

TM03/TE16

CONTROL LOGIC TEST PART 1
MD-11-DZTEA-A

EP-DZTEA-A-DL-A

JUN 1977

COPYRIGHT © 1977

digital

FICHE 1 OF 1

MADE IN USA

This microfiche card contains a grid of frames. The frames are arranged in approximately 12 rows and 10 columns. Each frame contains technical data, likely test results or control logic information, presented in a structured format with columns and rows of text. The data is too small to be legible in this image.

A small microfiche frame located in the bottom right corner of the card, containing technical data in a structured format.

.REM x

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZTEA-A-D
PRODUCT NAME: TMO3/TE16 CONTROL LOGIC TEST PART I
DATE CREATED: 21 FEB 77
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: J. G. ADAMS

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

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

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

COPYRIGHT (C) 1977, BY DIGITAL EQUIPMENT CORPORATION

CO1

TND3/TE16 CONTROL LOGIC TEST PART I
DZTEAA.P11 01-APR-77 12:00

MACY11 27(1006) 01-APR-77 12:02 PAGE 2

TABLE OF CONTENTS

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

1. ABSTRACT

THIS PROGRAM IS DESIGNED TO SEQUENTIALLY TEST ALL CONTROL LOGIC FUNCTIONALY OF THE TMD3. 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 TMD3 THE STRUCTURE AND ITS MAINTAINENCE MODES.

2. REQUIREMENTS (HARDWARE)

- A. ANY PDP-11 PROCESSOR
- B. BK OF CORE
- C. CONSOLE TTY
- D. TMD3 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 IC TO RESTART PROGRAM (2200)

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 TMDP CHAIN MODE THE
PROGRAM WILL NOT EST TMD3 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 RESPONSES 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> (RESPONSE) [LOCATION:]

TMD3-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:]
TMD3 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: XXXXXX 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 INTERVENTION 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 (M8909 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 (M8705)
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	MC	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	MC	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)
 UNEXPTED ERROR BITS

CS1	MC	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	MC	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: (SMD-SM5)

WHEN SMD-SM5 ARE SET TO ZERO (00), THE SCHEDULAR WILL EXECUTE ALL TESTS IN SEQUENCE. IF SMD-SM5 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 TMO3 AT THE ADDRESSES SPECIFIED BY THE OPERATOR. TEST OCCURS IMMEDIATELY AFTER DRIVE SELECTION.

PROGRAMMED SEQUENCE: FOR EACH TMO3 ADDRESS (0-7) THE C1 REGISTER IS READ, AND THE NON-EXISTANT DRIVE (NED) BIT IS CHECKED. NED IS SET WHEN THE TMO3 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	(MB1, 2, 3), (MBI3), (MBI4), (MBI5)
RH REGISTER SELECT	(BCTA)
TMO3 REGISTER SELECT	(MBI2)
MASSBUS REGISTER SELECT LINES	(MB1, 2)
PARITY TREE	(MBI4)
CPAR, ILR BITS	(MBI11)

LOGIC TEST #3: CONTROL BUS

PURPOSE: VERIFY THAT ALL CONTROL LINES PROPERLY TRANSMIT

K01

TMO3/TE16 CONTROL LOGIC TEST PART I
DZTEAA.P11 01-APR-77 12:00

MACY11 27(1006) 01-APR-77 12:02 PAGE 10

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
C-BUS MULTIPLEXERS
ERROR BIT
MCPE BIT

(MB1,2,3)
(MB13,4,5,8)(TCCM7)(MR)
(MB111)
(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
SLAVE ADDRESS BITS
SLAVE ADDRESS LINES
TE16 ADDRESS DECODE
SPR BIT

(MB12)
(MR6)
(M8907,2-2), (LAW6)
(LAW6)
(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
 M.R. FUNCTION DECODE
 M.R. MULTIPLEXOR

(MR2,3,5)
 (MR5)
 (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

CIRCUITS

PRINT REFERENCE

TMD3 REGISTER SELECT
 TC FLIP-FLOPS, MULTIPLEXERS

(MB12)
 (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

<u>CIRCUITS</u>	<u>PRINT REFERENCE</u>
TMD3 REGISTER SELECT	(MB12)
FRAME COUNT REGISTER	(MB18)
FRAME COUNT MULTIPLEXERS	(MB110)

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

<u>CIRCUITS</u>	<u>PRINT REFERENCE</u>
TMD3 REGISTER SELECTION	(MB12)
FUNCTION CODE FLOPS	(MB15)
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

CIRCUITPRINT REFERENCE

FCS
SET ILF
SET NEF
GO BIT
GO BIT MULTIPLEXER
SET ILR

M818
M817
M817
M815
M86
M812

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 TCCH7

LOGIC TEST #13: INTERRUPT TEST

PURPOSE: TO VERIFY THE OPERATION OF THE RM 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, MRL, 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, MRL, DPR, DRY, BOT ARE CHECKED.

LIKELY FAULT LOCATION: M8910, SLAVE CABLE, M8903

CIRCUIT

PRINT REFERENCE

MOL
 MRL
 DPR
 DRY
 BOT

LAM6, TCCM7, M8908, M9001YA, YC
 LAM8, TCCM7, M8908, M9001YA, YC
 TCCM7
 TCCM7
 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
ATA

LAW8, M8913, M8913YA, TCCM7
M813

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, MEL, DPR, DRY

LIKELY FAULT LOCATION: M8910, SLAVE CABLE, M8903

CIRCUIT

PRINT REFERENCE

SSC
EOT
SLA

LAW8, M8913, M8913YA, TCCM7
LAW6, TCCM7, M8908YA, M8913YA
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: M6909

CIRCUIT

PRINT REFERENCE

SET ILF DECODE
ILF FLOP
ILF MULTIPLEXER

M615, M617
M611
M610

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

CIRCUITPRINT REFERENCE

RMR DECODE
 RMR FLOP
 RMR MULTIPLEXER

MBI2
 MBI11
 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

CIRCUITPRINT REFERENCE

MASSBUS PARITY TREE
 CPAR FLOP
 CPAR MULTIPLEXER

MBI4
 MBI11
 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

<u>CIRCUIT</u>	<u>PRINT REFERENCE</u>
FORMAT BITS	MR6
ILF DECODE	BF3
ILF FLOP	MBI11
ILF MULTIPLEXERS	MBI10

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

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

LIKELY FAULT LOCATIONS: DBUS LINES, M8905, M8906

<u>CIRCUIT</u>	<u>PRINT REFERENCE</u>
MM CLK	MRS
WRT CLK GENERATION	TCCM4
DPAE FLOP	MBI11
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: M8909

CIRCUIT

PRINT REFERENCE

NEF FLOP
NEF MULTIPLEXER
SET NEF

MBI11
MBI10
MBI7

LOGIC TEST #26: FRAME COUNT ERROR

PROGRAMMED SEQUENCE: SET MC 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: M8909, MB CABLE, M8903, M8905

CIRCUITS

PRINT REFERENCE

RUN LINE
EBL PLS
FCE FLOP
SHUTDOWN LOGIC
MAINT. FUNCTION DECODE

MB1
MBI9
MBI11
TCCM5
MRS

LOGIC TEST #27: ILLEGAL REGISTER

PROGRAMMED SEQUENCE: IF THE RM 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
REGISTER SELECT DECODE
ILR FLOP

MB1, MB2
MBI2
MBI11

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 MRP 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 MRP 3 CODE INHIBITS THE MASSBUS MCLK RECEIVER CREATING A SITUATION WHERE SCLK IS NOT FOLLOWED BY A WRITE CLOCK.

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

CIRCUITS

PRINT REFERENCES

DTE FLOP
CRIPPLE OCCUPIED FUNCTION
MRP 3 FUNCTION
PREVIOUS OCCUPIED CHECK
CHECK FOR MCLK
MM CLK

MBI11
MRS
MRS
MBI7
BF2
MRS

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

OPI TIMER
OPI FLOP
OPI TIMER CONTROL

PRINT REFERENCES

TCCM5
MBI11
MBI7

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

UNSAFE FLOP
SET UNSAFE
MOL GENERATION

PRINT REFERENCES

MBI11
MBI7
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: MB909, MB903

CIRCUITS

PRINT REFERENCES

SPACE FUNCTION DECODE
PIP GENERATION
STATUS REGISTER

MB15
TCCM7
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: MB905, SLAVE BUS, MB911, MB903

CIRCUITS

PRINT REFERENCES

DENSITY BITS
DENSITY LINES
PES CIRCUIT
PES STATUS BIT

MR6
SBC
SC3
TCCM7

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

SETUP FORMAT AND MRP-3 ARE SET, READ COMMAND IS ISSUED,
TCW IS CHECKED. DRIVE IS INITIALIZED, TAPE CONTROL REG-
ISTER 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
FCS MULTIPLEXER

MB18
TCM7

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

CIRCUITSPRINT REFERENCES

ACCL BIT, MOTION DELAY COUNTER
CLOCK

TCCM3
SC2

LOGIC TEST 40: PE TAPE MARK (TM)PROGRAMMED SEQUENCE:

DRIVE IS INITIALIZED, WAM0 IS SET, WRITE TAPE MARK IS SET. AFTER DELAY TAPE MARK BIT IS CHECKED. WAM0 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

CIRCUITSPRINT REFERENCES

TAPE MARK DETECTOR
TAPE MARK MULTIPLEXER
ENVELOPE SIGNALS
WRITE DATA BUFFER
RDA MULTIPLEXERS
WRITE TAPE MARK FUNCTION
WAM0 SIGNAL

TCPE4, TCPES
TCCM7
DS 3, 5, 7
TCCM2
TCCM6
MBIS
MRS

M02

TMD3/TE16 CONTROL LOGIC TEST PART I
DZTEAA.P11 01-APR-77 12:00

MACY11 27(1006) 01-APR-77 12:02 PAGE 25

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
RSDO MULTIPLEXER
RDA MULTIPLEXERS
TM DETECTOR
ILLEGAL TAPE MARK FLOP

TCCM2
TCCM6
TCCM6
CNRZ4
CNRZ4

THE NEXT 5 TESTS CONSISTS OF WRITING ON TAPE USING MAIN-
TENANCE 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
CRC CHECK CIRCUIT

MRS
CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL)
WHICH ASSERTS MD(SB) 5L 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 TMO3
WRITE DATA LINES SHOULD BE ASSERTED.

LIKELY FAULT LOCATIONS: M8505, M8903, M8910, M8904

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
WRITE LINE DRIVERS
WRITE HEAD DRIVERS
LRC CHECKING

MRS
TCCM2
LAW3, 4
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 CORRECTABLE DATA ERROR. THE DEAD TRACK REGISTER IS CHECKED FOR BIT 1.

LIKELY FAULT LOCATIONS: M8905, M8901, M8902

CIRCUITS

PRINT REFERENCES

MM FUNCTION DECODE
 BIT STROBE CIRCUIT
 DEAD TRACK FLOP
 DEAD TRACK REGISTER

MRS
 DS4
 D55, TCPE2
 MR4

LOGIC TEST 45: PE INCORRECTABLE DATA

REPEAT OF TEST 44, EXCEPT THAT THE MAINT. MODE FUNCTION GOUNDS BITS STROBE FOR BITS 1, 2 AND THE HD 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

PEF
WRITE BUFFER
MM DECODE

PRINT REFERENCES

TCPE2
TCCM2
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 MB18

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
1172
1173

%

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

1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198

;FEB 77
;R. BARNES
;MCALL .SACT11,.SEOP,SCATCH,\$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


```

1245                                     ;REGISTER EQUIVS*****
1246                                     RO=%0
1247         000000                       R1=%1
1248         000001                       R2=%2
1249         000002                       R3=%3
1250         000003                       R4=%4
1251         000004                       R5=%5
1252         000005                       SP=%6
1253         000006                       PC=%7
1254         000007
1255
1256
1257                                     ;ACT11 HOOK *****
1258         000764                       $$VPC=.           ;SAVE CURRENT LOCATION CTR
1259         000046                       .=46
1260 000046 002502                       .WORD SENDAD      ;SET LOCATION 46
1261         000052                       .=52
1262 000052 000000                       .WORD 0           ;SET LOCATION 52 = 0
1263         000764                       .=$$VPC          ;RESTORE LOCATION CTR
1264
1265                                     ;TTY INTERRUPT VECTOR*****
1266
1267         000060                       .=60
1268 000060 016650                       .WORD TTINT      ;TTY INTERRUPT HEADER ADDRESS
1269 000062 000340                       .WORD 340       ;PRIORITY LEVEL 7
1270
1271                                     ;SOFTWARE SWITCH REGISTER*****
1272                                     ;USED IF HARDWARE SMR = 177777 OR NOT AVAILABLE
1273         000176                       .=176
1274 000176 000000                       SWREG: .WORD 0   ;SOFTWARE SWITCH REGISTER
1275
1276                                     ;START ADDRESS*****
1277         000200                       .=200
1278 000200 000137 001330                       JMP START ;PROGRAM START
1279
1280                                     ;RESTART ADDRESS*****
1281         000210                       .=210
1282 000210 000137 002066                       JMP ST2
1283
1284                                     ;TM03 INTERRUPT VECTOR*****
1285
1286         000224                       .=224
1287 000224 016640                       MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
1288 000226 000340
1289

```



```

1290
1291          000510
1292          .=510
1293          ;MASS BUS REGISTER EQUIVS*****
1294 000510 172440      CI:      172440
1295 000512 172442      MC:      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

```

;FLAGS AND COUNTERS*****

1334			
1335			
1336	000614	000000	T08:
1337	000616	000000	T18:
1338	000620	000000	HDRFL:
1339	000622	000000	EMADOR:
1340	000624	000000	DRVN:
1341	000626	000000	TRO0:
1342	000630	000000	TRO1:
1343	000632	000000	TRO2:
1344	000634	000000	TRO3:
1345	000636	000000	TRO4:
1346	000640	000000	TRO5:
1347	000642	000000	TRO6:
1348	000644	000000	TRO7:
1349	000646	000000	TR10:
1350	000650	000000	TR11:
1351	000652	000000	TR12:
1352	000654	000000	TR13:
1353	000656	000000	TR14:
1354	000660	000000	TR15:
1355	000662	000000	NRZOF:
1356	000664	000000	SLVN:
1357	000666	000000	PFLG:
1358	000670	000000	RTRN:
1359	000672	000000	ERADD:
1360	000674	000000	TEMP1:
1361	000676	000000	TEMP2:
1362	000700	000000	TEMP3:
1363	000702	000000	ITCNT:
1364	000704	000000	SAV1:
1365	000706	000000	SAV2:
1366	000710	000000	SAV3:
1367	000712	000000	SCOLP:
1368	000714	000000	ITRLP:
1369	000716	000000	EXFL:
1370	000720	000000	ATAF:
1371	000722	000000	SLAF:
1372	000724	000000	SSCF:
1373	000726	000000	ERRF:
1374	000730	000000	ASF:
1375	000732	000000	SCF:
1376	000734	000000	TREF:
1377	000736	000000	PEXFL:
1378	000740	000000	STFLG:
1379	000742	000000	LTADD:
1380	000744	000000	T24FL:
1381	000746	000000	ADDFL:
1382	000750	000000	WAM:
1383	000752	000000	FUN:
1384	000754	000000	DATC:
1385	000756	000000	WTAD:
1386	000760	000000	DATAD:
1387	000762	000000	RDAD:
1388	000764	000000	W2FLG:
1389	000766	000000	DERFL:

1390 000770 000000
 1391 000772 000000
 1392 000774 000000
 1393 000776 000000
 1394 001000 000000
 1395 001002 000000
 1396 001004 000000
 1397 001006 000000
 1398 001010 000000
 1399 001012 000000
 1400 001014 000000
 1401 001016 000000
 1402
 1403
 1404
 1405 001020 000000
 1406 001022 000000
 1407 001024 000000
 1408 001026 000000
 1409
 1410
 1411
 1412 001030 000005
 1413 001032 000005
 1414 001034 000012
 1415 001036 000012
 1416 001040 000000
 1417 001042 000017
 1418 001044 000017
 1419 001046 000017
 1420 001050 000017
 1421 001052 000000

PREFL: 0
 SERFL: 0
 CRCNT: 0
 UDES: 0
 WPGFL: 0
 PATRN: 0
 STATF: 0
 RDRVF: 0
 RCDP: 0
 STATC: 0
 SKAT: 0
 PCNTR: 0 ;PASS COUNTER

 ;EXPT WRAP STATUS*****
 WCS1: 0
 WCS2: 0
 WDS: 0
 WER: 0

 ;CORE DUMP PATTERNS*****
 WCDP2: 5
 5
 12
 12
 0
 WCDPO: 17
 17
 17
 17
 0

1422		
1423		
1424		
1425	001054	000000
1426	001056	000000
1427	001060	002552
1428	001062	002552
1429	001064	003026
1430	001066	003026
1431	001070	003232
1432	001072	003234
1433	001074	003414
1434	001076	003414
1435	001100	003714
1436	001102	003722
1437	001104	004104
1438	001106	004106
1439	001110	004220
1440	001112	004222
1441	001114	004334
1442	001116	004336
1443	001120	004460
1444	001122	004462
1445	001124	004704
1446	001126	004706
1447	001130	005100
1448	001132	005110
1449	001134	005202
1450	001136	005242
1451	001140	005312
1452	001142	005352
1453	001144	005422
1454	001146	005462
1455	001150	005532
1456	001152	005572
1457	001154	005642
1458	001156	005656
1459	001160	006004
1460	001162	006020
1461	001164	006150
1462	001166	006164
1463	001170	006274
1464	001172	006310
1465	001174	006424
1466	001176	006440
1467	001200	006750
1468	001202	006756
1469	001204	007102
1470	001206	007110
1471	001210	007316
1472	001212	007342
1473	001214	007434
1474	001216	007456
1475	001220	007752
1476	001222	007760
1477	001224	010564

;LOGIC TEST ENTRY TABLE*****

TSTTBL: 0
0
LT1
LT1
LT2
LT2
LT3
LT3IT
LT4
LT4
LT5
LT5IT
LT6
LT6IT
LT7
LT7IT
LT10
LT10IT
LT11
LT11IT
LT12
LT12IT
LT13
LT13IT
LT14
LT14IT
LT15
LT15IT
LT16
LT16IT
LT17
LT17IT
LT20
LT20IT
LT21
LT21IT
LT22
LT22IT
LT23
LT23IT
LT24
LT24IT
LT25
LT25IT
LT26
LT26IT
LT27
LT27IT
LT30
LT30IT
LT31
LT31IT
LT32

K03

TMD3/TE16 CONTROL LOGIC TEST PART I
DZTEAA.P11 01-APR-77 12:00.

