN TO SCHED
1744
```

```
                                                  ;LOGIC TEST 2: REGISTER ADDRESSING#########

003026  000240                          LT2:      NOP
003030  012777  000040  175462          LT2IT:    MOV     #40,@CS            ;INIT
003036  013777  000624  175454                    MOV     DRVN,@CS           ;SELECT DRIVE
003044  012737  022772  000622                    MOV     #MSLT2,EMADDR      ;SAVE LT2 HEADER ADDRESS
003052  012705  000510                            MOV     #C1,R5             ;SET ADDRESS OF FIRST REGISTER
003056  012700  000016                            MOV     #16,R0             ;SET NUMBER OF REGISTERS
003062  012702  000626                            MOV     #TR00,R2           ;SET START OF REGISTER BUFFER
003066  011501                          LT2A:     MOV     (R5),R1
003070  011112                                    MOV     (R1),(R2)          ;READ REGISTER
003072  032777  020000  175410                    BIT     #20000,@C1         ;SEE IF ERROR
003100  001402                                    BEQ     LT2B               ;IF NOT: BR
003102  004737  003132                            JSR     PC,LT2ER1          ;ELSE GO TO ERROR 1
003106  032777  000002  175410          LT2B:     BIT     #2,@ER             ;SEE IF ILR
003114  001402                                    BEQ     LT2C               ;IF NOT: BR
003116  004737  003150                            JSR     PC,LT2ER2          ;ELSE GO TO ERROR 2
003122  022225                          LT2C:     CMP     (R2)+,(R5)+        ;BUMP ADDRESS
003124  005300                                    DEC     R0
003126  001357                                    BNE     LT2A               ;CONTINUE FOR ALL REGISTERS
003130  000434                                    BR      LT2X
003132  012737  000002  000716          LT2ER1:   MOV     #2,EXFL            ;FLAG NOT EXPECTED
003140  012737  020646  000672                    MOV     #MSG4,ERADD        ;POINT TO CONTROLLER ERROR
003146  000415                                    BR      LT2ERG             ;GO TO ERROR
003150  012737  000002  000716          LT2ER2:   MOV     #2,EXFL            ;FLAG NOT EXPECTED
003156  012737  020664  000672                    MOV     #MSG5,ERADD        ;POINT TO DRIVE ERROR
003164  000406                                    BR      LT2ERG             ;GO TO ERROR
003166  012737  000001  000716          LT2ER3:   MOV     #1,EXFL            ;FLAG EXPECTED
003174  012737  020646  000672                    MOV     #MSG4,ERADD        ;POINT TO DRIVE
003202  012737  003216  000712          LT2ERG:   MOV     #LT2LP,SCOLP       ;SET SCOPE ADDRESS
003210  004737  014670                            JSR     PC,LTGER           ;GO PRINT
003214  000207                                    RTS     PC                 ;ELSE CONTINUE
003216  005726                          LT2LP:    TST     (SP)+              ;RESET STACK
003220  000722                                    BR      LT2A               ;LOOP
003222  004737  016240                  LT2X:     JSR     PC,ITER            ;GO SEE IF ITERATIONS
003226  000137  002344                            JMP     TSCD2              ;RETURN TO SCHED
```

```
1781                                                    ;LOGIC TEST 3: CONTROL BUS**********
1782
1783  003232  000240                          LT3:      NOP
1784  003234  012737  023051  000622         LT3IT:    MOV     #MSLT3,EMADDR   ;SET TEST HEADER
1785  003242  012701  000001                           MOV     #1,R1           ;PRESET PATTERN 1
1786  003246  012700  000020                           MOV     #20,R0          ;SET PATTERN CHANGE NUMBER
1787  003252  004737  016330                LT3A:     JSR     PC,INIT1        ;GO INIT
1788  003256  010177  175234                           MOV     R1,@FC          ;WRITE TO FC
1789  003262  032777  000010  175234                    BIT     #10,@ER         ;SEE IF CPAR (TM03)
1790  003270  001012                                    BNE     LT3ER1          ;IF SO: BR
1791  003272  017702  175220                LT3B:     MOV     @FC,R2          ;READ FC
1792  003276  032777  020000  175204                    BIT     #20000,@C1      ;SEE IF MCPE (RH)
1793  003304  001017                                    BNE     LT3ER2          ;IF SO: BR
1794  003306  005300                         LT3C:     DEC     R0              ;SEE IF DONE PATTERN CHANGES
1795  003310  001426                                    BEQ     LT3X            ;IF SO: BR
1796  003312  006301                                    ASL     R1              ;CHANGE PATTERN
1797  003314  000756                                    BR      LT3A            ;CONTINUE
1798  003316  012737  021177  000672         LT3ER1:   MOV     #MSG11,ERADD    ;SET ERROR CODE
1799  003324  012737  003252  000712                    MOV     #LT3A,$COLP     ;SET SCOPE ADDRESS
1800  003332  017702  175160                           MOV     @FC,R2          ;GET DATA
1801  003336  004737  015776                           JSR     PC,LTGER1       ;GO DO ERROR
1802  003342  000753                                    BR      LT3B
1803  003344  012737  021153  000672         LT3ER2:   MOV     #MSG10,ERADD    ;SET ERROR CODE
1804  003352  012737  003272  000712                    MOV     #LT3B,$COLP     ;SET SCOPE ADDRESS
1805  003360  004737  015776                           JSR     PC,LTGER1       ;GO DO ERROR
1806  003364  000750                                    BR      LT3C
1807  003366  105701                         LT3X:     TSTB    R1              ;SEE IF DONE PATTERN 2
1808  003370  100405                                    BMI     LT3XX           ;IF SO: BR
1809  003372  012701  000401                           MOV     #401,R1         ;SET PATTERN 2
1810  003376  012700  000010                           MOV     #10,R0          ;SET PATTERN CHANGE NUMBER
1811  003402  000723                                    BR      LT3A            ;DO PATTERN 2
1812  003404  004737  016240                LT3XX:    JSR     PC,ITER         ;GO SEE IF ITERATIONS
1813  003410  000137  002344                           JMP     TSCDL           ;RETURN TO SCHEDULAR
```

```
1814
1815                                            ;LOGIC TEST 4: SLAVE ADDRESSING**********
1816
1817  003414  013737  000664  000700  LT4:      MOV     SLVN,TEMP3
1818  003422  013701  000664                    MOV     SLVN,R1
1819  003426  005737  001014                    TST     SKAT            ;SEE IF SKIP ADDRESS TESTS
1820  003432  001403                            BEQ     1$              ;IF NOT: BR
1821  003434  005737  000740                    TST     STFLG           ;SEE IF SINGLE TEST
1822  003440  001523                            BEQ     LT4X            ;IF NOT: BR
1823  003442  032777  001000  175120  1$:       BIT     #1000,@SWR      ;BRANCH IF MAN INTERVETION
1824  003450  001430                            BEQ     LT4A            ;NOT SELECTED
1825  003452  012704  021002          LT4GO:    MOV     #MSG8A,R4
1826  003456  004737  016572                    JSR     PC,INST         ;PRINT TEST INSTRUCTIONS
1827  003462  012704  020741          LT4G:     MOV     #MSG8,R4
1828  003466  004737  017320                    JSR     PC,TTOUT        ;REQUEST SLAVE
1829  003472  012705  000700                    MOV     #TEMP3,R5
1830  003476  012701  000002                    MOV     #2,R1
1831  003502  012702  000007                    MOV     #7,R2
1832  003506  012703  000000                    MOV     #0,R3
1833  003 12  004737  016776                    JSR     PC,TTR          ;GET SLAVE NUMBER
1834  003516  005737  000674                    TST     TEMP1           ;SEE IF SLAVE
1835  003522  001472                            BEQ     LT4X            ;IF NOT: BR
1836  003524  005001                            CLR     R1              ;SELECT SLAVE 0
1837  003526  012700  000010                    MOV     #10,R0          ;SET NUMBER OF SLAVES
1838  003532  012777  000040  174760  LT4A:     MOV     #40,@CS         ;INIT
1839  003540  013777  000624  174752            MOV     DRVN,@CS        ;SELECT DRIVE
1840  003546  010177  174770                    MOV     R1,@TC          ;SELECT SLAVE
1841  003552  017703  174760                    MOV     @DT,R3          ;GET DT
1842  003556  020137  000700                    CMP     R1,TEMP3        ;SEE IF SHOULD HAVE SPR
1843  003562  001404                            BEQ     LT4B            ;IF SO: BR
1844  003564  032703  002000                    BIT     #2000,R3        ;SEE IF SPR
1845  003570  001420                            BEQ     LT4D            ;IF NOT: BR
1846  003572  000423                            BR      LT4ER1          ;GO TO ERROR 1
1847  003574  032703  002000          LT4B:     BIT     #2000,R3        ;SEE IF NO SLAVE PRESENT
1848  003600  001424                            BEQ     LT4ER2          ;(SPR=0)
1849  003602  012704  021654          LT4C:     MOV     #MSG30,R4
1850  003606  004737  017320                    JSR     PC,TTOUT        ;PRINT SERIAL NUMBER TAG
1851  003612  017703  174722                    MOV     @SN,R3
1852  003616  004737  017772                    JSR     PC,SNPT         ;PRINT SERIAL NUMBER
1853  003622  032777  001000  174740            BIT     #1000,@SWR      ;BRANCH IF NOT MANUAL INTERVENTION
1854  003630  001427                            BEQ     LT4X
1855  003632  005300                    LT4D:   DEC     R0
1856  003634  001712                            BEQ     LT4G            ;IF DONE ALL: BR
1857  003636  005201                            INC     R1              ;BUMP SLAVE
1858  0036 0  000734                            BR      LT4A            ;CONTINUE
1859  003642  012737  000001  000716  LT4ER1:   MOV     #1,EXFL         ;FLAG EXPT: NOT RECEIVED
1860  003650  000403                            BR      LT4ERG
1861  003652  012737  000002  000716  LT4ER2:   MOV     #2,EXFL         ;FLAG RECVD: NOT EXPT
1862  003660  012737  023133  000622  LT4ERG:   MOV     #MSLT4,EMADDR   ;SET  LT4 HEADER
1863  003666  012737  021131  000672            MOV     #MSG9,ERRDD     ;SET ERROR CONDITION
1864  003674  012737  003532  000712            MOV     #LT4A,SCOLP     ;SET SCOPE ADDRESS
1865  003702  004737  014670                    JSR     PC,LTGER        ;GO TO ERROR
1866  003706  000751                            BR      LT4D            ;IF NO SCOPE: BR
1867  003710  000137  002344          LT4X:     JMP     TSCD2           ;RETURN TO SCHED
1868
```

```
                                                ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST**********

003714  012737  023212  000622  LT5:    MOV     #MSLT5,EMADDR   ;SET TEST HEADER
003722  004737  016330          LT5IT:  JSR     PC,INIT1        ;GO INIT
003726  012700  000032                  MOV     #32,R0          ;SET LOOP FOR BITS 4-0
003732  005001                          CLR     R1              ;SET TEST WORD
003734  010177  174574          LT5A:   MOV     R1,@MR          ;SEND TEST WORD TO MR
003740  017702  174570                  MOV     @MR,R2          ;READ MR
003744  042702  177740                  BIC     #177740,R2      ;MASK BITS 4-0
003750  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
003752  001026                          BNE     LT5ER1
003754  005300          LT5B:   DEC     R0
003756  001402                          BEQ     LT5C            ;IF DONE LOOP: BR
003760  005201                          INC     R1              ;BUMP TEST WORD
003762  000764                          BR      LT5A            ;CONTINUE LOOP
003764  012701  000015          LT5C:   MOV     #15,R1          ;SET TEST WORD + WAM 3
003770  012700  001000                  MOV     #1000,R0        ;SET LOOP FOR BITS 15-7
003774  010177  174534          LT5D:   MOV     R1,@MR          ;LOAD MR
004000  017702  174530                  MOV     @MR,R2          ;READ MR
004004  042702  000140                  BIC     #140,R2         ;MASK OUT BITS 5,6
004010  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
004012  001401                          BEQ     LT5E            ;IF SO: BR
004014  000416                          BR      LT5ER2          ;ELSE GO TO ERR 2
004016  005300          LT5E:   DEC     R0
004020  001425                          BEQ     LT5X            ;IF DONE LOOP: BR
004022  062701  000200                  ADD     #200,R1         ;BUMP TEST WORD
004026  000762                          BR      LT5D            ;CONTINUE LOOP
004030  012737  021242  000672  LT5ER1: MOV     #MSG14,ERRDD    ;SET ERROR CODE
004036  012737  003734  000712          MOV     #LT5A,SCOLP     ;SET SCOPE ADDRESS
004044  004737  015776                  JSR     PC,LTGER1       ;GO TO ERROR
004050  000741                          BR      LT5B            ;CONTINUE
004052  012737  021257  000672  LT5ER2: MOV     #MSG15,ERRDD    ;SET ERROR CODE
004060  012737  003774  000712          MOV     #LT5D,SCOLP     ;SET SCOPE ADDRESS
004066  004737  015776                  JSR     PC,LTGER1       ;GO TO ERROR
004072  000751                          BR      LT5E            ;CONTINUE
004074  004737  016240          LT5X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
004100  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                        ;LOGIC TEST 6: TC REGISTER BIT TEST**********
                                                LT6:    NOP
004104  000240                                LT6IT:  MOV     #MSLT6,EMADDR   ;POINT TO LT6 HEADER
004106  012737  023256  000622                MOV     #3,R0           ;SET NUMBER OF TESTS
004114  012700  000003                        CLR     R1
004120  005001                                JSR     PC,INIT1        ;GO INIT
004122  004737  016330                        MOV     R1,@TC          ;WRITE TC
004126  010177  174410                        MOV     @TC,R2          ;READ TC
004132  017702  174404                        BIC     #160000,R2      ;MASK OUT SAC
004136  042702  160000                        CMP     R1,R2           ;SEE IF EXPT = RECVD
004142  020102                                BNE     LT6ER1          ;IF NOT: BR
004144  001010                        LT6D:   DEC     R0
004146  005300                                BEQ     LT6X            ;IF DONE ALL: BR
004150  001417                                CMP     #1,R0           ;SEE IF RESET TEST
004152  022700  000001                        BEQ     LT6A1           ;IF SO: BR
004156  001760                                MOV     #17777,R1       ;SET TEST WORD
004160  012701  017777                        BR      LT6A            ;DO SET TEST
004164  000756                        LT6ER1: MOV     #MSG18,ERADD    ;SET ERROR CODE
004166  012737  021305  000672                MOV     #LT6B,SCOLP     ;SET SCOPE ADDRES
004174  012737  004126  000712                JSR     PC,LTGER1       ;GO TO ERROR
004202  004737  015776                        BR      LT6D            ;CONTINUE
004206  000757                        LT6X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
004210  004737  016240                        JMP     TSCD2           ;RETURN TO SCHED
004214  000137  002344
```

```
1932                                                    ;LOGIC TEST 7: FRAME COUNT BIT TEST**********
1933
1934  004220  000240                          LT7:      NOP
1935  004222  012700  000003                  LT7IT:    MOV     #3,R0           ;SET TEST NUMBER
1936  004226  012737  023322  000622          LT7C:     MOV     #MSLT7,EMADDR   ;SET TEST HEADER
1937  004234  005001                                    CLR     R1              ;SET TEST WORD
1938  004236  004737  016330                  LT7A:     JSR     PC,INIT1        ;GO INIT
1939  004242  010177  174250                            MOV     R1,@FC          ;CLEAR FRAME COUNT
1940  004246  017702  174244                            MOV     @FC,R2          ;READ FC
1941  004252  020102                                    CMP     R1,R2           ;SEE IF EXPT = RECVD
1942  004254  001010                                    BNE     LT7ER1
1943  004256  005300                          LT7B:     DEC     R0              ;SEE IF DONE ALL
1944  004260  001417                                    BEQ     LT7X            ;IF SO: BR
1945  004262  022700  000001                            CMP     #1,R0           ;SEE IF RESET TEST
1946  004266  001757                                    BEQ     LT7C            ;IF SO: BR
1947  004270  012701  177777                            MOV     #-1,R1          ;SET TEST WORD TO -1
1948  004274  000760                                    BR      LT7A            ;CONTINUE
1949  004276  012737  021324  000672  LT7ER1:  MOV     #MSG19,ERADD    ;SET ERROR CODE
1950  004304  012737  004236  000712            MOV     #LT7A,SCOLP     ;SET SCOPE ADDRESS
1951  004312  004737  015776                    JSR     PC,LTGER1       ;GO PRINT ERROR
1952  004316  000757                            BR      LT7B            ;ELSE CONTINUE
1953  004320  012700  000003          LT7X:     MOV     #3,R0           ;RESET TEST AMT
1954  004324  004737  016240                    JSR     PC,ITER         ;GO SEE IF ITERATIONS
1955  004330  000137  002344                    JMP     TSCD2           ;RETURN TO SCHED
1956
```

```
                                                ;LOGIC TEST 10: FUNCTION CODE BIT TEST**********
                                        LT10:   NOP
004334  000240                                    
004336  012737  023366  000622        LT10IT: MOV     #MSLT10,EMADDR  ;SET TEST HEADER
004344  012700  000003                    MOV     #3,R0           ;SET NUMBER OF TESTS
004350  005001                          LT10A1: CLR     R1              ;SET TEST WORRD
004352  012777  000040  174140        LT10A:  MOV     #40,@CS         ;INIT
004360  013777  000624  174132                MOV     DRVN,@CS        ;SELECT DRIVE
004366  010177  174116                        MOV     R1,@C1          ;WRITE C1
004372  017702  174112                        MOV     @C1,R2          ;READ C1
004376  042702  177701                        BIC     #177701,R2      ;MASK FUNCTION CODE
004402  020102                              CMP     R1,R2           ;SEE IF EXPT = RECVD
004404  001010                              BNE     LT10E1
004406  005300                          LT10B:  DEC     R0
004410  001417                              BEQ     LT10X           ;IF DONE ALL: BR
004412  022700  000001                        CMP     #1,R0           ;SEE IF RESET TEST
004416  001754                              BEQ     LT10A1          ;IF SO: BR
004420  012701  000076                        MOV     #76,R1          ;SET TEST WORD
004424  000752                              BR      LT10A           ;DO SET TEST
004426  012737  021343  000672  LT10E1:  MOV     #MSG20,ERRDD    ;SET ERROR CODE
004434  012737  004352  000712                MOV     #LT10A,SCOLP    ;SET SCOPE ADDRESS
004442  004737  015776                        JSR     PC,LTGER1       ;GO PRINT ERROR
004446  000757                              BR      LT10B           ;ELSE CONTINUE
004450  004737  016240                LT10X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
004454  000137  002344                        JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 11: GO BIT SET RESET**********
                                        LT11:   NOP
004460  000240                                    
004462  012737  023441  000622          LT11IT: MOV     #MSLT11,EMADDR  ;SET TEST HEADER
004470  004737  016330                          JSR     PC,INIT1        ;GO INIT
004474  017702  174010                          MOV     @C1,R2          ;READ C1
004500  032702  000001                          BIT     #1,R2           ;SEE IF GO=0
004504  001030                                  BNE     LT11E1
004506  012777  000015  174020          LT11B:  MOV     #15,@MR         ;SELECT WAM 3
004514  005077  173776                          CLR     @FC             ;ASSURE FCS = 1
004520  052777  001700  174014                  BIS     #1700,@TC       ;ASSURE FMT OK
004526  012777  000071  173754                  MOV     #71,@C1         ;SET READ+GO
004534  017702  173750                          MOV     @C1,R2          ;READ C1
004540  032702  000001                          BIT     #1,R2           ;SEE IF GO =1
004544  001424                                  BEQ     LT11E2
004546  004737  016330          LT11C:  JSR     PC,INIT1        ;GO INIT
004552  017702  173732                          MOV     @C1,R2          ;READ C1
004556  032702  000001                          BIT     #1,R2           ;SEE IF GO=0
004562  001444                                  BEQ     LT11X           ;IF SO:BR
004564  000430                                  BR      LT11E3          ;ELSE GO TO ERROR 3
004566  012737  021375  000672  LT11E1: MOV     #MSG21,ERADD    ;SET ERROR CODE
004574  012702  000001                          MOV     #1,R2           ;SET RCVD
004600  005001                                  CLR     R1              ;SET EXPT
004602  012737  004462  000712                  MOV     #LT11IT,SCOLP           ;SET SCOPE ADDRESS
004610  004737  015776                          JSR     PC,LTGER1       ;GO PRINT ERROR
004614  000734                                  BR      LT11B           ;ELSE CONTINUE
004616  012737  021433  000672  LT11E2: MOV     #MSG22,ERADD    ;SET ERROR CODE
004624  005002                                  CLR     R2              ;SET RCVD
004626  012701  000001                          MOV     #1,R1           ;SET EXPT
004632  012737  004506  000712                  MOV     #LT11B,SCOLP    ;SET SCOPE ADDRESS
004640  004737  015776                          JSR     PC,LTGER1       ;GO PRINT ERROR
004644  000740                                  BR      LT11C           ;ELSE CONTINUE
004646  012737  021454  000672  LT11E3: MOV     #MSG23,ERADD    ;SET ERROR CODE
004654  005001                                  CLR     R1              ;SET EXPT
004656  012702  000001                          MOV     #1,R2           ;SET RCVD
004662  012737  004546  000712                  MOV     #LT11C,SCOLP    ;SET SCOPE ADDRESS
004670  004737  015776                          JSR     PC,LTGER1       ;GO PRINT ERROR
004674  004737  016240          LT11X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
004700  000137  002344                          JMP     TSCD2           ;RETURN TO SCHED
```

```
2022
2023                                                    ;LOGIC TEST 12: DRIVE READY BIT**********
2024
2025  004704  000240                          LT12:     NOP
2026  004706  012737  023506  000622  LT12IT: MOV       #MSLT12,EMADDR  ;SET TEST HEADER
2027  004714  004737  016330                  JSR       PC,INIT1        ;GO INIT
2028  004720  032777  000200  173574          BIT       #200,@DS        ;SEE IF DRY=1
2029  004726  001426                          BEQ       LT12E1
2030  004730  012777  000015  173576  LT12B:  MOV       #15,@MR         ;SET WAM3
2031  004736  005077  173554                  CLR       @FC             ;ASSURE FCS = 1
2032  004742  032777  001700  173572          BIT       #1700,@TC       ;ASSURE FMT OK
2033  004750  012777  000071  173532          MOV       #71,@C1         ;SET READ+GO
2034  004756  032777  000200  173536          BIT       #200,@DS        ;SEE IF DRY=0
2035  004764  001020                          BNE       LT12E2
2036  004766  004737  016330          LT12C:  JSR       PC,INIT1        ;GO INIT
2037  004772  032777  000200  173522          BIT       #200,@DS        ;SEE IF DRY=1
2038  005000  001033                          BNE       LT12X           ;IF SO: BR
2039  005002  000422                          BR        LT12E3          ;ELSE GO TO ERROR 3
2040  005004  012737  021507  000672  LT12E1: MOV       #MSG24,ERADD    ;SET ERROR CODE
2041  005012  012737  004706  000712          MOV       #LT12IT,SCOLP   ;SET SCOPE ADDRESS
2042  005020  004737  015770                  JSR       PC,LTGER2       ;GO TO ERROR
2043  005024  000741                          BR        LT12B           ;CONTINUE
2044  005026  012737  021535  000672  LT12E2: MOV       #MSG25,ERADD    ;SET ERROR CODE
2045  005034  012737  004730  000712          MOV       #LT12B,SCOLP    ;SET LOOP ADDRESS
2046  005042  004737  015770                  JSR       PC,LTGER2       ;GO PRINT ERROR
2047  005046  000747                          BR        LT12C           ;CONTINUE
2048  005050  012737  021564  000672  LT12E3: MOV       #MSG25A,ERADD   ;SET ERROR CODE
2049  005056  012737  004766  000712          MOV       #LT12C,SCOLP    ;SET ERROR LOOP
2050  005064  004737  015770                  JSR       PC,LTGER2       ;GET PRINT ERROR
2051  005070  004737  016240          LT12X:  JSR       PC,ITER         ;GO TO ITERATION SUBROUTINE
2052  005074  000137  002344                  JMP       TSCD2           ;RETURN TO SCHED
```

