TM03/TU45

CONTROL LOGIC TEST PART 1

CZTUOA0

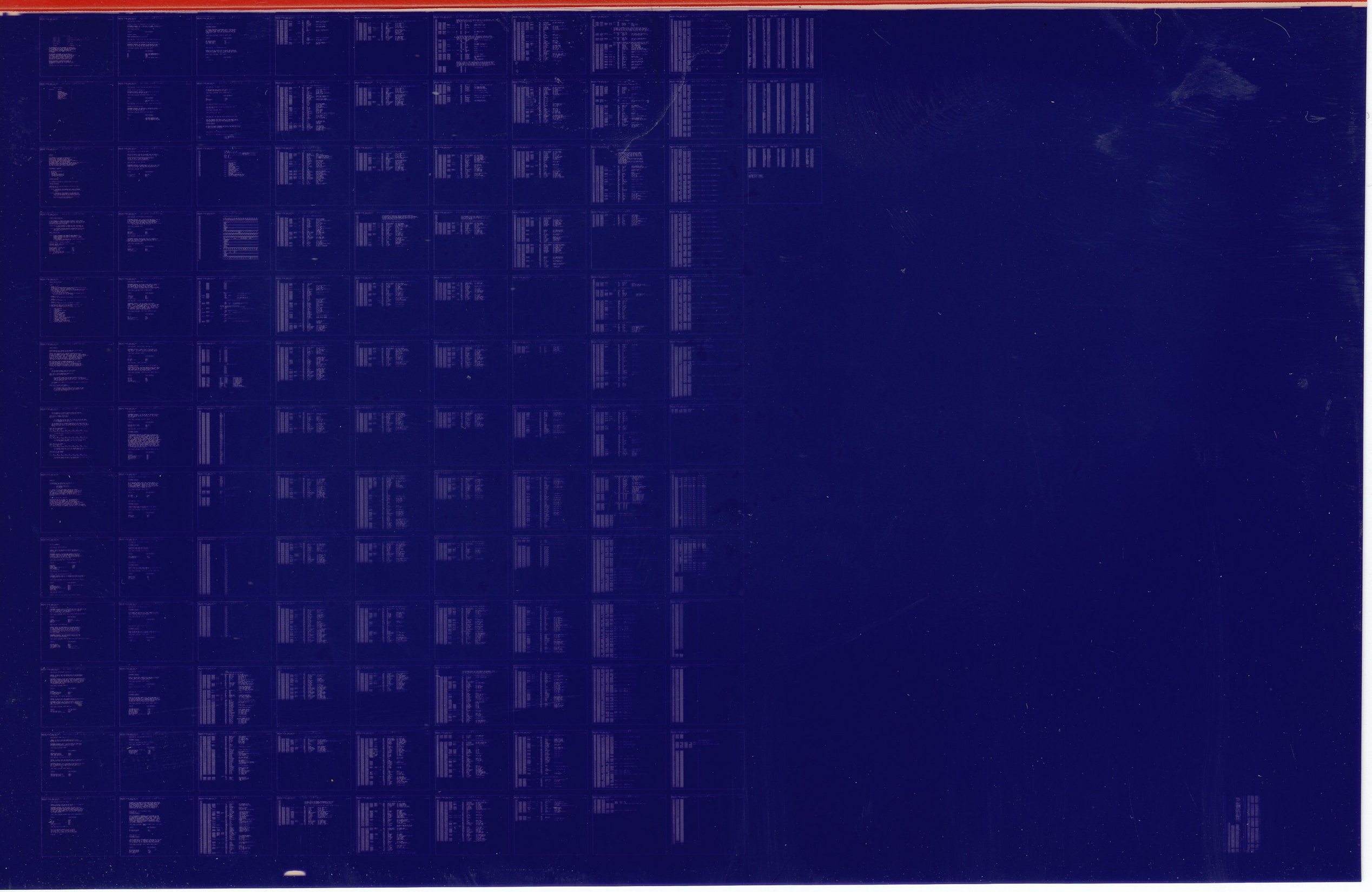
AH-E488A-MC

FICHE 1 OF 1

JUL 1978

digital

MADE IN USA



```
                .REM    %
```

IDENTIFICATION

PRODUCT CODE:	AC-E487A-MC
PRODUCT NAME:	CZTUOA0 TM03/TU45 CONTROL LOGIC TEST PART I
DATE CREATED:	25 MAY 1978
MAINTAINER:	CSS - NASHUA
AUTHOR:	J. G. ADAMS/R. J. COLLINS

TABLE OF CONTENTS

1. ABSTRACT

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

2. REQUIREMENTS (HARDWARE)

A. ANY PDP-11 PROCESSOR
B. 8K OF CORE
C. CONSOLE TTY
D. TM03 MAGTAPE CONTROLLER
E. MASSBUS CONTROLLER (RH)
F. TU45 MAGTAPE TRANSPORT

3. LOADING PROCEDURE

USE STANDARD PROCEDURE FOR LOADING BINARY PAPER TAPE.

4. STARTING PROCEDURE

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

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

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

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

4.1 AUTOMATIC MODE OPERATION

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

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

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

4.2 SAMPLE START AT 200

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

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

TM03-TU45 CONTROL LOGIC TEST- PART I (CZTUOA0)
***ASSURE TAPE IS AT BOT***
TYPE  C TO RESTART

REGISTER START: <172440> (CR)                   REGS:
VECTOR ADDRESS: <224> (CR)                      VECT:
TM03 DRIVE: <0> (CR)                            DRVN:
TU45 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 ITERVENTION TESTS
       0=INHIBIT MANUAL INTERVENTION
SW5-0: SELECT INDIVIDUAL TEST ** 00=DO ALL TESTS

6. ERROR PRINTOUTS

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

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

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

EXAMPLES:

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

```
LOGIC TEST 1: DRIVE ADDRESSING (M8939 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    WC     BA     FC     CS2    DS     ER     AS     MR     TC
004260 000000 033726 000000 000100 010600 000000 000000 177712 140300
```

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

```
LOGIC TEST 26: FCE (M8939)
ERR NOT SET
CS1    WC     BA     FC     CS2    DS     ER     AS     MR     TC
004260 000000 001376 000000 000100 110600 001000 000001 000000 100300
```