MACY11 27(1006) 01-APR-77 12:02 PAGE 36

1478	001226	010600	LT32IT
1479	001230	010726	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	
1510	001326	000051	

TADX: .WORD TEND
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$,@#4        ;SET NEW VECTOR
1519 001352 005037 000006      CLR      @#6           ;AND PSW
1520 001356 022777 177777 177204      CMP      @-1,@SWR      ;USE SOFTWARE SWITCH IF HARDWARE
1521 001364 001402          BEQ      2$           ;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
1534 001434 001413          BEQ      52$
1535 001436 000406          BR       51$
1536 001440 012737 000176 000570 50$: MOV      @SWREG,SWR  ; INVOKE SOFTWARE SWR
1537 001446 012777 100000 177114 51$: 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$: CMPB     @#,@#41    ; BRANCH IF NOT LOADED VIA TMDP
1542 001472 001004          BNE     5$
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      MOV     @#,@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
1553 001546 012701 000007          MOV     @7,R1        ; SET ADDRESS SAVE LOC
1554 001552 012702 176400          MOV     @176400,R2  ; SET SIZE OF RESPONSE
1555 001556 012703 172300          MOV     @172300,R3  ; SET UPPER LIMIT
1556 001562 004737 016776          JSR     PC,TTR       ; SET LOWER LIMIT
1557 001566 012704 022214          MOV     @MSG45,R4    ; GO GET RESPONSE
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	#INTINT,(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	#TOB,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 TMO3 DRIVE #
1582	001716	004737	017320	JSR	PC,TTOUT	
1583	001722	013703	000624	MOV	DRVN,R3	;GET CURRENT DRIVE #
1584	001726	004737	017446	JSR	PC,OC1P	;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,OC1P	;AND PRINT IT
1594	001776	012705	000664	MOV	#SLVN,R5	;TTR ROUTINE RETURNS RESPONSE TO (R5)
1595	002002	012701	000002	MOV	#2,R1	;LIMIT RESPONSE 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,OC1P	;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
1616 ;TEST SCHEDULAR*****
1617 002106 052777 000100 176456 TSCD: BIS #100,ATKS ;SET KEYBOARD IE BIT
1618 002114 005037 001000 CLR WPGFL ;CLEAR WRAP PATRN FLAG
1619 002120 005037 000740 CLR STFLG ;CLEAR SINGLE TEST FLAG
1620 002124 017700 176440 MOV #SWR,RO
1621 002130 042700 177700 BIC #177700,RO ;BRANCH IF SINGLE
1622 002134 001122 BNE TSCD ;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 IS: MOV #40,ACS ;: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 SDONE ;:FOR AVAILABILITY
1631 002202 013777 000624 176310 MOV DRVN,ACS ;:LOAD DRIVE #
1632 002210 005777 176274 TST AC1 ;:ACCESS DRIVE
1633 002214 032777 010000 176276 BIT #10000,ACS ;:BRANCH IF DRIVE NON EXISTANT
1634 002222 001356 BNE IS ;:(NED = 1)
1635 002224 005237 000664 NXTSLV: INC SLVN ;:STEP SLAVE # AND BRANCH
1636 002230 001011 BNE IS ;:IF NOT SLAVE 0
1637 002232 005737 000624 TST DRVN ;:BRANCH IF NOT DRIVE # 0
1638 002236 001006 BNE IS
1639 002240 122737 000006 000041 CMPB #6,#41 ;:BRANCH IF NOT THDP
1640 002246 001002 BNE IS
1641 002250 005237 000664 INC SLVN ;:STEP TO SLAVE # 1
1642 002254 022737 000010 000664 IS: CMP #10,SLVN ;:BRANCH IF ALL SLAVES TESTED
1643 002262 001733 BEQ NXTDRV ;:FOR AVAILABILITY
1644 002264 013777 000664 176250 MOV SLVN,ATC ;:LOAD SLAVE UNIT #
1645 002272 032777 002000 176236 BIT #2000,ATC ;: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 TSCD0: 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,RO ;:SET POINTER TO TEST
1653 002342 000110 JMP (RO) ;: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 WPGFL ;:CLEAR WRAP DATA GENERATOR FLAG
1658 002362 005737 000740 TSCD3: TST STFLG ;:SE IF SINGLE TEST
1659 002366 001750 BEQ TSCD0 ;:IF NOT: BR
1660 002370 017700 176174 MOV #SWR,RO
1661 002374 042700 177700 BIC #177700,RO ;: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,RO ;:SEE IF EXCEEDED TESTS
1665 002414 002410 BLT TEND ;:IF SO: BR
1666 002416 006300 ASL RO
1667 002420 006100 ROL RO ;:SET TABLE MODIFIER
1668 002422 012737 001054 000742 MOV #TSTTBL,LTADD
1669 002430 060037 000742 ADD RO,LTADD ;:SET TEST POINTER

```


1670	002434	000730			BR	TSCD1	
1671	002436	005737	001416		TEND: TST	CHNFLG	; BRANCH IF IN CHAIN MODE
1672	002442	001270			BNE	NXTSLV	; STEP TO NEXT SLAVE
1673	002444	012704	022032		SOONE: MOV	MSCH1, R4	
1674	002450	004737	017320		JSR	PC, TOUT	; 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	RO	; DELAY WAITING FOR
1678	002466	005300			IS: DEC	RO	
1679	002470	001376			BNE	IS	
1680	002472	013700	000042		MOV	2842, RO	; GET ACT11 RETURN ADDRESS
1681	002476	001405			BEQ	HERE	; BRANCH IF NOT ACT11
1682	002500	000005			RESET		
1683	002502	004710			SENDAD: JSR	PC, (RO)	
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			TENDX		
1690	002522	032777	010000	176040	BIT	810000, 2SWR	; SEE IF HALT ON PASS
1691	002530	001001			BNE	TENDX	; IF NOT: BR
1692	002532	000000			HALT		
1693	002534	012737	000001	001014	TENDX: MOV	81, SKAT	; SET SKIP ADDRESS TEST FLAG
1694	002542	005237	001016		INC	PCNTR	; BUMP PASS COUNTER
1695	002546	000137	002106		JMP	TSCD	; RESTART


```

;LOGIC TEST 2: REGISTER ADDRESSING*****
1745
1746
1747 003026 000240
1748 003030 012777 000040 175462
1749 003036 013777 000624 175454
1750 003044 012737 022772 000622
1751 003052 012705 000510
1752 003056 012700 000016
1753 003062 012702 000626
1754 003066 011501
1755 003070 011112
1756 003072 032777 020000 175410
1757 003100 001402
1758 003102 004737 003132
1759 003106 032777 000002 175410
1760 003114 001402
1761 003116 004737 003150
1762 003122 022225
1763 003124 005300
1764 003126 001357
1765 003130 000434
1766 003132 012737 000002 000716
1767 003140 012737 020646 000672
1768 003146 000415
1769 003150 012737 000002 000716
1770 003156 012737 020664 000672
1771 003164 000406
1772 003166 012737 000001 000716
1773 003174 012737 020646 000672
1774 003202 012737 003216 000712
1775 003210 004737 014670
1776 003214 000207
1777 003216 005726
1778 003220 000722
1779 003222 004737 016240
1780 003226 000137 002344

LT2: NOP
LT2IT: MOV #40,ACS
MOV DRVN,ACS
MOV #MSLT2,EMADDR
MOV #C1,R5
MOV #16,R0
MOV #TR00,R2
LT2A: MOV (R5),R1
MOV (R1),(R2)
BIT #20000,AC1
BEQ LT2B
JSR PC,LT2ER1
LT2B: BIT #2,ACR
BEQ LT2C
JSR PC,LT2ER2
LT2C: CMP (R2)+,(R5)+
DEC R0
BNE LT2A
BR LT2X
LT2ER1: MOV #2,EXFL
MOV #MSG4,ERADD
BR LT2ERG
LT2ER2: MOV #2,EXFL
MOV #MSG5,ERADD
BR LT2ERG
LT2ER3: MOV #1,EXFL
MOV #MSG4,ERADD
LT2ERG: MOV #LT2LP,SCOLP
JSR PC,LTGER
RTS PC
LT2LP: TST (SP)+
BR LT2A
LT2X: JSR PC,ITER
JMP TSCD2

;INIT
;SELECT DRIVE
;SAVE LT2 HEADER ADDRESS
;SET ADDRESS OF FIRST REGISTER
;SET NUMBER OF REGISTERS
;SET START OF REGISTER BUFFER
;READ REGISTER
;SEE IF ERROR
;IF NOT: BR
;ELSE GO TO ERROR 1
;SEE IF ILR
;IF NOT: BR
;ELSE GO TO ERROR 2
;BUMP ADDRESS
;CONTINUE FOR ALL REGISTERS
;FLAG NOT EXPECTED
;POINT TO CONTROLLER ERROR
;GO TO ERROR
;FLAG NOT EXPECTED
;POINT TO DRIVE ERROR
;GO TO ERROR
;FLAG EXPECTED
;POINT TO DRIVE
;SET SCOPE ADDRESS
;GO PRINT
;ELSE CONTINUE
;RESET STACK
;LOOP
;GO SEE IF ITERATIONS
;RETURN TO SCHED

```

```

;LOGIC TEST 3: CONTROL BUS*****
1781
1782
1783 003232 000240
1784 003234 012737 023051 000622 LT3: NOP
1785 003242 012701 000001 LT3IT: MOV #MSLT3,EMADDR ;SET TEST HEADER
1786 003246 012700 000020 MOV #1,R1 ;PRESET PATTERN 1
1787 003252 004737 016330 LT3A: JSR PC,INIT1 ;SET PATTERN CHANGE NUMBER
1788 003256 010177 175234 MOV R1,@FC ;GO INIT
1789 003262 032777 000010 175234 BIT #10,@ER ;WRITE TO FC
1790 003270 001012 BNE LT3ER1 ;SEE IF CPAR (TM03)
1791 003272 017702 175220 LT3B: MOV @FC,R2 ;IF SO: BR
1792 003276 032777 020000 175204 BIT #20000,@C1 ;READ FC
1793 003304 001017 BNE LT3ER2 ;SEE IF MCPE (RH)
1794 003306 005300 LT3C: DEC R0 ;IF SO: BR
1795 003310 001426 BEQ LT3X ;SEE IF DONE PATTERN CHANGES
1796 003312 006301 ASL R1 ;IF SO: BR
1797 003314 000756 BR LT3A ;CHANGE PATTERN
1798 003316 012737 021177 000672 LT3ER1: MOV #MSG11,ERADD ;CONTINUE
1799 003324 012737 003252 000712 MOV #LT3A,SCOLP ;SET ERROR CODE
1800 003332 017702 175160 MOV @FC,R2 ;SET SCOPE ADDRESS
1801 003336 004737 015776 JSR PC,LTGER1 ;GET DATA
1802 003342 000753 BR LT3B ;GO DO ERROR
1803 003344 012737 021153 000672 LT3ER2: MOV #MSG10,ERADD ;SET ERROR CODE
1804 003352 012737 003272 000712 MOV #LT3B,SCOLP ;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 TSCD2 ;RETURN TO SCHEDULAR

```

```

1814
1815
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  IS           ;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 IS:   BIT  #1000,JSWR  ;BRANCH IF MAN INTERVENTION
1824 003450 001430          BEQ  LT4A        ;NOT SELECTED
1825 003452 012704 021002          LT4G0: 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 003512 004737 016776          JSR  PC,TTR
1834 003516 005737 000674          TST  TEMP1
1835 003522 001472          BEQ  LT4X
1836 003524 005001          CLR  R1
1837 003526 012700 000010          MOV  #10,R0
1838 003532 012777 000040 174760 LT4A: MOV #40,ACS
1839 003540 013777 000624 174752 MOV  DRVN,ACS
1840 003546 010177 174770          MOV  R1,ATC
1841 003552 017703 174760          MOV  #01,R3
1842 003556 020137 000700          CMP  R1,TEMP3
1843 003562 001404          BEQ  LT4B
1844 003564 032703 002000          BIT  #2000,R3
1845 003570 001420          BEQ  LT4D
1846 003572 000423          BR   LT4ER1
1847 003574 032703 002000          LT4B: BIT  #2000,R3
1848 003600 001424          BEQ  LT4ER2
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
1853 003622 032777 001000 174740          BIT  #1000,JSWR
1854 003630 001427          BEQ  LT4X
1855 003632 005300          LT4D: DEC  R0
1856 003634 001712          BEQ  LT4G
1857 003636 005201          INC  R1
1858 003640 000734          BR   LT4A
1859 003642 012737 000001 000716 LT4ER1: MOV #1,EXFL
1860 003650 000403          BR   LT4ERG
1861 003652 012737 000002 000716 LT4ER2: MOV #2,EXFL
1862 003660 012737 023133 000622 LT4ERG: MOV #MSLT4,EMADDR
1863 003666 012737 021131 000672          MOV  #MSG9,ERADD
1864 003674 012737 003532 000712          MOV  #LT4A,SCOLP
1865 003702 004737 014670          JSR  PC,LTGER
1866 003706 000751          BR   LT4D
1867 003710 000137 002344          LT4X: JMP  TSCD2
1868

```

```

;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST*****
1869
1870
1871 003714 012737 023212 000622 LTS:  MOV #MSLT5,EMADDR ;SET TEST HEADER
1872 003722 004737 016330 LTSIT: JSR PC,INIT1 ;GO INIT
1873 003726 012700 000032          MOV #32,R0 ;SET LOOP FOR BITS 4-0
1874 003732 005001          CLR R1 ;SET TEST WORD
1875 003734 010177 174574 LTSA:  MOV R1,2MR ;SEND TEST WORD TO MR
1876 003740 017702 174570          MOV 2MR,R2 ;READ MR
1877 003744 042702 177740          BIC #177740,R2 ;MASK BITS 4-0
1878 003750 020102          CMP R1,R2 ;SEE IF EXPT = RECVD
1879 003752 001026          BNE LTSE1
1880 003754 005300 LTSB:  DEC R0
1881 003756 001402          BEQ LTSC ;IF DONE LOOP: BR
1882 003760 005201          INC R1 ;BUMP TEST WORD
1883 003762 000764          BR LTSA ;CONTINUE LOOP
1884 003764 012701 000015 LTS:  MOV #15,R1 ;SET TEST WORD + WAM 3
1885 003770 012700 001000 LTS:  MOV #1000,R0 ;SET LOOP FOR BITS 15-7
1886 003774 010177 174534 LTS:  MOV R1,2MR ;LOAD MR
1887 004000 017702 174530 LTS:  MOV 2MR,R2 ;READ MR
1888 004004 042702 000140          BIC #140,R2 ;MASK OUT BITS 5,6
1889 004010 020102          CMP R1,R2 ;SEE IF EXPT = RECVD
1890 004012 001401          BEQ LTSE ;IF SO: BR
1891 004014 000416          BR LTSE2 ;ELSE GO TO ERR 2
1892 004016 005300 LTS:  DEC R0
1893 004020 001425          BEQ LT5X ;IF DONE LOOP: BR
1894 004022 062701 000200          ADD #200,R1 ;BUMP TEST WORD
1895 004026 000762          BR LTSD ;CONTINUE LOOP
1896 004030 012737 021242 000672 LTSER1: MOV #MSG14,ERADD ;SET ERROR CODE
1897 004036 012737 003734 000712          MOV #LTS5,SCOLP ;SET SCOPE ADDRESS
1898 004044 004737 015776          JSR PC,LTGER1 ;GO TO ERROR
1899 004050 000741          BR LT5B ;CONTINUE
1900 004052 012737 021257 000672 LTSER2: MOV #MSG15,ERADD ;SET ERROR CODE
1901 004060 012737 003774 000712          MOV #LTS5,SCOLP ;SET SCOPE ADDRESS
1902 004066 004737 015776          JSR PC,LTGER1 ;GO TO ERROR
1903 004072 000751          BR LT5E ;CONTINUE
1904 004074 004737 016240 LTSX:  JSR PC,ITER ;GO SEE IF ITERATIONS
1905 004100 000137 002344          JMP TSCD2 ;RETURN TO SCHED
1906

```



```

;LOGIC TEST 7: FRAME COUNT BIT TEST*****
1932
1933
1934 004220 000240
1935 004222 012700 000003
1936 004226 012737 023322 000622
1937 004234 005001
1938 004236 004737 016330
1939 004242 010177 174250
1940 004246 017702 174244
1941 004252 020102
1942 004254 001010
1943 004256 005300
1944 004260 001417
1945 004262 022700 000001
1946 004266 001757
1947 004270 012701 177777
1948 004274 000760
1949 004276 012737 021324 000672
1950 004304 012737 004236 000712
1951 004312 004737 015776
1952 004316 000757
1953 004320 012700 000003
1954 004324 004737 016240
1955 004330 000137 002344
1956

LT7: NOP
LT7IT: MOV #3,R0 ;SET TEST NUMBER
LT7C: MOV #MSLT7,EMADDR ;SET TEST HEADER
CLR R1 ;SET TEST WORD
LT7A: JSR PC,INIT1 ;GO INIT
MOV R1,#FC ;CLEAR FRAME COUNT
MOV #FC,R2 ;READ FC
CMP R1,R2 ;SEE IF EXPT = RECVD
BNE LT7ER1
LT7B: DEC R0 ;SEE IF DONE ALL
BEQ LT7X ;IF SO: BR
CMP #1,R0 ;SEE IF RESET TEST
BEQ LT7C ;IF SO: BR
MOV #-1,R1 ;SET TEST WORD TO -1
BR LT7A ;CONTINUE
LT7ER1: MOV #MSG19,ERADD ;SET ERROR CODE
MOV #LT7A,SCOLP ;SET SCOPE ADDRESS
JSR PC,LTGER1 ;GO PRINT ERROR
BR LT7B ;ELSE CONTINUE
LT7X: MOV #3,R0 ;RESET TEST AMT
JSR PC,ITER ;GO SEE IF ITERATIONS
JMP TSCD2 ;RETURN TO SCHED
    