```
2053
2054                                            ;LOGIC TEST 13: INTERRUPT TEST**********
2055
2056  005100  005000                    LT13:   CLR     R0
2057  005102  012737  023557  000622            MOV     #MSLT13,EMADDR  ;SET TEST HEADER
2058  005110  004737  016330            LT13IT: JSR     PC,INIT1        ;GO INIT,SELECT DRIVE, SELECT ABOVE
2059  005114  012737  005172  000670            MOV     #LT13X,RTRN     ;SET RETURN ADDRESS
2060  005122  005077  173362                    CLR     @C1             ;CLEAR CS1
2061  005126  005077  173434                    CLR     @PSW            ;SET PRIORITY
2062  005132  052777  000100  173350            BIS     #100,@C1        ;BIT SET IE
2063  005140  005300                    LT13A:  DEC     R0
2064  005142  001376                            BNE     LT13A           ;AWAIT INTERRUPT
2065  005144  012777  000340  173414    LT13E1: MOV     #340,@PSW       ;RESET PRIORITY
2066  005152  012737  021611  000672            MOV     #MSG26,ERRDD    ;SET ERROR CODE
2067  005160  012737  005110  000712            MOV     #LT13IT,SCOLP   ;SET LOOP ADDRESS
2068  005166  004737  015770                    JSR     PC,LTGER2       ;GO PRINT ERROR
2069  005172  004737  016240            LT13X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
2070  005176  000137  002344                    JMP     TSCD2           ;RETURN TO SCHED
```

```
2071
2072                                        ;THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS.
2073                                        ;THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TE16
2074                                        ;CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.
2075
2076                                        ;LOGIC TEST 14: STATUS AT BOT ON LINE, LOADED, NO WRITE RING**********
2077
2078  005202  032777  001000  173360  LT14:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2079  005210  001005                    BNE     LT14A           ;IF NOT: BR
2080  005212  005737  000740          TST     STFLG           ;SEE IF SINGLE TEST
2081  005216  001433                    BEQ     LT14XX          ;IF NOT: BR
2082  005220  000137  016310          JMP     INHT            ;ELSE GO PRINT INHIB MSG
2083  005224  012737  023424  000622  LT14A:  MOV     #MSLT14,EMADDR  ;SET TEST HEADER
2084  005232  012704  025673          MOV     #MMSG1,R4       ;SET INSTRUCTION ONE
2085  005236  004737  016572          JSR     PC,INST         ;GO DO INSTRUCTION
2086  005242  004737  016330  LT14IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE + SLAVE
2087  005246  012701  014602          MOV     #14602,R1       ;SET TEST WORD
2088  005252  017702  173244          MOV     @DS,R2          ;ASSURE MOL,WRL,DPR,DRY,BOT
2089  005256  020102                    CMP     R1,R2
2090  005260  001410                    BEQ     LT14X           ;IF SO: BR
2091  005262  012737  005242  000712          MOV     #LT14IT,SCOLP   ;SET LOOP ADDRESS
2092  005270  012737  021640  000672          MOV     #MSG27,ERADD    ;SET ERROR CODE
2093  005276  004737  015776          JSR     PC,LTGER1       ;GO PRINT ERROR
2094  005302  004737  016240  LT14X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2095  005306  000137  002344  LT14XX: JMP     TSCD2           ;RETURN TO SCHED
2096
```

```
2097
2098                                                    ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING**********
2099
2100  005312  032777  001000  173250  LT15:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2101  005316  001005                  BNE     LT15A           ;IF NOT: BR
2102  005320  005737  000740          TST     STFLG           ;SEE IF SINGLE TEST
2103  005324  001433                  BEQ     LT15XX          ;IF NOT: BR
2104  005326  000137  016310          JMP     INHT            ;ELSE GO PRINT INHIB MSG
2105  005332  012737  023672  000622  LT15A:  MOV     #MSLT15,EMADDR  ;SET TEST HEADER
2106  005340  012704  025771          MOV     #MSG2,R4
2107  005344  004737  016572          JSR     PC,INS          ;PRINT INSTRUCTION
2108  005350  004737  016340  LT15IT: JSR     PC,INIT2        ;GO INIT, SELECT DRIVE, SLAV
2109  005354  012701  100700          MOV     #100700,R1      ;SET TEST WORD
2110  005360  017702  173134          MOV     @DS,R2          ;READ STATUS
2111  005364  020102                  CMP     R1,R2           ;SEE IF EXPT=RCVD
2112  005366  001410                  BEQ     LT15X
2113  005370  012737  005352  000712  MOV     #LT15IT,SCOLP   ;SET LOOP ADDRESS
2114  005376  012737  021640  000672  MOV     #MSG27,ERRDD    ;SET ERROR CODE
2115  005404  004737  015776          JSR     PC,LTGER1       ;GO PRINT ERROR
2116  005410  004737  016240  LT15X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2117  005414  000137  002344  LT15XX: JMP     TSC02           ;RETURN TO SCHED
```

```
2118
2119                                                    ;LOGIC TEST 16: STATUS AT EOT, OFFLINE LOADED, NO WRITE RING**********
2120
2121  005422  032777  001000  173140  LT16:   BIT     #1000,@SHR      ;SEE IF INHIB MAN TST
2122  005430  001005                          BNE     LT16A           ;IF NOT: BR
2123  005432  005737  000740                  TST     STFLG           ;SEE IF SINGLE TEST
2124  005436  001433                          BEQ     LT16XX          ;IF NOT: BR
2125  005440  000137  016310                  JMP     INHT            ;ELSE GO PRINT INHIB MSG
2126  005444  012737  023740  000622  LT16A:  MOV     #MSLT16,EMADDR  ;SET TEST HEADER
2127  005452  012704  026012                  MOV     #MSG3,R4
2128  005456  004737  016572                  JSR     PC,INST         ;GO PRINT INSTRUCTION
2129  005462  004737  016340          LT16IT: JSR     PC,INIT2        ;SELECT DRIVE,SLAVE
2130  005466  012701  116701                  MOV     #116701,R1      ;SET TEST WORD
2131  005472  017702  173024                  MOV     @0S,R2          ;READ STATUS
2132  005476  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2133  005500  001410                          BEQ     LT16X           ;IF SO: BR
2134  005502  012737  005462  000712          MOV     #LT16IT,SCOLP   ;SET LOOP ADDRESS
2135  005510  012737  021640  000672          MOV     #MSG27,ERADD    ;SET ERROR CODE
2136  005516  004737  015776                  JSR     PC,LTGER1       ;GO PRINT ERROR
2137  005522  004737  016240          LT16X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2138  005526  000137  002344          LT16XX: JMP     TSCD2           ;RETURN TO SCHED
2139
```

```
2140
2141                                                    ;LOGIC TEST 17: STATUS AT ON LINE, LOADED**********
2142
2143  005532  032777  001000  173030  LT17:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2144  005540  001005                    BNE     LT17A           ;IF NOT: BR
2145  005542  005737  000740                    TST     STFLG           ;SEE IF SINGLE TEST
2146  005546  001433                    BEQ     LT17XX          ;IF NOT: BR
2147  005550  000137  016310                    JMP     INMT            ;ELSE GO PRINT INHIB MSG
2148  005554  012737  024006  000622  LT17A:  MOV     #MSLT17,EMADDR  ;SET TEST HEADER
2149  005562  012704  026050                    MOV     #MMSG4,R4
2150  005566  004737  016572                    JSR     PC,IMST         ;GO PRINT INSTRUCTION
2151  005572  004737  016340          LT17IT: JSR     PC,INIT2        ;SELECT DRIVE, SLAVE
2152  005576  012701  110701                    MOV     #110701,R1      ;SET TEST WORD
2153  005602  017702  172714                    MOV     @0S,R2          ;READ STATUS
2154  005606  020102                    CMP     R1,R2           ;SEE IF EXPT=RCVD
2155  005610  001410                    BEQ     LT17X           ;IF SO: BR
2156  005612  012737  005572  000712            MOV     #LT17IT,SCOLP   ;SET LOOP ADDRESS
2157  005620  012737  021640  000672            MOV     #MSG27,ERRDD    ;SET ERROR CODE
2158  005626  004737  015776                    JSR     PC,LTGER1       ;YES PRINT ERROR
2159  005632  004737  016240          LT17X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2160  005636  000137  002344          LT17XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;THE FOLLOWING 11 TESTS WILL TEST ALL POSSIBLE ERROR BITS
                                                ;BY FORCING THEIR CONDITIONS THROUGH VARIOUS ILLEGAL PROGRAMMING
                                                ;SEQUENCES AND USING THE MAINTENANCE WILL MODES AVAILABLE WITH TM03
                                                ;FOR EACH ERROR CONDITION SET THE APPROPRIATE STATUS WILL BE
                                                ;CHECKED. IE: ERR, ATA, SLA, SC ETC.

                                                ;LOGIC TEST 20: ILLEGAL FUNCTION (ILF)**********

005642  012737  024054  000622  LT20:   MOV     #MSLT20,EMADDR  ;SET TEST HEADER
005650  012737  005670  000712          MOV     #LT20A,SCOLP    ;SET LOOP ADDRESS
005656  012700  000022          LT20IT: MOV     #22,R0          ;SET NUMBER OF ILL CODES
005662  012737  000544  000674          MOV     #ILFT,TEMP1     ;POINT TO START IF TABLE
005670  004737  016330          LT20A:  JSR     PC,INIT1        ;GO INIT, SELECT SLAVE + DRIVE
005674  012777  177777  172610          MOV     #-1,@WC         ;SET WC= -1
005702  012701  000001                  MOV     #1,R1           ;SET TEST WORD
005706  117777  172762  172574          MOVB    @TEMP1,@C1      ;SET ILL CODE
005714  017702  172604                  MOV     @ER,R2          ;READ ER
005720  030102                         BIT     R1,R2           ;SEE IF EXPT=RCVD
005722  001011                         BNE     LT20B           ;IF SO: BR
005724  012737  026353  000672          MOV     #TMS17,ERADD    ;SET ERROR CODE
005732  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
005740  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
005744  000404                         BR      LT20C
005746  020102          LT20B:  CMP     R1,R2           ;SEE UNEXPECTED ERRORS
005750  001402                         BEQ     LT20C           ;IF NOT: BR
005752  004737  014650                  JSR     PC,LTGER2       ;ELSE PRINT ERROR
005756  005300          LT20C:  DEC     R0              ;SEE IF DONE ALL ILL CODES
005760  001403                         BEQ     LT20X           ;IF SO: BR
005762  005237  000674                  INC     TEMP1           ;BUMP ADDRESS
005766  000740                         BR      LT20A           ;CONTINUE
005770  004737  016240          LT20X:  JSR     PC,ITER         ;GO SEE IF ITERATION
005774  004737  015330                  JSR     PC,DRVCLR
006000  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2194
2195                                        ;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)**********
2196
2197  006004  012737  024133  000622  LT21:   MOV     #MSLT21,EMADDR  ;SET TEST HEADER
2198  006012  012737  006020  000712          MOV     #LT21IT,SCOLP   ;SET SCOPE LOOP ADDRESS
2199  006020  004737  016330          LT21IT: JSR     PC,INIT         ;GO INIT, SELECT SLAVE, DRIVE
2200  006024  052777  000300  172510          BIS     #300,@TC        ;SET FORMAT
2201  006032  012777  000015  172474          MOV     #15,@MR         ;SET WAM3
2202  006040  012777  000071  172442          MOV     #71,@C1         ;SET READ+GO
2203  006046  005077  172444                  CLR     @FC             ;ATTEMPT WRITE TO FC
2204  006052  012701  000004                  MOV     #4,R1           ;SET TEST WORD
2205  006056  017702  172442                  MOV     @ER,R2          ;GET ER
2206  006062  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2207  006064  001011                          BNE     LT21A           ;IF SO: BR
2208  006066  012737  026367  000672          MOV     #TMS19,ERADD    ;SET ERROR CODE
2209  006074  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2210  006102  004737  014662                  JSR     PC,LTGERO       ;GO PRINT ERROR
2211  006106  000404                          BR      LT21B
2212  006110  020102          LT21A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2213  006112  001402                          BEQ     LT21B           ;IF NOT: BR
2214  006114  004737  014650                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2215  006120  004737  016240          LT21B:  JSR     PC,ITER         ;GO SEE IF ITERATION
2216  006124  012703  040000                  MOV     #40000,R3
2217  006130  005303          LT21XA: DEC     R3              ;DELAY FOR ALPHA
2218  006132  001376                          BNE     LT21XA
2219  006134  004737  014514                  JSR     PC,EORPA        ;GO DO EOR CLEAR
2220  006140  004737  015330                  JSR     PC,DRVCLR
2221  006144  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2222
2223                                                    ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)**********
2224
2225  006150  012737  024167  000622  LT22:   MOV     #MSLT22,EMADDR  ;SET TEST HEADER
2226  006156  012737  006164  000712          MOV     #LT22IT,SCOLP   ;SET SCOPE LOOP ADDRESS
2227  006164  004737  016330          LT22IT: JSR     PC,INIT1        ;INIT, SELECT SLAVE+DRIVE
2228  006170  052777  000020  172322          BIS     #20,@CS         ;ENABLE EVEN PARITY ON MB
2229  006176  012777  177777  172312          MOV     #-1,@FC         ;WRITE TO FC
2230  006204  012701  000010                  MOV     #10,R1          ;SET TEST WORD
2231  006210  042777  000020  172302          BIC     #20,@CS         ;RESET PARITY TO ODD
2232  006216  017702  172302                  MOV     @ER,R2          ;GET ER
2233  006222  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2234  006224  001011                          BNE     LT22A           ;IF SO: BR
2235  006226  012737  026375  000672          MOV     #TMS20,ERADD    ;SET ERROR CODE
2236  006234  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2237  006242  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
2238  006246  000404                          BR      LT22X
2239  006250  020102                  LT22A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2240  006252  001402                          BEQ     LT22X           ;IF NOT: BR
2241  006254  004737  014650                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2242  006260  004737  016240          LT22X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2243  006264  004737  015330                  JSR     PC,DRVCLR
2244  006270  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2245
2246                                        ;LOGIC TEST 23: FORMAT ERROR(FMT)**********
2247
2248  006274  012737  024224  000622  LT23:   MOV     #MSLT23,EMADDR  ;SET TEST HEADER
2249  006302  012737  006310  000712          MOV     #LT23IT,SCOLP   ;SET SCOPE ADDRESS
2250  006310  004737  016330          LT23IT: JSR     PC,INIT1        ;GO INIT SELECT DRIVE+SLAVE
2251  006314  042777  000360  172220          BIC     #360,@TC        ;SET ILLEGAL FORMAT
2252  006322  012701  000020                  MOV     #20,R1          ;SET TEST WORD
2253  006326  012777  000015  172200          MOV     #15,@MR         ;SET WCM 3
2254  006334  012777  000071  172146          MOV     #71,@C1         ;SET READ+GO
2255  006342  017702  172156                  MOV     @ER,R2          ;READ ER
2256  006346  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2257  006350  001011                          BNE     LT23A           ;IF SO: BR
2258  006352  012737  026404  000672          MOV     #TMS21,ERRDD    ;SET ERROR CODE
2259  006360  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2260  006366  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2261  006372  000404                          BR      LT23X
2262  006374  020102          LT23A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2263  006376  001402                          BEQ     LT23X           ;IF NOT: BR
2264  006400  004737  014650                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2265  006404  004737  016240          LT23X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2266  006410  004737  014514                  JSR     PC,EORPA
2267  006414  004737  015330                  JSR     PC,DRVCLR
2268  006420  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2269                                                    ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)**********
2270
2271  006424  012737  024266  000622  LT24:   MOV     #MSLT24,EMADDR  ;SET TEST HEADER
2272  006432  012737  006440  000712          MOV     #LT24IT,SCOLP   ;SET SCOPE ADDRESS
2273  006440  012737  000005  000606  LT24IT: MOV     #5,ITAMT
2274  006446  004737  016356                  JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2275  006452  052777  000300  172062          BIS     #300,@TC        ;SET NORMAL FORMAT
2276  006460  012777  026704  172026          MOV     #WDATA,@BA      ;SET BA
2277  006466  012777  177760  172022          MOV     #-20,@FC        ;SET FC
2278  006474  012777  177770  172010          MOV     #-10,@WC        ;SET WC
2279  006502  012777  000013  172024          MOV     #13,@MR         ;SELECT WAM 2
2280  006510  012777  000061  171772          MOV     #61,@C1         ;SET WRITE+GO
2281  006516  052777  000020  171774          BIS     #20,@CS         ;FORCE EVEN PARITY
2282  006524  012701  000040                  MOV     #40,R1          ;SET TEST WORD
2283  006530  012703  000004                  MOV     #4,R3
2284  006534  005000                          CLR     R0
2285  006536  005300                  1$:     DEC     R0
2286  006540  001376                          BNE     1$              ;DELAY
2287  006542  005303                          DEC     R3
2288  006544  001374                          BNE     1$
2289  006546  012700  000004                  MOV     #4,R0
2290  006552  012777  000013  171754  LT24A:  MOV     #13,@MR         ;CLOCK MR 4 TIMES
2291  006560  005300                          DEC     R0
2292  006562  022700  000002                  CMP     #2,R0           ;SEE IF DONE 1 BYTE
2293  006566  001002                          BNE     LT24B0          ;IF NOT: BR
2294  006570  017701  171740                  MOV     @MR,R1          ;ELSE GET BYTE 1
2295  006574  005700                  LT24B0: TST     R0              ;SEE IF BYTE 2
2296  006576  001365                          BNE     LT24B           ;IF NOT: BR
2297  006600  017704  171730                  MOV     @MR,R4          ;GET BYTE 2
2298  006604  005000                          CLR     R0
2299  006606  005300                  LT24C:  DEC     R0
2300  006610  001376                          BNE     LT24C           ;DELAY
2301  006612  032777  000040  171704          BIT     #40,@ER         ;SEE IF DPAR IS SET
2302  006620  001023                          BNE     LT24D           ;IF SO: BR
2303  006622  000301                          SWAB    R1
2304  006624  042701  177400                  BIC     #177400,R1      ;GET LOW BYTE
2305  006630  042704  000377                  BIC     #377,R4
2306  006634  050401                          BIS     R4,R1           ;GET HIGH BYTE
2307  006636  005237  000744                  INC     T24FL           ;SET T24 FLAG
2308  006642  012737  026412  000672          MOV     #TMS22,ERADD    ;SET ERROR CODE
2309  006650  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2310  006656  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2311  006662  005037  000744                  CLR     T24FL           ;CLEAR FLAG
2312  006666  000412                          BR      LT24X
2313  006670  012701  000050          LT24D:  MOV     #50,R1
2314  006674  017702  171624                  MOV     @ER,R2          ;GET ERROR REGISTER
2315  006700  042702  020000                  BIC     #20000,R2       ;MASK OPI
2316  006704  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2317  006706  001402                          BEQ     LT24X           ;IF NOT: BR
2318  006710  004737  014650                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2319  006714  042777  000020  171576  LT24X:  BIC     #20,@CS         ;RESET EVEN PARITY
2320  006722  004737  014514                  JSR     PC,EORPA        ;GO DO EOR CLEAR
2321  006726  004737  015330                  JSR     PC,DRVCLR       ;GO SEE IF DRIVE CLEAR OK
2322  006732  004737  016240                  JSR     PC,ITER         ;GO SEE IF ITERATION
2323  006736  012737  000020  000606          MOV     #20,ITAMT
2324  006744  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2325
2326                                                ;LOGIC TEST 25: NON-EXECUTABLE FUNCTION(NEF)##########
2327
2328  006750  012737  024326  000622  LT25:   MOV     #MSLT25,EMADDR  ;SET TEST HEADER
2329  006756  004737  016356          LT25IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2330  006762  052777  000300  171552          BIS     #300,@TC        ;SET NORMAL FORMAT
2331  006770  012777  177777  171520          MOV     #-1,@FC         ;SET ITLLEGAL FC
2332  006776  012777  000013  171530          MOV     #13,@MR         ;SET WCM 2
2333  007004  012777  000061  171476          MOV     #61,@C1         ;LOAD WRITE+GO
2334  007012  012701  004000                  MOV     #4000,R1        ;SET TEST WORD
2335  007016  017702  171502                  MOV     @ER,R2          ;GET ER
2336  007022  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2337  007024  001014                          BNE     LT25A           ;IF SO: BR
2338  007026  012737  006756  000712          MOV     #LT25IT,SCOLP   ;SET LOOP ADDRESS
2339  007034  012737  026500  000672          MOV     #TMS31,ERADD    ;SET ERROR CODE
2340  007042  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
2341  007050  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
2342  007054  000404                          BR      LT25X
2343  007056  020102                  LT25A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2344  007060  001402                          BEQ     LT25X           ;IF NOT: BR
2345  007062  004737  014650                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2346  007066  004737  016240          LT25X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2347  007072  004737  015330                  JSR     PC,DRVCLR
2348  007076  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2349
2350                                                    ;LOGIC TEST 26: FRAME COUNT ERROR(FCE)**********
2351
2352  007102  012737  024362  000622  LT26:   MOV     #MSLT26,EMADDR  ;SET TEST HEADER
2353  007110  004737  016356          LT26IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2354  007114  005000                          CLR     R0
2355  007116  005300                  1$:     DEC     R0
2356  007120  001376                          BNE     1$              ;AWAIT OPI RESET
2357  007122  052777  000300  171412        BIS     #300,@TC        ;SET NORMAL FORMAT
2358  007130  012777  177770  171364        MOV     #-10,@WC        ;SET WC=-10
2359  007136  012777  177760  171352        MOV     #-20,@FC        ;SET FC=-20
2360  007144  012777  000013  171362        MOV     #13,@MR         ;SET WAM 3
2361  007152  012777  000061  171330        MOV     #61,@C1         ;LOAD WRITE+GO
2362  007160  012701  001000                MOV     #1000,R1        ;SET TEST WORD
2363  007164  005000                          CLR     R0
2364  007166  005300                  2$:     DEC     R0
2365  007170  001376                          BNE     2$              ;DELAY
2366  007172  012777  000025  171334        MOV     #25,@MR         ;LOAD MM EOR CLEAR
2367  007200  105077  171330                CLRB    @MR             ;RESET MR
2368  007204  012703  000004                MOV     #4,R3
2369  007210  005000                          CLR     R0
2370  007212  032777  001000  171304  3$:     BIT     #1000,@ER       ;SEE IF FCE SET
2371  007220  001022                          BNE     4$              ;IF SO: BR
2372  007222  005300                          DEC     R0
2373  007224  001372                          BNE     3$              ;DELAY
2374  007226  005303                          DEC     R3
2375  007230  001370                          BNE     3$
2376  007232  017702  171266                MOV     @ER,R2          ;GET ER
2377  007236  012737  007110  000712        MOV     #LT26IT,SCOLP   ;SET SCOPE ADDRESS
2378  007244  012737  026457  000672        MOV     #TMS28,ERADD
2379  007252  012737  000001  000716        MOV     #1,EXFL         ;SET EXPT FLG
2380  007260  004737  014662                JSR     PC,LTGER0       ;GO PRINT ERROR
2381  007264  000406                          BR      LT26X
2382  007266  017702  171232          4$:     MOV     @ER,R2          ;GET ERROR REGISTER
2383  007272  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2384  007274  001402                          BEQ     LT26X           ;IF NOT: BR
2385  007276  004737  014650                JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2386  007302  004737  016240          LT26X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2387  007306  004737  015330                JSR     PC,DRVCLR
2388  007312  000137  002344                JMP     TSCD2           ;RETURN TO SCHED
```