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

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

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

```
LOGIC TEST 32: UNS (M8939)
NOT RESET BY DRIVE CLEAR

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

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

7. OPERATION

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

1. LOAD ADDRESS 200 OR 210
2. SET SWITCHES FOR DESIRED TEST CYCLE
3. PRESS START

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

SINGLE TEST SELECTION: (SW0-SW5)

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

8. SUB TEST SUMMARIES

LOGIC TEST #1: DRIVE ADDRESSING

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

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

LIKELY FAULT LOCATIONS: M5904,CABLE,M5903,M8939

CIRCUITS	PRINT REFERENCES
RH-DS BITS	(CSRB)
RH-NED BIT	(CSRB)
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,M8939,M8905-YB,M8933

CIRCUITS	PRINT REFERENCE
C-LINES	(MB1,2,3),(MBI3),(MBI4),(MBI5)
RH REGISTER SELECT	(BCTA)
TM03 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

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,M8939,M8905-YB,M8933

CIRCUITS	PRINT REFERENCE
C-LINES	(MB1,2,3)
C-BUS MULTIPLEXERS	(MBI3,4,5,8)(TCCM7)(MR)
ERROR BIT	(MBI11)
MCPE BIT	(PACA)

LOGIC TEST #4: SLAVE ADDRESSING

PURPOSE: VERIFY THE FUNCTIONING OF THE SLAVE ADDRESS BITS IN
THE TAPE CONTROL REGISTER THE SLAVE ADDRESS BUS LINES,
THE ADDRESS DECODE CIRCUIT IN THE TU45 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-YB,M8937,CABLE,M9001,M8928,M9001YA,M8933

CIRCUITS	PRINT REFERENCE
REGISTER SELECT	(MBI2)
SLAVE ADDRESS BITS	(MR6)
SLAVE ADDRESS LINES	(M8937,2-2),(LAW6)
TU45 ADDRESS DECODE	(LAW6)
SPR BIT	(LAW6)(M9001YA)(TCCM7)

LOGIC TEST #5: MAINTENANCE REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8905-YB

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

LOGIC TEST #6: TAPE CONTROL REGISTER BITS

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

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

LIKELY FAULT LOCATIONS: M8939,M8905-YB

CURCUITS	PRINT REFERENCE
TM03 REGISTER SELECT	(MBI2)
TC FLIP-FLOPS, MULTIPLEXERS	(MR6)

LOGIC TEST #7: FRAME COUNT BIT TEST

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

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

LIKELY FAULT LOCATIONS: M8939

CIRCUITS	PRINT REFERENCE
TM03 REGISTER SELECT	(MBI2)
FRAME COUNT REGISTER	(MBI8)
FRAME COUNT MULTIPLEXERS	(MBI10)

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: M8939, M8905-YB

CIRCUITS	PRINT REFERENCE
TM03 REGISTER SELECTION	(MBI2)
FUNCTION CODE FLOPS	(MBI5)
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),M8939,M8905-YB

CIRCUIT	PRINT REFERENCE
FCS	MBI8
SET ILF	MBI7
SET NEF	MBI7
GO BIT	MBI5
GO BIT MULTIPLEXER	MR6
SET ILR	MBI2

LOGIC TEST #12: DRIVE READY BIT

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

PRINT REF TCCM7

LOGIC TEST #13: INTERRUPT TEST

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

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

LIKELY FAULT LOCATION:

CIRCUITS	PRINT REFERENCE
INTERRUPT CONTROL	BCTF

MANUAL INTERVENTION TESTS 14,15,16,17

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