```



```

;LOGIC TEST 10: FUNCTION CODE BIT TEST*****
1957
1958
1959 004334 000240 LT10: NOP
1960 004336 012737 023366 000622 LT10IT: MOV #MSLT10,EMADDR ;SET TEST HEADER
1961 004344 012700 000003 MOV #3,R0 ;SET NUMBER OF TESTS
1962 004350 005001 LT10A1: CLR R1 ;SET TEST WORD
1963 004352 012777 000040 174140 LT10A: MOV #40,ACS ;INIT
1964 004360 013777 000624 174132 MOV DRVN,ACS ;SELECT DRIVE
1965 004366 010177 174116 MOV R1,AC1 ;WRITE C1
1966 004372 017702 174112 MOV AC1,R2 ;READ C1
1967 004376 042702 177701 BIC #177701,R2 ;MASK FUNCTION CODE
1968 004402 020102 CMP R1,R2 ;SEE IF EXPT = RECVD
1969 004404 001010 BNE LT10E1
1970 004406 005300 LT10B: DEC R0
1971 004410 001417 BEQ LT10X ;IF DONE ALL: BR
1972 004412 022700 000001 CMP #1,R0 ;SEE IF RESET TEST
1973 004416 001754 BEQ LT10A1 ;IF SO: BR
1974 004420 012701 000076 MOV #76,R1 ;SET TEST WORD
1975 004424 000752 BR LT10A ;DO SET TEST
1976 004426 012737 021343 000672 LT10E1: MOV #MSG20,ERADD ;SET ERROR CODE
1977 004434 012737 004352 000712 MOV #LT10A,SCOLP ;SET SCOPE ADDRESS
1978 004442 004737 015776 JSR PC,LTGER1 ;GO PRINT ERROR
1979 004446 000757 BR LT10B ;ELSE CONTINUE
1980 004450 004737 016240 LT10X: JSR PC,ITER ;GO SEE IF ITERATIONS
1981 004454 000137 002344 JMP TSCD2 ;RETURN TO SCHED

```

```

1982
1983
1984
1985 004460 000240
1986 004462 012737 023441 000622 LT11: NOP
1987 004470 004737 016330 LT11IT: MOV #MSLT11,EMADDR ;SET TEST HEADER
1988 004474 017702 174010 JSR PC,INIT1 ;GO INIT
1989 004500 032702 000001 BIT #1,R2 ;READ C1
1990 004504 001030 BNE LT11E1 ;SEE IF GO=0
1991 004506 012777 000015 174020 LT11B: MOV #15,SPR ;SELECT WAM 3
1992 004514 005077 173776 CLR #FC ;ASSURE FCS = 1
1993 004520 052777 001700 174014 BIS #1700,ATC ;ASSURE FMT OK
1994 004526 012777 000071 173754 MOV #71,AC1 ;SET READ+GO
1995 004534 017702 173750 MOV #AC1,R2 ;READ C1
1996 004540 032702 000001 BIT #1,R2 ;SEE IF GO =1
1997 004544 001424 BEQ LT11E2
1998 004546 004737 016330 LT11C: JSR PC,INIT1 ;GO INIT
1999 004552 017702 173732 MOV #AC1,R2 ;READ C1
2000 004556 032702 000001 BIT #1,R2 ;SEE IF GO=0
2001 004562 001444 BEQ LT11X ;IF S0:BR
2002 004564 000430 BR LT11E3 ;ELSE GO TO ERROR 3
2003 004566 012737 021375 000672 LT11E1: MOV #MSG21,ERADD ;SET ERROR CODE
2004 004574 012702 000001 MOV #1,R2 ;SET RCVD
2005 004600 005001 CLR R1 ;SET EXPT
2006 004602 012737 004462 000712 MOV #LT11IT,SCOLP ;SET SCOPE ADDRESS
2007 004610 004737 015776 JSR PC,LTGER1 ;GO PRINT ERROR
2008 004614 000734 BR LT11B ;ELSE CONTINUE
2009 004616 012737 021433 000672 LT11E2: MOV #MSG22,ERADD ;SET ERROR CODE
2010 004624 005002 CLR R2 ;SET RCVD
2011 004626 012701 000001 MOV #1,R1 ;SET EXPT
2012 004632 012737 004506 000712 MOV #LT11B,SCOLP ;SET SCOPE ADDRESS
2013 004640 004737 015776 JSR PC,LTGER1 ;GO PRINT ERROR
2014 004644 000740 BR LT11C ;ELSE CONTINUE
2015 004646 012737 021454 000672 LT11E3: MOV #MSG23,ERADD ;SET ERROR CODE
2016 004654 005001 CLR R1 ;SET EXPT
2017 004656 012702 000001 MOV #1,R2 ;SET RCVD
2018 004662 012737 004546 000712 MOV #LT11C,SCOLP ;SET SCOPE ADDRESS
2019 004670 004737 015776 JSR PC,LTGER1 ;GO PRINT ERROR
2020 004674 004737 016240 LT11X: JSR PC,ITER ;GO SEE IF ITERATIONS
2021 004700 000137 002344 JMP TSCD2 ;RETURN TO SCHED

```

```

2022
2023
2024
2025
2026 004704 000240
2027 004706 012737 023506 000622
2028 004714 004737 016330
2029 004720 032777 000200 173574
2030 004726 001426
2031 004730 012777 000015 173576
2032 004736 005077 173554
2033 004742 052777 001700 173572
2034 004750 012777 000071 173532
2035 004756 032777 000200 173536
2036 004764 001020
2037 004766 004737 016330
2038 004772 032777 000200 173522
2039 005000 001033
2040 005002 000422
2041 005004 012737 021507 000672
2042 005012 012737 004706 000712
2043 005020 004737 015770
2044 005024 000741
2045 005026 012737 021535 000672
2046 005034 012737 004730 000712
2047 005042 004737 015770
2048 005046 000747
2049 005050 012737 021564 000672
2050 005056 012737 004766 000712
2051 005064 004737 015770
2052 005070 004737 016240
2053 005074 000137 002344

;LOGIC TEST 12: DRIVE READY BIT*****

LT12: NOP
LT12IT: MOV #MSLT12,EMADDR ;SET TEST HEADER
JSR PC,INIT1 ;GO INIT
BIT #200,JD5 ;SEE IF DRY=1
BEQ LT12E1
LT12B: MOV #15,AMR ;SET WAM3
CLR #FC ;ASSURE FCS = 1
BYS #1700,ATC ;ASSURE FMT OK
MOV #71,AC1 ;SET READ+GO
BIT #200,JD5 ;SEE IF DRY=0
BNE LT12E2
LT12C: JSR PC,INIT1 ;GO INIT
BIT #200,JD5 ;SEE IF DRY=1
BNE LT12X ;IF SO: BR
BR LT12E3 ;ELSE GO TO ERROR 3
LT12E1: MOV #MSG24,ERADD ;SET ERROR CODE
MOV #LT12IT,SCOLP ;SET SCOPE ADDRESS
JSR PC,LTGER2 ;GO TO ERROR
BR LT12B ;CONTINUE
LT12E2: MOV #MSG25,ERADD ;SET ERROR CODE
MOV #LT12B,SCOLP ;SET LOOP ADDRESS
JSR PC,LTGER2 ;GO PRINT ERROR
BR LT12C ;CONTINUE
LT12E3: MOV #MSG25A,ERADD ;SET ERROR CODE
MOV #LT12C,SCOLP ;SET ERROR LOOP
JSR PC,LTGER2 ;GO PRINT ERROR
LT12X: JSR PC,ITER ;GO TO ITERATION SUBROUTINE
JMP TSCD2 ;RETURN TO SCHED

```

```

2053
2054
2055
2056 005100 005000
2057 005102 012737 023557 000622
2058 005110 004737 016330
2059 005114 012737 005172 000670
2060 005122 005077 173362
2061 005126 005077 173434
2062 005132 052777 000100 173350
2063 005140 005300
2064 005142 001376
2065 005144 012777 000340 173414
2066 005152 012737 021611 000672
2067 005160 012737 005110 000712
2068 005166 004737 015770
2069 005172 004737 016240
2070 005176 000137 002344

;LOGIC TEST 13: INTERRUPT TEST*****
LT13: CLR R0
MOV #MSLT13,EMADDR ;SET TEST HEADER
LT13IT: JSR PC,INIT1 ;GO INIT,SELECT DRIVE, SELECT ABOVE
MOV #LT13X,RTRN ;SET RETURN ADDRESS
CLR @C1 ;CLEAR CS1
CLR @PSW ;SET PRIORITY
BIS #100,@C1 ;BIT SET IE
LT13A: DEC R0
BNE LT13A ;AWAIT INTERRUPT
LT13E1: MOV #340,@PSW ;RESET PRIORITY
MOV #MSG26,ERADD ;SET ERROR CODE
MOV #LT13IT,SCOLP ;SET LOOP ADDRESS
LT13X: JSR PC,LTGER2 ;GO PRINT ERROR
JSR PC,ITER ;GO TO ITERATION SUBROUTINE
JMP TSCD2 ;RETURN TO SCHED
  
```

2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096

005202 032777 001000 173360
005210 001005
005212 005737 000740
005216 001433
005220 000137 016310
005224 012737 023234 000622
005232 012704 025673
005236 004737 016572
005242 004737 016330
005246 012701 014602
005252 017702 173244
005256 020102
005260 001410
005262 012737 005242 000712
005270 012737 021640 000672
005276 004737 015776
005302 004737 016240
005306 000137 002344

LT14: BIT #1000,JSWR
BNE LT14A
TST STFLG
BEQ LT14XX
JMP INMT
LT14A: MOV #MSLT14,EMADDR
MOV #MSG1,R4
LT14IT: JSR PC,INST
JSR PC,INIT1
MOV #14602,R1
MOV #05,R2
CMP R1,R2
BEQ LT14X
MOV #LT14IT,SCOLP
MOV #MSG27,ERADD
LT14X: JSR PC,LTGER1
LT14XX: JSR PC,ITER
JMP TSCD2

```

;THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS.
;THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TE16
;CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.

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

;SEE IF INHIB MAN TST
;IF NOT: BR
;SEE IF SINGLE TEST
;IF NOT: BR
;ELSE GO PRINT INHIB MSG
;SET TEST HEADER
;SET INSTRUCTION ONE
;GO DO INSTRUCTION
;INIT, SELECT DRIVE + SLAVE
;SET TEST WORD
;ASSURE MOL,WRL,DPR,DRY,BOT

;IF SO: BR
;SET LOOP ADDRESS
;SET ERROR CODE
;GO PRINT ERROR
;GO SEE IF ITERATION
;RETURN TO SCHED
    
```

```

2097
2098
2099
2100 005312 032777 001000 173250 LT15: BIT 01000,JSNR ;SEE IF INHIB MAN TST
2101 005320 001005 BNE LT15A ;IF NOT: BR
2102 005322 005737 000740 TST STFLG ;SEE IF SINGLE TEST
2103 005324 001433 BEQ LT15XX ;IF NOT: BR
2104 005330 000137 016310 JMP INMT ;ELSE GO PRINT INHIB MSG
2105 005334 012737 023672 000622 LT15A: MOV @MSLT15,EMADDR ;SET TEST HEADER
2106 005342 012704 025771 MOV @MSC2,R4
2107 005346 004737 016572 JSR PC,INST ;PRINT INSTRUCTION
2108 005352 004737 016340 LT15IT: JSR PC,INIT2 ;GO INIT, SELECT DRIVE, SLAV
2109 005356 012701 100700 MOV @100700,R1 ;SET TEST WORD
2110 005362 017702 173134 MOV @DS,R2 ;READ STATUS
2111 005366 020102 CMP R1,R2 ;SEE OF EXPT=RCVD
2112 005370 001410 BEQ LT15X
2113 005372 012737 005352 000712 MOV @LT15IT,SCOLP ;SET LOOP ADDRESS
2114 005400 012737 021640 000672 MOV @MSC27,ERADD ;SET ERROR CODE
2115 005406 004737 015776 JSR PC,LTGER1 ;GO PRINT ERROR
2116 005412 004737 016240 LT15X: JSR PC,ITER ;GO SEE IF ITERATIONS
2117 005416 000137 002344 LT15XX: JMP TSCD2 ;RETURN TO SCHED

```

```

2118
2119
2120 ;LOGIC TEST 16: STATUS AT EOT, OFFLINE LOADED, NO WRITE RING*****
2121 005422 032777 001000 173140 LT16: BIT #1000,JSWR ;SEE IF INHIB MAN TST
2122 005430 001005 ;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 INMT ;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 #DS,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
2142
2143 005532 032777 001000 173030 LT17: BIT #1000,JSWR ;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 #MSG4,R4
2150 005566 004737 016572 JSR PC,INST ;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 #DS,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,ERADD ;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

```


2161
2162
2163
2164
2165
2166
2167
2168
2169 005642 012737 024054 000622
2170 005650 012737 005670 000712
2171 005656 012700 000022
2172 005663 012737 000544 000674
2173 005670 004737 016330
2174 005674 012777 177777 172610
2175 005702 012701 000001
2176 005706 117777 172762 172574
2177 005714 017702 172604
2178 005720 030102
2179 005722 001011
2180 005724 012737 026353 000672
2181 005732 012737 000001 000716
2182 005740 004737 014662
2183 005744 000404
2184 005746 020102
2185 005750 001402
2186 005752 004737 014650
2187 005756 005300
2188 005760 001403
2189 005762 005237 000674
2190 005766 000740
2191 005770 004737 016240
2192 005774 004737 015330
2193 006000 000137 002344

; THE FOLLOWING 11 TESTS WILL TEST ALL POSSIBLE ERROR BITS
; BY FORCING THEIR CONDITIONS THROUGH VARIOUS ILLEGAL PROGRAMMING
; SEQUENCES AND USING THE MAINTENANCE MILL 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)*****

```

LT20:  MOV      #MSLT20,EMADDR      ;SET TEST HEADER
        MOV      #LT20A,SCOLP      ;SET LOOP ADDRESS
LT20IT: MOV      #22,R0             ;SET NUMBER OF ILL CODES
        MOV      #ILFT,TEMP1       ;POINT TO START IF TABLE
LT20A:  JSR      PC,INIT1           ;GO INIT, SELECT SLAVE + DRIVE
        MOV      #-1,WC            ;SET WC= -1
        MOV      #1,R1             ;SET TEST WORD
        MOV      @TEMP1,@C1        ;SET ILL CODE
        MOV      @ER,R2            ;READ ER
        BIT      R1,R2             ;SEE IF EXPT=RCVD
        BNE      LT20B            ;IF SO: BR
        MOV      #TMS17,ERADD      ;SET ERROR CODE
        MOV      #1,EXFL           ;SET EXPT FLG
        JSR      PC,LTGER0         ;GO PRINT ERROR
        BR       LT20C
LT20B:  CMP      R1,R2             ;SEE UNEXPECTED ERRORS
        BEQ      LT20C            ;IF NOT: BR
        JSR      PC,LTGER3         ;ELSE PRINT ERROR
LT20C:  DEC      R0               ;SEE IF DONE ALL ILL CODES
        BEQ      LT20X            ;IF SO: BR
        INC      TEMP1            ;BUMP ADDRESS
        BR       LT20A            ;CONTINUE
LT20X:  JSR      PC,ITER           ;GO SEE IF ITERATION
        JSR      PC,DRVCLR        ;
        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,INIT1 ;GO INIT, SELECT SLAVE, DRIVE
2200 006024 052777 000300 BIS #300,ATC ;SET FORMAT
2201 006032 012777 000015 172510 MOV #15,AMR ;SET WAM3
2202 006040 012777 000071 172474 MOV #71,AC1 ;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,LTGER0 ;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,SCS ;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,SCS ;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,ATC ;SET ILLEGAL FORMAT
2252 006322 012701 000020 MOV #20,R1 ;SET TEST WORD
2253 006326 012777 000015 172200 MOV #15,AMR ;SET HAM 3
2254 006334 012777 000071 172146 MOV #71,PC1 ;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,ERADD ;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

```

```

;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)*****
2269
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,ATC ;SET NORMAL FORMAT
2276 006460 012777 026704 172026      MOV #MDATA,2BA ;SET BA
2277 006466 012777 177760 172022      MOV #20,2FC ;SET FC
2278 006474 012777 177770 172010      MOV #10,2WC ;SET WC
2279 006502 012777 000013 172024      MOV #13,2MR ;SELECT WAM 2
2280 006510 012777 000061 171772      MOV #61,2C1 ;SET WRITE+GO
2281 006516 052777 000020 171774      BIS #20,2CS ;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      1S: DEC R0
2286 006540 001376      BNE 1S ;DELAY
2287 006542 005303      DEC R3
2288 006544 001374      BNE 1S
2289 006546 012700 000004      MOV #4,R0
2290 006552 012777 000013 171754 LT24B: MOV #13,2MR ;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 2MR,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 2MR,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 #40,2ER ;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,LTGER0 ;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 2ER,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 #20,2CS ;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 TSC02 ;RETURN TO SCHED

```