```
2389
2390                                                    ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)**********
2391
2392  007316  022737  172400  000510  LT27:   CMP     #172400,C1      ;SEE IF ADDRESSES OPEN
2393  007324  001041                  BNE     LT27XX          ;IF NOT: BR
2394  007326  012737  007352  000712          MOV     #LT27A,SCOLP    ;SET SCOPE ADDRESS
2395  007334  012737  024416  000622          MOV     #MSLT27,EMADDR  ;SET TEST HEADER
2396  007342  012700  000020          LT27IT: MOV     #20,R0          ;SET NUMBER OF ILR TESTS
2397  007346  012701  172434                  MOV     #172434,R1      ;SET FIRST ILR ADDRESS
2398  007352  004737  016356          LT27A:  JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2399  007356  011103                  MOV     (R1),R3         ;ATTEMPT ILR READ
2400  007360  032777  000002  171136          BIT     #2,@ER          ;SEE IF ILR=1
2401  007366  001010                  BNE     LT27B           ;IF SO: BR
2402  007370  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT-NOT RCVD FLAG
2403  007376  012737  026301  000672          MOV     #TMS10,ERADD    ;SET ERROR CODE
2404  007404  004737  014670                  JSR     PC,LTGER        ;GO PRINT ERROR
2405  007410  005300                  LT27B:  DEC     R0              ;SEE IF DONE ALL
2406  007412  001402                  BEQ     LT27X           ;IF SO: BR
2407  007414  005721                  TST     (R1)+           ;BUMP ADDRESS
2408  007416  000755                  BR      LT27A           ;CONTINUE TESTS
2409  007420  004737  016240          LT27X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2410  007424  004737  015330                  JSR     PC,DRVCLR
2411  007430  000137  002344          LT27XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2412
2413                                                    ;LOGIC TEST 30: DRIVE TIMING ERROR**********
2414
2415  007434  012737  026506  000672  LT30:   MOV     #TMS32,ERRDD    ;SET ERROR CODE
2416  007442  012737  024452  000622          MOV     #MSLT30,EMADDR  ;SET TEST HEADER
2417  007450  012737  007456  000712          MOV     #LT30IT,SCOLP   ;SET SCOPE ADDRESS
2418  007456  004737  016356          LT30IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE + SLAVE
2419  007462  052777  000300  171052          BIS     #300,@TC        ;SET NORMAL FORMAT
2420  007470  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2421  007474  012777  000017  171032          MOV     #17,@MR         ;CRIPPLE OCCUPIED
2422  007502  005077  171010                  CLR     @FC             ;SET FC3
2423  007506  012777  000061  170774          MOV     #61,@C1         ;LOAD WRITE+GO
2424  007514  032777  010000  171002          BIT     #10000,@ER      ;SEE IF DTE SET
2425  007522  001005                          BNE     LT30A           ;IF SO: BR
2426  007524  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2427  007532  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2428  007536  004737  016356          LT30A:  JSR     PC,INIT3        ;GO INIT SELECT DRIVE,SLAVE
2429  007542  052777  000300  170772          BIS     #300,@TC        ;SET FORMAT
2430  007550  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2431  007554  005077  170736                  CLR     @FC             ;SET FCS
2432  007560  012777  000015  170746          MOV     #15,@MR         ;SET WRAP 3
2433  007566  012777  000061  170714          MOV     #61,@C1         ;LOAD WRITE+GO
2434  007574  012704  040000                  MOV     #40000,R4
2435  007600  005777  170736          LT30B:  TST     @TC             ;SEE IF ALPHA
2436  007604  100015                          BPL     LT30C           ;AWAIT ALPHA
2437  007606  005300                          DEC     R0
2438  007610  001373                          BNE     LT30B
2439  007612  013704  000622                  MOV     EMADDR,R4
2440  007616  004737  017320                  JSR     PC,TTOUT        ;PRINT HEADER
2441  007622  012704  022374                  MOV     #MSG50,R4
2442  007626  004737  017320                  JSR     PC,TTOUT        ;PRINT ALPHA ERROR
2443  007632  004737  016210                  JSR     PC,SCOPE
2444  007636  000435                          BR      LT30X
2445  007640  012777  000015  170666  LT30C:  MOV     #15,@MR         ;CLOCK MR
2446  007646  012777  000015  170660          MOV     #15,@MR         ;CLOCK MR
2447  007654  005000                          CLR     R0
2448  007656  005300                  LT30D:  DEC     R0
2449  007660  001376                          BNE     LT30D           ;DELAY
2450  007662  032777  010000  170634          BIT     #10000,@ER      ;SEE IF DTE SET
2451  007670  001006                          BNE     LT30E           ;IF SO: BR
2452  007672  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2453  007700  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2454  007704  000412                          BR      LT30X
2455  007706  012701  010000          LT30E:  MOV     #10000,R1       ;SET TEST WORD
2456  007712  017702  170606                  MOV     @ER,R2          ;GET ERROR REGISTER
2457  007716  042702  020100                  BIC     #20100,R2       ;MASK OPI AND VPE
2458  007722  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2459  007724  001402                          BEQ     LT30X           ;IF NOT: BR
2460  007726  004737  014650                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2461  007732  004737  016240          LT30X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2462  007736  004737  014514                  JSR     PC,EORPA        ;GO CLEAR GO BIT
2463  007742  004737  015330                  JSR     PC,DRVCLR
2464  007746  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
2465
```

```
2466
2467                                                    ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)**********
2468
2469  007752  012737  024510  000622  LT31:   MOV     #MSLT31,EMADDR  ;SET TEST HEADER
2470  007760  012737  007760  000712  LT31IT: MOV     #LT31IT,SCOLP   ;SET SCOPE ADDRESS
2471  007766  012737  026522  000672          MOV     #TMS33A,ERADD   ;SET ERROR MSG HDR
2472  007774  012737  000002  000606          MOV     #2,ITAMT        ;SET REDUCED ITER COUNT
2473  010002  004737  016356                  JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2474  010006  005000                          CLR     R0
2475  010010  005300                  1$:     DEC     R0
2476  010012  001376                          BNE     1$              ;AWAIT OPI RESET
2477  010014  052777  000300  170520          BIS     #300,@TC        ;SET FORMAT
2478  010022  012777  000013  170504          MOV     #13,@MR         ;SET WRAP 2
2479  010030  005077  170462                  CLR     @FC             ;SET FRAME COUNT
2480  010034  012705  020000                  MOV     #20000,R5       ;SET TEST BIT (OPI)
2481  010040  012702  010056                  MOV     #2$,R2          ;SET RETURN ADDRESS FROM TIMER
2482  010044  004737  010256                  JSR     PC,TIMON        ;START TIMER
2483  010050  012777  000061  170432          MOV     #61,@C1         ;LOAD WRITE+GO
2484  010056  030577  170442          2$:     BIT     R5,@ER          ;BRANCH WHEN OPI SETS
2485  010062  001002                          BNE     3$
2486  010064  000163  010350                  JMP     TIMER(R3)       ;GO TO TIMER & RETURN TO 2$ ABOVE
2487  010070  017702  170430          3$:     MOV     @ER,R2          ;GET ERROR REGISTER
2488  010074  020502                          CMP     R5,R2           ;SEE IF UNEXPECTED ERRORS
2489  010076  001403                          BEQ     4$              ;IF NOT: BR
2490  010100  004737  014650                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2491  010104  000453                          BR      LT31X
2492  010106  004737  010442          4$:     JSR     PC,TIMOK        ;GO CHECK TIME FOR OPI TO SET
2493  010112  102450                          BVS     LT31X           ;BRANCH IF TIME WAS INCORRECT
2494
2495  010114  012737  010130  000712          MOV     #LT31A,SCOLP    ;SET SCOPE LOOP
2496  010122  012737  026536  000672          MOV     #TMS338,ERADD   ;SET ERROR MSG HEADER
2497  010130  004737  016356          LT31A:  JSR     PC,INIT3        ;GO INIT
2498  010134  005000                          CLR     R0
2499  010136  005300                  1$:     DEC     R0              ;WAIT FOR OPI TO CLEAR
2500  010140  001376                          BNE     1$
2501  010142  052777  000300  170372          BIS     #300,@TC        ;SET FORMAT
2502  010150  012777  000015  170356          MOV     #15,@MR         ;SET WRAP 3
2503  010156  012702  010200                  MOV     #2$,R2          ;SET RETURN ADDRESS FROM TIMER
2504  010162  012705  020000                  MOV     #20000,R5       ;SET TEST WORD
2505  010166  004737  010256                  JSR     PC,TIMON        ;START TIMER
2506  010172  012777  000071  170310          MOV     #71,@C1         ;LOAD READ+GO
2507  010200  030577  170320          2$:     BIT     R5,@ER          ;BRANCH WHEN OPI SETS
2508  010204  001002                          BNE     3$
2509  010206  000163  010350                  JMP     TIMER(R3)       ;GO TO TIMER
2510  010212  017702  170306          3$:     MOV     @ER,R2          ;GET ERROR REGISTER
2511  010216  020502                          CMP     R5,R2           ;SEE IF UNEXPECTED ERRORS
2512  010220  001403                          BEQ     4$
2513  010222  004737  014650                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2514  010226  000402                          BR      LT31X           ;EXIT TEST
2515  010230  004737  010442          4$:     JSR     PC,TIMOK        ;GO CHECK TIME
2516  010234  004737  016240          LT31X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2517  010240  004737  015330                  JSR     PC,DRVCLR
2518  010244  012737  000020  000606          MOV     #20,ITAMT
2519  010252  000137  002344                  JMP     TSC02           ;RETURN TO SCHED
2520
2521                                            ;ROUTINE TO START THE TIMER. THE TIMER IS AN OSCILLATOR IN THE MAINT-
```

```
                                        ;ENHANCE REGISTER (BIT 6) THAT TOGGLES EVERY 56 (10) MICROSECONDS. THIS
                                        ;ROUTINE WAITS FOR THE OSCILLATOR TO TOGGLE AND RETURN WITH R3 INDICATING
                                        ;THE STATE OF THE OSCILLATOR.
010256  005000                  TIMON:  CLR     R0              ;CLEAR TICK COUNT
010260  005001                          CLR     R1
010262  012703  000024                  MOV     #24,R3          ;PRESET INDEX TO TIMER
010266  032777  000100  170240          BIT     #100,@MR        ;BRANCH IF OSC CLEAR
010274  001405                          BEQ     2$
010276  032777  000100  170230  1$:     BIT     #100,@MR        ;WAIT FOR OSC TO CLEAR
010304  001374                          BNE     1$
010306  000405                          BR      4$              ;EXIT

010310  005403                  2$:     NEG     R3              ;SET INDEX TO TIMER
010312  032777  000100  170214  3$:     BIT     #100,@MR        ;WAIT FOR OSC TO SET
010320  001774                          BEQ     3$
010322  000207                  4$:     RTS     PC              ;RETURN

                                        ;THIS ROUTINE TIMES AN EVENT. EACH TIME THE OSCILLATOR BIT CHANGES
                                        ;STATE THE TICK COUNT IN R1 & R0 IS INCREMENTED. THE ROUTINE IS CALLED
                                        ;USING R3 AS AN INDEX TO INDICATE THE OSCILLATORS PAST STATE. WHEN
                                        ;THE OSC BIT CHANGES STATE R3 IS NEGATED.
010324  032777  000100  170202  TIMER1: BIT     #100,@MR        ;BRANCH IF OSC HAS CHANGED STATE
010332  001406                          BEQ     TIMER
010334  000112                          JMP     (R2)            ;RETURN
        010350                          .=TIMER1+24
010350  005403                  TIMER:  NEG     R3              ;SET INDEX TO OTHER STATE
010352  062700  000001                  ADD     #1,R0           ;INCREMENT TICK COUNT
010356  005501                          ADC     R1
010360  022701  000003                  CMP     #3,R1           ;BRANCH IF TIMER OVERFLOWS
010364  001410                          BEQ     TIMOVF
010366  000112                          JMP     (R2)            ;RETURN
        010374                          .=TIMER +24
010374  032777  000100  170132  TIMER0: BIT     #100,@MR        ;BRANCH IF OSC SET
010402  001362                          BNE     TIMER
010404  000112                          JMP     (R2)            ;RETURN

010406  013704  000622          TIMOVF: MOV     EMADDR,R4       ;TYPE TEST HEADER
010412  004737  017320                  JSR     PC,TTOUT
010416  013704  000672                  MOV     ERADD,R4        ;GET ERROR MSG ADDRESS
010422  004737  017320                  JSR     PC,TTOUT        ;AND TYPE IT
010426  012704  026616                  MOV     #TMS33E,R4      ;TYPE
010432  004737  017320                  JSR     PC,TTOUT        ;'TIMER OVERFLOWED'
010436  000137  010234                  JMP     LT31X           ;GO EXIT TEST

                                ;ROUTINE TO CHECK IF TIME IS WITHIN LIMITS. IF NOT THE ROUTINE RETURNS
                                ;WITH THE 'V' BIT SET. THE LIMITS WERE SLECTED BY DIVIDING THE TIME
                                ;IN MICROSECONDS BY 448. THE LOWER LIMIT IS 5,500,000 USECS (5.5 SECS);
                                ;THE UPPER LIMIT IS 8,500,000 USECS (8.5 SECS). THE 448 IS DERIVED FROM
                                ;56 USECS/TICK TIMES THE DIVISION BY 8 BY THE TIMOK ROUTINE.
010442  000240                  TIMOK:  NOP
010444  006201                          ASR     R1              ;DIVIDE COUNT BY 8
010446  006000                          ROR     R0
010450  006201                          ASR     R1
010452  006000                          ROR     R0
010454  006201                          ASR     R1
010456  006000                          ROR     R0
```

```
2578  010460  020027  027764            CMP     R0,#12276.      ;BRANCH IF GREATER THAN LOWER LIMIT
2579  010464  101016                    BHI     1$
2580  010466  013704  000622            MOV     EMADDR,R4       ;GET ERROR MSG HEADER
2581  010472  004737  017320            JSR     PC,TTOUT        ;TYPE ERROR MSG HEADER
2582  010476  013704  000672            MOV     ERADD,R4        ;GET ERROR DESCRIPTOR MSG
2583  010502  004737  017320            JSR     PC,TTOUT
2584  010506  012704  026551            MOV     #TMS33C,R4      ;TYPE 'OCCURED TOO SOON'
2585  010512  004737  017320            JSR     PC,TTOUT
2586  010516  000262                    SEV                     ;SET 'V' TO INDICATE ERROR
2587  010520  000420                    BR      2$
2588
2589  010522  020027  045035    1$:     CMP     R0,#18973.      ;BRANCH IF LESS THAN UPPER LIMIT
2590  010526  003415                    BLE     2$
2591  010530  013704  000622            MOV     EMADDR,R4       ;GET ERROR MSG HEADER
2592  010534  004737  017320            JSR     PC,TTOUT
2593  010540  013704  000672            MOV     ERADD,R4
2594  010544  004737  017320            JSR     PC,TTOUT        ;TYPE ERROR MSG HEADER
2595  010550  012704  026573            MOV     #TMS33D,R4      ;TYPE 'OCCURED TOO LATE'
2596  010554  004737  017320            JSR     PC,TTOUT
2597  010560  000262                    SEV
2598  010562  000207            2$:     RTS     PC
```

```
                                                ;LOGIC TEST 32: UNSAFE(UNS)**********
010564  012737  024544  000622  LT32:   MOV     #MSLT32,EMADDR  ;SET TEST HEADER
010572  012737  010600  000712          MOV     #LT32IT,SCOLP   ;SET SCOPE ADDRESS
010600  004737  016356          LT32IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE +SLAVE
010604  013700  000664                  MOV     SLVN,R0         ;GET SLAVE NUMBER
010610  005100                          COM     R0              ;SET NONEXISTANT SLAVE
010612  042700  177770                  BIC     #177770,R0      ;MASK SLAVE NUMBER
010616  052700  000300                  BIS     #300,R0         ;SET FORMAT
010622  010077  167714                  MOV     R0,@TC          ;SELECT ILLEGAL SLAVE
010626  032777  002000  167702          BIT     #2000,@DT       ;EXIT TEST IF SALVE AVAILABLE
010634  001032                          BNE     LT32XX
010636  012777  000071  167644          MOV     #71,@C1         ;LOAD READ+GO
010644  012701  040000                  MOV     #40000,R1       ;SET TEST WORD
010650  017702  167650                  MOV     @ER,R2          ;READ ER
010654  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
010656  001011                          BNE     1$              ;IF SO: BR
010660  012737  026636  000672          MOV     #TMS34,ERADD    ;SET ERROR CODE
010666  012737  000001  000716          MOV     #1,EXFL         ;SET ERROR CODE
010674  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
010700  000404                          BR      LT32X
010702  020102          1$:     CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
010704  001402                          BEQ     LT32X           ;IF NOT: BR
010706  004737  014650                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
010712  004737  016240          LT32X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
010716  004737  015330                  JSR     PC,DRVCLR
010722  000137  002344          LT32XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2629
2630                                    ;THE FOLLOWING 6 TESTS WILL LOOK AT VARIOUS BITS IN THE
2631                                    ;DRIVE STATUS(DS) AND TAPE CONTROL(TC)
2632                                    ;REGISTERS BY FORCING CERTAIN CONDITONS WHICH DO NOT
2633                                    ;REQUIRE TAPE MOVEMENT.
2634
2635                                    ;LOGIC TEST 33: POSITIONING IN PROGRESS(PIP)**********
2636
2637  010726  012737  024600  000622  LT33:   MOV     #MSLT33,EMADDR  ;SET TEST HEADER
2638  010734  012737  010742  000712          MOV     #LT33IT,SCOLP   ;SET SCOPE ADDRESS
2639  010742  004737  016356          LT33IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2640  010746  012777  000013  167560          MOV     #13,@MR         ;SET WAM 2
2641  010754  012777  177777  167534          MOV     #-1,@FC         ;SET FCS
2642  010762  012777  000031  167520          MOV     #31,@C1         ;LOAD SPACE FORWARD+GO
2643  010770  032777  020000  167524          BIT     #20000,@DS      ;SEE IF PIP=1
2644  010776  001010                          BNE     LT33X           ;IF SO: BR
2645  011000  012737  026331  000672          MOV     #TMS14,ERADD    ;SET ERROR CODE
2646  011006  012737  000001  000716          MOV     #1,EXFL         ;SET ERROR CODE
2647  011014  004737  014662                  JSR     PC,LTGERO       ;GO PRINT ERROR
2648  011020  004737  016240          LT33X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2649  011024  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2650
2651                                                    ;LOGIC TEST 34: PHASE ENCODED STATUS(PES)********
2652
2653  011030  012737  026251  000672  LT34:   MOV     #TMS6,ERRDD     ;SET ERROR CODE
2654  011036  012737  024634  000622          MOV     #MSLT34,EMADDR  ;SET TEST HEADER
2655  011044  012700  000004          LT34IT: MOV     #4,R0
2656  011050  004737  016356          LT34A1: JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2657  011054  042777  003400  167460          BIC     #3400,@TC       ;SELECT NRZI
2658  011062  052777  001400  167452          BIS     #1400,@TC
2659  011070  032777  000040  167424  LT34A:  BIT     #40,@DS         ;SEE IF PES=0
2660  011076  001410                          BEQ     LT34B           ;IF SO: BR
2661  011100  012737  000002  000716          MOV     #2,EXFL         ;SET RCVD-NOT EXPT
2662  011106  012737  011050  000712          MOV     #LT34A1,SCOLP   ;SET SCOPE ADDRESS
2663  011114  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2664  011120  004737  016372          LT34B:  JSR     PC,INIT4
2665  011124  032777  000040  167370  LT34C:  BIT     #40,@DS         ;SEE IF PES=1
2666  011132  001010                          BNE     LT34X           ;IF SO: BR
2667  011134  012737  011124  000712          MOV     #LT34C,SCOLP    ;SET SCOPE ADDRESS
2668  011142  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT-NOT RCVD FLAG
2669  011150  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2670  011154  004737  016240          LT34X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2671  011160  000137  002344          LT34XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2672
2673                                                    ;LOGIC TEST 35: SLAVE ADDRESS CHANGE (SAC)**********
2674
2675  011164  012737  026661  000672  LT35:   MOV     #TMS37,ERADD
2676  011172  012737  024670  000622          MOV     #MSLT35,EMADDR
2677  011200  004737  016356          LT35IT: JSR     PC,INIT3        ;INIT SELECT DRIVE, SLAVE
2678  011204  032777  000020  167310  1$:     BIT     #20,@DS         ;SEE IF SDWN IS RESET
2679  011212  001374                          BNE     1$              ;IF NOT: BR
2680  011214  052777  000300  167320          BIS     #300,@TC        ;SET FORMAT
2681  011222  012777  000015  167304          MOV     #15,@MR         ;SET WAM 3
2682  011230  012777  000071  167252          MOV     #71,@C1         ;LOAD READ+GO
2683  011236  032777  020000  167276          BIT     #20000,@TC      ;SEE IF SAC=0
2684  011244  001410                          BEQ     LT35A           ;IF SO: BR
2685  011246  012737  000002  000716          MOV     #2,EXFL         ;SET RCV-NOT EXPT FLAG
2686  011254  012737  011200  000712          MOV     #LT35IT,SCOLP   ;SET SCOPE ADDRESS
2687  011262  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2688  011266  004737  016356          LT35A:  JSR     PC,INIT3        ;INIT
2689  011272  005277  167244                  INC     @TC             ;BUMP SLAVE ADDRESS
2690  011276  032777  020000  167236          BIT     #20000,@TC      ;SEE IF SAC=1
2691  011304  001010                          BNE     LT35X           ;IF SO: BR
2692  011306  012737  011266  000712          MOV     #LT35A,SCOLP    ;SET SCOPE ADDRESS
2693  011314  012737  000001  000716          MOV     #1,EXFL         ;SE EXPT-NOT RCVD FLAG
2694  011322  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2695  011326  004737  016240          LT35X:  JSR     PC,ITER
2696  011332  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2697
2698                                                    ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)**********
2699
2700  011336  012737  024732  000622  LT36:   MOV     #MSLT36,EMADDR
2701  011344  012737  026667  000672          MOV     #TMS38,ERADD    ;SET ERROR CODE
2702  011352  004737  016356          LT36IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2703  011356  032777  040000  167156          BIT     #40000,@TC      ;SEE IF FCS=0
2704  011364  001410                          BEQ     1$              ;IF SO: BR
2705  011366  012737  011352  000712          MOV     #LT36IT,SCOLP   ;SET SCOPE ADDRESS
2706  011374  012737  000002  000716          MOV     #2,EXFL         ;SET RCVD-NOT EXPT
2707  011402  004737  014662                  JSR     PC,LTGERO       ;GO PRINT ERROR
2708  011406  004737  016356          1$:     JSR     PC,INIT3        ;INIT
2709  011412  005077  167100                  CLR     @FC             ;WRITE TO FC
2710  011416  032777  040000  167116          BIT     #40000,@TC      ;SEE IF FCS=1
2711  011424  001010                          BNE     LT36X           ;IF SO: BR
2712  011426  012737  011406  000712          MOV     #1$,SCOLP       ;SET SCOPE ADDRESS
2713  011434  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT-NOT RCVD
2714  011442  004737  014662                  JSR     PC,LTGERO       ;GO PRINT ERROR
2715  011446  004737  016240          LT36X:  JSR     PC,ITER
2716  011452  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2717
2718                                            ;LOGIC TEST 37: ACCELERATION(ACCL)**********
2719
2720  011456  012737  024774  000622  LT37:   MOV     #MSLT37,EMADDR
2721  011464  012737  026675  000672          MOV     #TMS39,ERADD    ;SET ERROR CODE
2722  011472  004737  016356          LT37IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2723  011476  052777  000300  167036          BIS     #300,@TC        ;SET FORMAT
2724  011504  005777  167032                  TST     @TC             ;SEE IF ACCL=1
2725  011510  100410                          BMI     LT37A           ;IF SO: BR
2726  011512  012737  000001  000716          MOV     #1,EXFL
2727  011520  012737  011472  000712          MOV     #LT37IT,SCOLP   ;SET SCOPE ADDRESS
2728  011526  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2729  011532  004737  016356          LT37A:  JSR     PC,INIT3        ;INIT
2730  011536  052777  000300  166776          BIS     #300,@TC        ;SET FORMAT
2731  011544  012777  000015  166762          MOV     #15,@MR         ;SET WAM 3
2732  011552  012777  000071  166730          MOV     #71,@C1         ;LOAD READ+GO
2733  011560  012700  100000                  MOV     #100000,R0      ;SET ACCL DELAY
2734  011564  005777  166752          LT37B:  TST     @TC             ;SEE IF ACCL=0
2735  011570  100012                          BPL     LT37X           ;IF SO: BR
2736  011572  005300                          DEC     R0
2737  011574  001373                          BNE     LT37B           ;DELAY
2738  011576  012737  011532  000712          MOV     #LT37A,SCOLP    ;SET SCOPE ADDRESS
2739  011604  012737  000002  000716          MOV     #2,EXFL
2740  011612  004737  014662                  JSR     PC,LTGERD       ;GO PRINT ERROR
2741  011616  004737  016240          LT37X:  JSR     PC,ITER
2742  011622  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2743
2744                                                    ;LOGIC TEST 40: PE TAPE MARK (TM)**********
2745
2746  011626  012737  011642  000712  LT40:   MOV     #LT40IT,SCOLP   ;SET SCOPE ADDRESS
2747  011634  012737  025037  000622          MOV     #MSLT40,EMADDR
2748  011642  004737  016372          LT40IT: JSR     PC,INIT4        ;INIT, SELECT DRIVE+SLAVE
2749  011646  005000                  1$:     CLR     R0
2750  011650  005300                          DEC     R0
2751  011652  001376                          BNE     1$              ;DELAY FOR OPI RESET
2752  011654  052777  002300  166660          BIS     #2300,@TC
2753  011662  012777  000007  166644          MOV     #7,@MR          ;SET WAM 0
2754  011670  012777  000027  166612          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
2755  011676  012700  100000                  MOV     #100000,R0      ;SET DELAY
2756  011702  032777  000004  166612  2$:     BIT     #4,@DS          ;SEE IF TM=1
2757  011710  001012                          BNE     LT40X           ;IF SO: BR
2758  011712  005300                          DEC     R0
2759  011714  001372                          BNE     2$              ;DELAY
2760  011716  012737  026227  000672          MOV     #TMS3,ERADD
2761  011724  012737  000001  000716          MOV     #1,EXFL
2762  011732  004737  014662                  JSR     PC,LTGERR       ;GO PRINT ERROR
2763  011736  004737  016240          LT40X:  JSR     PC,ITER
2764  011742  000137  002344          LT40XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2765
2766                                                    ;LOGIC TEST 41: NRZ TAPE MARK (TM,VPE,ITM)**********
2767
2768  011746  012737  011762  000712  LT41:   MOV     #LT41IT,SCOLP   ;SET SCOPE ADDRESS
2769  011754  012737  025104  000622          MOV     #MSLT41,EMADDR
2770  011762  004737  016356          LT41IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE,SLAVE
2771  011766  052777  001700  166546          BIS     #1700,@TC       ;SET NRZ+NORMAL FORMAT
2772  011774  012777  177760  166514          MOV     #-20,@FC        ;SET FCS
2773  012002  012777  000007  166524          MOV     #7,@MR          ;SET WAM 0
2774  012010  012777  000027  166472          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
2775  012016  005000                          CLR     R0
2776  012020  032777  000004  166474  1$:     BIT     #4,@DS          ;SEE IF TM=1
2777  012026  001012                          BNE     2$              ;IF SO: BR
2778  012030  005300                          DEC     R0
2779  012032  001372                          BNE     1$              ;DELAY
2780  012034  012737  026227  000672          MOV     #TMS3,ERRDD     ;SET ERROR CODE
2781  012042  012737  000001  000716          MOV     #1,EXFL
2782  012050  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
2783  012054  032777  002000  166442  2$:     BIT     #2000,@ER       ;SEE IF ITM=1
2784  012062  001010                          BNE     3$              ;IF SO: BR
2785  012064  012737  026472  000672          MOV     #TMS30,ERRDD    ;SET ERROR CODE
2786  012072  012737  000001  000716          MOV     #1,EXFL
2787  012100  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
2788  012104  032777  000100  166412  3$:     BIT     #100,@ER        ;SEE IF VPE=1
2789  012112  001011                          BNE     4$              ;IF SO: BR
2790  012114  012737  026457  000672          MOV     #TMS28,ERRDD    ;SET ERROR CODE
2791  012122  012737  000001  000716          MOV     #1,EXFL
2792  012130  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
2793  012134  000410                          BR      LT41X
2794  012136  012701  002100          4$:     MOV     #2100,R1        ;SET EXPT ERROR BITS
2795  012142  017702  166356                  MOV     @ER,R2          ;GET ERROR REGISTER
2796  012146  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2797  012150  001402                          BEQ     LT41X           ;IF NOT: BR
2798  012152  004737  014650                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2799  012156  005002          LT41X:  CLR     R2              ;SET TIMER
2800  012160  032777  000200  166334  1$:     BIT     #200,@DS        ;SEE IF DRY SET
2801  012166  001002                          BNE     2$              ;IF SO: BR
2802  012170  005302                          DEC     R2              ;AWAIT DRY
2803  012172  001372                          BNE     1$              ;DELAY
2804  012174  004737  016240          2$:     JSR     PC,ITER         ;GO SEE IF ITERATIONS
2805  012200  004737  015330                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
2806  012204  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2807
2808                                        ;THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT.  EACH
2809                                        ;TEST WILL PERFORM A TAPE WRITE WHILE IN A PARTICULAR MAINTENANCE
2810                                        ;MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.
2811
2812                                        ;LOGIC TEST 42: CYCLIC REDUNDANCY ERROR(CRC)**********
2813
2814  012210  012737  001700  000776  LT42:   MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
2815  012216  004737  014464                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2816  012222  012700  001000                  MOV     #1000,R0
2817  012226  005300                  1$:     DEC     R0
2818  012230  001376                          BNE     1$              ;PAUSE
2819  012232  012737  025153  000622          MOV     #MSLT42,EMADDR
2820  012240  012737  012246  000712          MOV     #LT42IT,SCOLP   ;SET SCOPE ADDRESS
2821  012246  004737  016406          LT42IT: JSR     PC,INIT         ;INIT SELECT DRIVE+SLAVE
2822  012252  012777  177770  166232          MOV     #-10,@WC
2823  012260  012777  177760  166230          MOV     #-20,@FC        ;SET FC=20
2824  012266  012777  026704  166220          MOV     #WDATA,@BA      ;SET BUS ADDRESS
2825  012274  012777  000007  166232          MOV     #7,@MR          ;SET MM CODE
2826  012302  012777  000061  166200          MOV     #61,@C1         ;LOAD WRITE+GO
2827  012310  005000                          CLR     R0
2828  012312  032777  000200  166202  LT42A:  BIT     #200,@DS        ;SEE IF DRY=1
2829  012320  001002                          BNE     LT42B           ;IF SO: BR
2830  012322  005300                          DEC     R0
2831  012324  001372                          BNE     LT42A           ;DELAY
2832  012326  022777  000200  166170  LT42B:  CMP     #200,@ER        ;SEE IF LRC ERROR ONLY
2833  012334  001007                          BNE     LT42B1          ;IF NOT: BR
2834  012336  017702  166166                  MOV     @CC,R2          ;GET CHECK CHAR
2835  012342  042702  177000                  BIC     #177000,R2      ;MASK CRC
2836  012346  022702  000777                  CMP     #777,R2         ;SEE IF SETUP CRC IS CORRECT
2837  012352  001410                          BEQ     LT42B2          ;IF SO: BR
2838  012354  004737  014650          LT42B1: JSR     PC,LTGER3       ;ELSE PRINT ERROR SETUP
2839  012360  012704  022455                  MOV     #MSG55,R4
2840  012364  004737  017320                  JSR     PC,TTOUT        ;PRINT SETUP ERROR MSG
2841  012370  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
2842  012374  004737  016406          LT42B2: JSR     PC,INIT         ;GO INIT
2843  012400  012777  177770  166104          MOV     #-10,@WC        ;SET WC
2844  012406  012777  177760  166102          MOV     #-20,@FC        ;SET FC
2845  012414  012777  026704  166072          MOV     #WDATA,@BA      ;SET BA
2846  012422  012777  000021  166104          MOV     #21,@MR         ;SET MM
2847  012430  012777  000061  166052          MOV     #61,@C1         ;LOAD WRITE+GO
2848  012436  005000                          CLR     R0
2849  012440  032777  000200  166054  LT42C:  BIT     #200,@DS        ;SEE IF DRY
2850  012446  001002                          BNE     LT42D           ;IF SO: BR
2851  012450  005300                          DEC     R0
2852  012452  001372                          BNE     LT42C           ;AWAIT DRY
2853  012454  005777  166044          LT42D:  TST     @ER             ;SEE IF CRC=1
2854  012460  100411                          BMI     LT42E           ;IF SO: BR
2855  012462  012737  026653  000672          MOV     #TMS36,ERADD    ;SET ERROR CODE
2856  012470  012737  000001  000716          MOV     #1,EXFL
2857  012476  004737  014662                  JSR     PC,LTGERO       ;GO PRINT ERROR
2858  012502  000410                          BR      LT42X
2859  012504  012701  100200          LT42E:  MOV     #100200,R1      ;SET EXPT ERROR BITS
2860  012510  017702  166010                  MOV     @ER,R2          ;GET ERROR REGISTER
2861  012514  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2862  012516  001402                          BEQ     LT42X           ;IF NOT: BR
```