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

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

LIKELY FAULT LOCATION: M8928,SLAVE CABLE, M8933

CIRCUIT	PRINT REFERENCE
MOL	LAW6,TCCM7,M8908,M9001YA,YC
WRL	LAW8,TCCM7,M8908,M9001YA,YC
DPR	TCCM7
DRY	TCCM7
BOT	LAW6,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: M8928,M8933,M8939,SLAVE CABLE

CIRCUIT	PRINT REFERENCE
SSC	LAW8,M8913,M8913YA,TCCM7
ATA	MBI3

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

PURPOSE: TO TEST EOT,SSC,SLA

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

LIKELY FAULT LOCATION: M8928,SLAVE CABLE,M8933

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

LOGIC TEST #17: STATUS AT ONLINE LOADED

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

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

LOGIC TEST #20: ILLEGAL FUNCTION

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

LIKELY FAULT LOCATION: M8939

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

LOGIC TEST #21: REGISTER MODIFICATION REFUSED

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

LIKELY FAULT LOCATION: M8939

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

LOGIC TEST #22: CONTROL BUS PARITY (CPAR)

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

LIKELY FAULT LOCATIONS: M8939

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

LOGIC TEST #23: FORMAT ERROR (FMT)

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

LIKELY FAULT LOCATIONS: M8905-YB,M8906,M8939

CIRCUIT	PRINT REFERENCE
FORMAT BITS	MR6
ILF DECODE	BF3
ILF FLOP	MBI11
ILF MULTIPLEXERS	MBI10

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

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

LIKELY FAULT LOCATIONS: DBUS LINES, M8905-YB, M8906

CIRCUIT	PRINT REFERENCE
MM CLK	MR5
WRT CLK GENERATION	TCCM4
DPAR 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: M8939

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

LOGIC TEST #26: FRAME COUNT ERROR

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

LIKELY FAULT LOCATIONS: M8939, MB CABLE, M8933, M8905-YB

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

H 2

LOGIC TEST #27: ILLEGAL REGISTER

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

LIKELY FAULT LOCATIONS: MASSBUSS, M8939

CIRCUITS	PRINT REFERENCE
REGISTER SELECT LINES	MB1, MB2
REGISTER SELECT DECODE	MBI2
ILR FLOP	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 WRP 3 CODE IS LOADED INTO MR. WRITE COMMAND AND GO BIT ARE SET. AFTER DELAY FOR ACCELERATION, THE MR CLOCK IS GENERATED AND ANOTHER CHECK IS MADE FOR DTE. FINAL CHECK IS MADE FOR ERRORS OTHER THAN OPI. THE FIRST MAINTENANCE REGISTER CODE WHICH CRIPPLES THE OCCUPIED RECEIVER CAUSES OCCUPIED TO BE ASSERTED AND TESTS THE CIRCUITRY WHICH CHECKS FOR OCCUPIED WHEN A DATA TRANSFER COMMAND IS INITIATED. THE SECOND TEST UTILIZES THE FACT THAT THE WRP 3 CODE INHIBITS THE MASSBUS WCLK RECEIVER CREATING A SITUATION WHERE SCLK IS NOT FOLLOWED BY A WRITE CLOCK.

LIKELY FAULT LOCATIONS: M8939, M8905-YB, M8906, MB CABLES

CIRCUITS	PRINT REFERENCES
DTE FLOP	MBI11
CRIPPLE OCCUPIED FUNCTION	MR5
WRP 3 FUNCTION	MR5
PREVIOUS OCCUPIED CHECK	MBI7
CHECK FOR WCLK	BF2
MM CLK	MR5