```

2325
2326
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,ITC ;SET NORMAL FORMAT
2331 006770 012777 177777 171520      MOV #-1,FC ;SET ITLLEGAL FC
2332 006776 012777 000013 171530      MOV #13,MR ;SET WAM 2
2333 007004 012777 000061 171476      MOV #61,PC1 ;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
2351
2352
2353 007102 012737 024362 000622 LT26: MOV #MSLT26,EMADDR ;SET TEST HEADER
2354 007110 004737 016356 LT26IT: JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2355 007114 005000 CLR RO
2356 007116 005300 1S: DEC RO
2357 007120 001376 BNE 1S ;AWAIT OPI RESET
2358 007122 052777 000300 171412 BIS #300,2TC ;SET NORMAL FORMAT
2359 007130 012777 177770 171354 MOV #10,2MC ;SET MC=-10
2360 007136 012777 177760 171352 MOV #20,2FC ;SET FC=-20
2361 007144 012777 000013 171362 MOV #13,2MR ;SET WAM 3
2362 007152 012777 000061 171330 MOV #61,2C1 ;LOAD WRITE+GO
2363 007160 012701 001000 MOV #1000,R1 ;SET TEST WORD
2364 007164 005000 CLR RO
2365 007166 005300 2S: DEC RO
2366 007170 001376 BNE 2S ;DELAY
2367 007172 012777 000025 171334 MOV #25,2MR ;LOAD MM EOR CLEAR
2368 007200 105077 171330 CLR# 2MR ;RESET MR
2369 007204 012703 000004 MOV #4,R3
2370 007210 005000 CLR RO
2371 007212 032777 001000 171304 3S: BIT #1000,2ER ;SEE IF FCE SET
2372 007220 001022 BNE 4S ;IF 50: BR
2373 007222 005300 DEC RO
2374 007224 001372 BNE 3S ;DELAY
2375 007226 005303 DEC R3
2376 007230 001370 BNE 3S
2377 007232 017702 171266 MOV 2ER,R2 ;GET ER
2378 007236 012737 007110 000712 MOV #LT26IT,SCOLP ;SET SCOPE ADDRESS
2379 007244 012737 026457 000672 MOV #TMS28,ERADD
2380 007252 012737 000001 000716 MOV #1,EXFL ;SET EXPT FLG
2381 007260 004737 014662 JSR PC,LTGER0 ;GO PRINT ERROR
2382 007264 000406 BR LT26X
2383 007266 017702 171232 4S: MOV 2ER,R2 ;GET ERROR REGISTER
2384 007272 020102 CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2385 007274 001402 BEQ LT26X ;IF NOT: BR
2386 007276 004737 014650 JSR PC,LTGER3 ;ELSE GO PRINT ERROR
2387 007302 004737 016240 LT26X: JSR PC,ITER ;GO SEE IF ITERATION
2388 007306 004737 015330 JSR PC,DRVCLR
2389 007312 000137 002344 JMP TSCD2 ;RETURN TO SCHED
  
```

```

2389
2390
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,DR          ;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

```


2465
2464
2463
2462
2461
2460
2459
2458
2457
2456
2455
2454
2453
2452
2451
2450
2449
2448
2447
2446
2445
2444
2443
2442
2441
2440
2439
2438
2437
2436
2435
2434
2433
2432
2431
2430
2429
2428
2427
2426
2425
2424
2423
2422
2421
2420
2419
2418
2417
2416
2415
2414
2413
2412

;LOGIC TEST 30: DRIVE TIMING ERROR*****

```

007434 012737 026506 000672 LT30:  MOV  #TMS32,ERADD ;SET ERROR CODE
007442 012737 024452 000622  MOV  #MSLT30,EMADDR ;SET TEST HEADER
007450 012737 007456 000712  MOV  #LT30IT,SCOLP ;SET SCOPE ADDRESS
007456 004737 016356  LT30IT: JSR  PC,INIT3 ;INIT, SELECT DRIVE + SLAVE
007462 052777 000300 171052  BIS  #300,ATC ;SET NORMAL FORMAT
007470 012701 010000  MOV  #10000,R1 ;SET TEST WORD
007474 012777 000017 171032  MOV  #17,AMR ;CRIPPLE OCCUPIED
007502 005077 171010  CLR  #FC ;SET FC3
007506 012777 000061 170774  MOV  #61,AC1 ;LOAD WRITE+GO
007514 032777 010000 171002  BIT  #10000,AMR ;SEE IF DTE SET
007522 001005  BNE  LT30A ;IF S0: BR
007524 012737 000001 000716  MOV  #1,EXFL ;SET EXPT FLG
007532 004737 014662  JSR  PC,LTGERO ;GO PRINT ERROR
007536 004737 016356  LT30A: JSR  PC,INIT3 ;GO INIT SELECT DRIVE, SLAVE
007542 052777 000300 170772  BIS  #300,ATC ;SET FORMAT
007550 012701 010000  MOV  #10000,R1 ;SET TEST WORD
007554 005077 170736  CLR  #FC ;SET FCS
007560 012777 000015 170746  MOV  #15,AMR ;SET WRAP 3
007566 012777 000061 170714  MOV  #61,AC1 ;LOAD WRITE+GO
007574 012704 040000  MOV  #40000,R4
007600 005777 170736  LT30B: TST  #ATC ;SEE IF ALPHA
007604 100015  BPL  LT30C ;AWAIT ALPHA
007606 005300  DEC  RO
007610 001373  BNE  LT30B
007612 013704 000622  MOV  EMADDR,R4
007616 004737 017320  JSR  PC,TTOUT ;PRINT HEADER
007622 012704 022374  MOV  #MSG50,R4 ;PRINT ALPHA ERROR
007626 004737 017320  JSR  PC,TTOUT
007632 004737 016210  JSR  PC,SCOPE
007636 000435  BR   LT30X
007640 012777 000015 170666  LT30C: MOV  #15,AMR ;CLOCK MR
007646 012777 000015 170660  MOV  #15,AMR ;CLOCK MR
007654 005000  CLR  RO
007656 005300  LT30D: DEC  RO
007660 001376  BNE  LT30D ;DELAY
007662 032777 010000 170634  BIT  #10000,AMR ;SEE IF DTE SET
007670 001006  BNE  LT30E ;IF S0: BR
007672 012737 000001 000716  MOV  #1,EXFL ;SET EXPT FLG
007700 004737 014662  JSR  PC,LTGERO ;GO PRINT ERROR
007704 000412  BR   LT30X
007706 012701 010000  LT30E: MOV  #10000,R1 ;SET TEST WORD
007712 017702 170606  MOV  #R2 ;GET ERROR REGISTER
007716 042702 020100  BIC  #20100,R2 ;MASK OPI AND VPE
007722 020102  CMP  R1,R2 ;SEE IF UNEXPECTED ERRORS
007724 001402  BEQ  LT30X ;IF NOT: BR
007726 004737 014650  JSR  PC,LTGER3 ;ELSE GO PRINT ERROR
007732 004737 016240  LT30X: JSR  PC,ITER ;GO SEE IF ITERATION
007736 004737 014514  JSR  PC,EORPA ;GO CLEAR GO BIT
007742 004737 015330  JSR  PC,DRVCLR
007746 000137 002344  JMP  TSCD2 ;RETURN TO SCHED
    
```

```

;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)*****
2566
2567
2568
2569 007752 012737 024510 000622 LT31: MOV #MSLT31,EMADDR ;SET TEST HEADER
2570 007760 012737 007760 000712 LT31IT: MOV #LT31IT,SCOLP ;SET SCOPE ADDRESS
2571 007766 012737 026522 000672 MOV #TMS33A,ERADD ;SET ERROR MSG HDR
2572 007774 012737 000002 000606 MOV #2,ITAMT ;SET REDUCED ITER COUNT
2573 010002 004737 016356 JSR PC,INIT3 ;INIT, SELECT DRIVE+SLAVE
2574 010006 005000 CLR RO
2575 010010 005300 1S: DEC RO
2576 010012 001376 BNE 1S ;AWAIT OPI RESET
2577 010014 052777 000300 170520 BIS #300,ATC ;SET FORMAT
2578 010022 012777 000013 170504 MOV #13,AMR ;SET WAM 2
2579 010030 005077 CLR #PC ;SET FRAME COUNT
2580 010034 012705 020000 MOV #20000,R5 ;SET TEST BIT (OPI)
2581 010040 012702 010056 MOV #25,R2 ;SET RETURN ADDRESS FROM TIMER
2582 010044 004737 010256 JSR PC,TIMON ;START TIMER
2583 010050 012777 000061 170432 MOV #61,AC1 ;LOAD WRITE+GO
2584 010056 030577 170442 2S: BIT R5,2ER ;BRANCH WHEN OPI SETS
2585 010062 001002 BNE 3S
2586 010064 000163 010350 JMP TIMER(R3) ;GO TO TIMER & RETURN TO 2S ABOVE
2587 010070 017702 170430 3S: MOV 2ER,R2 ;GET ERROR REGISTER
2588 010074 020502 CMP R5,R2 ;SEE IF UNEXPECTED ERRORS
2589 010076 001403 BEQ 4S ;IF NOT: BR
2590 010100 004737 014650 JSR PC,LTGER3 ;ELSE PRINT ERROR
2591 010104 000453 BR LT31X
2592 010106 004737 010442 4S: JSR PC,TIMOK ;GO CHECK TIME FOR OPI TO SET
2593 010112 102450 BVS LT31X ;BRANCH IF TIME WAS INCORRECT
2594
2595 010114 012737 010130 000712 MOV #LT31A,SCOLP ;SET SCOPE LOOP
2596 010122 012737 026536 000672 LT31A: MOV #TMS33B,ERADD ;SET ERROR MSG HEADER
2597 010130 004737 016356 JSR PC,INIT3 ;GO INIT
2598 010134 005000 CLR RO
2599 010136 005300 1S: DEC RO ;WAIT FOR OPI TO CLEAR
2600 010140 001376 BNE 1S
2601 010142 052777 000300 170372 BIS #300,ATC ;SET FORMAT
2602 010150 012777 000015 170356 MOV #15,AMR ;SET WRAP 3
2603 010156 012702 010200 MOV #25,R2 ;SET RETURN ADDRESS FROM TIMER
2604 010162 012705 020000 MOV #20000,R5 ;SET TEST WORD
2605 010166 004737 010256 JSR PC,TIMON ;START TIMER
2606 010172 012777 000071 170310 MOV #71,AC1 ;LOAD READ+GO
2607 010200 030577 170320 2S: BIT R5,2ER ;BRANCH WHEN OPI SETS
2608 010204 001002 BNE 3S
2609 010206 000163 010350 JMP TIMER(R3) ;GO TO TIMER
2610 010212 017702 170306 3S: MOV 2ER,R2 ;GET ERROR REGISTER
2611 010216 020502 CMP R5,R2 ;SEE IF UNEXPECTED ERRORS
2612 010220 001403 BEQ 4S ;ELSE PRINT ERROR
2613 010222 004737 014650 JSR PC,LTGER3 ;EXIT TEST
2614 010226 000402 BR LT31X ;EXIT TEST
2615 010230 004737 010442 4S: JSR PC,TIMOK ;GO CHECK TIME
2616 010234 004737 016240 LT31X: JSR PC,ITER ;GO SEE IF ITERATIONS
2617 010240 004737 015330 JSR PC,DRVCLR
2618 010244 012737 000020 000606 MOV #20,ITAMT
2619 010252 000137 002344 JMP TSC02 ;RETURN TO SCHED

```

;ROUTINE TO START THE TIMER. THE TIMER IS AN OSCILLATOR IN THE MAINT-

010256
010257
010258
010259
010260
010261
010262
010263
010264
010265
010266
010267
010268
010269
010270
010271
010272
010273
010274
010275
010276
010277
010278
010279
010280
010281
010282
010283
010284
010285
010286
010287
010288
010289
010290
010291
010292
010293
010294
010295
010296
010297
010298
010299
010300
010301
010302
010303
010304
010305
010306
010307
010308
010309
010310
010311
010312
010313
010314
010315
010316
010317
010318
010319
010320
010321
010322
010323
010324
010325
010326
010327
010328
010329
010330
010331
010332
010333
010334
010335
010336
010337
010338
010339
010340
010341
010342
010343
010344
010345
010346
010347
010348
010349
010350
010351
010352
010353
010354
010355
010356
010357
010358
010359
010360
010361
010362
010363
010364
010365
010366
010367
010368
010369
010370
010371
010372
010373
010374
010375
010376
010377
010378
010379
010380
010381
010382
010383
010384
010385
010386
010387
010388
010389
010390
010391
010392
010393
010394
010395
010396
010397
010398
010399
010400
010401
010402
010403
010404
010405
010406
010407
010408
010409
010410
010411
010412
010413
010414
010415
010416
010417
010418
010419
010420
010421
010422
010423
010424
010425
010426
010427
010428
010429
010430
010431
010432
010433
010434
010435
010436
010437
010438
010439
010440
010441
010442
010443
010444
010445
010446
010447
010448
010449
010450
010451
010452
010453
010454
010455
010456

005000
005001
012703 000024
032777 000100 170240
001405
032777 000100 170230
001374
000405

005403
032777 000100 170214
001774
000207

032777 000100 170202
001405
000112
010350
005403
062700 000001
005501
022701 000003
001410
000112
010374
032777 000100 170132
001362
000112

013704 000622
004737 017320
013704 000672
004737 017320
012704 026616
004737 017320
000137 010234

000240
006201
006000
006201
006000
006201
006000

; ENANCE 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.

```
TIMON: CLR R0 ;CLEAR TICK COUNT
        CLR R1
        MOV #24,R3 ;PRESET INDEX TO TIMER
        BIT #100,2MR ;BRANCH IF OSC CLEAR
        BEQ 25
        BIT #100,2MR ;WAIT FOR OSC TO CLEAR
        BNE 15
        BR 45 ;EXIT

        NEG R3 ;SET INDEX TO TIMER
        BIT #100,2MR ;WAIT FOR OSC TO SET
        BEQ 35
        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.

```
TIMER1: BIT #100,2MR ;BRANCH IF OSC HAS CHANGED STATE
        BEQ TIMER
        JMP (R2) ;RETURN
        . =TIMER1+24

TIMER: NEG R3 ;SET INDEX TO OTHER STATE
        ADD #1,R0 ;INCREMENT TICK COUNT
        ADC R1
        CMP #3,R1 ;BRANCH IF TIMER OVERFLOWS
        BEQ TIMOVF
        JMP (R2) ;RETURN
        . =TIMER +24

TIMER0: BIT #100,2MR ;BRANCH IF OSC SET
        BNE TIMER
        JMP (R2) ;RETURN
```

```
TIMOVF: MOV ERADDR,R4 ;TYPE TEST HEADER
        JSR PC,TTOUT
        MOV ERADD,R4 ;GET ERROR MSG ADDRESS
        JSR PC,TTOUT ;AND TYPE IT
        MOV #TMS33E,R4 ;TYPE
        JSR PC,TTOUT ;'TIMER OVERFLOWED'
        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.

```
TIMOK: NOP
        ASR R1 ;DIVIDE COUNT BY 8
        ROR R0
        ASR R1
        ROR R0
        ASR R1
        ROR R0
```

2578	010460	020027	027764		CMP	RO, #12276.	;BRANCH IF GREATER THAN LOWER LIMIT
2579	010464	101016			BHI	1S	
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	2S	
2588							
2589	010522	020027	045035	1S:	CMP	RO, #18973.	;BRANCH IF LESS THAN UPPER LIMIT
2590	010526	003415			BLE	2S	
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		2S:	RTS	PC	

```

2599
2600
2601
2602 ;LOGIC TEST 32: UNSAFE(UNS)*****
2603
2604 010564 012737 024544 000622 LT32: MOV #MSLT32,EMADDR ;SET TEST HEADER
2605 010572 012737 010600 000712 LT32IT: MOV #LT32IT,SCOLP ;SET SCOPE ADDRESS
2606 010600 004737 016356 LT32IT: JSR PC,INIT3 ;INIT, SELECT DRIVE +SLAVE
2607 010604 013700 000664 LT32IT: MOV SLVN,RO ;GET SLAVE NUMBER
2608 010610 005100 LT32IT: COM RO ;SET NONEXISTANT SLAVE
2609 010612 042700 177770 LT32IT: BIC #177770,RO ;MASK SLAVE NUMBER
2610 010616 052700 000300 LT32IT: BIS #300,RO ;SET FORMAT
2611 010622 010077 167714 LT32IT: MOV RO,ATC ;SELECT ILLEGAL SLAVE
2612 010626 032777 002000 167702 LT32IT: BIT #2000,ATC ;EXIT TEST IF SALVE AVAILABLE
2613 010634 001032 LT32IT: BNE LT32XX
2614 010636 012777 000071 167644 LT32IT: MOV #71,AC1 ;LOAD READ+GO
2615 010644 012701 040000 LT32IT: MOV #40000,R1 ;SET TEST WORD
2616 010650 017702 167650 LT32IT: MOV #ER,R2 ;READ ER
2617 010654 030102 LT32IT: BIT R1,R2 ;SEE IF EXPT=RCVD
2618 010656 001011 LT32IT: BNE IS ;IF SO: BR
2619 010660 012737 026636 000672 LT32IT: MOV #TMS34,ERADD ;SET ERROR CODE
2620 010666 012737 000001 000716 LT32IT: MOV #1,EXFL ;SET ERROR CODE
2621 010674 004737 014662 LT32IT: JSR PC,LTGER0 ;GO PRINT ERROR
2622 010700 000404 LT32IT: BR LT32X
2623 010702 020102 LT32IT: IS: CMP R1,R2 ;SEE IF UNEXPECTED ERRORS
2624 010704 001402 LT32IT: BEQ LT32X ;IF NOT: BR
2625 010706 004737 014650 LT32IT: JSR PC,LTGER3 ;ELSE PRINT ERROR
2626 010712 004737 016240 LT32X: JSR PC,ITER ;GO SEE IF ITERATIONS
2627 010716 004737 015330 LT32X: JSR PC,DRVCLR
2628 010722 000137 002344 LT32XX: JMP TSCD2 ;RETURN TO SCHED

```

2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649

: THE FOLLOWING 6 TESTS WILL LOCK AT VARIOUS BITS IN THE
: DRIVE STATUS (DS) AND TAPE CONTROL (TC)
: REGISTERS BY FORCING CERTAIN CONDITONS WHICH DO NOT
: REQUIRE TAPE MOVEMENT.

; LOGIC TEST 33: POSITIONING IN PROGRESS (PIP)*****

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,AMR	:SET WAM 2
2641	010754	012777	177777	167534		MOV	#-1,FC	:SET FCS
2642	010762	012777	000031	167520		MOV	#31,AC1	: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,LTGERD	: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, ERADD ;SET ERROR CODE
2654 011036 012737 024634 000622 LT34: 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 LT34A1: BIC #3400, JTC ;SELECT NRZI
2658 011062 052777 001400 167452 LT34A1: BIS #1400, JTC
2659 011070 032777 000040 167424 LT34A: BIT #40, JDS ;SEE IF PES=0
2660 011076 001410 BEQ LT34B ;IF SO: BR
2661 011100 012737 000002 000716 LT34B: MOV #2, EXFL ;SET RCVD-NOT EXPT
2662 011106 012737 011050 000712 LT34A1: MOV #LT34A1, SCOLP ;SET SCOPE ADDRESS
2663 011114 004737 014662 LT34B: JSR PC, LTGERO ;GO PRINT ERROR
2664 011120 004737 016372 LT34B: JSR PC, INIT4
2665 011124 032777 000040 167370 LT34C: BIT #40, JDS ;SEE IF PES=1
2666 011132 001010 BNE LT34X ;IF SO: BR
2667 011134 012737 011124 000712 LT34C: MOV #LT34C, SCOLP ;SET SCOPE ADDRESS
2668 011142 012737 000001 000716 LT34C: MOV #1, EXFL ;SET EXPT-NOT RCVD FLAG
2669 011150 004737 014662 LT34C: JSR PC, LTGERO ;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
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 IS: BIT #20,SDS ;SEE IF SDWN IS RESET
2679 011212 001374 BNE IS ;IF NOT: BR
2680 011214 052777 000300 167320 BIS #300,ATC ;SET FORMAT
2681 011222 012777 000015 167304 MOV #15,AMR ;SET WAM 3
2682 011230 012777 000071 167252 MOV #71,AC1 ;LOAD READ+GO
2683 011236 032777 020000 167276 BIT #20000,ATC ;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,LTGERO ;GO PRINT ERROR
2688 011266 004737 016356 LT35A: JSR PC,INIT3 ;INIT
2689 011272 005277 167244 INC ATC ;BUMP SLAVE ADDRESS
2690 011276 032777 020000 167236 BIT #20000,ATC ;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,LTGERO ;GO PRINT ERROR
2695 011326 004737 016240 LT35X: JSR PC,ITER
2696 011332 000137 002344 JMP TSCD2 ;RETURN TO SCHED