```
 2863  012520  004737  014650              JSR     PC,LTGER3       ;ELSE PRINT ERROR
 2864  012524  004737  016240      LT42X:  JSR     PC,ITER         ;DO ITERATIONS
 2865  012530  004737  015330              JSR     PC,DRVCLR
 2866  012534  000137  002344              JMP     TSCD2           ;RETURN TO SCHED
 2867
 2868                                      ;LOGIC TEST 43: LONGITUALINAL REDUNDANCY(LRC)**********
 2869
 2870  012540  012737  001700  000776  LT43:   MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
 2871  012546  004737  014464              JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
 2872  012552  012737  012566  000712      MOV     #LT43IT,SCOLP   ;SET SCOPE ADDRESS
 2873  012560  012737  025207  000622      MOV     #MSLT43,EMADDR
 2874  012566  004737  016406      LT43I1: JSR     PC,INIT         ;INIT, SELECT DRIVE+SLAVE
 2875  012572  005001                      CLR     R1
 2876  012574  005301              1$:     DEC     R1              ;DELAY
 2877  012576  001376                      BNE     1$
 2878  012600  012777  000023  165726      MOV     #23,@MR         ;SET MM
 2879  012606  012777  177770  165676      MOV     #-10,@WC        ;SET WC
 2880  012614  012777  177760  165674      MOV     #-20,@FC        ;SET FC
 2881  012622  012777  026704  165664      MOV     #WDATA,@BA      ;SET BA
 2882  012630  012777  000061  165652      MOV     #61,@CS         ;LOAD WRITE+GO
 2883  012636  005000                      CLR     R0
 2884  012640  032777  000200  165654  LT43C:  BIT     #200,@DS        ;SEE IF DRY
 2885  012646  001002                      BNE     LT43D           ;IF SO: BR
 2886  012650  005300                      DEC     R0
 2887  012652  001372                      BNE     LT43C           ;AWAIT DRY
 2888  012654  032777  000200  165642  LT43D:  BIT     #200,@ER        ;SEE IF LRC=1
 2889  012662  001011                      BNE     LT43E           ;IF SO: BR
 2890  012664  012737  026443  000672      MOV     #TMS26,ERADD    ;SET ERROR CODE
 2891  012672  012737  000001  000716      MOV     #1,EXFL
 2892  012700  004737  014662              JSR     PC,LTGER0       ;GO PRINT
 2893  012704  000425                      BR      LT43X
 2894  012706  017702  165622      LT43E:  MOV     @MR,R2
 2895  012712  042702  000177              BIC     #177,R2         ;MASK LRC
 2896  012716  012701  157600              MOV     #157600,R1      ;SET EXPT LRC
 2897  012722  020102                      CMP     R1,R2           ;SEE IF EXPT = RCVD
 2898  012724  001405                      BEQ     LT43F           ;IF SO: BR
 2899  012726  012737  022442  000672      MOV     #MSG53,ERADD    ;SET ERROR CODE
 2900  012734  004737  015776              JSR     PC,LTGER1       ;PRINT ERROR
 2901  012740  017702  165560      LT43F:  MOV     @ER,R2          ;GET ERROR REGISTER
 2902  012744  012701  000200              MOV     #200,R1         ;SET EXPT ERROR BITS
 2903  012750  020102                      CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
 2904  012752  001402                      BEQ     LT43X           ;IF NOT: BR
 2905  012754  004737  014650              JSR     PC,LTGER3       ;ELSE PRINT ERROR
 2906  012760  004737  016240      LT43X:  JSR     PC,ITER
 2907  012764  004737  015330              JSR     PC,DRVCLR
 2908  012770  000137  002344              JMP     TSCD2           ;RETURN TO SCHED
```

```
2909                                                    ;LOGIC TEST 44: PE CORRECTABLE DATA (CORR)***********
2910
2911  012774  012737  002300  000776  LT44:   MOV     #2300,@DES      ;SET UNIT DESRIPTION = PE
2912  013002  004737  014464                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2913  013006  012737  025243  000622          MOV     #MSLT44,EMADDR  ;SET HEADER
2914  013014  012737  013022  000712          MOV     #LT44IT,SCOLP   ;SET SCOP
2915  013022  004737  016406          LT44IT: JSR     PC,INIT         ;GO INITIALIZE
2916  013026  012777  177600  165456          MOV     #-200,@WC       ;SET WC=200
2917  013034  012777  177400  165454          MOV     #-400,@FC       ;SET FC=400
2918  013042  012777  026704  165444          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
2919  013050  012777  000061  165432          MOV     #61,@CI         ;LOAD WRITE AND GO
2920  013056  005000                          CLR     R0
2921  013060  005777  165432          LT44A:  TST     @FC             ;SEE IF FC=0
2922  013064  001402                          BEQ     LT44A1          ;IF SO:BR
2923  013066  005300                          DEC     R0
2924  013070  001373                          BNE     LT44A           ;AWAIT FC=0
2925  013072  012777  000021  165434  LT44A1: MOV     #21,@MR         ;SET MAINT MODE
2926  013100  005000                          CLR     R0
2927  013102  032777  000200  165412  LT44B:  BIT     #200,@DS        ;SEE IF DRY
2928  013110  001002                          BNE     LT44C           ;IF SO :BR
2929  013112  005300                          DEC     R0
2930  013114  001372                          BNE     LT44B           ;AWAIT DRY
2931  013116  005777  165402          LT44C:  TST     @ER             ;SEE IF CORR=1
2932  013122  100410                          BMI     LT44D           ;IF SO: BR
2933  013124  012737  026644  000672          MOV     #TMS35,ERADD    ;ELSE SET ERROR CODE
2934  013132  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
2935  013140  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
2936  013144  000240          LT44D:  NOP
2937  013146  122777  000002  165354  LT44E:  CMPB    #2,@CC          ;SEE IF DEAD TRACK BIT 1
2938  013154  001414                          BEQ     LT44F           ;IF SO: BR
2939  013156  117702  165346                  MOVB    @CC,R2          ;ELSE SAVE RECVD
2940  013162  042702  177000                  BIC     #177000,R2      ;MASK OUT CRC
2941  013166  112701  000002                  MOVB    #2,R1           ;SAVE EXPT
2942  013172  012737  022051  000672          MOV     #MSG42,ERADD    ;SET ERROR CODE
2943  013200  004737  015776                  JSR     PC,LTGER1       ;GO PRINT ERROR
2944  013204  000410                          BR      LT44X
2945  013206  017702  165312          LT44F:  MOV     @ER,R2          ;GET ERROR REGISTER
2946  013212  012701  100000                  MOV     #100000,R1      ;SET EXPT ERROR BITS
2947  013216  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2948  013220  001402                          BEQ     LT44X           ;IF SO: BR
2949  013222  004737  014650                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2950  013226  004737  016240          LT44X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2951  013232  004737  015330                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
2952  013236  000137  002344          LT44XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2953
2954                                        ;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)***********
2955
2956  013242  012737  002300  000776  LT45:   MOV     #2300,@DES      ;SET UNIT DESCRIPTION = PE
2957  013250  004737  014464                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2958  013254  012737  026421  000622          MOV     #MSLT45,ERADDR
2959  013262  012737  013270  000712          MOV     #LT45IT,SCOLP
2960  013270  004737  016406          LT45IT: JSR     PC,INIT         ;INIT SELECT DRIVE SLAVE
2961  013274  012777  177600  165210          MOV     #-200,@WC       ;SET WC=200
2962  013302  012777  177400  165206          MOV     #-400,@FC       ;SET FC=400
2963  013310  012777  026704  165176          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
2964  013316  012777  000061  165164          MOV     #61,@CI         ;LOAD WRITE+GO
2965  013324  005000                          CLR     R0
2966  013330  005777  165164          LT45E:  TST     @FC             ;AWAIT FC=0
2967  013334  001402                          BEQ     LT45E1
2968  013336  005300                          DEC     R0
2969  013340  001373                          BNE     LT45E           ;AWAIT FC=0
2970  013342  012777  000023  165166  LT45E1: MOV     #23,@MR         ;SET MAINT CODE
2971  013350  005000                          CLR     R0
2972  013352  032777  000200  165144  LT45A:  BIT     #200,@DS        ;SEE IF DRY IS SET
2973  013360  001002                          BNE     LT45B           ;IF SO: BR
2974  013362  005300                          DEC     R0
2975  013364  001372                          BNE     LT45A           ;AWAIT DRY
2976  013366  032777  000100  165132  LT45B:  BIT     #100,@ER        ;SEE IF ERR=1
2977  013374  001010                          BNE     LT45D           ;IF SO:BR
2978  013376  012737  026421  000672          MOV     #TMS23,ERADD    ;SET ERROR CODE
2979  013404  012737  000001  000716          MOV     #1,EXFL
2980  013412  004737  014662                  JSR     PC,LTGER0       ;GO PRINT ERROR
2981  013416  017702  165110          LT45D:  MOV     @CC,R2          ;GET CHECK CHAR
2982  013424  042702  177000                  BIC     #177000,R2      ;MASK CHECK CHAR
2983  013430  012701  000046                  MOV     #46,R1          ;SET EXPT CK
2984  013434  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
2985  013436  001405                          BEQ     LT45F           ;IF SO: BR
2986  013440  012737  022454  000672          MOV     #MSG54,ERADD
2987  013446  004737  015776                  JSR     PC,LTGER1       ;ELSE GO PRINT ERROR
2988  013452  017702  165052          LT45F:  MOV     @ER,R2
2989  013456  042702  120600                  BIC     #120600,R2      ;MASK OPI ,NSG, CORR, AND PEF
2990  013462  012701  000100                  MOV     #100,R1         ;SET EXPT ERROR BITS
2991  013466  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2992  013470  001402                          BEQ     LT45X           ;IF NOT: BR
2993  013472  004737  014650                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2994  013476  004737  016240          LT45X:  JSR     PC,ITER
2995  013502  004737  015330                  JSR     PC,DRVCLR
2996  013506  000137  002344          LT45XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2997
2998                                                ;LOGIC TEST 46: PE FORMAT ERROR(PEF,NSG)**********
2999
3000  013506  012737  002300  000776  LT46:   MOV     #2300,@DES      ;SET UNIT DESCRIPTION = PE
3001  013514  004737  014464          JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
3002  013520  012737  025405  000622  MOV     #MSLT46,ERADDR  ;SET HEADER
3003  013526  012737  013534  000712  MOV     #LT46IT,SCOLP   ;SET SCOPE ADDRESS
3004  013534  004737  016406          JSR     PC,INIT         ;INITIALIZE
3005  013540  012777  177770  164744  MOV     #-10,@WC        ;SET WC=10
3006  013546  012777  177760  164742  MOV     #-20,@FC        ;SET FC=20
3007  013554  012777  026704  164732  MOV     #DATA,@BA       ;SET BA=START OF WRITE BUFFER
3008  013562  012777  000061  164720  MOV     #61,@CS         ;LOAD WRITE+GO
3009  013570  005777  164722  LT46A:  TST     @FC
3010  013574  001375          BNE     LT46A           ;AWAIT FC=0
3011  013576  032777  000100  164730  1$:     BIT     #100,@MR
3012  013604  001774          BEQ     1$              ;DELAY
3013  013606  032777  000100  164720  2$:     BIT     #100,@MR
3014  013614  001374          BNE     2$
3015  013616  032777  000100  164710  3$:     BIT     #100,@MR
3016  013624  001774          BEQ     3$
3017  013626  012777  000027  164700          MOV     #27,@MR         ;SET MM CODE TO KILL PEF
3018  013634  005000          CLR     R0              ;INIT TIMING LOOP
3019  013636  032777  000200  164656  LT46B:  BIT     #200,@DS        ;SEE IF DRY SET
3020  013644  001002          BNE     LT46C           ;IF SO: BR
3021  013646  005300          DEC     R0
3022  013650  001372          BNE     LT46B           ;AWAIT DRY
3023  013652  032777  000200  164644  LT46C:  BIT     #200,@ER        ;SEE IF PEF SET
3024  013660  001011          BNE     LT46D           ;IF SO: BR
3025  013662  012737  026435  000672          MOV     #TMS25,ERADD    ;SET ERROR TAG
3026  013670  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
3027  013676  004737  014662          JSR     PC,LTGER0       ;GO PRINT ERROR
3028  013702  000412          BR      LT46X
3029  013704  017702  164614  LT46D:  MOV     @ER,R2          ;GET ERROR REGISTER
3030  013710  042702  020000          BIC     #20000,R2       ;CLEAR OPI BIT (MAY OR MAY NOT SET)
3031  013714  012701  000600          MOV     #600,R1         ;SET EXPT ERROR BITS
3032  013720  020102          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
3033  013722  001402          BEQ     LT46X           ;IF NOT: BR
3034  013724  004737  014650          JSR     PC,LTGER3       ;ELSE PRINT ERROR
3035  013730  004737  016240  LT46X:  JSR     PC,ITER
3036  013734  004737  015330          JSR     PC,DRVCLR
3037  013740  000137  002344  LT46XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
3038                                                    ;LOGIC TEST 47: FRAME COUNT OVERFLOW(M8905)*******
3039
3040  013744  012737  025441  000622  LT47:   MOV     #MSLT47,EMADDR  ;SET TEST HEADER
3041  013752  012737  013760  000712          MOV     #LT47IT,SCOLP   ;SET SCOPE ADDRESS
3042  013760  004737  016356          LT47IT: JSR     PC,INIT3        ;GO INIT
3043  013764  012777  177770  164520          MOV     #-10,@WC        ;SET WC = 10
3044  013772  012777  177760  164516          MOV     #-20,@FC        ;SET FC = 20
3045  014000  052777  001700  164534          BIS     #1700,@TC       ;SET TO NRZ, NORMAL, 000
3046  014006  012777  026704  164500          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3047  014014  012777  000013  164512          MOV     #13,@MR         ;SET WRAP 2
3048  014022  012777  000061  164460          MOV     #61,@C1         ;LOAD WRITE+GO
3049  014030  012700  040000                  MOV     #40000,R0
3050  014034  005777  164502          LT47A:  TST     @TC             ;SEE IF ALPHA
3051  014040  100002                          BPL     LT47B           ;IF SO: BR
3052  014042  005300                          DEC     R0
3053  014044  001373                          BNE     LT47A           ;AWAIT ALPHA
3054  014046  012700  000020          LT47B:  MOV     #20,R0          ;SET CLK CNT
3055  014052  052777  000040  164454  LT47C:  BIS     #40,@MR
3056  014060  042777  000040  164446          BIC     #40,@MR         ;CLOCK MR
3057  014066  005300                          DEC     R0
3058  014070  001370                          BNE     LT47C           ;IF NOT DONE ALL: BR
3059  014072  017702  164420                  MOV     @FC,R2
3060  014076  005001                          CLR     R1              ;SET TEST WORD
3061  014100  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
3062  014102  001410                          BEQ     LT47X           ;IF SO: BR
3063  014104  012737  021324  000672          MOV     #MSG19,ERADD    ;SET ERROR CODE
3064  014112  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
3065  014120  004737  015776                  JSR     PC,LTGER1       ;GO PRINT ERROR
3066  014124  004737  016240          LT47X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
3067  014130  000137  002344                  JMP     TSCD2           ;RETURN TO SCHEDULAR
3068
```