LOGIC TEST 31: OPERATION INCOMPLETE (OPI)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8939

CIRCUITS	PRINT REFERENCES
OPI TIMER	TCCM5
OPI FLOP	MBI11
OPI TIMER CONTROL	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: M8939, M8928, SLAVE CABLE

CIRCUITS	PRINT REFERENCES
UNSAFE FLOP	MBI11
SET UNSAFE	MBI7
MOL GENERATION	LAW6

LOGIC TEST 33: POSITIONING IN PROGRESS (PIP)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8939, M8933

CIRCUITS	PRINT REFERENCES
SPACE FUNCTION DECODE	MBI5
PIP GENERATION	TCCM7
STATUS REGISTER	TCCM7

LOGIC TEST 34: PHASE-ENCODED STATUS (PES)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, SLAVE BUS, M8928, M8933

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

LOGIC TEST 35: TAPE CONTROL WRITE (TCW)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATION: M8905-YB

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: M8939, M8933

CIRCUITS	PRINT REFERENCES
FCS BIT	MBI8
FCS MULTIPLEXER	TCCM7

LOGIC TEST 37: ACCELERATION (ACCL)

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8933, M8928

CIRCUITS	PRINT REFERENCES
ACCL BIT, MOTION DELAY COUNTER	TCCM3
CLOCK	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 M8932.

LIKELY FAULT LOCATIONS: M8932, M8901, M8933, M8905-YB

CIRCUITS	PRINT REFERENCES
TAPE MARK DETECTOR	TCPE4, TCPE5
TAPE MARK MULTIPLEXER	TCCM7
ENVELOPE SIGNALS	DS 3, 5, 7
WRITE DATA BUFFER	TCCM2
RDA MULTIPLEXERS	TCCM6
WRITE TAPE MARK FUNCTION	MBI5
WAM0 SIGNAL	MR5

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

PROGRAMMED SEQUENCE:

SAME AS TEST 40 EXCEPT NRZ DENSITY IS SELECTED.

LIKELY FAULT LOCATIONS: M8933, M8934

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

THE NEXT 5 TESTS CONSISTS OF WRITING ON TAPE USING 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 WRAPO 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-YB, M8934, G056, SLAVE CABLE,
                         M8928

CIRCUITS	PRINT REFERENCES
MM FUNCTION DECODE	MR5
CRC CHECK CIRCUIT	CNRZ3

LOGIC TEST 43: LRC

PROGRAMMED SEQUENCE:

A WRITE OPERATION IS PERFORMED WITH A MM FUNCTION (INC TMRL)
WHICH ASSERTS WD(SB) 5L THROUGHOUT THE RECORD. ALL ONES
DATA IS USED SO THAT THE FUNCTION SHOULD NOT INTERFERE
WITH THE RECORD. THE LRC CHARACTER SHOULD NOT BE EFFECTED.

LIKELY FAULT LOCATIONS: M8505, M8933, M8928, M8934

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

LOGIC TEST 44: PE CORRECTABLE DATA

PROGRAMMED SEQUENCE:

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

LIKELY FAULT LOCATIONS: M8905-YB, M8901, M8932

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

LOGIC TEST 45: PE INCORRECTABLE DATA

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

LIKELY FAULT LOCATIONS: M8932, 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: M8932, M8933, M8905-YB

CIRCUITS	PRINT REFERENCES
PEF.	TCPE2
WRITE BUFFER	TCCM2
MM DECODE	MR5

LOGIC TEST 47: FRAME COUNT OVERFLOW

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

LIKELY FAULT LOCATION: M8939

FRAME COUNT REGISTER MBI8

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

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

PROGRAM SEQUENCE:

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

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

THIS TEST IS THE COMPLEMENT OF LOGIC TEST 50 ABOVE.

%