```



```

2697
2698
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,ATC ;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$: JSR PC,INIT3 ;INIT
2709 011412 005077 167100      CLR @FC ;WRITE TO FC
2710 011416 032777 040000 167116      BIT #40000,ATC ;SEE IF FCS=1
2711 011424 001010      BNE LT36X ;IF SO: BR
2712 011426 012737 011406 000712      MOV #1$,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
2719
2720 011456 012737 024774 000622 LT37: MOV #MSLT37,EMADDR
2721 011464 012737 026675 000672 LT37: 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,ATC ;SET FORMAT
2724 011504 005777 167032 TST ATC ;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,ATC ;SET FORMAT
2731 011544 012777 000015 166762 MOV #15,AMR ;SET WAM 3
2732 011552 012777 000071 166730 MOV #71,AC1 ;LOAD READ+GO
2733 011560 012700 100000 MOV #100000,RO ;SET ACCL DELAY
2734 011564 005777 166752 LT37B: TST ATC ;SEE IF ACCL=0
2735 011570 100012 BPL LT37X ;IF SO: BR
2736 011572 005300 DEC RO
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 CLR RO
2750 011650 005300 1S: DEC RO
2751 011652 001376 BNE 1S ;DELAY FOR OPI RESET
2752 011654 052777 002300 166660 BIS #2300,ATC
2753 011662 012777 000007 166644 MOV #7,AMR ;SET WAM 0
2754 011670 012777 000027 166612 MOV #27,AC1 ;LOAD WRITE TAPE MARK+GO
2755 011676 012700 100000 MOV #100000,RO ;SET DELAY
2756 011702 032777 000004 166612 2S: BIT #4,ADS ;SEE IF TM=1
2757 011710 001012 BNE LT40X ;IF SO: BR
2758 011712 005300 DEC RO
2759 011714 001372 BNE 2S ;DELAY
2760 011716 012737 026227 000672 MOV #TMS3,ERADD
2761 011724 012737 000001 000716 MOV #1,EXFL
2762 011732 004737 014662 JSR PC,LTGERO ;GO PRINT ERROR
2763 011736 004737 016240 LT40X: JSR PC,ITER
2764 011742 000137 002344 LT40XX: JMP TSC02 ;RETURN TO SCHED
  
```

```

2765
2766
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,ATC ;SET NRZ+NORMAL FORMAT
2772 011774 012777 177760 166514 MOV #20,ATC ;SET FCS
2773 012002 012777 000007 166524 MOV #7,ATR ;SET WAM 0
2774 012010 012777 000027 166472 MOV #27,ATC1 ;LOAD WRITE TAPE MARK+GO
2775 012016 005000 CLR R0
2776 012020 032777 000004 166474 1S: BIT #4,ADS ;SEE IF TM=1
2777 012026 001012 BNE 2S ;IF S0: BR
2778 012030 005300 DEC R0
2779 012032 001372 BNE 1S ;DELAY
2780 012034 012737 026227 000672 MOV #TMS3,ERADD ;SET ERROR CODE
2781 012042 012737 000001 000716 MOV #1,EXFL
2782 012050 004737 014662 JSR PC,LTGERO ;GO PRINT ERROR
2783 012054 032777 002000 166442 2S: BIT #2000,ZER ;SEE IF ITM=1
2784 012062 001010 BNE 3S ;IF S0: BR
2785 012064 012737 026472 000672 MOV #TMS30,ERADD ;SET ERROR CODE
2786 012072 012737 000001 000716 MOV #1,EXFL
2787 012100 004737 014662 JSR PC,LTGERO ;GO PRINT ERROR
2788 012104 032777 000100 166412 3S: BIT #100,ZER ;SEE IF VPE=1
2789 012112 001011 BNE 4S ;IF S0: BR
2790 012114 012737 026457 000672 MOV #TMS28,ERADD ;SET ERROR CODE
2791 012122 012737 000001 000716 MOV #1,EXFL
2792 012130 004737 014662 JSR PC,LTGERO ;GO PRINT ERROR
2793 012134 000410 BR LT41X
2794 012136 012701 002100 4S: MOV #2100,R1 ;SET EXPT ERROR BITS
2795 012142 017702 166356 MOV ZER,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 CLR R2 ;SET TIMER
2800 012160 032777 000200 166334 1S: BIT #200,ADS ;SEE IF DRY SET
2801 012166 001002 BNE 2S ;IF S0: BR
2802 012170 005302 DEC R2 ;AWAIT DRY
2803 012172 001372 BNE 1S ;DELAY
2804 012174 004737 016240 2S: 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
2809
2810
2811
2812
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 012228 005300 1S: DEC R0
2818 012230 001376 BNE 1S ;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, 2MC
2823 012260 012777 177760 166230 MOV #-20, 2FC ;SET FC=20
2824 012266 012777 026704 166220 MOV #MDATA, 2BA ;SET BUS ADDRESS
2825 012274 012777 000007 166232 MOV #7, 2MR ;SET MM CODE
2826 012302 012777 000061 166200 MOV #61, 2C1 ;LOAD WRITE+GO
2827 012310 005000 CLR R0
2828 012312 032777 000200 166202 LT42A: BIT #200, 2DS ;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, 2ER ;SEE IF LRC ERROR ONLY
2833 012334 001007 BNE LT42B1 ;IF NOT: BR
2834 012336 017702 166166 MOV 2CC, 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 022465 MOV #MSG55, R4
2840 012364 004737 017320 JSR PC, TOUT ;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, 2MC ;SET MC
2844 012406 012777 177760 166102 MOV #-20, 2FC ;SET FC
2845 012414 012777 026704 166072 MOV #MDATA, 2BA ;SET BA
2846 012422 012777 000021 166104 MOV #21, 2MR ;SET MM
2847 012430 012777 000061 166052 MOV #61, 2C1 ;LOAD WRITE+GO
2848 012436 005000 CLR R0
2849 012440 032777 000200 166054 LT42C: BIT #200, 2DS ;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 2ER ;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, LTGER0 ;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 2ER, 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          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: LONGITUDINAL 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          LT43IT: JSR   PC,INIT   ;INIT, SELECT DRIVE+SLAVE
2875 012572 005001          CLR    R1
2876 012574 005301          IS:    DEC    R1      ;DELAY
2877 012576 001376          BNE    IS
2878 012600 012777 000023 165726  MOV    #23,MMR      ;SET MM
2879 012606 012777 177770 165676  MOV    #-10,MC      ;SET MC
2880 012614 012777 177760 165674  MOV    #-20,FC      ;SET FC
2881 012622 012777 026704 165664  MOV    #MDATA,BA    ;SET BA
2882 012630 012777 000061 165652  MOV    #61,AC1     ;LOAD WRITE+GO
2883 012636 005000          CLR    R0
2884 012640 032777 000200 165654  LT43C: BIT    #200,ADS  ;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,GER  ;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    MMR,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    GER,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

```

;LOGIC TEST 44: PE CORRECTABLE DATA (CORR)*****

```

2909
2910
2911 012774 012737 002300 000776 LT44: MOV #2300, UDES ;SET UNIT DESCRIPTION = 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, ZWC ;SET WC=200
2917 013034 012777 177400 165454 MOV #400, ZFC ;SET FC=400
2918 013042 012777 026704 165444 MOV #WDATA, ZBA ;SET BA=START OF WRITE BUFFER
2919 013050 012777 000061 165432 MOV #61, ZC1 ;LOAD WRITE AND GO
2920 013056 005000 CLR RO
2921 013060 005777 165432 LT44A: TST ZFC ;SEE IF FC=0
2922 013064 001402 BEQ LT44A1 ;IF SO: BR
2923 013066 005300 DEC RO
2924 013070 001373 BNE LT44A ;AWAIT FC=0
2925 013072 012777 000021 165434 LT44A1: MOV #21, ZMR ;SET MAINT MODE
2926 013100 005000 CLR RO
2927 013102 032777 000200 165412 LT44B: BIT #200, ZDS ;SEE IF DRY
2928 013110 001002 BNE LT44C ;IF SO :BR
2929 013112 005300 DEC RO
2930 013114 001372 BNE LT44B ;AWAIT DRY
2931 013116 005777 165402 LT44C: TST ZER ;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, LTGERO ;GO PRINT ERROR
2936 013144 000240 LT44D: NOP
2937 013146 122777 000002 165354 LT44E: CMPB #2, ZCC ;SEE IF DEAD TRACK BIT 1
2938 013154 001414 BEQ LT44F ;IF SO: BR
2939 013156 117702 165346 MOVB ZCC, 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 ZER, 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

```

```

;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)*****
2963 013242 012737 002300 000776 LT45: MOV 82300, UDES ;SET UNIT DESCRIPTION = PE
2964 013243 004737 014464 JSR PC, STATIC ;GO SEE IF STATIC ONLY
2965 013244 012737 025323 000622 MOV 8MSLT45, EMADDR
2966 013245 012737 013270 000712 MOV 8LT45IT, SCOLP
2967 013270 004737 016406 LT45IT: JSR PC, INIT ;INIT SELECT DRIVE SLAVE
2968 013274 012777 177600 165210 MOV 8-200, 2MC ;SET MC=200
2969 013302 012777 177400 165206 MOV 8-400, 2FC ;SET FC=400
2970 013310 012777 026704 165176 MOV 8NDATA, 2BA ;SET BA=START OF WRITE BUFFER
2971 013316 012777 000061 165164 MOV 861, 2C1 ;LOAD WRITE+GO
2972 013324 005000 CLR R0
2973 013326 005777 165164 LT45E: TST 2FC ;AWAIT FC=0
2974 013332 001402 BEQ LT45E1
2975 013334 005300 DEC R0
2976 013336 001373 BNE LT45E ;AWAIT FC=0
2977 013340 012777 000023 165166 LT45E1: MOV 823, 2MR ;SET MAINT CODE
2978 013346 005000 CLR R0
2979 013350 032777 000200 165144 LT45A: BIT 8200, 2DS ;SEE IF DRY IS SET
2980 013356 001002 SNE LT45B ;IF SO: BR
2981 013360 005300 DEC R0
2982 013362 001372 BNE LT45A ;AWAIT DRY
2983 013364 032777 000100 165132 LT45B: BIT 8100, 2ER ;SEE IF INC=1
2984 013372 001010 BNE LT45D ;IF SO: BR
2985 013374 012737 026421 000672 MOV 8TMS23, ERADD ;SET ERROR CODE
2986 013402 012737 000001 000716 MOV 81, EXFL
2987 013410 004737 014662 JSR PC, LTGERD ;GO PRINT ERROR
2988 013414 017702 165110 LT45D: MOV 2CC, R2 ;GET CHECK CHAR
2989 013420 042702 177000 BIC 8177000, R2 ;MASK CHECK CHAR
2990 013424 012701 000046 MOV 846, R1 ;SET EXPT CK
2991 013430 020102 CMP R1, R2 ;SEE IF EXPT = RCVD
2992 013432 001405 BEQ LT45F ;IF SO: BR
2993 013434 012737 022454 000672 MOV 8MSG54, ERADD ;ELSE GO PRINT ERROR
2994 013442 004737 015776 LT45F: JSR PC, LTGER1
2995 013446 017702 165052 MOV 2ER, R2 ;MASK OPI, MSG, CORR, AND PEF
2996 013452 042702 120600 BIC 8120600, R2 ;SET EXPT ERROR BITS
2997 013456 012701 000100 MOV 8100, R1 ;SEE IF UNEXPECTED ERRORS
2998 013462 020102 CMP R1, R2 ;IF NOT: BR
2999 013464 001402 BEQ LT45X ;ELSE PRINT ERROR
3000 013466 004737 014650 LT45X: JSR PC, LTGER3
3001 013472 004737 016240 LT45X: JSR PC, ITER
3002 013476 004737 015330 LT45X: JSR PC, DRVCLR
3003 013502 000137 002344 LT45XX: JMP TSCD2 ;RETURN TO SCHED

```



```

2997
2998
2999
3000 013506 012737 002300 000776 LT46: MOV      #2300, UDES      ;SET UNIT DESCRIPTION = PE
3001 013514 004737 014464          JSR      PC, STATIC    ;GO SEE IF STATIC ONLY
3002 013520 012737 025405 000622          MOV      #MSLT46, EMADDR ;SET HEADER
3003 013526 012737 013534 000712          MOV      #LT46IT, SCOLP ;SET SCOPE ADDRESS
3004 013534 004737 016406          LT46IT: JSR      PC, INIT    ;INITIALIZE
3005 013540 012777 177770 164744          MOV      #10, #MC      ;SET MC=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, #C1      ;LOAD WRITE+GO
3009 013570 005777 164722          LT46A: TST      #FC
3010 013574 001375          BNE      LT46A        ;AWAIT FC=0
3011 013576 032777 000100 164730 1S:  BIT      #100, #MR
3012 013604 001774          BEQ      1S
3013 013606 032777 000100 164720 2S:  BIT      #100, #MR
3014 013614 001374          BNE      2S
3015 013616 032777 000100 164710 3S:  BIT      #100, #MR
3016 013624 001774          BEQ      3S
3017 013626 012777 000027 164700          MOV      #27, #MR
3018 013634 005000          CLR      RO
3019 013636 032777 000200 164656 LT46B: BIT      #200, #DS
3020 013644 001002          BNE      LT46C
3021 013646 005300          DEC      RO
3022 013650 001372          BNE      LT46B
3023 013652 032777 000200 164644 LT46C: BIT      #200, #ER
3024 013660 001011          BNE      LT46D
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, LTGERO    ;GO PRINT ERROR
3028 013702 000412          BR
3029 013704 017702 164614          LT46D: MOV      #ER, R2
3030 013710 042702 020000          BIC      #20000, R2
3031 013714 012701 000600          MOV      #600, R1
3032 013720 020102          CMP      R1, R2
3033 013722 001402          BEQ      LT46X
3034 013724 004737 014650          JSR      PC, LTGER3
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

```



```

3069
3070 ;LOGIC TEST 50: NEF WHEN WRITING PE ON NRZ SELECTED SLAVE
3071
3072 014134 012737 025506 000622 LT50: MOV #SLT50,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,ATC ;CLEAR DENSITY BITS
3076 014162 052777 002300 164352 BIS #2300,ATC ;SET DENSITY = PE
3077 014170 012777 177770 164314 MOV #10,AWC ;SET WORD COUNT
3078 014176 012777 177760 164312 MOV #20,AWC ;SET FRAME COUNT
3079 014204 012777 026704 164302 MOV #WDATA,AWA ;SET BUS ADDRESS
3080 014212 012777 000013 164314 MOV #13,AWR ;SET WRAP 2
3081 014220 012777 000061 164262 MOV #61,AC1 ;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 JFR,R2 ;GET ERROR REGISTER
3087 014244 030102 BIT R1,R2 ;BRANCH IF NEF BIT SET
3088 014246 001006 BNE IS
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 IS: 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 TSCD2
3098 ;LOGIC TEST 51: NEF WHEN WRITING NRZ ON PE SELECTED SLAVE
3099
3100 014310 012737 025566 000622 LT51: MOV #SLT51,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,ATC ;CLEAR DENSITY BITS
3104 014336 052777 001300 164176 BIS #1300,ATC ;SET DENSITY = NRZ
3105 014344 012777 177770 164140 MOV #10,AWC ;SET WORD COUNT
3106 014352 012777 177760 164136 MOV #20,AWC ;SET FRAME COUNT
3107 014360 012777 026704 164126 MOV #WDATA,AWA ;SET BUS ADDRESS
3108 014366 012777 000013 164140 MOV #13,AWR ;SET WRAP 2
3109 014374 012777 000061 164106 MOV #61,AC1 ;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 JFR,R2 ;GET ERROR REGISTER
3115 014420 030102 BIT R1,R2 ;BRANCH IF NEF SET
3116 014422 001006 BNE IS
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 IS: CMP R1,R2 ;BRANCH IF NO UNEXPECTED
3121 014442 001402 BEQ LT51X ;ERROR BITS WERE SET
3122 014444 004737 014650 LT51X: JSR PC,LTGER3 ;ITERATE TEST
3123 014450 004737 016240 JSR PC,ITER ;CLEAR DRIVE
3124 014454 004737 015330 JSR PC,DRVCLR

```

F07

TMO3/TE16 CONTROL LOGIC TEST PART I
DZTEAA.P11 01-APR-77 12:00

MACY11 27(1006) 01-APR-77 12:02 PAGE 83

3125 014460 000137 002344

JMP TSC02

;RETURN TO SCHEDULER

```

3126
3127
3128 014464 005737 000740
3129 014470 001006
3130 014472 005737 001012
3131 014476 001403
3132 014500 005726
3133 014502 000137 002344
3134 014506 005037 001006
3135 014512 000207
3136