```
3069
3070                                                ;LOGIC TEST 50: NEF WHEN WRITING PE ON NRZ SELECTED SLAVE
3071
3072  014134  012737  025506  000622  LT50:   MOV     #MSLT50,EMADDR  ;SET TEST HEADER
3073  014142  012737  014150  000712          MOV     #LT50IT,SCOLP
3074  014150  004737  016356          LT50IT: JSR     PC,INIT3        ;SET SLAVE = NRZ
3075  014154  042777  003400  164360          BIC     #3400,@TC       ;CLEAR DENSITY BITS
3076  014162  052777  002300  164352          BIS     #2300,@TC       ;SET DENSITY = PE
3077  014170  012777  177770  164314          MOV     #-10,@WC        ;SET WORD COUNT
3078  014176  012777  177760  164312          MOV     #-20,@FC        ;SET FRAME COUNT
3079  014204  012777  026704  164302          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3080  014212  012777  000013  164314          MOV     #13,@MR         ;SET WRAP 2
3081  014220  012777  000061  164262          MOV     #61,@C1         ;LOAD WRITE COMMAND
3082  014226  000240                          NOP
3083  014230  000240                          NOP
3084  014232  000240                          NOP
3085  014234  012701  004000                  MOV     #4000,R1        ;SET EXPECTED RESULT
3086  014240  017702  164260                  MOV     @ER,R2          ;GET ERROR REGISTER
3087  014244  030102                          BIT     R1,R2           ;BRANCH IF NEF BIT SET
3088  014246  001006                          BNE     1$
3089  014250  012737  000001  000716          MOV     #1,EXFL         ;SET EXPECTED FLAG
3090  014256  004737  014662                  JSR     PC,LTGER0       ;PRINT ERROR
3091  014262  000404                          BR      LT50X
3092  014264  020102                  1$:     CMP     R1,R2           ;BRANCH IF NO UNEXPECTED ERROR
3093  014266  001402                          BEQ     LT50X           ;BITS WERE SET
3094  014270  004737  014650                  JSR     PC,LTGER3       ;PRINT ERROR MSG
3095  014274  004737  016240          LT50X:  JSR     PC,ITER         ;ITERATE TEST
3096  014300  004737  015330                  JSR     PC,DRVCLR       ;RESET DRIVE
3097  014304  000137  002344                  JMP     TSC02
3098                                          ;LOGIC TEST 51: NEF WHEN WRITING NRZ ON PE SELECTED SLAVE
3099
3100  014310  012737  025566  000622  LT51:   MOV     #MSLT51,EMADDR  ;SET ERROR MSG HEADER
3101  014316  012737  014324  000712          MOV     #LT51IT,SCOLP   ;SET SCOPE LOOP ADDRESS
3102  014324  004737  016372          LT51IT: JSR     PC,INIT4        ;SET SLAVE = PE
3103  014330  042777  002300  164204          BIC     #2300,@TC       ;CLEAR DENSITY BITS
3104  014336  052777  001300  164176          BIS     #1300,@TC       ;SET DENSITY = NRZ
3105  014344  012777  177770  164140          MOV     #-10,@WC        ;SET WORD COUNT
3106  014352  012777  177760  164136          MOV     #-20,@FC        ;SET FRAME COUNT
3107  014360  012777  026704  164126          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3108  014366  012777  000013  164140          MOV     #13,@MR         ;SET WRAP 2
3109  014374  012777  000061  164106          MOV     #61,@C1         ;SET WRITE COMMAND AND GO
3110  014402  000240                          NOP
3111  014404  000240                          NOP
3112  014406  000240                          NOP
3113  014410  012701  004000                  MOV     #4000,R1        ;SET EXPECTED RESULT
3114  014414  017702  164104                  MOV     @ER,R2          ;GET ERROR REGISTER
3115  014420  030102                          BIT     R1,R2           ;BRANCH IF NEF SET
3116  014422  001006                          BNE     1$
3117  014424  012737  000001  000716          MOV     #1,EXFL         ;SET EXPECTED FLAG
3118  014432  004737  014662                  JSR     PC,LTGER0       ;PRINT ERROR MSG
3119  014436  000404                          BR      LT51X
3120  014440  020102                  1$:     CMP     R1,R2           ;BRANCH IF NO UNEXPECTED
3121  014442  001402                          BEQ     LT51X           ;ERROR BITS WERE SET
3122  014444  004737  014650                  JSR     PC,LTGER3
3123  014450  004737  016240          LT51X:  JSR     PC,ITER         ;ITERATE TEST
3124  014454  004737  015330                  JSR     PC,DRVCLR       ;CLEAR DRIVE
```

```
  3125  014460  000137  002344                JMP     TSCD2           ;RETURN TO SCHEDULER
```

```
3126                                            ;STATIC TESTS ONLY SUBROUTINE*********
3127
3128  014464  005737  000740          STATIC:   TST     STFLG           ;SEE IF SINGLE TEST ONLY
3129  014470  001006                            BNE     1$              ;IF SO: BR
3130  014472  005737  001012                    TST     STATC           ;SEE IF STATIC ONLY
3131  014476  001403                            BEQ     1$              ;IF NOT: BR
3132  014500  005726                            TST     (SP)+           ;RESET STACK
3133  014502  000137  002344                    JMP     TSCD2           ;RETURN TO SCHEDULAR
3134  014506  005037  001006          1$:       CLR     RDRVF
3135  014512  000207                            RTS     PC              ;RETURN TO TEST
3136
```

```
3137
3138                                        ;END OF RECORD FORCE SUBROUTINE*****
3139
3140  014514  017700  164014        EORPA:  MOV     @MR,R0          ;GET MAINT REG
3141  014520  042700  000036                BIC     #36,R0          ;CLEAR CURRENT OP CODE
3142  014524  052700  000024                BIS     #24,R0          ;SET EOR CLEAR OP CODE
3143  014530  010077  164000                MOV     R0,@MR          ;DO EOR
3144  014534  042777  000037  163772        BIC     #37,@MR         ;CLEAR EOR AND MM
3145  014542  005000                        CLR     R0
3146  014544  012701  000002                MOV     #2,R1
3147  014550  032777  000001  163732 EORP1: BIT     #1,@C1          ;SEE IF GO GONE
3148  014556  001430                        BEQ     EORP2           ;IF SO: BR
3149  014560  005300                        DEC     R0
3150  014562  001372                        BNE     EORP1           ;AWAIT GO RESET
3151  014564  005301                        DEC     R1
3152  014566  001370                        BNE     EORP1
3153  014570  032777  020000  163772        BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3154  014576  001020                        BNE     EORP2           ;IF SO: BR
3155  014600  005737  000620                TST     HDRFL           ;SEE IF DONE HEADER
3156  014604  001004                        BNE     EORP1A          ;IF SO: BR
3157  014606  013704  000622                MOV     EMADDR,R4
3158  014612  004737  017320                JSR     PC,TTOUT        ;PRINT HEADER
3159  014616  012704  027317        EORP1A: MOV     #MSG31,R4
3160  014622  004737  017320                JSR     PC,TTOUT        ;PRINT EOR GO BIT ERROR
3161  014626  032777  100000  163734        BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
3162  014634  001401                        BEQ     EORP2           ;IF NOT: BR
3163  014636  000000                        HALT
3164  014640  000240                EORP2:  NOP
3165  014642  005037  000676        EORPX:  CLR     TEMP2           ;CLEAR FLAG
3166  014646  000207                        RTS     PC              ;RETURN
3167
```

```
                                                ;LOGIC TEST ADDRESSING ERROR SUBROUTINE**********
                                        LTGER3: CLR     EXFL
                                                MOV     #MSG51,ERADD
                                        LTGER0: MOV     #1,ADDFL        ;SET NO ADDRESS FLAG
                                        LTGER:  NOP
                                                CLR     PFLG            ;CLEAR PRINT FLAG
                                                BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
                                                BNE     LTGX            ;IF NOT: BR
                                        LTGA:   TST     HDRFL           ;SEE IF PRINTED HEADER
                                                BNE     LTGA1           ;IF SO: BR
                                                MOV     EMADDR,R4
                                                JSR     PC,TTOUT        ;PRINT TEST HEADER
                                        LTGA1:  MOV     #1,HDRFL        ;SET HEADER FLAG
                                                MOV     ERADD,R4
                                                JSR     PC,TTOUT        ;PRINT CONDITION ERROR
                                                TST     ADDFL
                                                BNE     LTGA2
                                                MOV     R1,R3
                                                JSR     PC,OCTP         ;PRINT ADDRESS
                                        LTGA2:  TST     EXFL
                                                BEQ     LTGC            ;IF NO STATUS: BR
                                                MOV     #MSG6,R4
                                                CMP     #1,EXFL         ;EXPT-NOT RCVD
                                                BEQ     LTGB
                                                MOV     #MSG7,R4        ;RCVD-NOT EXPT
                                        LTGB:   JSR     PC,TTOUT        ;PRINT STATUS
                                        LTGC:   INC     PFLG
                                                TST     ADDFL           ;SEE IF ADD TST
                                                BEQ     LTGD            ;IF SO: BR
                                                TST     T24FL           ;SEE IF TEST 24
                                                BEQ     LTGC0           ;IF NOT: BR
                                                MOV     #MSG27,R4
                                                JSR     PC,TTOUT        ;PRINT DATA TAG
                                                MOV     #MSG12,R4
                                                JSR     PC,TTOUT        ;PRINT EXPT TAG
                                                MOV     #-1,R3
                                                JSR     PC,OCTPE        ;PRINT EXPT
                                                MOV     #MSG13,R4
                                                JSR     PC,TTOUT        ;PRINT RCVD TAG
                                                MOV     R1,R3           ;GET RCVD
                                                JSR     PC,OCTPE        ;PRINT RCVD
                                        LTGC0:  JSR     PC,REGP         ;PRINT REGISTERS
                                        LTGD:   BIT     #4000,@SWR
                                                BNE     LTGX
                                                MOV     #MSG16,R4
                                                JSR     PC,TTOUT
                                                MOV     ITCNT,R3        ;PRINT ITERATION
                                                JSR     PC,OCTP
                                        LTGX:   TST     @SWR
                                                BPL     LTGXA           ;IF NOT STOP ON ERROR: BR
                                                HALT
                                        LTGXA:  TST     PFLG
                                                BNE     LTGXX           ;IF PRINTED: BR
                                                BIT     #20000,@SWR
                                                BEQ     LTGA
```

```
  3224  015160  005037  000746        LTGXX:  CLR     ADDFL           ;CLEAR ADDRESS FLAG
  3225  015164  005037  000716                CLR     EXFL
  3226  015170  000137  016210                JMP     SCOPE
  3227
  3228                                ;SUBROUTINE TO PRINT MAJOR REGISTERS**********
  3229
  3230  015174  000240                REGP:   NOP
  3231  015176  012704  022236                MOV     #MSG46,R4
  3232  015202  004737  017320                JSR     PC,TTOUT        ;PRINT REGISTER HEADER
  3233  015206  017703  163276                MOV     @C1,R3
  3234  015212  004737  017436                JSR     PC,OCTPE
  3235  015216  017703  163270                MOV     @4C,R3
  3236  015222  004737  017436                JSR     PC,OCTPE
  3237  015226  017703  163262                MOV     @8A,R3
  3238  015232  004737  017436                JSR     PC,OCTPE
  3239  015236  017703  163254                MOV     @FC,R3
  3240  015242  004737  017436                JSR     PC,OCTPE
  3241  015246  017703  163246                MOV     @CS,R3
  3242  015252  004737  017436                JSR     PC,OCTPE
  3243  015256  017703  163240                MOV     @DS,R3
  3244  015262  004737  017436                JSR     PC,OCTPE                ;PRINT REGISTERS
  3245  015266  017703  163232                MOV     @ER,R3
  3246  015272  004737  017436                JSR     PC,OCTPE
  3247  015276  017703  163224                MOV     @AS,R3
  3248  015302  004737  017436                JSR     PC,OCTPE
  3249  015306  017703  163222                MOV     @MR,R3
  3250  015312  004737  017436                JSR     PC,OCTPE
  3251  015316  017703  163220                MOV     @TC,R3
  3252  015322  004737  017436                JSR     PC,OCTPE
  3253  015326  000207                        RTS     PC
  3254
  3255
```

```
                                                ;DRIVE CLEAR SUBROUTINE*********
                                        DRVCLR: NOP
015330  000240                                  MOV     #40000,R4
015332  012704  040000                          DEC     R4
015336  005304                          DCD:    BNE     DCD             ;DELAY
015340  001376                                  CLR     PFLG
015342  005037  000666                          JSR     PC,ATTN         ;GO SEE OF ATTN SET
015346  004737  015564                          MOV     #11,@C1         ;ISSUE DRIVE CLEAR
015352  012777  000011  163130                  CLR     R0
015360  005000                          DCA:    BIT     #200,@DS        ;SEE IF DRY
015362  032777  000200  163132                  BNE     DCA0
015370  001002                                  DEC     R0
015372  005300                                  BNE     DCA             ;WAIT FOR DRY
015374  001372                          DCA0:   BIT     #40000,@DS      ;SEE IF ERR RESET
015376  032777  040000  163116                  BNE     DCE             ;IF NOT: BR
015404  001022                                  TST     @ER             ;SEE IF ERROR REGISTER RESET
015406  005777  163112                          BNE     DCE             ;IF NOT: BR
015412  001017                                  TST     @DS             ;SEE IF ATA RESET
015414  005777  163102                          BMI     DCE             ;IF NOT: BR
015420  100414                                  MOV     #1,R3           ;SET TEST BIT
015422  012703  000001                          MOV     DRVN,R4         ;GET DRIVE NUMBER & BRANCH
015426  013704  000624                          BEQ     DCC             ;IF DRIVE 0
015432  001403                          DCB:    ASL     R3              ;POSITION TEST BIT PER DRIVE NUMBER
015434  006303                                  DEC     R4              ;SEE IF DONE
015436  005304                                  BNE     DCB             ;IF NOT: BR
015440  001375                          DCC:    BIT     R3,@AS          ;SEE IF ATTEN IS RESET
015442  030377  163060                          BNE     DCE             ;IF NOT: BR
015446  001001                                  RTS     PC              ;RETURN
015450  000207

015452  000240                          DCE:    NOP
015454  032777  020000  163106                  BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
015462  001017                                  BNE     DCEX            ;IF SO: BR
015464  005737  000620                          TST     HDRFL           ;SEE IF PRINT HEADER
015470  001004                                  BNE     DCEA            ;IF NOT: BR
015472  013704  000622                          MOV     EMADDR,R4
015476  004737  017320                          JSR     PC,TTOUT        ;PRINT HEADER
015502  012704  022342                  DCEA:   MOV     #MSG47,R4
015506  004737  017320                          JSR     PC,TTOUT        ;PRINT DRIVE CLEAR ERROR
015512  004737  015174                          JSR     PC,REGP         ;PRINT REGISTERS
015516  005237  000666                          INC     PFLG            ;SET PRINTED FLAG
015522  005777  163042                  DCEX:   TST     @SWR            ;SEE IF HALT ON ERROR
015526  100001                                  BPL     DCEXA           ;IF NOT: BR
015530  000000                                  HALT
015532  005737  000666                  DCEXA:  TST     PFLG            ;SEE IF HAVE PRINTED
015536  001004                                  BNE     DCEXX           ;IF SO: BR
015540  032777  020000  163022                  BIT     #20000,@SWR     ;BRANCH IF ERROR
015546  001741                                  BEQ     DCE             ;PRINTOUT DESIRED
015550  000240                          DCEXX:  NOP
015552  012737  015330  000712                  MOV     #DRVCLR,SCOLP   ;SET SCOPE LOOP ADDRESS
015560  000137  016210                          JMP     SCOPE           ;GO DO SCOPE LOOP
```

```
                                                ;COMPOSITE ERROR CHECK SUBROUTINE**********
                                        ATTN:   NOP
015564  000240
015566  005777  162730                          TST     @DS             ;SEE IF ATA SET
015572  001004                                  BNE     ATTA            ;IF SO: BR
015574  012737  021700  000700                  MOV     #MSG32,TEMP3
015602  000427                                  BR      ATTP            ;ELSE PRINT ERROR
015604  032777  040000  162710        ATTA:     BIT     #40000,@DS      ;SEE IF COMPOSITE ERROR SET
015612  001004                                  BNE     ATTB            ;IF SO: BR
015614  012737  021662  000700                  MOV     #MSG31,TEMP3
015622  000417                                  BR      ATTP            ;ELSE PRINT ERROR
015624  012703  000001                ATTB:     MOV     #1,R3           ;SET TEST BIT
015630  012737  021716  000700                  MOV     #MSG33,TEMP3
015636  013704  000624                          MOV     DRVN,R4         ;GET DRIVE NUMBER & BRANCH
015642  001403                                  BEQ     ATTD            ;IF DRIVE 0
015644  006303                ATTC:     ASL     R3                      ;POSITION TEST BIT
015646  005304                                  DEC     R4              ;SEE IF DONE
015650  001375                                  BNE     ATTC            ;IF NOT: BR
015652  030377  162650                ATTD:     BIT     R3,@AS          ;SEE IF ATTEN SUMMARY SET
015656  001401                                  BEQ     ATTP            ;IF NOT: BR
015660  000207                                  RTS     PC              ;ELSE RETURN
015662  032777  020000  162700        ATTP:     BIT     #20000,@SWR     ;SEE IF PRINT INHIBIT
015670  001021                                  BNE     ATTX            ;IF SO: BR
015672  005737  000620                          TST     HDRFL           ;SEE IF DONE HEADER
015676  001004                                  BNE     ATTPA           ;IF SO: BR
015700  013704  000622                          MOV     EMADDR,R4
015704  004737  017320                          JSR     PC,TTOUT        ;PRINT HEADER
015710  013704  000700                ATTPA:    MOV     TEMP3,R4
015714  004737  017320                          JSR     PC,TTOUT        ;PRINT ERROR TYPE
015720  004737  015174                          JSR     PC,REGP         ;PRINT REGISTERS
015724  005237  000666                          INC     PFLG            ;SET PRINT FLAG
015730  005237  000620                          INC     HDRFL           ;SET HEADER FLAG
015734  005777  162630                ATTX:     TST     @SWR            ;SEE IF HALT ON ERROR
015740  100001                                  BPL     ATTXA           ;IF NOT: BR
015742  000000                                  HALT
015744  005737  000666                ATTXA:    TST     PFLG            ;SEE IF DONE PRINT
015750  001004                                  BNE     ATTXX           ;IF SO: BR
015752  032777  020000  162610                  BIT     #20000,@SWR     ;BRANCH IF NO ERROR
015760  001740                                  BEQ     ATTP            ;PRINTOUT DESIRED
015762  005037  000666                ATTXX:    CLR     PFLG            ;CLEAR PRINT FLAG
015766  000207                                  RTS     PC              ;RETURN
```

```
                                                        ;LOGIC TEST REGISTER BIT ERRuR SUBROUTIINE**********
                                                        
015770  012737  000001  000736  LTGER2: MOV     #1,PEXFL        ;SET FLAG
015776  000240                  LTGER1: NOP
016000  005037  000666                  CLR     PFLG            ;CLEAR PRINT FLAG
016004  032777  020000  162556          BIT     #20000,@SWR     ;BRANCH IF ERROR
016012  001055                          BNE     LTG1X           ;PRINTOUT DESIRED
016014  005737  000620          LTG1A:  TST     HDRFL           ;SEE IF PRINT HEADER
016020  001004                          BNE     LTG1B           ;IF NOT: BR
016022  013704  000622                  MOV     EMADDR,R4
016026  004737  017320                  JSR     PC,TTOUT        ;PRINT HEADER
016032  012737  000001  000620  LTG1B:  MOV     #1,HDRFL        ;SET FLAG
016040  013704  000672                  MOV     ERADD,R4
016044  004737  017320                  JSR     PC,TTOUT        ;PRINT ERROR CODE
016050  005737  000736                  TST     PEXFL           ;SEE IF PRINT EXPT-RCVD
016054  001016                          BNE     LTG1T           ;IF NOT: BR
016056  012704  021224                  MOV     #MSG12,R4
016062  004737  017320                  JSR     PC,TTOUT        ;PRINT EXPT TAG
016066  010103                          MOV     R1,R3
016070  004737  017446                  JSR     PC,OCTP         ;PRINT EXPT
016074  012704  021233                  MOV     #MSG13,R4
016100  004737  017320                  JSR     PC,TTOUT        ;PRINT RCVD TAG
016104  010203                          MOV     R2,R3
016106  004737  017446                  JSR     PC,OCTP         ;PRINT RCVD
016112  032777  004000  162450  LTG1T:  BIT     #4000,@SWR
016120  001010                          BNE     LTG1C
016122  012704  021275                  MOV     #MSG16,R4
016126  004737  017320                  JSR     PC,TTOUT
016132  013703  000702                  MOV     ITCNT,R3
016136  004737  017446                  JSR     PC,OCTP         ;PRINT ITERATION
016142  005237  000666          LTG1C:  INC     PFLG
016146  000240                  LTG1X:  NOP
016150  005777  162414                  TST     @SWR
016154  100001                          BPL     LTG1X1          ;IF NOT STOP ON ERROR: BR
016156  000000                          HALT
016160  005737  000666          LTG1X1: TST     PFLG
016164  001004                          BNE     LTG1XX          ;IF HAVE PRINTED: BR
016166  032777  020000  162374          BIT     #20000,@SWR
016174  001707                          BEQ     LTG1A
016176  000240                  LTG1XX: NOP
016200  005037  000736                  CLR     PEXFL           ;CLEAR EXPT-RCVD FLAG
016204  000137  016210                  JMP     SCOPE           ;GO TO SCOPE

                                                        ;SCOPE LOOP ON ERROR SUBROUTINE**********

016210  000240                  SCOPE:  NOP
016212  032777  040000  162350          BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
016220  001001                          BNE     1$              ;IF SO: BR
016222  000207                          RTS     PC              ;ELSE EXIT
016224  000240                  1$:     NOP
016226  005726                          TST     (SP)+           ;RESET STACK
016230  000240                          NOP
016232  000240                          NOP
016234  000177  162452                  JMP     @SCOLP          ;LOOP ON ERROR
```

```
3404                                                    ;TEST ITERATION SUBROUTINE**********
3405
3406  016240  032777  004000  162322  ITER:   BIT     #4000,@SWR      ;SEE IF ITERATIONS
3407  016246  001403                  BEQ     2$              ;IF SO: BR
3408  016250  005037  000702          1$:     CLR     ITCNT           ;CLEAR ITERATION COUNTER
3409  016254  000207                  RTS     PC              ;ELSE EXIT
3410  016256  005737  001016          2$:     TST     PCNTR           ;NO SUBTEST ITERATIONS ON FIRST PASS
3411  016262  001772                  BEQ     1$
3412  016264  005237  000702                  INC     ITCNT           ;BUMP COUNTER
3413  016270  023737  000702  000606          CMP     ITCNT,ITAMT     ;SEE IF DONE ALL
3414  016276  001764                  BEQ     1$              ;IF SO: BR
3415  016300  005726                  TST     (SP)+           ;RESET STACK
3416  016302  017700  162406          MOV     @ITRLP,R0       ;SET ITERATION POINTER
3417  016306  000110                  JMP     (R0)            ;GO ITERATE
3418
3419                                                    ;MANUAL INTERVENTION INHIBIT*********
3420
3421  016310  000240          INMT:   NOP
3422  016312  012704  022066          MOV     #MSG43,R4
3423  016316  004737  017320          JSR     PC,TTOUT        ;GO PRINT INHIB MSG
3424  016322  000000                  HALT
3425  016324  000137  002344          JMP     TSCD2           ;RETURN TO SCHED
3426
```

```
3427
3428                                        ;INITIALIZE SUBROUTINE**********
3429
3430  016330  000240                INIT1:  NOP
3431  016332  012777  000040  162160        MOV     #40,@CS         ;INIT
3432  016340  013777  000624  162152  INIT2: MOV     DRVN,@CS        ;SELECT DRIVE
3433  016346  013777  000664  162166        MOV     SLVN,@TC        ;SELECT SLAVE
3434  016354  000207                        RTS     PC              ;RETURN
3435
3436                                ;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
3437                                ;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
3438                                ;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
3439                                ;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.
3440
3441                                ;SET SLAVE IN NRZ OFF BOT
3442  016356  013746  000776        INIT3:  MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESCRIPTION
3443  016362  012737  001400  000776        MOV     #1400,UDES      ;SET UNIT DESCRIPTION = NRZ
3444  016370  000410                        BR      INIT5           ;GO TO INIT5 ROUTINE
3445
3446                                ;SET SLAVE IN PE OFF BOT
3447  016372  013746  000776        INIT4:  MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESRIPTION
3448  016376  012737  002000  000776        MOV     #2000,UDES      ;SET UNIT DESRIPTION = PE
3449  016404  000402                        BR      INIT5           ;GO DO IT
3450
3451                                ;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
3452                                ;IT IS ENTERED AT INIT5 WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
3453  016406  013746  000776        INIT:   MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESRIPTION
3454  016412  012777  000040  162100  INIT5: MOV     #40,@CS         ;INIT CONTROLLER
3455  016420  013777  000624  162072        MOV     DRVN,@CS        ;SELECT TM03 DRIVE
3456  016426  013777  000664  162106        MOV     SLVN,@TC        ;SELECT TE16 SLAVE
3457  016434  013746  000776                MOV     UDES,-(SP)      ;GET SLAVE DESCRIPTION
3458  016440  042716  174377                BIC     #174377,(SP)    ;CLEAR ALL BUT DENSITY SELECT BITS
3459  016444  022726  001400                CMP     #1400,(SP)+     ;BRANCH IF REQUESTING PE MODE
3460  016450  001005                        BNE     1$
3461  016452  032777  000040  162042        BIT     #40,@DS         ;BRANCH IF SLAVE IS IN NRZ MODE
3462  016460  001420                        BEQ     4$              ;(PES = 0)
3463  016462  000404                        BR      2$
3464  016464  032777  000040  162030  1$:   BIT     #40,@DS         ;BRANCH IF SLAVE IS IN PE MODE
3465  016472  001013                        BNE     4$
3466  016474  012777  000007  162006  2$:   MOV     #7,@C1          ;REWIND SLAVE
3467  016502  032777  000200  162012  20$:  BIT     #200,@DS        ;WAIT FOR READY
3468  016510  001774                        BEQ     20$
3469  016512  032777  020000  162002  3$:   BIT     #20000,@DS      ;WAIT UNTIL PIP CLEARS
3470  016520  001374                        BNE     3$
3471  016522  053777  000776  162012  4$:   BIS     UDES,@TC        ;LOAD SLAVE DESRIPTION
3472  016530  032777  000002  161764        BIT     #2,@DS          ;BRANCH IF NOT AT BOT
3473  016536  001407                        BEQ     6$
3474  016540  012777  000025  161742        MOV     #25,@C1         ;ERASE TO GET OFF BOT
3475  016546  032777  000200  161746  5$:   BIT     #200,@DS        ;WAIT FOR READY
3476  016554  001774                        BEQ     5$
3477  016556  012777  000011  161724  6$:   MOV     #11,@C1         ;RESET DRIVE
3478  016564  012637  000776                MOV     (SP)+,UDES      ;RESTORE UNIT DESRIPTION
3479  016570  000207                        RTS     PC              ;RETURN
3480
3481
3482
```