```

;STATIC TESTS ONLY SUBROUTINE*****

```

STATIC: TST STFLG ;SEE IF SINGLE TEST ONLY
        BNE IS ;IF SO: BR
        TST STATC ;SEE IF STATIC ONLY
        BEQ IS ;IF NOT: BR
        TST (SP)+ ;RESET STACK
        JMP TSCD2 ;RETURN TO SCHEDULAR
IS: CLR RDRVF
     RTS PC ;RETURN TO TEST

```

```

3137
3138
3139
3140 014514 017700 164014
3141 014520 042700 000036
3142 014524 052700 000024
3143 014530 010077 164000
3144 014534 042777 000037 163772
3145 014542 005000
3146 014544 012701 000002
3147 014550 032777 000001 163732 EORP1:
3148 014556 001430
3149 014560 005300
3150 014562 001372
3151 014564 005301
3152 014566 001370
3153 014570 032777 020000 163772
3154 014576 001020
3155 014600 005737 000620
3156 014604 001004
3157 014606 013704 000622
3158 014612 004737 017320
3159 014616 012704 027317 EORP1A:
3160 014622 004737 017320
3161 014626 032777 100000 163734
3162 014634 001401
3163 014636 000000
3164 014640 000240 EORP2:
3165 014642 005037 000676 EORPX:
3166 014646 000207
3167

```

;END OF RECORD FORCE SUBROUTINE*****

```

EORPA: MOV 2MR, R0 ;GET MAINT REG
BIC 236, R0 ;CLEAR CURRENT OP CODE
BIS 224, R0 ;SET EOR CLEAR OP CODE
MOV R0, 2MR ;DO EOR
BIC 237, 2MR ;CLEAR EOR AND MM
CLR R0
MOV 22, R1
EORP1: BIT 21, 2C1 ;SEE IF GO GONE
BEQ EORP2 ;IF SO: BR
DEC R0
BNE EORP1 ;AWAIT GO RESET
DEC R1
BNE EORP1
BIT 20000, 2SWR ;SEE IF ERROR PRINT INHIBIT
BNE EORP2 ;IF SO: BR
TST HDRFL ;SEE IF DONE HEADER
BNE EORP1A ;IF SO: BR
MOV EMADDR, R4
JSR PC, TTOUT ;PRINT HEADER
EORP1A: MOV 2MSG31, R4
JSR PC, TTOUT ;PRINT EOR GO BIT ERROR
BIT 210000, 2SWR ;SEE IF HALT ON ERROR
BEQ EORP2 ;IF NOT: BR
EORP2: NOP
EORPX: CLR TEMP2 ;CLEAR FLAG
RTS PC ;RETURN

```

```

;LOGIC TEST ADDRESSING ERROR SUBROUTINE*****
3168
3169
3170 014650 005037 000716 LTGER3: CLR EXFL
3171 014654 012737 022413 000672 MOV #MSG51,ERADD
3172 014662 012737 000001 000746 LTGER0: MOV #1,ADDFL ;SET NO ADDRESS FLAG
3173 014670 000240 LTGER: NOP
3174 014672 005037 000666 CLR PFLG ;CLEAR PRINT FLAG
3175 014676 032777 020000 163664 BIT #20000,JSWR ;SEE IF SHOULD PRINT
3176 014704 001112 BNE LTGX ;IF NOT: BR
3177 014706 005737 000620 LTGA: TST HDRFL ;SEE IF PRINTED HEADER
3178 014712 001004 BNE LTGA1 ;IF SO: BR
3179 014714 013704 000622 MOV EMADDR,R4
3180 014720 004737 017320 JSR PC,TTOUT ;PRINT TEST HEADER
3181 014724 012737 000001 000620 LTGA1: MOV #1,HDRFL ;SET HEADER FLAG
3182 014732 013704 000672 MOV ERADD,R4
3183 014736 004737 017320 JSR PC,TTOUT ;PRINT CONDITION ERROR
3184 014742 005737 000746 TST ADDFL
3185 014746 001003 BNE LTGA2
3186 014750 010103 MOV R1,R3
3187 014752 004737 017446 JSR PC,OCTP ;PRINT ADDRESS
3188 014756 005737 000716 LTGA2: TST EXFL
3189 014762 001412 BEQ LTGC ;IF NO STATUS: BR
3190 014764 012704 020704 MOV #MSG6,R4
3191 014770 022737 000001 000716 CMP #1,EXFL ;EXPT-NOT RCVD
3192 014776 001402 BEQ LTGB
3193 015000 012704 020723 MOV #MSG7,R4 ;RCVD-NOT EXPT
3194 015004 004737 017320 LTGB: JSR PC,TTOUT ;PRINT STATUS
3195 015010 005237 000666 LTGC: INC PFLG
3196 015014 005737 000746 TST ADDFL ;SEE IF ADD TST
3197 015020 001430 BEQ LTGD ;IF SO: BR
3198 015022 005737 000744 TST T24FL ;SEE IF TEST 24
3199 015026 001423 BEQ LTGCO ;IF NOT: BR
3200 015030 012704 027304 MOV #MSG27,R4
3201 015034 004737 017320 JSR PC,TTOUT ;PRINT DATA TAG
3202 015040 012704 021224 MOV #MSG12,R4
3203 015044 004737 017320 JSR PC,TTOUT ;PRINT EXPT TAG
3204 015050 012703 177777 MOV #-1,R3
3205 015054 004737 017436 JSR PC,OCTPE ;PRINT EXPT
3206 015060 012704 021233 MOV #MSG13,R4
3207 015064 004737 017320 JSR PC,TTOUT ;PRINT RCVD TAG
3208 015070 010103 MOV R1,R3 ;GET RCVD
3209 015072 004737 017436 JSR PC,OCTPE ;PRINT RCVD
3210 015076 004737 015174 LTGCO: JSR PC,REGP ;PRINT REGISTERS
3211 015102 032777 004000 163460 LTGD: BIT #4000,JSWR
3212 015110 001010 BNE LTGX
3213 015112 012704 021275 MOV #MSG16,R4
3214 015116 004737 017320 JSR PC,TTOUT
3215 015122 013703 000702 MOV ITCNT,R3 ;PRINT ITERATION
3216 015126 004737 017446 JSR PC,OCTP
3217 015132 005777 163432 LTGX: TST JSWR
3218 015136 100001 BPL LTGXA ;IF NOT STOP ON ERROR: BR
3219 015140 000000 HALT
3220 015142 005737 000666 LTGXA: TST PFLG
3221 015146 001004 BNE LTGXX ;IF PRINTED: BR
3222 015150 032777 020000 163412 BIT #20000,JSWR
3223 015156 001653 BEQ LTGA

```

```

3224 015160 005037 000746
3225 015164 005037 000716
3226 015170 000137 016210
3227
3228
3229
3230 015174 000240
3231 015176 012704 022236
3232 015202 004737 017320
3233 015206 017703 163276
3234 015212 004737 017436
3235 015216 017703 163270
3236 015222 004737 017436
3237 015226 017703 163262
3238 015232 004737 017436
3239 015236 017703 163254
3240 015242 004737 017436
3241 015246 017703 163246
3242 015252 004737 017436
3243 015256 017703 163240
3244 015262 004737 017436
3245 015266 017703 163232
3246 015272 004737 017436
3247 015276 017703 163224
3248 015302 004737 017436
3249 015306 017703 163222
3250 015312 004737 017436
3251 015316 017703 163220
3252 015322 004737 017436
3253 015326 000207
3254
3255

```

```

LTGXX: CLR      ADOFL      ;CLEAR ADDRESS FLAG
        CLR      EXFL
        JMP      SCOPE

```

;SUBROUTINE TO PRINT MAJOR REGISTERS*****

```

REGP:  NOP
        MOV      @MSG46,R4
        JSR      PC,TTOUT      ;PRINT REGISTER HEADER
        MOV      @C1,R3
        JSR      PC, OCTPE
        MOV      @WC,R3
        JSR      PC, OCTPE
        MOV      @BA,R3
        JSR      PC, OCTPE
        MOV      @FC,R3
        JSR      PC, OCTPE
        MOV      @CS,R3
        JSR      PC, OCTPE
        MOV      @DS,R3
        JSR      PC, OCTPE
        MOV      @ER,R3
        JSR      PC, OCTPE
        MOV      @AS,R3
        JSR      PC, OCTPE
        MOV      @MR,R3
        JSR      PC, OCTPE
        MOV      @TC,R3
        JSR      PC, OCTPE
        RTS      PC

```

;PRINT REGISTERS


```

;DRIVE CLEAR SUBROUTINE*****
3256
3257
3258 015330 000240          DRVCLR: NOP
3259 015332 012704 040000      MOV      #40000,R4
3260 015336 005304          DCD:    DEC      R4
3261 015340 001376          BNE     DCD           ;DELAY
3262 015342 005037 000666      CLR     PFLG
3263 015346 004737 015564      JSR    PC,ATTN       ;GO SEE OF ATTN SET
3264 015352 012777 000011 163130  MOV     #11,2C1      ;ISSUE DRIVE CLEAR
3265 015360 005000          CLR     R0
3266 015362 032777 000200 163132  DCA:    BIT     #200,2DS  ;SEE IF DRY
3267 015370 001002          BNE     DCA0
3268 015372 005300          DEC     R0
3269 015374 001372          BNE     DCA           ;WAIT FOR DRY
3270 015376 032777 040000 163116  DCA0:   BIT     #40000,2DS  ;SEE IF ERR RESET
3271 015404 001022          BNE     DCE           ;IF NOT: BR
3272 015406 005777 163112      TST    2ER           ;SEE IF ERROR REGISTER RESET
3273 015412 001017          BNE     DCE           ;IF NOT: BR
3274 015414 005777 163102      TST    2DS           ;SEE IF ATA RESET
3275 015420 100414          BMI     DCE           ;IF NOT: BR
3276 015422 012703 000001      MOV     #1,R3        ;SET TEST BIT
3277 015426 013704 000624      MOV     DRVN,R4     ;GET DRIVE NUMBER & BRANCH
3278 015432 001403          BEQ    DCC
3279 015434 006303          DCB:   ASL     R3    ;POSITION TEST BIT PER DRIVE NUMBER
3280 015436 005304          DEC     R4
3281 015440 001375          BNE     DCB
3282 015442 030377 163060      DCC:   BIT     R3,2AS ;SEE IF ATTN IS RESET
3283 015446 001001          BNE     DCE
3284 015450 000207          RTS     PC           ;RETURN
3285
3286 015452 000240          DCE:   NOP
3287 015454 032777 020000 163106  BIT     #20000,2SWR ;SEE IF ERROR PRINT INHIBIT
3288 015462 001017          BNE     DCEX
3289 015464 005737 000620      TST    HDRFL        ;IF SO: BR
3290 015470 001004          BNE     DCEA        ;SEE IF PRINT HEADER
3291 015472 013704 000622      MOV     EMADDR,R4   ;IF NOT: BR
3292 015476 004737 017320      JSR    PC,TTOUT     ;PRINT HEADER
3293 015502 012704 022342      DCEA:  MOV     #MSG47,R4
3294 015506 004737 017320      JSR    PC,TTOUT     ;PRINT DRIVE CLEAR ERROR
3295 015512 004737 015174      JSR    PC,REGP      ;PRINT REGISTERS
3296 015516 005237 000666      INC     PFLG        ;SET PRINTED FLAG
3297 015522 005777 163042      DCEX:  TST    2SWR    ;SEE IF HALT ON ERROR
3298 015526 100001          BPL    DCEXA       ;IF NOT: BR
3299 015530 000000          HALT
3300 015532 005737 000666      DCEXA: TST    PFLG
3301 015536 001004          BNE     DCEXX
3302 015540 032777 020000 163022  BIT     #20000,2SWR ;SEE IF HAVE PRINTED
3303 015546 001741          BEQ    DCE           ;IF SO: BR
3304 015550 000240          DCEXX: NOP
3305 015552 012737 015330 000712  MOV     #DRVCLR,SCOLP ;SET SCOPE LOOP ADDRESS
3306 015560 000137 016210      JMP    SCOPE        ;GO DO SCOPE LOOP

```

```

;COMPOSITE ERROR CHECK SUBROUTINE*****
3307
3308
3309 015564 000240          ATTN:  NOP
3310 015566 005777 162730  TST      2DS          ;SEE IF ATA SET
3311 015572 001004          BNE     ATTA         ;IF SO: BR
3312 015574 012737 021700 000700  MOV     #MSG32,TEMP3
3313 015602 000427          BR      ATTP         ;ELSE PRINT ERROR
3314 015604 032777 040000 162710  ATTA:  BIT     #40000,2DS ;SEE IF COMPOSITE ERROR SET
3315 015612 001004          BNE     ATTB         ;IF SO: BR
3316 015614 012737 021662 000700  MOV     #MSG31,TEMP3
3317 015622 000417          BR      ATTP         ;ELSE PRINT ERROR
3318 015624 012703 000001          ATTB:  MOV     #1,R3   ;SET TEST BIT
3319 015630 012737 021716 000700  MOV     #MSG33,TEMP3
3320 015636 013704 000624          MOV     DRVN,R4     ;GET DRIVE NUMBER & BRANCH
3321 015642 001403          BEQ     ATTD         ;IF DRIVE 0
3322 015644 006303          ATTC:  ASL     R3     ;POSITION TEST BIT
3323 015646 005304          DEC     R4          ;SEE IF DONE
3324 015650 001375          BNE     ATTC         ;IF NOT: BR
3325 015652 030377 162650          ATTD:  BIT     R3,2AS ;SEE IF ATTN SUMMARY SET
3326 015656 001401          BEQ     ATTP         ;IF NOT: BR
3327 015660 000207          RTS     PC          ;ELSE RETURN
3328 015662 032777 020000 162700  ATTP:  BIT     #20000,2SWR ;SEE IF PRINT INHIBIT
3329 015670 001021          BNE     ATTX         ;IF SO: BR
3330 015672 005737 000620          TST     HDRFL       ;SEE IF DONE HEADER
3331 015676 001004          BNE     ATTPA        ;IF SO: BR
3332 015700 013704 000622          MOV     EMADDR,R4
3333 015704 004737 017320          JSR     PC,TTOUT    ;PRINT HEADER
3334 015710 013704 000700          ATTPA: MOV     TEMP3,R4
3335 015714 004737 017320          JSR     PC,TTOUT    ;PRINT ERROR TYPE
3336 015720 004737 015174          JSR     PC,REGP     ;PRINT REGISTERS
3337 015724 005237 000666          INC     PFLG        ;SET PRINT FLAG
3338 015730 005237 000620          INC     HDRFL       ;SET HEADER FLAG
3339 015734 005777 162630          ATTX:  TST     2SWR  ;SEE IF HALT ON ERROR
3340 015740 100001          BPL     ATTXA        ;IF NOT: BR
3341 015742 000000          HALT
3342 015744 005737 000666          ATTXA: TST     PFLG   ;SEE IF DONE PRINT
3343 015750 001004          BNE     ATTX         ;IF SO: BR
3344 015752 032777 020000 162610  BIT     #20000,2SWR ;BRANCH IF NO ERROR
3345 015760 001740          BEQ     ATTP         ;PRINTOUT DESIRED
3346 015762 005037 000666          ATXX:  CLR     PFLG  ;CLEAR PRINT FLAG
3347 015766 000207          RTS     PC          ;RETURN

```

```

;LOGIC TEST REGISTER BIT ERROR SUBROUTINE*****
3348
3349
3350 015770 012737 000001 000736 LTGER2: MOV #1,PEXFL ;SET FLAG
3351 015776 000240 LTGER1: NOP
3352 016000 005037 000666 CLR PFLG ;CLEAR PRINT FLAG
3353 016004 032777 020000 162556 BIT #20000,JSWR ;BRANCH IF ERROR
3354 016012 001055 BNE LTGIX ;PRINTOUT DESIRED
3355 016014 005737 000620 LTG1A: TST HDRFL ;SEE IF PRINT HEADER
3356 016020 001004 BNE LTG1B ;IF NOT: BR
3357 016022 013704 000622 MOV EMADR,R4
3358 016026 004737 017320 JSR PC,TTOUT ;PRINT HEADER
3359 016032 012737 000001 000620 LTG1B: MOV #1,HDRFL ;SET FLAG
3360 016040 013704 000672 MOV ERADR,R4
3361 016044 004737 017320 JSR PC,TTOUT ;PRINT ERROR CODE
3362 016050 005737 000736 TST PEXFL ;SEE IF PRINT EXPT-RCVD
3363 016054 001016 BNE LTG1T ;IF NOT: BR
3364 016056 012704 021224 MOV #MSG12,R4
3365 016062 004737 017320 JSR PC,TTOUT ;PRINT EXPT TAG
3366 016066 010103 MOV R1,R3
3367 016070 004737 017446 JSR PC,OCTP ;PRINT EXPT
3368 016074 012704 021233 MOV #MSG13,R4
3369 016100 004737 017320 JSR PC,TTOUT ;PRINT RCVD TAG
3370 016104 010203 MOV R2,R3
3371 016106 004737 017446 JSR PC,OCTP ;PRINT RCVD
3372 016112 032777 004000 162450 LTG1T: BIT #4000,JSWR
3373 016120 001010 BNE LTG1C
3374 016122 012704 021275 MOV #MSG16,R4
3375 016126 004737 017320 JSR PC,TTOUT
3376 016132 013703 000702 MOV ITCNT,R3
3377 016136 004737 017446 JSR PC,OCTP ;PRINT ITERATION
3378 016142 005237 000666 LTG1C: INC PFLG
3379 016146 000240 LTG1X: NOP
3380 016150 005777 162414 TST JSWR
3381 016154 100001 BPL LTG1X1 ;IF NOT STOP ON ERROR: BR
3382 016156 000000 HALT
3383 016160 005737 000666 LTG1X1: TST PFLG
3384 016164 001004 BNE LTG1XX ;IF HAVE PRINTED: BR
3385 016166 032777 020000 162374 BIT #20000,JSWR
3386 016174 001707 BEQ LTG1A
3387 016176 000240 LTG1XX: NOP
3388 016200 005037 000736 CLR PEXFL ;CLEAR EXPT-RCVD FLAG
3389 016204 000137 016210 JMP SCOPE ;GO TO SCOPE
3390
3391
3392 ;SCOPE LOOP ON ERROR SUBROUTINE*****
3393
3394 016210 000240 SCOPE: NOP
3395 016212 032777 040000 162350 BIT #40000,JSWR ;SEE IF LOOP ON ERROR
3396 016220 001001 BNE IS ;IF SO: BR
3397 016222 000207 RTS PC ;ELSE EXIT
3398 016224 000240 IS: NOP
3399 016226 005726 TST (SP)+ ;RESET STACK
3400 016230 000240 NOP
3401 016232 000240 NOP
3402 016234 000177 162452 JMP JSCLP ;LOOP ON ERROR
3403