```
3483                                                    ;MANUAL INSTRUCTION SUBROUTINE**********
3484
3485  016572  000240                          INST:     NOP
3486  016574  004737  017320                           JSR     PC,TTOUT        ;PRINT INSTRUCTION
3487  016600  012704  025646                           MOV     #MMSG0,R4
3488  016604  004737  017320                           JSR     PC,TTOUT        ;PRINT REPLY
3489  016610  012705  000700                           MOV     #TEMP3,R5
3490  016614  012701  000001                           MOV     #1,R1
3491  016620  012702  177777                           MOV     #-1,R2
3492  016624  012703  000000                           MOV     #0,R3
3493  016630  004737  016776                           JSR     PC,TTR          ;AWAIT REPLY
3494  016634  000240                                   NOP
3495  016636  000207                                   RTS     PC              ;EXIT
3496
3497                                                    ;MAG TAPE INTERRUPT HANDLER**********
3498
3499  016640  000240                          MTINT:    NOP
3500  016642  013716  000670                           MOV     RTRN,(SP)       ;SET RETURN FROM INTERUPT ADDRESS
3501  016646  000002                                   RTI                     ;RETURN
3502
3503                                                    ;TTY INTERRUPT HANDLER**********
3504
3505  016650  017746  161720                TTINT:    MOV     @TKB,-(SP)      ;GET CHARACTER
3506  016654  042716  000200                           BIC     #200,(SP)       ;CLEAR PARITY BIT
3507  016660  122716  000003                           CMPB    #3,(SP)         ;BRANCH IF NOT CONTROL C
3508  016664  001010                                   BNE     1$
3509  016666  005737  001416                           TST     CHNFLG          ;INHIBIT ↑C IF IN CHAIN MODE
3510  016672  001005                                   BNE     1$
3511  016674  005077  161666                           CLR     @PSW            ;CLEAR PSW
3512  016700  000005                                   RESET
3513  016702  000137  000200                           JMP     @#200           ;RESTART
3514  016706  122716  000001                1$:       CMPB    #1,(SP)         ;BRANCH IF NOT ↑A
3515  016712  001017                                   BNE     2$
3516  016714  022737  000176  000570                   CMP     #SWREG,SWR      ;BRANCH IF USING HARWARE SWR
3517  016722  001016                                   BNE     3$
3518  016724  012737  177570  000570                   MOV     #177570,SWR     ;INVOKE HARDWARE SWR
3519  016732  004737  020172                           JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3520  016736  012704  022670                           MOV     #MSG63,R4       ;TYPE 'HARDWARE SWR IN USE'
3521  016742  004737  017320                           JSR     PC,TTOUT
3522  016746  004737  020214                           JSR     PC,.RESTORE
3523  016752  122716  000007                2$:       CMPB    #7,(SP)         ;BRANCH IF NOT ↑G
3524  016756  001005                                   BNE     4$
3525  016760  012737  000176  000570        3$:       MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
3526  016766  004737  020074                           JSR     PC,GTSWR        ;GET SWITCHES
3527  016772  005726                        4$:       TST     (SP)+           ;POP CHARACTER OFF STACK
3528  016774  000002                                   RTI                     ;RETURN
3529
```

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

016776  010146                  TTR:    MOV     R1,-(SP)        ;SAVE CHARACTER COUNT
017000  011601                  10$:    MOV     (SP),R1         ;RESTORE CHARACTER COUNT (FOR ↑U)
017002  005037  000674                  CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
017006  005000                          CLR     R0
017010  004737  017256          1$:     JSR     PC,TTIN         ;GO READ CHARACTER
017014  122737  000003  000616          CMPB    #3,TIB          ;BRANCH IF NOT ↑C
017022  001003                          BNE     11$
017024  000005                          RESET                   ;RESET
017026  000137  000200                  JMP     @#200           ;RESTART PROGRAM
017032  122737  000015  000616  11$:     CMPB    #15,TIB         ;SEE IF CR
017040  001004                          BNE     2$              ;IF NOT: BR
017042  005737  000674                  TST     TEMP1           ;SEE IF FIRST CHARACTER
017046  001471                          BEQ     9$              ;IF SO: BR
017050  000457                          BR      6$
017052  122737  000025  000616  2$:     CMPB    #25,TIB         ;BRANCH IF NOT CONTROL U
017060  001005                          BNE     21$
017062  012704  022616                  MOV     #MSG59,R4       ;TYPE <CR><LF>
017066  004737  017320                  JSR     PC,TTOUT
017072  000742                          BR      10$
017074  122737  000177  000616  21$:    CMPB    #177,TIB        ;BRANCH IF NOT 'RUBOUT'
017102  001012                          BNE     3$
017104  000241                          CLC
017106  006000                          ROR     R0              ;REMOVE LAST CHAR
017110  006200                          ASR     R0
017112  006200                          ASR     R0
017114  012704  022620                  MOV     #MSG60,R4       ;TYPE '\'
017120  004737  017320                  JSR     PC,TTOUT
017124  005201                          INC     R1              ;DECREMENT CHARS RECEIVED COUNT
017126  000730                          BR      1$
017130  122737  000060  000616  3$:     CMPB    #60,TIB         ;SEE IF CHAR IS LESS THAN 0
017136  101402                          BLOS    4$              ;IF NOT: BR
017140  000137  017236                  JMP     TIMER           ;ELSE GO TO ERROR
017144  122737  000070  000616  4$:     CMPB    #70,TIB         ;SEE IF CHAR IS GREATER THAN 7
017152  101002                          BHI     5$              ;IF NOT: BR
017154  000137  017236                  JMP     TIMER           ;ELSE GO TO ERROR
017160  005237  000674          5$:     INC     TEMP1           ;SET FIRST CHARACTER FLAG
017164  006300                          ASL     R0
017166  006300                          ASL     R0              ;SHIFT 3 LEFT
017170  006300                          ASL     R0
```

```
3586  017172  042737  177770  000616          BIC     #177770,TIB     ;STRIP ASCII
3587  017200  053700  000616                  BIS     TIB,R0          ;LOAD CHARACTER
3588  017204  005301                          DEC     R1              ;SEE IF DONE
3589  017206  001300                          BNE     1$              ;IF NOT: BR
3590  017210  020002                  6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
3591  017212  101402                          BLOS    7$              ;IF NOT: BR
3592  017214  000137  017236                  JMP     TIMER           ;ELSE GO TO ERROR
3593  017220  020300                  7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
3594  017222  101402                          BLOS    8$              ;IF NOT: BR
3595  017224  000137  017236                  JMP     TIMER           ;ELSE GO TO ERROR
3596  017230  010015                  8$:     MOV     R0,(R5)         ;LOAD VALUE
3597  017232  005726                  9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
3598  017234  000207                          RTS     PC              ;EXIT
3599
```

```
3600
3601                                            ;TTY ENTRY ERROR SUBROUTINE**********
3602
3603  017236  012704  022026          TINER:    MOV     #MSG40,R4
3604  017242  004737  017320                    JSR     PC,TTOUT        ;PRINT?
3605  017246  005726                            TST     (SP)+           ;POP CHAR COUNT OFF STACK
3606  017250  162716  000020                    SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
3607  017254  000207                            RTS     PC              ;REDO VALUE ENTRY
3608
3609                                            ;TTY READ SUBROUTINE**********
3610
3611  017256  005277  161310          TTIN:     INC     @TKS
3612  017262  105777  161304          1$:       TSTB    @TKS
3613  017266  100375                            BPL     1$
3614  017270  017737  161300  000616            MOV     @TKB,TIB
3615  017276  042737  000200  000616            BIC     #200,TIB                ;STRIP PARITY BIT
3616  017304  013737  000616  000614            MOV     TIB,TOB                 ;MOVE CHAR TO TTY OUTPUT BFR
3617  017312  004737  017420                    JSR     PC,TOG                  ;AND ECHO IT
3618  017316  000207                            RTS     PC
3619
3620                                            ;TTY OUTPUT SUBROUTINE***********
3621
3622  017320  112437  000614          TTOUT:    MOVB    (R4)+,TOB
3623  017324  122737  000043  000614            CMPB    #43,TOB
3624  017332  001440                            BEQ     TEX
3625  017334  122737  000045  000614            CMPB    #45,TOB
3626  017342  001403                            BEQ     1$
3627  017344  004737  017420                    JSR     PC,TOG
3628  017350  000763                            BR      TTOUT
3629  017352  112737  000015  000614  1$:       MOVB    #15,TOB
3630  017360  004737  017420                    JSR     PC,TOG
3631  017364  012703  000004                    MOV     #4,R3
3632  017370  005037  000614          2$:       CLR     TOB
3633  017374  004737  017420                    JSR     PC,TOG
3634  017400  005303                            DEC     R3
3635  017402  001372                            BNE     2$                      ;DO FILLERS
3636  017404  112737  000012  000614            MOVB    #12,TOB
3637  017412  004737  017420                    JSR     PC,TOG
3638  017416  000740                            BR      TTOUT
3639  017420  105777  161152          TOG:      TSTB    @TPS
3640  017424  100375                            BPL     TOG
3641  017426  113777  000614  161144            MOVB    TOB,@TPB
3642  017434  000207                TEX:        RTS     PC
3643
3644
3645                                            ;OCTAL OUTPUT SUBROUTINE***********
3646
3647  017436  012737  000001  017666  OCTPE:    MOV     #1,OFL
3648  017444  000402                            BR      OCTPE1
3649  017446  005037  017666          OCTP:     CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
3650  017452  010304                OCTPE1:     MOV     R3,R4           ;SEE IF NUMBER IS ZERO
3651  017454  001006                            BNE     OCTPO           ;IF NOT ZERO: BR
3652  017456  005737  017666                    TST     OFL             ;SEE IF PRINT ALL 0
3653  017462  001003                            BNE     OCTPO           ;IF SO: BR
3654  017464  004737  017646                    JSR     PC,OCTPG1       ;ELSE PRINT ZERO
3655  017470  000447                            BR      OCTP3           ;SPACE AND EXIT
```

```
3656  017472  032704  100000          OCTPO:  BIT     #100000,R4      ;SEE IF MSD = 1
3657  017476  001405                          BEQ     OCTP1           ;IF NOT: BR
3658  017500  012704  000001                  MOV     #1,R4
3659  017504  004737  017624                  JSR     PC,OCTPG        ;PRINT 1
3660  017510  000403                          BR      OCTP2
3661  017512  005004                  OCTP1:  CLR     R4
3662  017514  004737  017624                  JSR     PC,OCTPG        ;PRINT 0
3663  017520  010304                  OCTP2:  MOV     R3,R4
3664  017522  006004                          ROR     R4
3665  017524  006004                          ROR     R4
3666  017526  006004                          ROR     R4              ;POSITION DIGIT
3667  017530  006004                          ROR     R4
3668  017532  000304                          SWAB    R4
3669  017534  004737  017624                  JSR     PC,OCTPG        ;PRINT DIGIT 2
3670  017540  010304                          MOV     R3,R4
3671  017542  006004                          ROR     R4
3672  017544  000304                          SWAB    R4
3673  017546  004737  017624                  JSR     PC,OCTPG        ;PRINT DIGIT 3
3674  017552  010304                          MOV     R3,R4
3675  017554  006104                          ROL     R4
3676  017556  006104                          ROL     R4
3677  017560  000304                          SWAB    R4
3678  017562  004737  017624                  JSR     PC,OCTPG        ;PRINT DIGIT 4
3679  017566  010304                          MOV     R3,R4
3680  017570  006004                          ROR     R4
3681  017572  006004                          ROR     R4
3682  017574  006004                          ROR     R4
3683  017576  004737  017624                  JSR     PC,OCTPG
3684  017602  010304                          MOV     R3,R4
3685  017604  004737  017624                  JSR     PC,OCTPG        ;PRINT DIGIT 5
3686  017610  012737  000240  000614  OCTP3:  MOV     #240,TOB
3687  017616  004737  017420                  JSR     PC,TOG          ;PRINT SPACE
3688  017622  000207                          RTS     PC              ;EXIT
3689
3690  017624  042704  177770          OCTPG:  BIC     #177770,R4
3691  017630  001004                          BNE     OCTPGO
3692  017632  005737  017666                  TST     OFL
3693  017636  001001                          BNE     OCTPGO
3694  017640  000207                          RTS     PC
3695
3696  017642  005237  017666          OCTPGO: INC     OFL
3697  017646  052704  000260          OCTPG1: BIS     #260,R4
3698  017652  010437  000614                  MOV     R4,TOB
3699  017656  004737  017420                  JSR     PC,TOG
3700  017662  010304                          MOV     R3,R4
3701  017664  000207                          RTS     PC
3702  017666  000000                  OFL:    0                       ;FIRST CHAR FLAG
3703
```

```
3704
3705                                        ;DATA CHARACTER OUTPUT SUBROUTINE***********
3706
3707  017670  012704  000010        DOUT:   MOV     #10,R4          ;SET NUMBER TO PRINT
3708  017674  110337  000614                MOVB    R3,@#TOB
3709  017700  105777  160672        1$:     TSTB    @#TPS
3710  017704  100375                        BPL     1$
3711  017706  132737  000200  000614        BITB    #200,@#TOB
3712  017714  001404                        BEQ     2$
3713  017716  012777  000061  160654        MOV     #061,@#TPB
3714  017724  000403                        BR      3$
3715  017726  012777  000060  160644  2$:   MOV     #060,@#TPB
3716  017734  006337  000614        3$:     ASL     @#TOB
3717  017740  005304                        DEC     R4
3718  017742  001356                        BNE     1$
3719  017744  000207                        RTS     PC
3720
3721  017746  013703  000700        DOUTD:  MOV     @#TEMP3,R3
3722  017752  000303                        SWAB    R3
3723  017754  004737  017670                JSR     PC,DOUT
3724  017760  013703  000700                MOV     @#TEMP3,R3
3725  017764  004737  017670                JSR     PC,DOUT
3726  017770  000207                        RTS     PC
3727
3728                                        ;TE16 SERIAL NUMBER PRINT SUBROUTINE**********
3729
3730  017772  010304        SNPT:   MOV     R3,R4
3731  017774  000304                SWAB    R4
3732  017776  006004                ROR     R4
3733  020000  006004                ROR     R4
3734  020002  006004                ROR     R4
3735  020004  006004                ROR     R4              ;GET FIRST DIGIT
3736  020006  004737  020050        JSR     PC,SNPG         ;PRINT
3737  020012  010304                MOV     R3,R4
3738  020014  000304                SWAB    R4              ;GET SECOND DIGIT
3739  020016  004737  020050        JSR     PC,SNPG         ;PRINT
3740  020022  010304                MOV     R3,R4
3741  020024  006004                ROR     R4
3742  020026  006004                ROR     R4
3743  020030  006004                ROR     R4
3744  020032  006004                ROR     R4
3745  020034  004737  020050        JSR     PC,SNPG         ;PRINT THIRD DIGIT
3746  020040  010304                MOV     R3,R4
3747  020042  004737  020050        JSR     PC,SNPG         ;PRINT FOURTH DIGIT
3748  020046  000207                RTS     PC              ;EXIT
3749  020050  012737  000260  000614  SNPG:  MOV     #260,@#TOB      ;SET BASE = 0
3750  020056  042704  177760                BIC     #177760,R4      ;MASK DIGIT
3751  020062  050437  000614                BIS     R4,@#TOB        ;SET ASCII
3752  020066  004737  017420                JSR     PC,TOG          ;TYPE DIGIT
3753  020072  000207                        RTS     PC              ;RETURN
```

```
                                        ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
                                        ;IF A CONTROL G (↑G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
020074  022737  000176  000570  GTSWR:  CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
020102  001032                          BNE     1$              ;NOT INVOKED
020104  004737  020172                  JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
020110  012704  026173                  MOV     #SWSWR,R4       ;TYPE 'SWR = '
020114  004737  017320                  JSR     PC,TTOUT
020120  017703  160444                  MOV     @SWR,R3         ;GET CURRENT VALUE
020124  004737  017436                  JSR     PC,OCTPE        ;AND TYPE IT
020130  012704  026203                  MOV     #SWNEW,R4       ;ASK FOR NEW VALUE
020134  004737  017320                  JSR     PC,TTOUT
020140  013705  000570                  MOV     SWR,R5          ;NEW VALUE WILL BE RETURNED IN (R5)
020144  012701  000007                  MOV     #7,R1           ;LIMIT TO 7 CHARACTERS
020150  012702  177777                  MOV     #177777,R2      ;LIMIT RESPONSE TO BETWEEN
020154  012703  000000                  MOV     #0,R3           ;0 AND 177777
020160  004737  016776                  JSR     PC,TTR          ;GET RESPONSE
020164  004737  020214                  JSR     PC,.RESTORE     ;RESTORE REGISTERS
020170  000207                  1$:     RTS     PC              ;RETURN TO CALLER
                                        ;;ROUTINE TO SAVE REGISTERS ON THE STACK
020172  010546                  .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
020174  010446                          MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
020176  010346                          MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
020200  010246                          MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
020202  010146                          MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
020204  010046                          MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
020206  016646  000014                  MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
020212  000207                          RTS     PC              ;;RETURN TO CALLER

                                        ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
020214  012666  000014          .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
020220  012600                          MOV     (SP)+,%0
020222  012601                          MOV     (SP)+,%1
020224  012602                          MOV     (SP)+,%2
020226  012603                          MOV     (SP)+,%3
020230  012604                          MOV     (SP)+,%4
020232  012605                          MOV     (SP)+,%5
020234  000207                          RTS     PC              ;;RETURN


                                        ;MESSAGE TABLE**********

020236  022445  046524  031460  MSG1:   .ASCII  /%%TM03-TE16 CONTROL LOGIC TEST- PART I (DZTEA-A)/
020244  052055  030505  020066
020252  047503  052116  047522
020260  020114  047514  044507
020266  020103  042524  052123
020274  020055  040520  052122
020302  044440  024040  055104
020310  042524  026501  024501
020316  025045  025052  051501          .ASCII  /%***ASSURE TAPE IS AT BOT***/
020324  052523  042522  052040
020332  050101  020105  051511
020340  040440  020124  047502
020346  025124  025052
020352  052045  050131  020105          .ASCII  /%TYPE <CR> TO TERMINATE RESPONSE & ↑C TO RESTART%%/
```

```
3810  020360  041474  037122  052040
3811  020366  020117  042524  046522
3812  020374  047111  052101  020105
3813  020402  042522  050123  047117
3814  020410  042523  023040  057040
3815  020416  020103  047524  051040
3816  020424  051505  040524  052122
3817  020432  021445
3818  020434  042045  044522  042526  MSG2:   .ASCII  /%DRIVE NUMBER OR (CR) WHEN DONE #/
3819  020442  047040  046525  042502
3820  020450  020122  051117  024040
3821  020456  051103  020051  044127
3822  020464  047105  042040  047117
3823  020472  020105      043
3824  020475      045  043045  051117  MSG2A:  .ASCII  /%%FOR DRIVE ADDRESS TEST;/
3825  020502  042040  044522  042526
3826  020510  040440  042104  042522
3827  020516  051523  052040  051505
3828  020524  035524
3829  020526  020045  047105  042524          .ASCII  /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
3830  020534  020122  054105  052120
3831  020542  042040  044522  042526
3832  020550  047040  046525  042502
3833  020556  026122  040440  046114
3834  020564  047440  044124  051105
3835  020572  020123  044123  052517
3836  020600  042114  041040  020105
3837  020606  047516  026516  054105
3838  020614  051511  020524  052116
3839  020622  021456
3840  020624  047045  047117  042455  MSG3:   .ASCII  /%NON-EXIST DRIVE #/
3841  020632  044530  052123  042040
3842  020640  044522  042526  021440
3843  020646  051045  020110  042504  MSG4:   .ASCII  /%RH DETECTED #/
3844  020654  042524  052103  042105
3845  020662  021440
3846  020664  052045  030115  020063  MSG5:   .ASCII  /%TM03 DETECTED #/
3847  020672  042504  042524  052103
3848  020700  042105  021440
3849  020704  054105  052120  047055  MSG6:   .ASCII  /EXPT-NOT RECVD#/
3850  020712  052117  051040  041505
3851  020720  042126      043
3852  020723      122  053103  026504  MSG7:   .ASCII  /RCVD-NOT EXPT#/
3853  020730  047516  020124  054105
3854  020736  052120      043
3855  020741      045  046123  053101  MSG8:   .ASCII  /%SLAVE NUMBER OR (CR) WHEN DONE #/
3856  020746  020105  052516  041115
3857  020754  051105  047440  020122
3858  020762  041450  024522  053440
3859  020770  042510  020116  047504
3860  020776  042516  021440
3861  021002  022445  047506  020122  MSG8A:  .ASCII  /%%FOR SLAVE ADDRESS TEST;/
3862  021010  046123  053101  020105
3863  021016  042101  051104  051505
3864  021024  020123  042524  052123
3865  021032      073
```

```
3866  021033  045  042440  052116          .ASCII  /% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.*/
3867  021040  051105  042440  050130
3868  021046  020124  046123  053101
3869  021054  020105  047516  041115
3870  021062  041105  040054  046101
3871  021070  020114  052117  042510
3872  021076  051522  051440  047510
3873  021104  046123  020104  042502
3874  021112  047040  047117  042455
3875  021120  044440  052123  047101
3876  021126  027124  043
3877  021131  045  047516  026516  MSG9:   .ASCII  /%NON-EXIST SLAVE */
3878  021136  054105  051511  020124
3879  021144  046123  053101  020105
3880  021152  043
3881  021153  045  042522  042101  MSG10:  .ASCII  /%READ CONT BUS PAR */
3882  021160  041440  047117  020124
3883  021166  052502  020123  040520
3884  021174  020122  043
3885  021177  045  051127  052111  MSG11:  .ASCII  /%WRITE CONT BUS PAR */
3886  021204  020105  047503  052116
3887  021212  041040  051525  050040
3888  021220  051101  021440
3889  021224  042440  050130  020124  MSG12:  .ASCII  / EXPT */
3890  021232  043
3891  021233  040  041522  042126  MSG13:  .ASCII  / RCVD */
3892  021240  021440
3893  021242  046445  020122  044502  MSG14:  .ASCII  /%MR BITS 4-0*/
3894  021250  051524  032040  030055
3895  021256  043
3896  021257  045  051115  041040  MSG15:  .ASCII  /%MR BITS 15-7*/
3897  021264  052111  020123  032461
3898  021272  033455  043
3899  021275  045  052111  051105  MSG16:  .ASCII  /%ITER: */
3900  021302  020072  043
3901  021305  045  041524  041040  MSG18:  .ASCII  /%TC BITS 12-0 */
3902  021312  052111  020123  031061
3903  021320  030055  021440
3904  021324  043045  020103  044502  MSG19:  .ASCII  /%FC BITS 15-0 */
3905  021332  051524  030440  026465
3906  021340  020060  043
3907  021343  045  052506  020116  MSG20:  .ASCII  /%FUN CODE BITS 5-1 OF C1 */
3908  021350  047503  042504  041040
3909  021356  052111  020123  026465
3910  021364  020061  043117  041440
3911  021372  020061  043
3912  021375  045  047507  041040  MSG21:  .ASCII  /%GO BIT NOT CORRECT AT START */
3913  021402  052111  047040  052117
3914  021410  041440  051117  042522
3915  021416  052103  040440  020124
3916  021424  052123  051101  020124
3917  021432  043
3918  021433  045  047507  041040  MSG22:  .ASCII  /%GO BIT NOT SET */
3919  021440  052111  047040  052117
3920  021446  051440  052105  021440
3921  021454  043445  020117  044502  MSG23:  .ASCII  /%GO BIT NOT RESET BY INIT */
```

```
3922  021462  020124  047516  020124
3923  021470  042522  042523  020124
3924  021476  054502  044440  044516
3925  021504  020124     043
3926  021507     045  051104  020131  MSG24:  .ASCII  /%DRY NOT SET BY INIT #/
3927  021514  047516  020124  042523
3928  021522  020124  054502  044440
3929  021530  044516  020124     043
3930  021536     045  051104  020131  MSG25:  .ASCII  /%DRY NOT RESET BY GO=1#/
3931  021542  047516  020124  042522
3932  021550  042523  020124  054502
3933  021556  043440  020517  021461
3934  021564  042045  054522  047040  MSG25A: .ASCII  /%DRY NOT SET BY GO=0#/
3935  021572  052117  051440  052105
3936  021600  041040  020131  047507
3937  021606  030075     043
3938  021611     045  047516  044440  MSG26:  .ASCII  /%NO INTERRUPT RETURNED#/
3939  021616  052116  051105  052522
3940  021624  052120  051040  052105
3941  021632  051125  042516  021504
3942  021640  041045  042101  051440  MSG27:  .ASCII  /%BAD STATUS#/
3943  021646  040524  052524  021523
3944  021654  051445  035116  021440  MSG30:  .ASCII  /%SN: #/
3945  021662  042445  051122  047040  MSG31:  .ASCII  /%ERR NOT SET #/
3946  021670  052117  051440  052105
3947  021676  021440
3948  021700  040445  040524  047040  MSG32:  .ASCII  /%ATA NOT SET #/
3949  021706  052117  051440  052105
3950  021714  021440
3951  021716  040445  020123  044502  MSG33:  .ASCII  /%AS BIT NOT SET #/
3952  021724  020124  047516  020124
3953  021732  042523  020124     043
3954  021737     045  041523  047040  MSG34:  .ASCII  /%SC NOT SET #/
3955  021744  052117  051440  052105
3956  021752  021440
3957  021754  052045  042522  047040  MSG35:  .ASCII  /%TRE NOT SET #/
3958  021762  052117  051440  052105
3959  021770  021440
3960  021772  051445  040514  047040  MSG36:  .ASCII  /%SLA NOT SET #/
3961  022000  052117  051440  052105
3962  022006  021440
3963  022010  051445  041523  047040  MSG37:  .ASCII  /%SSC NOT SET #/
3964  022016  052117  051440  052105
3965  022024  021440
3966  022026  037440  021440          MSG40:  .ASCII  / ? #/
3967  022032  022445  047105  020104  MSG41:  .ASCII  /%%END OF PASS #/
3968  022040  043117  050040  051501
3969  022046  020123     043
3970  022051     045  042504  042101  MSG42:  .ASCII  /%DEAD TRACK #/
3971  022056  052040  040522  045503
3972  022064  021440
3973  022066  022445  040515  052516  MSG43:  .ASCII  /%%MANUAL TESTS (14-17) INHIBITED: HALT%/
3974  022074  046101  052040  051505
3975  022102  051524  024040  032061
3976  022110  030455  024467  044440
3977  022116  044116  041111  052111
```

```
  3978  022124  042105  020072  040510
  3979  022132  052114      045
  3980  022136      122  051505  046105          .ASCII  /RESELECT AND PRESS CONTINUE%*/
  3981  022144  041505  020124  047101
  3982  022152  020104  051120  051505
  3983  022160  020123  047503  052116
  3984  022166  047111  042525  021445
  3985  022172  051045  043505  051511  MSG44:  .ASCII  /%REGISTER START: */
  3986  022200  042524  020122  052123
  3987  022206  051101  035124  021440
  3988  022214  053045  041505  047524  MSG45:  .ASCII  /%VECTOR ADDRESS: */
  3989  022222  020122  042101  051104
  3990  022230  051505  035123  021440
  3991  022236  041445  030523  020040  MSG46:  .ASCII  /%CS1    WC     BA     FC     CS2    DS     ER     AS/
  3992  022244  020040  041527  020040
  3993  022252  020040  041040  020101
  3994  022260  020040  020040  041506
  3995  022266  020040  020040  041440
  3996  022274  031123  020040  020040
  3997  022302  051504  020040  020040
  3998  022310  042440  020122  020040
  3999  022316  020040  051501
  4000  022322  020040  020040  046440          .ASCII  /     MR     TC%*/
  4001  022330  020122  020040  020040
  4002  022336  041524  021445
  4003  022342  047045  052117  051040  MSG47:  .ASCII  /%NOT RESET BY DRIVE CLEAR*/
  4004  022350  051505  052105  041040
  4005  022356  020131  051104  053111
  4006  022364  020105  046103  040505
  4007  022372  021522
  4008  022374  040445  050114  040510  MSG50:  .ASCII  /%ALPHA NOT SET*/
  4009  022402  047040  052117  051440
  4010  022410  052105      043
  4011  022413      045  047125  054105  MSG51:  .ASCII  /%UNEXPECTED ERROR BITS*/
  4012  022420  042520  052103  042105
  4013  022426  042440  051122  051117
  4014  022434  041040  052111  021523
  4015  022442  041045  042101  046040  MSG53:  .ASCII  /%BAD LRC */
  4016  022450  041522  021440
  4017  022454  041045  042101  041440  MSG54:  .ASCII  /%BAD CK */
  4018  022462  020113      043
  4019  022465      045  042523  052524  MSG55:  .ASCII  /%SETUP ERROR: CHECK WRAP 0 WITH TEST 50*/
  4020  022472  020120  051105  047522
  4021  022500  035122  041440  042510
  4022  022506  045503  053440  040522
  4023  022514  020120  020060  044527
  4024  022522  044124  052040  051505
  4025  022530  020124  030065      043
  4026  022535      045  052123  052101  MSG56:  .ASCII  /%STATIC TESTS ONLY: */
  4027  022542  041511  052040  051505
  4028  022550  051524  047440  046116
  4029  022556  035131  021440
  4030  022562  052045  047515  020063  MSG57:  .ASCII  /%TM03 DRIVE: */
  4031  022570  051104  053111  035105
  4032  022576  021440
  4033  022600  052045  030505  020066  MSG58:  .ASCII  /%TE16 SLAVE: */
```

```
4034  022606  046123  053101  035105
4035  022614  021440
4036  022616  021445                          MSG59:  .ASCII  /%#/
4037  022620  021534                          MSG60:  .ASCII  /\#/
4038  022622  051045  046505  053117  MSG62:  .ASCII  /%REMOVE TMDP FROM SLAVE TO BE TESTED%#/
4039  022630  020105  046524  050104
4040  022636  043040  047522  020115
4041  022644  046123  053101  020105
4042  022652  047524  041040  020105
4043  022660  042524  052123  042105
4044  022666  021445
4045  022670  044045  051101  053504  MSG63:  .ASCII  /%HARDWARE SWR IN USE%#/
4046  022676  051101  020105  053523
4047  022704  020122  047111  052440
4048  022712  042523  021445
```

```
                                                        ;TEST HEADER**********
                                                
022716  022445  047514  044507  MSLT1:  .ASCII  /%%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)*/
027724  020103  042524  052123
027732  030440  020072  051104
027740  052111  020105  042101
027746  051104  021505  044523
027754  043516  024440  034115
027762  030071  020071  044122
027770  021451
027772  022445  047514  044507  MSLT2:  .ASCII  /%%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)*/
030000  020103  042524  052123
030006  031040  020072  042522
030014  044507  042123  051105
030022  040440  042104  042522
030030  051523  047111  020107
030036  046450  034470  034460
030044  051040  024510  000043
030051  000045  046045  043517  MSLT3:  .ASCII  /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905 M8909)*/
030056  041511  052040  051505
030064  020124  035063  041440
030072  047117  051124  046117
030100  041040  051525  052040
030106  051505  020124  051050
030114  020110  034115  030071
030122  020065  034115  030071
030130  024471  000043
030133  000045  046045  043517  MSLT4:  .ASCII  /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905 M8903)*/
030140  041511  052040  051505
030146  020124  035064  051440
030154  040514  042526  040440
030162  042104  042522  051523
030170  047111  020107  046450
030176  034470  032460  046440
030204  034470  031460  021451
030212  022445  047514  044507  MSLT5:  .ASCII  /%%LOGIC TEST 5: MR BIT TEST (M8905)*/
030220  020103  042524  052123
030226  032440  020072  051115
030234  041040  052111  052040
030242  051505  020124  046450
030250  034470  032460  021451
030256  022445  047514  044507  MSLT6:  .ASCII  /%%LOGIC TEST 6: TC BIT TEST (M8905)*/
030264  020103  042524  052123
030272  033040  020072  041524
030300  041040  052111  052040
030306  051505  020124  046450
030314  034470  032460  021451
030322  022445  047514  044507  MSLT7:  .ASCII  /%%LOGIC TEST 7: FC BIT TEST (M8905)*/
030330  020103  042524  052123
030336  033440  020072  041506
030344  041040  052111  052040
030352  051505  020124  046450
030360  034470  032460  021451
030366  022445  047514  044507  MSLT10: .ASCII  /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905)*/
030374  020103  042524  052123
030402  030440  035060  043040
```

```
4105  023410  047125  052103  047511
4106  023416  020116  044502  020124
4107  023424  042524  052123  024040
4108  023432  034115  030071  024465
4109  023440     043
4110  023441     045  046045  043517  MSLT11: .ASCII  /%%LOGIC TEST 11: GO BIT TEST (M8909)#/
4111  023446  041511  052040  051505
4112  023454  020124  030461  020072
4113  023462  047507  041040  052111
4114  023470  052040  051505  020124
4115  023476  046450  034470  034460
4116  023504  021451
4117  023506  022445  047514  044507  MSLT12: .ASCII  /%%LOGIC TEST 12: DRIVE READY BIT (M8909)#/
4118  023514  020103  042524  052123
4119  023522  030440  035062  042040
4120  023530  044522  042526  051040
4121  023536  040505  054504  041040
4122  023544  052111  024040  034115
4123  023552  030071  024471     043
4124  023557     045  046045  043517  MSLT13: .ASCII  /%%LOGIC TEST 13: INTERRUPT TEST (RH)#/
4125  023564  041511  052040  051505
4126  023572  020124  031461  020072
4127  023600  047111  042524  051122
4128  023606  050125  020124  042524
4129  023614  052123  024040  044122
4130  023622  021451
4131  023624  022445  047514  044507  MSLT14: .ASCII  /%%LOGIC TEST 14: MANUAL STATUS TEST 1#/
4132  023632  020103  042524  052123
4133  023640  030440  035064  046440
4134  023646  047101  040525  020114
4135  023654  052123  052101  051525
4136  023662  052040  051505  020124
4137  023670  021461
4138  023672  022445  047514  044507  MSLT15: .ASCII  /%%LOGIC TEST 15: MANUAL STATUS TEST 2#/
4139  023700  020103  042524  052123
4140  023706  030440  035065  046440
4141  023714  047101  040525  020114
4142  023722  052123  052101  051525
4143  023730  052040  051505  020124
4144  023736  021462
4145  023740  022445  047514  044507  MSLT16: .ASCII  /%%LOGIC TEST 16: MANUAL STATUS TEST 3#/
4146  023746  020103  042524  052123
4147  023754  030440  035066  046440
4148  023762  047101  040525  020114
4149  023770  052123  052101  051525
4150  023776  052040  051505  020124
4151  024004  021463
4152  024006  022445  047514  044507  MSLT17: .ASCII  /%%LOGIC TEST 17: MANUAL STATUS TEST 4#/
4153  024014  020103  042524  052123
4154  024022  030440  035067  046440
4155  024030  047101  040525  020114
4156  024036  052123  052101  051525
4157  024044  052040  051505  020124
4158  024052  021464
4159  024054  022445  047514  044507  MSLT20: .ASCII  /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8909)#/
4160  024062  020103  042524  052123
```

```
4161  024070  031040  035060  044440
4162  024076  046114  043505  046101
4163  024104  043040  047125  052103
4164  024112  047511  020116  042524
4165  024120  052123  024040  034115
4166  024126  030071  024471      043
4167  024133      045  046045  043517  MSLT21: .ASCII  /%%LOGIC TEST 21: RMR(M8909)@/
4168  024140  041511  052040  051505
4169  024146  020124  030462  020072
4170  024154  046522  024122  034115
4171  024162  030071  024471      043
4172  024167      045  046045  043517  MSLT22: .ASCII  /%%LOGIC TEST 22: CPAR(M8909)@/
4173  024174  041511  052040  051505
4174  024202  020124  031062  020072
4175  024210  050103  051101  046450
4176  024216  034470  034460  021451
4177  024224  022445  047514  044507  MSLT23: .ASCII  /%%LOGIC TEST 23: FMT(M8905 M8906)@/
4178  024232  020103  042524  052123
4179  024240  031040  035063  043040
4180  024246  052115  046450  034470
4181  024254  032460  046440  034470
4182  024262  033060  021451
4183  024266  022445  047514  044507  MSLT24: .ASCII  /%%LOGIC TEST 24: DPAR(M8906 RH)@/
4184  024274  020103  042524  052123
4185  024302  031040  035064  042040
4186  024310  040520  024122  034115
4187  024316  030071  020066  044122
4188  024324  021451
4189  024326  022445  047514  044507  MSLT25: .ASCII  /%%LOGIC TEST 25: NEF(M8909)@/
4190  024334  020103  042524  052123
4191  024342  031040  035065  047040
4192  024350  043105  046450  034470
4193  024356  034460  021451
4194  024362  022445  047514  044507  MSLT26: .ASCII  /%%LOGIC TEST 26: FCE(M8909)@/
4195  024370  020103  042524  052123
4196  024376  031040  035066  043040
4197  024404  042503  046450  034470
4198  024412  034460  021451
4199  024416  022445  047514  044507  MSLT27: .ASCII  /%%LOGIC TEST 27: ILR(M8909)@/
4200  024424  020103  042524  052123
4201  024432  031040  035067  044440
4202  024440  051114  046450  034470
4203  024446  034460  021451
4204  024452  022445  047514  044507  MSLT30: .ASCII  /%%LOGIC TEST 30:DTE(M8906 RH)@/
4205  024460  020103  042524  052123
4206  024466  031440  035060  052104
4207  024474  044105  034115  030071
4208  024502  020066  044122  021451
4209  024510  022445  047514  044507  MSLT31: .ASCII  /%%LOGIC TEST 31: OPI(M8903)@/
4210  024516  020103  042524  052123
4211  024524  031440  035061  047440
4212  024532  044520  046450  034470
4213  024540  031460  021451
4214  024544  022445  047514  044507  MSLT32: .ASCII  /%%LOGIC TEST 32: UNS(M8909)@/
4215  024552  020103  042524  052123
4216  024560  031440  035062  052440
```

```
4217  024566  051516  046450  034470
4218  024574  034460  021451
4219  024600  022445  047514  044507  MSLT33: .ASCII  /%%LOGIC TEST 33: PIP(M8909)#/
4220  024606  020103  042524  052123
4221  024614  031440  035063  050040
4222  024622  050111  046450  034470
4223  024630  034460  021451
4224  024634  022445  047514  044507  MSLT34: .ASCII  /%%LOGIC TEST 34: PES(M8911)#/
4225  024642  020103  042524  052123
4226  024650  031440  035064  050040
4227  024656  051505  046450  034470
4228  024664  030461  021451
4229  024670  022445  047514  044507  MSLT35: .ASCII  /%%LOGIC TEST 35: SAC(M8903 M8905)#/
4230  024676  020103  042524  052123
4231  024704  031440  035065  051440
4232  024712  041501  046450  034470
4233  024720  031460  046440  034470
4234  024726  032460  021451
4235  024732  022445  047514  044507  MSLT36: .ASCII  /%%LOGIC TEST 36: FCS(M8903 M8905)#/
4236  024740  020103  042524  052123
4237  024746  031440  035066  043040
4238  024754  051503  046450  034470
4239  024762  031460  046440  034470
4240  024770  032460  021451
4241  024774  022445  047514  044507  MSLT37: .ASCII  /%%LOGIC TEST 37: ACCL(M8903 M8905)#/
4242  025002  020103  042524  052123
4243  025010  031440  035067  040440
4244  025016  041503  024114  034115
4245  025024  030071  020063  034115
4246  025032  030071  024465      043
4247  025037      045  046045  043517  MSLT40: .ASCII  /%%LOGIC TEST 40: PE TAPE MARK(M8902)#/
4248  025044  041511  052040  051505
4249  025052  020124  030064  020072
4250  025060  042520  052040  050101
4251  025066  020105  040515  045522
4252  025074  046450  034470  031060
4253  025102  021451
4254  025104  022445  047514  044507  MSLT41: .ASCII  /%%LOGIC TEST 41: NRZ TAPE MARK (M8904)#/
4255  025112  020103  042524  052123
4256  025120  032040  035061  047040
4257  025126  055122  052040  050101
4258  025134  020105  040515  045522
4259  025142  024040  034115  030071
4260  025150  024464      043
4261  025153      045  046045  043517  MSLT42: .ASCII  /%%LOGIC TEST 42: CRC(M8904)#/
4262  025160  041511  052040  051505
4263  025166  020124  031064  020072
4264  025174  051103  024103  034115
4265  025202  030071  024464      043
4266  025207      045  046045  043517  MSLT43: .ASCII  /%%LOGIC TEST 43: LRC(M8904)#/
4267  025214  041511  052040  051505
4268  025222  020124  031464  020072
4269  025230  051114  024103  034115
4270  025236  030071  024464      043
4271  025243      045  046045  043517  MSLT44: .ASCII  /%%LOGIC TEST 44: CORRECTABLE DATA (M8902 M8901)#/
4272  025250  041511  052040  051505
```

```
  4273  025256  020124  032064  020072
  4274  025264  047503  051122  041505
  4275  025272  040524  046102  020105
  4276  025300  040504  040524  024040
  4277  025306  034115  030071  020062
  4278  025314  034115  030071  024461
  4279  025322      043
  4280  025323      045  046045  043517  MSLT45: .ASCII  /%%LOGIC TEST 45: INCORRECTABLE DATA (M8902 M8904)#/
  4281  025330  041511  052040  051505
  4282  025336  020124  032464  020072
  4283  025344  047111  047503  051122
  4284  025352  041505  040524  046102
  4285  025360  020105  040504  040524
  4286  025366  024040  034115  030071
  4287  025374  020062  034115  030071
  4288  025402  024464      043
  4289  025405      045  046045  043517  MSLT46: .ASCII  /%%LOGIC TEST 46: PEF(M8902)#/
  4290  025412  041511  052040  051505
  4291  025420  020124  033064  020072
  4292  025426  042520  024106  034115
  4293  025434  030071  024462      043
  4294  025441      045  046045  043517  MSLT47: .ASCII  /%%LOGIC TEST 47: FC OVERFLOW (M8905)#/
  4295  025446  041511  052040  051505
  4296  025454  020124  033464  020072
  4297  025462  041506  047440  042526
  4298  025470  043122  047514  020127
  4299  025476  046450  034470  032460
  4300  025504  021451
  4301  025506  022445  047514  044507  MSLT50: .ASCII  /%%LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/
  4302  025514  020103  042524  052123
  4303  025522  032440  035060  047040
  4304  025530  043105  053440  042510
  4305  025536  020116  051127  052111
  4306  025544  020105  042520  047440
  4307  025552  020116  051116  020132
  4308  025560  046123  053101  021505
  4309  025566  022445  047514  044507  MSLT51: .ASCII  /%%LOGIC TEST 51: NEF WHEN WRITE NRZ ON PE SLAVE#/
  4310  025574  020103  042524  052123
  4311  025602  032440  035061  047040
  4312  025610  043105  053440  042510
  4313  025616  020116  051127  052111
  4314  025624  020105  051116  020132
  4315  025632  047117  050040  020105
  4316  025640  046123  053101  021505
```

```
4317
4318                                                    ;MANUAL INSTRUCTION**********
4319
4320  025646  052045  050131  020105  MMSG0:  .ASCII  /%TYPE CR WHEN READY;*/
4321  025654  051103  053440  042510
4322  025662  020116  042522  042101
4323  025670  035531     043
4324  025673     045  046445  052517  MMSG1:  .ASCII  /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:*/
4325  025700  052116  052040  050101
4326  025706  020105  044527  044124
4327  025714  047040  020117  051127
4328  025722  052111  020105  044522
4329  025730  043516  020054  047514
4330  025736  042101  052040  020117
4331  025744  047502  026124  051440
4332  025752  052105  052040  020117
4333  025760  047117  046040  047111
4334  025766  035105     043
4335  025771     045  042523  020124  MMSG2:  .ASCII  /%SET TO OFFLINE:*/
4336  025776  047524  047440  043106
4337  026004  044514  042516  021472
4338  026012  046445  053117  020105  MMSG3:  .ASCII  /%MOVE FORWARD TO EOT, ONLINE:*/
4339  026020  047506  053522  051101
4340  026026  020104  047524  042440
4341  026034  052117  020054  047117
4342  026042  044514  042516  021472
4343  026050  047445  043106  046040  MMSG4:  .ASCII  /%OFF LINE REVERSE PAST EOT, INSERT WRITE RING, ON LINE*/
4344  026056  047111  020105  042522
4345  026064  042526  051522  020105
4346  026072  040520  052123  042440
4347  026100  052117  020054  047111
4348  026106  042523  052122  053440
4349  026114  044522  042524  051040
4350  026122  047111  026107  047440
4351  026130  020116  044514  042516
4352  026136     043
4353  026137     045  046445  053117  MMSG5:  .ASCII  /%%MOVE TAPE TO BOT; ON LINE*/
4354  026144  020105  040524  042520
4355  026152  052040  020117  047502
4356  026160  035524  047440  020116
4357  026166  044514  042516     043
```

```
4358
4359                                                    ;TAG MESSAGE
4360
4361  026173     045  053523  020122  SMSWR:  .ASCII  /%SWR = #/
4362  026200  020075     043
4363  026203     040  042516  020127  SMNEW:  .ASCII  / NEW = #/
4364  026210  020075     043
4365  026213     045  046123  020101  TMS1:   .ASCII  /%SLA #/
4366  026220     043
4367  026221     045  047502  020124  TMS2:   .ASCII  /%BOT #/
4368  026226     043
4369  026227     045  046524  021440  TMS3:   .ASCII  /%TM #/
4370  026234  044445  041104  021440  TMS4:   .ASCII  /%IDB #/
4371  026242  051445  053504  020116  TMS5:   .ASCII  /%SDWN #/
4372  026250     043
4373  026251     045  042520  020123  TMS6:   .ASCII  /%PES #/
4374  026256     043
4375  026257     045  051523  020103  TMS7:   .ASCII  /%SSC #/
4376  026264     043
4377  026265     045  051104  020131  TMS8:   .ASCII  /%DRY #/
4378  026272     043
4379  026273     045  050104  020122  TMS9:   .ASCII  /%DPR #/
4380  026300     043
4381  026301     045  052116  020114  TMS10:  .ASCII  /%NTL #/
4382  026306     043
4383  026307     045  047505  020124  TMS11:  .ASCII  /%EOT #/
4384  026314     043
4385  026315     045  051127  020114  TMS12:  .ASCII  /%WRL #/
4386  026322     043
4387  026323     045  047515  020114  TMS13:  .ASCII  /%MOL #/
4388  026330     043
4389  026331     045  044520  020120  TMS14:  .ASCII  /%PIP #/
4390  026336     043
4391  026337     045  051105  020122  TMS15:  .ASCII  /%ERR #/
4392  026344     043
4393  026345     045  052101  020101  TMS16:  .ASCII  /%ATA #/
4394  026352     043
4395  026353     045  046111  020106  TMS17:  .ASCII  /%ILF #/
4396  026360     043
4397  026361     045  046111  020122  TMS18:  .ASCII  /%ILR #/
4398  026366     043
4399  026367     045  046522  020122  TMS19:  .ASCII  /%RMR #/
4400  026374     043
4401  026375     045  050103  051101  TMS20:  .ASCII  /%CPAR #/
4402  026402  021440
4403  026404  043045  052115  021440  TMS21:  .ASCII  /%FMT #/
4404  026412  042045  040520  020122  TMS22:  .ASCII  /%DPAR #/
4405  026420     043
4406  026421     045  047111  020103  TMS23:  .ASCII  /%INC #/
4407  026426     043
4408  026427     045  050126  020105  TMS24:  .ASCII  /%VPE #/
4409  026434     043
4410  026435     045  042520  020106  TMS25:  .ASCII  /%PEF #/
4411  026442     043
4412  026443     045  051114  020103  TMS26:  .ASCII  /%LRC #/
4413  026450     043
```

```
4414  026451  045  051516  020107  TMS27:  .ASCII  /%NSG #/
4415  026456  043
4416  026457  045  041506  020105  TMS28:  .ASCII  /%FCE #/
4417  026464  043
4418  026465  045  051503  021440  TMS29:  .ASCII  /%CS #/
4419  026472  044445  046524  021440  TMS30:  .ASCII  /%ITM #/
4420  026500  047045  043105  021440  TMS31:  .ASCII  /%NEF #/
4421  026506  042045  042524  021440  TMS32:  .ASCII  /%DTE #/
4422  026514  047445  044520  021440  TMS33:  .ASCII  /%OPI #/
4423  026522  053445  044522  042524  TMS33A: .ASCII  /%WRITE OPI #/
4424  026530  047440  044520  021440
4425  026536  051045  040505  020104  TMS33B: .ASCII  /%READ OPI #/
4426  026544  050117  020111  043
4427  026551  040  041517  052503  TMS33C: .ASCII  / OCCURED TO SOON%#/
4428  026556  042522  020104  047524
4429  026564  051440  047517  022516
4430  026572  043
4431  026573  040  041517  052503  TMS33D: .ASCII  / OCCURRED TO LATE%#/
4432  026600  051122  042105  052040
4433  026606  020117  040514  042524
4434  026614  021445
4435  026616  043040  044501  042514  TMS33E: .ASCII  / FAILED TO SET%#/
4436  026624  020104  047524  051440
4437  026632  052105  021445
4438  026636  052445  051516  021440  TMS34:  .ASCII  /%UNS #/
4439  026644  041445  051117  020122  TMS35:  .ASCII  /%CORR #/
4440  026652  043
4441  026653  045  051103  020103  TMS36:  .ASCII  /%CRC #/
4442  026660  043
4443  026661  045  040523  020103  TMS37:  .ASCII  /%SAC #/
4444  026666  043
4445  026667  045  041506  020123  TMS38:  .ASCII  /%FCS #/
4446  026674  043
4447  026675  045  041501  046103  TMS39:  .ASCII  /%ACCL #/
4448  026702  021440
4449
4450                                        .EVEN
4451                                        ;WRITE BUFFER
4452
4453  026704  000100              WDATA:
4454  026704  177777                      -1
4455  026706  177777                      -1
4456  026710  177777                      -1
4457  026712  177777                      -1
4458  026714  177777                      -1
4459  026716  177777                      -1
4460  026720  177777                      -1
4461  026722  177777                      -1
4462  026724  177777                      -1
4463  026726  177777                      -1
4464  026730  177777                      -1
4465  026732  177777                      -1
4466  026734  177777                      -1
4467  026736  177777                      -1
4468  026740  177777                      -1
4469  026742  177777                      -1
```

```
4470  026744  177777                    -1
4471  026746  177777                    -1
4472  026750  177777                    -1
4473  026752  177777                    -1
4474  026754  177777                    -1
4475  026756  177777                    -1
4476  026760  177777                    -1
4477  026762  177777                    -1
4478  026764  177777                    -1
4479  026766  177777                    -1
4480  026770  177777                    -1
4481  026772  177777                    -1
4482  026774  177777                    -1
4483  026776  177777                    -1
4484  027000  177777                    -1
4485  027002  177777                    -1
4486  027004  177777                    -1
4487  027006  177777                    -1
4488  027010  177777                    -1
4489  027012  177777                    -1
4490  027014  177777                    -1
4491  027016  177777                    -1
4492  027020  177777                    -1
4493  027022  177777                    -1
4494  027024  177777                    -1
4495  027026  177777                    -1
4496  027030  177777                    -1
4497  027032  177777                    -1
4498  027034  177777                    -1
4499  027036  177777                    -1
4500  027040  177777                    -1
4501  027042  177777                    -1
4502  027044  177777                    -1
4503  027046  177777                    -1
4504  027050  177777                    -1
4505  027052  177777                    -1
4506  027054  177777                    -1
4507  027056  177777                    -1
4508  027060  177777                    -1
4509  027062  177777                    -1
4510  027064  177777                    -1
4511  027066  177777                    -1
4512  027070  177777                    -1
4513  027072  177777                    -1
4514  027074  177777                    -1
4515  027076  177777                    -1
4516  027100  177777                    -1
4517  027102  177777                    -1
4518
4519
4520                                    ;READ BUFFER
4521
4522  027104  000100            RDATA:
4523  027104  000000                    0
4524  027106  000000                    0
4525  027110  000000                    0
```

```
4526  027112  000000                0
4527  027114  000000                0
4528  027116  000000                0
4529  027120  000000                0
4530  027122  000000                0
4531  027124  000000                0
4532  027126  000000                0
4533  027130  000000                0
4534  027132  000000                0
4535  027134  000000                0
4536  027136  000000                0
4537  027140  000000                0
4538  027142  000000                0
4539  027144  000000                0
4540  027146  000000                0
4541  027150  000000                0
4542  027152  000000                0
4543  027154  000000                0
4544  027156  000000                0
4545  027160  000000                0
4546  027162  000000                0
4547  027164  000000                0
4548  027166  000000                0
4549  027170  000000                0
4550  027172  000000                0
4551  027174  000000                0
4552  027176  000000                0
4553  027200  000000                0
4554  027202  000000                0
4555  027204  000000                0
4556  027206  000000                0
4557  027210  000000                0
4558  027212  000000                0
4559  027214  000000                0
4560  027216  000000                0
4561  027220  000000                0
4562  027222  000000                0
4563  027224  000000                0
4564  027226  000000                0
4565  027230  000000                0
4566  027232  000000                0
4567  027234  000000                0
4568  027236  000000                0
4569  027240  000000                0
4570  027242  000000                0
4571  027244  000000                0
4572  027246  000000                0
4573  027250  000000                0
4574  027252  000000                0
4575  027254  000000                0
4576  027256  000000                0
4577  027260  000000                0
4578  027262  000000                0
4579  027264  000000                0
4580  027266  000000                0
4581  027270  000000                0
```

```
4582  027272  000000                          0
4583  027274  000000                          0
4584  027276  000000                          0
4585  027300  000000                          0
4586  027302  000000                          0
4587
4588                                          ;WRAP AROUND MESSAGES******
4589
4590  027304  242045  052101  020101  WMSG27: .ASCII  /%DATA PAT:#/
4591  027312  040520  035124  000043
4592  027317  000045  047505  020122  WMSG31: .ASCII  /%EOR CLEAR DID NOT CLEAR GO%#/
4593  027324  046103  040505  020122
4594  027332  044504  020104  047516
4595  027340  020124  046103  040505
4596  027346  020122  047507  021445
4597
4598                                          .EVEN
4599  027354  000000                  PRE:    0
4600  027356  000000                          0
4601  027360  000000                          0
4602  027362  000000                          0
4603  027364  000000                          0
4604  027366  000000                          0
4605  027370  000000                          0
4606  027372  000000                          0
4607  027374  000000                          0
4608  027376  000000                          0
4609  027400  000000                          0
4610  027402  000000                          0
4611  027404  000000                          0
4612  027406  000000                          0
4613  027410  000000                          0
4614  027412  000000                          0
4615  027414  000000                          0
4616  027416  000000                          0
4617  027420  000000                          0
4618  027422  000000                          0
4619  027424  000000                          0
4620  027426  000000                          0
4621  027430  000000                          0
4622  027432  000000                          0
4623  027434  000000                          0
4624  027436  000000                          0
4625  027440  000000                          0
4626  027442  000000                          0
4627  027444  000000                          0
4628  027446  000000                          0
4629  027450  000000                          0
4630  027452  000000                          0
4631  027454  000000                          0
4632  027456  000000                          0
4633  027460  000000                          0
4634  027462  000000                          0
4635  027464  000000                          0
4636  027466  000000                          0
4637  027470  000000                          0
```

```
4638  027472  000000                          0
4639  027474  000000                          0
4640  027476  000000                POST:     0
4641  027500  000000                          0
4642  027502  000000                          0
4643  027504  000000                          0
4644  027506  000000                          0
4645  027510  000000                          0
4646  027512  000000                          0
4647  027514  000000                          0
4648  027516  000000                          0
4649  027520  000000                          0
4650  027522  000000                          0
4651  027524  000000                          0
4652  027526  000000                          0
4653  027530  000000                          0
4654  027532  000000                          0
4655  027534  000000                          0
4656  027536  000000                          0
4657  027540  000000                          0
4658  027542  000000                          0
4659  027544  000000                          0
4660  027546  000000                          0
4661  027550  000000                          0
4662  027552  000000                          0
4663  027554  000000                          0
4664  027556  000000                          0
4665  027560  000000                          0
4666  027562  000000                          0
4667  027564  000000                          0
4668  027566  000000                          0
4669  027570  000000                          0
4670  027572  000000                          0
4671  027574  000000                          0
4672  027576  000000                          0
4673  027600  000000                          0
4674  027602  000000                          0
4675  027604  000000                          0
4676  027606  000000                          0
4677  027610  000000                          0
4678  027612  000000                          0
4679  027614  000000                          0
4680  027616  000000                          0
4681  027620  000000              WBUFF:      0
4682          030232                          .=.+410
4683  030232  000000              RBUFF:      0
4684
4685          000001                          .END
```

```
ADDFL   000746
AS      000526
ASF     000730
ATAF    000720
ATTA    015604
ATTB    015624
ATTC    015644
ATTD    015652
ATTN    015564
ATTP    015662
ATTPA   015710
ATTX    015734
ATTXA   015744
ATTXX   015762
BA      000514
CC      000530
CHNFLG  001416
CRCNT   000774
CS      000520
C1      000510
DATAD   000760
DATC    000754
DB      000532
DCA     015362
DCAD    015376
DCB     015434
DCC     015442
DCD     015336
DCE     015452
DCEA    015502
DCEX    015522
DCEXA   015532
DCEXX   015550
DERFL   000766
DOUT    017670
DOUTD   017746
DRVCLR  015330
DRVN    000624
DRVTP   000604
DS      000522
DT      000536
EMADDR  000622
E[illegible]A  014514
E[illegible]X  014642
E[illegible]1  014550
E[illegible]1A  014616
E[illegible]2  014640
ER      000[illegible]24
ER[illegible]00  00[illegible]52
E[illegible]F  000[illegible]26
EXFL    000[illegible]16
FC      000[illegible]16
FUN     000[illegible]52
GTSWR   020074
HDRFL   000620
HERE    002512
ILFT    000544
INIT    016406
INIT1   016330
INIT2   016340
INIT3   016356
INIT4   016372
INIT5   016412
INMT    016310
INST    016572
ITAMT   000606
ITCNT   000702
ITER    016240
ITRLP   000714
LTADD   000742
LTGA    014706
LTGA1   014724
LTGA2   014756
LTGB    015004
LTGC    015010
LTGCO   015076
LTGD    015102
LTGER   014670
LTGERO  014662
LTGER1  015776
LTGER2  015770
LTGER3  014650
LTGX    015132
LTGXA   015142
LTGXX   015160
LTG1A   016014
LTG1B   016032
LTG1C   016142
LTG1T   016112
LTG1X   016146
LTG1XX  016176
LTG1X1  016160
LTI     002552
LTIA    002676
LTIB    002744
LTIC    002752
LTIER   002762
LTIER1  002772
LTIER2  003000
LTIG    002620
LTIGO   002610
LTIX    003022
LTI0    004334
LTI0A   004352
LTI0A1  004350
LTI0B   004406
LTI0E1  004426
LTI0IT  004336
LTI0X   004450
LTI1    004460
LTI1B   004506
LTI1C   004546
LTI1E1  004566
LTI1E2  004616
LTI1E3  004646
LTI1IT  004462
LTI1X   004674
LTI2    004704
LTI2A   004730
LTI2C   004766
LTI2E1  005004
LTI2E2  005026
LTI2E3  005050
LTI2IT  004706
LTI2X   005070
LTI3    005100
LTI3A   005140
LTI3E1  005144
LTI3IT  005110
LTI3X   005172
LTI4    005202
LTI4A   005224
LTI4IT  005242
LTI4X   005302
LTI4XX  005306
LTI5    005312
LTI5A   005334
LTI5IT  005352
LTI5X   005412
LTI5XX  005416
LTI6    005422
LTI6A   005444
LTI6IT  005462
LTI6X   005522
LTI6XX  005526
LTI7    005532
LTI7A   005554
LTI7IT  005572
LTI7X   005632
LTI7XX  005636
LTI2    003026
LT2A    003066
LT2B    003106
LT2C    003122
LT2ERG  003202
LT2ER1  003132
LT2ER2  003150
LT2ER3  003166
LT2IT   003030
LT2LP   003216
LT2X    003222
LT20    005642
LT209   005670
LT20I   005746
LT20C   005756
LT20IT  005656
LT20X   005770
LT21    006004
LT21A   006110
LT21B   006120
LT21IT  006020
LT21XA  006130
LT22    006150
LT22A   006250
LT22IT  006164
LT22X   006260
LT23    006274
LT23A   006374
LT23IT  006310
LT23X   006404
LT24    006424
LT24B   006552
LT24BO  006574
LT24C   006606
LT24D   006670
LT24IT  006440
LT24X   006714
LT25    006750
LT25A   007056
LT25IT  006756
LT25X   007066
LT26    007102
LT26IT  007110
LT26X   007302
LT27    007316
LT27A   007352
LT27B   007410
LT27IT  007342
LT27X   007420
LT27XX  007430
LT3     003232
LT3A    003252
LT3B    003272
LT3C    003306
LT3ER1  003316
LT3ER2  003344
LT3IT   003234
LT3X    003366
LT3XX   003404
LT30    007434
LT30A   007536
LT30B   007600
LT30C   007640
LT30D   007656
LT30E   007706
LT30IT  007456
LT30X   007732
LT31    007752
LT31A   010130
LT31IT  007760
LT31X   010234
LT32    010564
LT32IT  010600
LT32X   010712
LT32XX  010722
LT33    010726
LT33IT  010742
LT33X   011020
LT34    011030
LT34A   011070
LT34A1  011050
LT34B   011120
LT34C   011124
LT34IT  011044
LT34X   011154
LT34XX  011160
LT35    011164
LT35A   011266
LT35IT  011200
LT35X   011326
LT36    011336
LT36IT  011352
LT36X   011446
LT37    011456
LT37A   011532
LT37B   011564
LT37IT  011472
LT37X   011616
LT4     003414
LT4A    003532
LT4B    003574
LT4C    003602
LT4D    003632
LT4ERG  003660
LT4ER1  003642
LT4ER2  003652
LT4G    003462
LT4GO   003452
LT4X    003710
LT4D    011626
LT4OIT  011642
LT4OX   011736
LT4OXX  011742
LT41    011746
LT41IT  011762
```

```
LT41X   012156
LT42    012210
LT42A   012312
LT42B   012326
LT42B1  012354
LT42B2  012374
LT42C   012440
LT42D   012454
LT42E   012504
LT42IT  012546
LT42X   012624
LT43    012640
LT43C   012640
LT43D   012654
LT43E   012706
LT43F   012740
LT43IT  012566
LT43X   012760
LT44    012774
LT44A   013060
LT44A1  013072
LT44B   013102
LT44C   013116
LT44D   013144
LT44E   013146
LT44F   013206
LT44IT  013022
LT44X   013226
LT44XX  013236
LT45    013242
LT45A   013350
LT45B   013364
LT45D   013414
LT45E   013326
LT45E1  013340
LT45F   013446
LT45IT  013270
LT45X   013472
LT45XX  013502
LT46    013506
LT46A   013570
LT46B   013636
LT46C   013652
LT46D   013704
LT46IT  013534
LT46X   013730
LT46XX  013740
LT47    013744
LT47A   014034
LT47B   014046
LT47C   014052
LT47IT  013760
LT47X   014124
LT5     003714
LT5A    003734
LT5B    003754
LT5C    003764
LT5D    003774
LT5E    004016
LT5ER1  004030
LT5ER2  004052
LT5IT   003722
LT5X    004074
LT50    014134
LT50IT  014150
LT50X   014274
LT51    014310
LT51IT  014324
LT51X   014450
LT6     004104
LT6A    004122
LT6A1   004120
LT6B    004126
LT6D    004146
LT6ER1  004166
LT6IT   004106
LT6X    004210
LT7     004220
LT7A    004236
LT7B    004256
LT7C    004226
LT7ER1  004276
LT7IT   004222
LT7X    004320
MMSG0   025646
MMSG1   025673
MMSG2   025771
MMSG3   026012
MMSG4   026050
MMSG5   026137
MR      000534
MSG1    020236
MSG10   021153
MSG11   021177
MSG12   021224
MSG13   021233
MSG14   021242
MSG15   021257
MSG16   021275
MSG18   021305
MSG19   021324
MSG2    020434
MSG2A   020475
MSG20   021343
MSG21   021375
MSG22   021433
MSG23   021454
MSG24   021507
MSG25   021535
MSG25A  021564
MSG26   021611
MSG27   021640
MSG3    020624
MSG30   021654
MSG31   021662
MSG32   021700
MSG33   021716
MSG34   021737
MSG35   021754
MSG36   021772
MSG37   022010
MSG4    020646
MSG40   022026
MSG41   022032
MSG42   022051
MSG43   022066
MSG44   022172
MSG45   022214
MSG46   022236
MSG47   022342
MSG5    020664
MSG50   022374
MSG51   022413
MSG53   022442
MSG54   022454
MSG55   022465
MSG56   022535
MSG57   022562
MSG58   022600
MSG59   022616
MSG6    020704
MSG60   022620
MSG62   022622
MSG63   022670
MSG7    020723
MSG8    020741
MSG8A   021002
MSG9    021131
MSLT1   022716
MSLT10  023366
MSLT11  023441
MSLT12  023506
MSLT13  023557
MSLT14  023624
MSLT15  023672
MSLT16  023740
MSLT17  024006
MSLT2   022772
MSLT20  024054
MSLT21  024133
MSLT22  024167
MSLT23  024224
MSLT24  024266
MSLT25  024326
MSLT26  024362
MSLT27  024416
MSLT3   023051
MSLT30  024452
MSLT31  024510
MSLT32  024544
MSLT33  024600
MSLT34  024634
MSLT35  024670
MSLT36  024732
MSLT37  024774
MSLT4   023133
MSLT40  025037
MSLT41  025104
MSLT42  025153
MSLT43  025207
MSLT44  025243
MSLT45  025323
MSLT46  025405
MSLT47  025441
MSLT5   023212
MSLT50  025506
MSLT51  025566
MSLT6   023256
MSLT7   023322
MTINT   016640
NRZOF   000662
NXTDRV  002152
NXTSLV  002224
OCTP    017446
OCTPE   017436
OCTPE1  017452
OCTPG   017624
OCTPGO  017642
OCTPG1  017646
OCTPO   017472
OCTP1   017512
OCTP2   017520
OCTP3   017610
OFL     017666
PATRN   001002
PCNTR   001016
PEXFL   000736
PFLG    000666
POST    027476
PRE     027354
PREFL   000770
PSW     000566
RBUFF   030232
RCOP    001010
RDAD    000762
RDATA   027104
RDRVF   001006
REGP    015174
REGS    000612
RTRN    000570
SAV1    000704
SAV2    000706
SAV3    000710
SCF     000732
SCOLP   000712
SCOPE   016210
SERFL   000772
SERNUM  000602
SKAT    001014
SLAF    000722
SLVN    000664
SN      000540
SNPG    020050
SNPT    017772
SSCF    000724
START   001330
STATC   001012
STATF   001004
STATIC  014464
STFLG   000740
STSCD   002402
STO     001662
ST1     001704
ST2     002066
SWR     000570
SWREG   000176
TADX    001324
TC      000542
TEMP1   000674
TEMP2   000676
TEMP3   000700
TEND    002436
TENDX   002534
TEX     017434
TIB     000616
TIMER   010350
TIMER0  010374
TIMER1  010324
TIMOK   010442
TIMON   010256
TIMOVF  010406
TINER   017236
TKB     000574
TKS     000572
TLAST   001326
```

```
TMS1    026213          TMS28   026457          TMS6    026251          TR13    000654          WCOP0   001042
TMS10   026301          TMS29   026465          TMS7    026257          TR14    000656          WCOP2   001030
TMS11   026307          TMS3    026227          TMS8    026265          TR15    000660          WCS1    001020
TMS12   026315          TMS30   026472          TMS9    026273          TSCD    002106          WCS2    001022
TMS13   026323          TMS31   026500          TO8     000614          TSCDA   002302          WDATA   026704
TMS14   026331          TMS32   026506          TOG     017420          TSCDO   002310          WDS     001024
TMS15   026337          TMS33   026514          TP8     000600          TSCD1   002316          WER     001026
TMS16   026345          TMS33A  026522          TPS     000576          TSCD2   002344          WMSG27  027304
TMS17   026353          TMS33B  026536          TREF    000734          TSCD3   002362          WMSG31  027317
TMS18   026361          TMS33C  026551          TR00    000626          TSTTBL  001054          WPGFL   001000
TMS19   026367          TMS33D  026573          TR01    000630          TTIN    017256          WTAD    000756
TMS2    026221          TMS33E  026616          TR02    000632          TTINT   016650          W2FLG   000764
TMS20   026375          TMS34   026636          TR03    000634          TTOUT   017320          $DONE   002444
TMS21   026404          TMS35   026644          TR04    000636          TTR     016776          $ENDAD  002502
TMS22   026412          TMS36   026653          TR05    000640          T24FL   000744          $MNEW   026203
TMS23   026421          TMS37   026661          TR06    000642          UDES    000776          $MSWR   026173
TMS24   026427          TMS38   026667          TR07    000644          VECT    000610          $SVPC = 000764
TMS25   026435          TMS39   026675          TR10    000646          WAM     000750          .       = 030234
TMS26   026443          TMS4    026234          TR11    000650          WBUFF   027620          .RESTO  020214
TMS27   026451          TMS5    026242          TR12    000652          WC      000512          .SAVE   020172

. ABS.  030234      000

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 DZTEAA,DZTEAA/SOL=DZTEAA
 RUN-TIME: 4 8 .6 SECONDS
 RUN-TIME RATIO: 200/13=14.8
 CORE USED:  7K  (13 PAGES)
```