```

```

3404                                     ;TEST ITERATION SUBROUTINE*****
3405
3406 016240 032777 004000 162322 ITER: BIT #4000,JSWR ;SEE IF ITERATIONS
3407 016246 001403 BEQ 2S ;IF SO: BR
3408 016250 005037 000702 1S: CLR ITCNT ;CLEAR ITERATION COUNTER
3409 016254 000207 RTS PC ;ELSE EXIT
3410 016256 005737 001016 2S: TST PCNTR ;NO SUBTEST ITERATIONS ON FIRST PASS
3411 016262 001772 BEQ 1S
3412 016264 005237 000702 INC ITCNT ;BUMP COUNTER
3413 016270 023737 000702 000606 CMP ITCNT,ITAMT ;SEE IF DONE ALL
3414 016276 001764 BEQ 1S ;IF SO: BR
3415 016300 005726 TST (SP)+ ;RESET STACK
3416 016302 017700 162406 MOV @ITRPL,RO ;SET ITERATION POINTER
3417 016306 000110 JMP (RO) ;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
3429
3430 016330 000240
3431 016332 012777 000040 162160
3432 016340 013777 000624 162152
3433 016346 013777 000664 162166
3434 016354 000207
3435
3436
3437
3438
3439
3440
3441
3442 016356 013746 000776
3443 016362 012737 001400 000776
3444 016370 000410
3445
3446
3447 016372 013746 000776
3448 016376 012737 002000 000776
3449 016404 000402
3450
3451
3452
3453 016406 013746 000776
3454 016412 012777 000040 162100
3455 016420 013777 000624 162072
3456 016426 013777 000664 162106
3457 016434 013746 000776
3458 016440 042716 174377
3459 016444 022726 001400
3460 016450 001005
3461 016452 032777 000040 162042
3462 016460 001420
3463 016462 000404
3464 016464 032777 000040 162030 1$:
3465 016472 001013
3466 016474 012777 000007 162006 2$:
3467 016502 032777 000200 162012 20$:
3468 016510 001774
3469 016512 032777 020000 162002 3$:
3470 016520 001374
3471 016522 053777 000776 162012 4$:
3472 016530 032777 000002 161764
3473 016536 001407
3474 016540 012777 000025 161742
3475 016546 032777 000200 161746 5$:
3476 016554 001774
3477 016556 012777 000011 161724 6$:
3478 016564 012637 000776
3479 016570 000207
3480
3481
3482
;INITIALIZE SUBROUTINE*****
INIT1: NOP
MOV #40,2CS ;INIT
INIT2: MOV DRVN,2CS ;SELECT DRIVE
MOV SLVN,2TC ;SELECT SLAVE
RTS PC ;RETURN
;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.
;SET SLAVE IN NRZ OFF BOT
INIT3: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
MOV #1400,UDES ;SET UNIT DESCRIPTION = NRZ
BR INITS ;GO TO INITS ROUTINE
;SET SLAVE IN PE OFF BOT
INIT4: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
MOV #2000,UDES ;SET UNIT DESCRIPTION = PE
BR INITS ;GO DO IT
;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
;IT IS ENTERED AT INITS WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
INIT: MOV UDES,-(SP) ;SAVE TEST'S UNIT DESCRIPTION
INITS: MOV #40,2CS ;INIT CONTROLLER
MOV DRVN,2CS ;SELECT TMD3 DRIVE
MOV SLVN,2TC ;SELECT TE16 SLAVE
MOV UDES,-(SP) ;GET SLAVE DESCRIPTION
BIC #174377,(SP) ;CLEAR ALL BUT DENSITY SELECT BITS
CMP #1400,(SP)+ ;BRANCH IF REQUESTING PE MODE
BNE 1$
BIT #40,2DS ;BRANCH IF SLAVE IS IN NRZ MODE
BEQ 4$ ;(PES = 0)
BR 2$
BIT #40,2DS ;BRANCH IF SLAVE IS IN PE MODE
BNE 4$
MOV #7,2C1 ;REWIND SLAVE
BIT #200,2DS ;WAIT FOR READY
BEQ 20$
BIT #20000,2DS ;WAIT UNTIL PIP CLEARS
BNE 3$
BIS UDES,2TC ;LOAD SLAVE DESCRIPTION
BIT #2,2DS ;BRANCH IF NOT AT BOT
BEQ 6$
MOV #25,2C1 ;ERASE TO GET OFF BOT
BIT #200,2DS ;WAIT FOR READY
BEQ 5$
MOV #11,2C1 ;RESET DRIVE
MOV (SP)+,UDES ;RESTORE UNIT DESCRIPTION
RTS PC ;RETURN

```

```

3483                                     ;MANUAL INSTRUCTION SUBROUTINE*****
3484
3485 016572 000240          INST:  NOP
3486 016574 004737 017320      JSR    PC, TTOUT          ;PRINT INSTRUCTION
3487 016600 012704 025646      MOV    #MSG0, 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 HARDWARE 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

```

3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584
3585

```
*****
: 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		10S:	MOV	(SP), R1		; RESTORE CHARACTER COUNT (FOR ↑U)
017002	005037	000674		CLR	TEMP1		; CLEAR FIRST CHARACTER FLAG
017006	005000			CLR	R0		
017010	004737	017256	1S:	JSR	PC, TTIN		; GO READ CHARACTER
017014	122737	000003 000616		CMPB	#3, TIB		; BRANCH IF NOT ↑C
017022	001003			BNE	11S		
017024	000005			RESET			; RESET
017026	000137	000200		JMP	#200		; RESTART PROGRAM
017032	122737	000015 000616	11S:	CMPB	#15, TIB		; SEE IF CR
017040	001004			BNE	2S		; IF NOT: BR
017042	005737	000674		TST	TEMP1		; SEE IF FIRST CHARACTER
017046	001471			BEQ	9S		; IF SO: BR
017050	000457			BR	6S		
017052	122737	000025 000616	2S:	CMPB	#25, TIB		; BRANCH IF NOT CONTROL U
017060	001005			BNE	21S		
017062	012704	022616		MOV	#MSG59, R4		; TYPE <CR><LF>
017066	004737	017320		JSR	PC, TTOUT		
017072	000742			BR	10S		
017074	122737	000177 000616	21S:	CMPB	#177, TIB		; BRANCH IF NOT 'RUBOUT'
017102	001012			BNE	3S		
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	1S		
017130	122737	000060 000616	3S:	CMPB	#60, TIB		; SEE IF CHAR IS LESS THAN 0
017136	101402			BLOS	4S		; IF NOT: BR
017140	000137	017236		JMP	TINER		; ELSE GO TO ERROR
017144	122737	000070 000616	4S:	CMPB	#70, TIB		; SEE IF CHAR IS GREATER THAN 7
017152	101002			BHI	5S		; IF NOT: BR
017154	000137	017236		JMP	TINER		; ELSE GO TO ERROR
017160	005237	000674	5S:	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,T1B	:STRIP ASCII
3587	017200	053700	000616		BIS	T1B,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	T1NER	: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	T1NER	: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
3602
3603 017236 012704 022026
3604 017242 004737 017320
3605 017246 005726
3606 017250 162716 000020
3607 017254 000207
3608
3609
3610
3611 017256 005277 161310
3612 017262 105777 161304
3613 017266 100375
3614 017270 017737 161300 000616
3615 017276 042737 000200 000616
3616 017304 013737 000616 000614
3617 017312 004737 017420
3618 017316 000207
3619
3620
3621
3622 017320 112437 000614
3623 017324 122737 000043 000614
3624 017332 001440
3625 017334 122737 000045 000614
3626 017342 001403
3627 017344 004737 017420
3628 017350 000763
3629 017352 112737 000015 000614
3630 017360 004737 017420
3631 017364 012703 000004
3632 017370 005037 000614
3633 017374 004737 017420
3634 017400 005303
3635 017402 001372
3636 017404 112737 000012 000614
3637 017412 004737 017420
3638 017416 000740
3639 017420 105777 161152
3640 017424 100375
3641 017426 113777 000614 161144
3642 017434 000207
3643
3644
3645
3646
3647 017436 012737 000001 017666
3648 017444 000402
3649 017446 005037 017666
3650 017452 010304
3651 017454 001006
3652 017456 005737 017666
3653 017462 001003
3654 017464 004737 017646
3655 017470 000447

;TTY ENTRY ERROR SUBROUTINE*****
TINER: MOV #MSG40,R4
        JSR PC,TTOUT ;PRINT?
        TST (SP)+ ;POP CHAR COUNT OFF STACK
        SUB #20,(SP) ;RESET SP TO START OF VALUE ROUTINE
        RTS PC ;REDO VALUE ENTRY

;TTY READ SUBROUTINE*****
TTIN: INC @TKS
1S: TSTB @TKS
    BPL 1S
    MOV @TKB,TIB
    BIC #200,TIB ;STRIP PARITY BIT
    MOV TIB,TOB ;MOVE CHAR TO TTY OUTPUT BFR
    JSR PC,TOG ;AND ECHO IT
    RTS PC

;TTY OUTPUT SUBROUTINE*****
TTOUT: MOVB (R4)+,TOB
        CMPB #43,TOB
        BEQ TEX
        CMPB #45,TOB
        BEQ 1S
        JSR PC,TOG
        BR TTOUT
1S: MOVB #15,TOB
    JSR PC,TOG
    MOV #4,R3
2S: CLR TOB
    JSR PC,TOG
    DEC R3
    BNE 2S ;DO FILLERS
    MOVB #12,TOB
    JSR PC,TOG
    BR TTOUT
TOG: TSTB @TPS
    BPL TOG
    MOVB TOB,@TPB
    RTS PC
TEX:

;OCTAL OUTPUT SUBROUTINE*****
OCTPE: MOV #1,OFL
        BR OCTPE1
OCTP: CLR OFL ;CLEAR FLAG FOR LEADING ZERO
OCTPE1: MOV R3,R4 ;SEE IF NUMBER IS ZERO
        BNE OCTPO ;IF NOT ZERO: BR
        TST OFL ;SEE IF PRINT ALL 0
        BNE OCTPO ;IF SO: BR
        JSR PC,OCTPG1 ;ELSE PRINT ZERO
        BR OCTP3 ;SPACE AND EXIT
    
```

3656	017472	032704	100000		OCTP0:	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	OCTPG0	
3692	017632	005737	017666			TST	OFL	
3693	017636	001001				BNE	OCTPG0	
3694	017640	000207				RTS	PC	
3695								
3696	017642	005237	017666		OCTPG0:	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$: MOV #060,@TPB
3716 017734 005337 000614 3$: 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

```

```

3754
3755 ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
3756 ;IF A CONTROL G (↑G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
3757 020074 022737 000176 000570 GTSWR: CMP #SWREG,SWR ;BRANCH IF SOFTWARE SWR
3758 020102 001032 BNE IS ;NOT INVOKED
3759 020104 004737 020172 JSR PC,SAVE ;SAVE REGISTERS ON THE STACK
3760 020110 012704 026173 MOV #SMSWR,R4 ;TYPE 'SWR = '
3761 020114 004737 017320 JSR PC,TTOUT
3762 020120 017703 160444 MOV #SMR,R3 ;GET CURRENT VALUE
3763 020124 004737 017436 JSR PC,OCPE ;AND TYPE IT
3764 020130 012704 026203 MOV #SMNEW,R4 ;ASK FOR NEW VALUE
3765 020134 004737 017320 JSR PC,TTOUT
3766 020140 013705 000570 MOV SWR,R5 ;NEW VALUE WILL BE RETURNED IN (R5)
3767 020144 012701 000007 MOV #7,R1 ;LIMIT TO 7 CHARACTERS
3768 020150 012702 177777 MOV #177777,R2 ;LIMIT RESPONSE TO BETWEEN
3769 020154 012703 000000 MOV #0,R3 ;0 AND 177777
3770 020160 004737 016776 JSR PC,TTR ;GET RESPONSE
3771 020164 004737 020214 JSR PC,.RESTORE ;RESTORE REGISTERS
3772 020170 000207 IS: RTS PC ;RETURN TO CALLER
3773
3774 020172 010546 ;:ROUTINE TO SAVE REGISTERS ON THE STACK
3775 020174 010446 .SAVE: MOV %5,-(SP) ;:R5 IS SAVED AT 12(SP)
3776 020176 010346 MOV %4,-(SP) ;:R4 IS SAVED AT 10(SP)
3777 020200 010246 MOV %3,-(SP) ;:R3 IS SAVED AT 6(SP)
3778 020202 010146 MOV %2,-(SP) ;:R2 IS SAVED AT 4(SP)
3779 020204 010046 MOV %1,-(SP) ;:R1 IS SAVED AT 2(SP)
3780 020206 016646 000014 MOV %0,-(SP) ;:R0 IS SAVED AT (SP)
3781 020212 000207 MOV 14(SP),-(SP) ;:PUSH RETURN PC ON THE STACK
3782 RTS PC ;:RETURN TO CALLER
3783
3784 020214 012666 000014 ;:ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
3785 020220 012600 .RESTORE:MOV (SP)+,14(SP) ;:STORE RETURN PC ON STACK
3786 020222 012601 MOV (SP)+,%0
3787 020224 012602 MOV (SP)+,%1
3788 020226 012603 MOV (SP)+,%2
3789 020230 012604 MOV (SP)+,%3
3790 020232 012605 MOV (SP)+,%4
3791 020234 000207 RTS PC ;:RETURN
3792
3793 ;MESSAGE TABLE*****
3794
3795 MSG1: .ASCII /%TMD3-TE16 CONTROL LOGIC TEST- PART I (DZTEA-A)/
3796 020236 022445 046524 031460
3797 020244 052055 030505 020066
3798 020252 047503 052116 047522
3799 020260 020114 047514 044507
3800 020266 020103 042524 052123
3801 020274 020055 040520 052122
3802 020302 044440 024040 055104
3803 020310 042524 026501 024501
3804 020316 025045 025052 051501 .ASCII /%***ASSURE TAPE IS AT BOT***/
3805 020324 052523 042522 052040
3806 020332 050101 020105 051511
3807 020340 040440 020124 047502
3808 020346 025124 025052
3809 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
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
3825	020502	042040	044522	042526
3826	020510	040440	042104	042522
3827	020516	051523	052040	051505
3828	020524	035524		
3829	020526	020045	047105	042524
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	040524	052116
3839	020622	021456		
3840	020624	047045	047117	042455
3841	020632	044530	052123	042040
3842	020640	044522	042526	021440
3843	020646	051045	020110	042504
3844	020654	042524	052103	042105
3845	020662	021440		
3846	020664	052045	030115	020063
3847	020672	042504	042524	052103
3848	020700	042105	021440	
3849	020704	054105	052120	047055
3850	020712	052117	051040	041505
3851	020720	042126	043	
3852	020723	122	053103	026504
3853	020730	047516	020124	054105
3854	020736	052120	043	
3855	020741	045	046123	053101
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
3862	021010	046123	053101	020105
3863	021016	042101	051104	051505
3864	021024	020123	042524	052123
3865	021032	073		

MSG2: .ASCII /%DRIVE NUMBER OR (CR) WHEN DONE #/

MSG2A: .ASCII /%FOR DRIVE ADDRESS TEST; /

.ASCII /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT. #/

MSG3: .ASCII /%NON-EXIST DRIVE #/

MSG4: .ASCII /%RH DETECTED #/

MSG5: .ASCII /%TMO3 DETECTED #/

MSG6: .ASCII /EXPT-NOT RECVD#/

MSG7: .ASCII /RCVD-NOT EXPT#/

MSG8: .ASCII /%SLAVE NUMBER OR (CR) WHEN DONE #/

MSG8A: .ASCII /%FOR SLAVE ADDRESS TEST; /

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	052516	041115			
3870	021062	051105	020054	046101			
3871	021070	020114	052117	042510			
3872	021076	051522	051440	047510			
3873	021104	046123	020104	042502			
3874	021112	047040	047117	042455			
3875	021120	044530	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	036517	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	

M08

TM03/TE16 CONTROL LOGIC TEST PART I
DZTEAA.P11 01-APR-77 12:00

MACY11 27(1006) 01-APR-77 12:02 PAGE 103

3978	022124	042105	020072	040510	
3979	022132	052114	045		
3980	022135	122	051505	046105	.ASCII /RESELECT AND PRESS CONTINUE%/
3981	022143	041505	020124	047101	
3982	022150	020104	051120	051505	
3983	022156	020123	047503	052116	
3984	022164	047111	042523	021445	
3985	022173	051045	043503	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	022267	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		
4037	022620	021534		
4038	022622	051045	046505	053117
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
4046	022676	051101	020105	053523
4047	022704	020122	047111	052440
4048	022712	042523	021445	

MSG59: .ASCII /%#/

MSG60: .ASCII /\#/

MSG62: .ASCII /%REMOVE TMDP FROM SLAVE TO BE TESTED%/

MSG63: .ASCII /%HARDWARE SWR IN USE%/

;TEST HEADER*****

4049				
4050				
4051	022716	022445	047514	044507
4052	022724	020103	042524	052123
4053	022732	030440	020072	051104
4054	022740	053111	020105	042101
4055	022746	051104	051505	044523
4056	022754	043516	024040	034115
4057	022762	030071	020071	044122
4058	022770	021451		
4059	022772	022445	047514	044507
4060	023000	020103	042524	052123
4061	023006	031040	020072	042522
4062	023014	044507	052123	051105
4063	023022	040440	042104	042522
4064	023030	051523	047111	020107
4065	023036	046450	034470	034460
4066	023044	051040	024510	043
4067	023051	045	046045	043517
4068	023056	041511	052040	051505
4069	023064	020124	035063	041440
4070	023072	047117	051124	046117
4071	023100	041040	051525	052040
4072	023106	051505	020124	051050
4073	023114	020110	034115	030071
4074	023122	020065	034115	030071
4075	023130	024471	043	
4076	023133	045	046045	043517
4077	023140	041511	052040	051505
4078	023146	020124	035064	051440
4079	023154	040514	042526	040440
4080	023162	042104	042522	051523
4081	023170	047111	020107	046450
4082	023176	034470	032460	046440
4083	023204	034470	031460	021451
4084	023212	022445	047514	044507
4085	023220	020103	042524	052123
4086	023226	032440	020072	051115
4087	023234	041040	052111	052040
4088	023242	051505	020124	046450
4089	023250	034470	032460	021451
4090	023256	022445	047514	044507
4091	023264	020103	042524	052123
4092	023272	033040	020072	041524
4093	023300	041040	052111	052040
4094	023306	051505	020124	046450
4095	023314	034470	032460	021451
4096	023322	022445	047514	044507
4097	023330	020103	042524	052123
4098	023336	033440	020072	041506
4099	023344	041040	052111	052040
4100	023352	051505	020124	046450
4101	023360	034470	032460	021451
4102	023366	022445	047514	044507
4103	023374	020103	042524	052123
4104	023402	030440	035060	043040

MSLT1: .ASCII /%%LOGIC TEST 1: DRIVE ADDRESSING (M8909 RH)*/

MSLT2: .ASCII /%%LOGIC TEST 2: REGISTER ADDRESSING (M8909 RH)*/

MSLT3: .ASCII /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905 M8909)*/

MSLT4: .ASCII /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905 M8903)*/

MSLT5: .ASCII /%%LOGIC TEST 5: MR BIT TEST (M8905)*/

MSLT6: .ASCII /%%LOGIC TEST 6: TC BIT TEST (M8905)*/

MSLT7: .ASCII /%%LOGIC TEST 7: FC BIT TEST (M8905)*/

MSLT10: .ASCII /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905)*/

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	024105	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	

TMO3/TE16 CONTROL LOGIC TEST PART I
DZTEAA.P11 01-APR-77 12:00

MACY11 27(1006) 01-APR-77 12:02 PAGE 108

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
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
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
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
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
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

MSLT45: .ASCII /%:LOGIC TEST 45: INCORRECTABLE DATA (M8902 M8904)#/

MSLT46: .ASCII /%:LOGIC TEST 46: PEF(M8902)#/

MSLT47: .ASCII /%:LOGIC TEST 47: FC OVERFLOW (M8905)#/

MSLT50: .ASCII /%:LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/

MSLT51: .ASCII /%:LOGIC TEST 51: NEF WHEN WRITE NRZ ON PE SLAVE#/

4317
4318
4319
4320 025646 052045 050131 020105
4321 025654 051103 053440 042510
4322 025662 020116 042522 042101
4323 025670 035531 043
4324 025673 045 046445 052517
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
4336 025776 047524 047440 043106
4337 026004 044514 042516 021472
4338 026012 046445 053117 020105
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
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
4354 026144 020105 040524 042520
4355 026152 052040 020117 047502
4356 026160 035524 047440 020116
4357 026166 044514 042516 043

;MANUAL INSTRUCTION*****

MMSG0: .ASCII /%TYPE CR WHEN READY;#/
MMSG1: .ASCII /%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:##/

MMSG2: .ASCII /%SET TO OFFLINE:##/

MMSG3: .ASCII /%MOVE FORWARD TO EOT, ONLINE:##/

MMSG4: .ASCII /%OFF LINE REVERSE PAST EOT, INSERT WRITE RING, ON LINE##/

MMSG5: .ASCII /%MOVE TAPE TO BOT; ON LINE##/

```

4358
4359
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

```

;TAG MESSAGE

414	026451	045	051516	020107	TMS27:	.ASCII	/XNSG #/
415	026456	043					
416	026457	045	041506	020105	TMS28:	.ASCII	/XFCE #/
417	026464	043					
418	026465	045	051503	021440	TMS29:	.ASCII	/XCS #/
419	026472	044445	046524	021440	TMS30:	.ASCII	/XITM #/
420	026500	047045	043105	021440	TMS31:	.ASCII	/XNEF #/
421	026506	042045	042524	021440	TMS32:	.ASCII	/XDTE #/
422	026514	047445	044520	021440	TMS33:	.ASCII	/XOPI #/
423	026522	053445	044522	042524	TMS33A:	.ASCII	/XWRITE OPI #/
424	026530	047440	044520	021440			
425	026536	051045	040505	020104	TMS33B:	.ASCII	/XREAD OPI #/
426	026544	050117	020111	043			
427	026551	040	041517	052503	TMS33C:	.ASCII	/ OCCURED TO SOONX#/
428	026556	042522	020104	047524			
429	026564	051440	047517	022516			
430	026572	043					
431	026573	040	041517	052503	TMS33D:	.ASCII	/ OCCURRED TO LATEX#/
432	026600	051122	042105	052040			
433	026606	020117	040514	042524			
434	026614	021445					
435	026616	043040	044501	042514	TMS33E:	.ASCII	/ FAILED TO SETX#/
436	026624	020104	047524	051440			
437	026632	052105	021445				
438	026636	052445	051516	021440	TMS34:	.ASCII	/XUNS #/
439	026644	041445	051117	020122	TMS35:	.ASCII	/XCORR #/
440	026652	043					
441	026653	045	051103	020103	TMS36:	.ASCII	/XCRC #/
442	026660	043					
443	026661	045	040523	020103	TMS37:	.ASCII	/XSAC #/
444	026666	043					
445	026667	045	041506	020123	TMS38:	.ASCII	/XFCS #/
446	026674	043					
447	026675	045	041501	046103	TMS39:	.ASCII	/XACCL #/
448	026702	021440					

.EVEN
;WRITE BUFFER

WDATA:

453	026704	000100					
454	026704	177777					-1
455	026706	177777					-1
456	026710	177777					-1
457	026712	177777					-1
458	026714	177777					-1
459	026716	177777					-1
460	026720	177777					-1
461	026722	177777					-1
462	026724	177777					-1
463	026726	177777					-1
464	026730	177777					-1
465	026732	177777					-1
466	026734	177777					-1
467	026736	177777					-1
468	026740	177777					-1
469	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

;READ BUFFER

RDATA:

4522	027104	000100	0
4523	027104	000000	0
4524	027106	000000	0
4525	027110	000000	0

ADDFL	000746	GTSWR	020074	LT10E1	004426	LT21P	003216	LT30C	007640
AS	000526	HDRFL	000620	LT10IT	004336	LT2X	003222	LT30D	007656
ASF	000730	HERE	002512	LT10X	004450	LT20	005642	LT30E	007706
ATAF	000720	ILFT	000544	LT11	004460	LT20A	005670	LT30IT	007456
ATTA	015604	INIT	016406	LT11B	004506	LT20B	005746	LT30X	007732
ATTB	015624	INIT1	016330	LT11C	004546	LT20C	005756	LT31	007752
ATTC	015644	INIT2	016340	LT11E1	004566	LT20IT	005656	LT31A	010130
ATTD	015652	INIT3	016356	LT11E2	004616	LT20X	005770	LT31IT	007760
ATTN	015564	INIT4	016372	LT11E3	004646	LT21	006004	LT31X	010234
ATTP	015662	INIT5	016412	LT11IT	004462	LT21A	006110	LT32	010564
ATTPA	015710	INMT	016310	LT11X	004674	LT21B	006120	LT32IT	010600
ATTX	015734	INST	016572	LT12	004704	LT21IT	006020	LT32X	010712
ATTXA	015744	ITAMT	000606	LT12	004730	LT21XA	006130	LT32XX	010722
ATTXX	015762	ITCNT	000702	LT12C	004766	LT22	006150	LT33	010726
BA	000514	ITER	016240	LT12E1	005004	LT22A	006250	LT33IT	010742
CC	000530	ITRLP	000714	LT12E2	005026	LT22IT	006164	LT33X	011020
CHNFLG	001416	LTADD	000742	LT12E3	005050	LT22X	006260	LT34	011030
CRCNT	000774	LTGA	014706	LT12IT	004706	LT23	006274	LT34A	011070
CS	000520	LTGA1	014724	LT12X	005070	LT23A	006374	LT34A1	011050
CI	000510	LTGA2	014756	LT13	005100	LT23IT	006310	LT34B	011120
DATAD	000760	LTGB	015004	LT13A	005140	LT23X	006404	LT34C	011124
DATC	000754	LTGC	015010	LT13E1	005144	LT24	006424	LT34IT	011044
DB	000532	LTGCO	015076	LT13IT	005110	LT24B	006552	LT34X	011154
DCA	015362	LTGD	015102	LT13X	005172	LT24B0	006574	LT34XX	011160
DCA0	015376	LTGER	014670	LT14	005202	LT24C	006606	LT35	011164
DCB	015434	LTGER0	014662	LT14A	005224	LT24D	006670	LT35A	011266
DCC	015442	LTGER1	015776	LT14IT	005242	LT24IT	006440	LT35IT	011200
DCD	015336	LTGER2	015770	LT14X	005302	LT24X	006714	LT35X	011326
DCE	015452	LTGER3	014650	LT14XX	005306	LT25	006750	LT36	011336
DCEA	015502	LTGX	015132	LT15	005312	LT25A	007056	LT36IT	011352
DCEX	015522	LTGXA	015142	LT15A	005334	LT25IT	006756	LT36X	011446
DCEXA	015532	LTGXX	015160	LT15IT	005352	LT25X	007066	LT37	011456
DCEXX	015550	LTG1A	016014	LT15X	005412	LT26	007102	LT37A	011532
DERFL	000766	LTG1B	016032	LT15XX	005416	LT26IT	007110	LT37B	011564
DOUT	017670	LTG1C	016142	LT16	005422	LT26X	007302	LT37IT	011472
DOUTD	017746	LTG1T	016112	LT16A	005444	LT27	007316	LT37X	011616
DRVCLR	015330	LTG1X	016146	LT16IT	005462	LT27A	007352	LT4	003414
DRVN	000624	LTG1XX	016176	LT16X	005522	LT27B	007410	LT4A	003532
DRVTP	000604	LTG1X1	016160	LT16XX	005526	LT27IT	007342	LT4B	003574
DS	000522	LT1	002552	LT17	005532	LT27X	007420	LT4C	003602
DT	000536	LT1A	002676	LT17A	005554	LT27XX	007430	LT4D	003632
EMADD	000622	LT1B	002744	LT17IT	005572	LT3	003232	LT4ERG	003660
EORPA	014514	LT1C	002752	LT17X	005632	LT3A	003252	LT4ER1	003642
EORPX	014642	LT1ER	002762	LT17XX	005636	LT3B	003272	LT4ER2	003652
EORP1	014550	LT1ER1	002772	LT2	003026	LT3C	003306	LT4G	003462
EORP1A	014616	LT1ER2	003000	LT2A	003066	LT3ER1	003316	LT4G0	003452
EORP2	014640	LT1G	002620	LT2B	003106	LT3ER2	003344	LT4X	003710
ER	000524	LT1G0	002610	LT2C	003122	LT3IT	003234	LT40	011626
ERADD	000672	LT1X	003022	LT2ERG	003202	LT3X	003366	LT40IT	011642
ERRF	000726	LT10	004334	LT2ER1	003132	LT3XX	003404	LT40X	011736
EXFL	000716	LT10A	004352	LT2ER2	003150	LT30	007434	LT40XX	011742
FC	000516	LT10A1	004350	LT2ER3	003166	LT30A	007536	LT41	011746
FUN	000752	LT10B	004406	LT2IT	003030	LT30B	007600	LT41IT	011762

LT41X	012156	LT5	003714	MSG23	021454	MSLT21	024133	RBUFF	030232
LT42	012210	LT5A	003734	MSG24	021507	MSLT22	024167	RCOP	001010
LT42A	012312	LT5B	003754	MSG25	021535	MSLT23	024224	RDAO	000762
LT42B	012326	LT5C	003764	MSG25A	021564	MSLT24	024256	RDATA	027104
LT42B1	012354	LT5D	003774	MSG26	021611	MSLT25	024326	RDRVF	001006
LT42B2	012374	LT5E	004016	MSG27	021640	MSLT26	024362	REGP	015174
LT42C	012440	LT5ER1	004030	MSG28	020624	MSLT27	024416	REGS	000612
LT42D	012454	LT5ER2	004052	MSG29	021654	MSLT3	023051	RTRN	000570
LT42E	012504	LT5IT	003722	MSG30	021662	MSLT30	024452	SAV1	000704
LT42IT	012246	LT5X	004074	MSG31	021700	MSLT31	024510	SAV2	000706
LT42X	012524	LT50	014134	MSG32	021716	MSLT32	024544	SAV3	000710
LT43	012540	LT50IT	014150	MSG33	021737	MSLT33	024600	SCF	000732
LT43C	012640	LT50X	014274	MSG34	021754	MSLT34	024634	SCOLP	000712
LT43D	012654	LT51	014310	MSG35	021772	MSLT35	024670	SCOPE	016210
LT43E	012706	LT51IT	014324	MSG36	022010	MSLT36	024732	SERFL	000772
LT43F	012740	LT51X	014450	MSG37	020646	MSLT37	024774	SERNUM	000602
LT43IT	012566	LT6	004104	MSG40	022026	MSLT4	023133	SKAT	001014
LT43X	012760	LT6A	004122	MSG41	022032	MSLT40	025037	SLAF	000722
LT44	012774	LT6A1	004120	MSG42	022051	MSLT41	025104	SLVN	000664
LT44A	013060	LT6B	004126	MSG43	022066	MSLT42	025153	SN	000540
LT44A1	013072	LT6D	004146	MSG44	022172	MSLT43	025207	SNPG	020050
LT44B	013102	LT6ER1	004166	MSG45	022214	MSLT44	025243	SNPT	017772
LT44C	013116	LT6IT	004106	MSG46	022236	MSLT45	025323	SSCF	000724
LT44D	013144	LT6X	004210	MSG47	022342	MSLT46	025405	START	001330
LT44E	013146	LT7	004220	MSG5	020664	MSLT47	025441	STATC	001012
LT44F	013206	LT7A	004236	MSG50	022374	MSLT5	023212	STATF	001004
LT44IT	013022	LT7B	004256	MSG51	022413	MSLT50	025506	STATIC	014464
LT44X	013226	LT7C	004226	MSG53	022442	MSLT51	025566	STFLG	000740
LT44XX	013236	LT7ER1	004276	MSG54	022454	MSLT6	023256	STSCD	002402
LT45	013242	LT7IT	004222	MSG55	022465	MSLT7	023322	STO	001662
LT45A	013350	LT7X	004320	MSG56	022535	MTINT	016640	ST1	001704
LT45B	013364	MMSG0	025646	MSG57	022562	NRZOF	000662	ST2	002066
LT45D	013414	MMSG1	025673	MSG58	022600	NXTDRV	002152	SWR	000570
LT45E	013326	MMSG2	025771	MSG59	022616	NXTSLV	002224	SWREG	000176
LT45E1	013340	MMSG3	026012	MSG6	020704	OCTP	017446	TADX	001324
LT45F	013446	MMSG4	026050	MSG60	022620	OCTPE	017436	TC	000542
LT45IT	013270	MMSG5	026137	MSG62	022622	OCTPE1	017452	TEMP1	000674
LT45X	013472	MR	000534	MSG63	022670	OCTPG	017624	TEMP2	000676
LT45XX	013502	MSG1	020236	MSG7	020723	OCTPG0	017642	TEMP3	000700
LT46	013506	MSG10	021153	MSG8	020741	OCTPG1	017646	TEND	002436
LT46A	013570	MSG11	021177	MSG8A	021002	OCTP0	017472	TENDX	002534
LT46B	013636	MSG12	021224	MSG9	021131	OCTP1	017512	TEX	017434
LT46C	013652	MSG13	021233	MSLT1	022716	OCTP2	017520	TIB	000616
LT46D	013704	MSG14	021242	MSLT10	023366	OCTP3	017610	TIMER	010350
LT46IT	013534	MSG15	021257	MSLT11	023441	OFL	017666	TIMERO	010374
LT46X	013730	MSG16	021275	MSLT12	023506	PATRN	001002	TIMER1	010324
LT46XX	013740	MSG18	021305	MSLT13	023557	PCNTR	001016	TIMOK	010442
LT47	013744	MSG19	021324	MSLT14	023624	PEXFL	000736	TIMON	010256
LT47A	014034	MSG2	020434	MSLT15	023672	PFLG	000666	TIMOVF	010406
LT47B	014046	MSG2A	020475	MSLT16	023740	POST	027476	TINER	017236
LT47C	014052	MSG20	021343	MSLT17	024006	PRE	027354	TKB	000574
LT47IT	013760	MSG21	021375	MSLT2	022772	PREFL	000770	TKS	000572
LT47X	014124	MSG22	021433	MSLT20	024054	PSW	000566	TLAST	001326

TMS1	026213	TMS28	026457	TMS6	026251	TR13	000654	WCPO	001042
TMS10	026301	TMS29	026465	TMS7	026257	TR14	000656	WCP2	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	T08	000614	TSCDA	002302	WDATA	026704
TMS14	026331	TMS32	026506	T0G	017420	TSCD0	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	TRO0	000626	TSTTBL	001054	WPGFL	001000
TMS19	026367	TMS33D	026573	TRO1	000630	TTIN	017256	WTAD	000756
TMS2	026221	TMS33E	026616	TRO2	000632	TTINT	016650	W2FLG	000764
TMS20	026375	TMS34	026636	TRO3	000634	TTOUT	017320	SDONE	002444
TMS21	026404	TMS35	026644	TRO4	000636	TTR	016776	SENDAD	002502
TMS22	026412	TMS36	026653	TRO5	000640	T24FL	000744	SMNEW	026203
TMS23	026421	TMS37	026661	TRO6	000642	UDES	000776	SMSMR	026173
TMS24	026427	TMS38	026667	TRO7	000644	VECT	000610	SSVPC =	000764
TMS25	026435	TMS39	026675	TR10	000646	WAM	000750	=	030234
TMS26	026443	TMS4	026234	TR11	000650	WBUF	027620	.RESTO	020214
TMS27	026451	TMS5	026242	TR12	000652	WC	000512	.SAVE	020172

. ABS. 030234 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0DZTEAA,DZTEAA/SOL+DZTEAA
RUN-TIME: 4 8 .6 SECONDS
RUN-TIME RATIO: 200/13=14.8
CORE USED: 7K (13 PAGES)