```
1171                                    .LIST   BIN,LOC,SEQ
1172                                    .TITLE TM03/TU45 CONTROL LOGIC TEST PART I
1173                                    ;CZTUOA0
```

```
                        ;25 MAY 78
                        ;J.G. ADAMS/R. J. COLLINS
                        ;REVISED JUN 1977 BY J. G. ADAMS ;CHANGED MODULE REFERENCES TO
                                                         ;TM03 MODULES
                        .MCALL  .$ACT11,.$EOP,$CATCH,$SAVE,$RESTORE,$CHAIN,$CHNMODE
                        .NLIST  MC
                        .LIST   ME
                        .ENABLE ABS,AMA


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

```
                        ;TU45 REGISTER BITS**********

                        ;15 ;14 ;13 ;12 ;11 ;10 ;09 ;08 ;07 ;06 ;05 ;04 ;03 ;02 ;01 ;00 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CS1:                   ;SC ;TRE;MCP;SPR;DNA;PSL;A17;A16;RDY;IE ;FUN;FUN;FUN;FUN;FUN;GO ;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;WC:                    ; WORD                                                          ;
                        ; COUNT                                                         ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;BA:                    ; BUS                                                           ;
                        ; ADDRESS                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;FC:                    ; FRAME                                                         ;
                        ; COUNT                                                         ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CS2:                   ;DLT;WCE;UPE;NED;NEM;PGE;MXF;MDP;OR ;IR ;CLR;PAT;BAI;U2 ;U1 ;U0 ;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;ERR;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DS:                    ;ATA;ERR;PIP;MOL;WRL;EOT;SPR;DPR;DRY;SSC;PES;SDN;IDB;TMK;BOT;SLA;
                        ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;ER:                    ;CDE;UNS;OPI;DTE;NEF;CS ;FCE;NSG;PEF;INC;DAT;FMT;CNT;RMR;ILR;ILF;
                        ;CRC;   ;   ;   ;ITM;   ;   ;LRC;VPE;BPE;   ;BPE;   ;   ;   ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;ATS:                   ; ATTENTION                                                     ;
                        ; SUMMARY                                                       ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;CC:                    ; CHECK                                                         ;
                        ; CHARACTER                                                     ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DB:                    ; DATA                                                          ;
                        ; BUFFER                                                        ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;MR:                    ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;DB ;WRT;MM ;OP ;OP ;OP ;OP ;MM ;
                        ; 7 ; 6 ; 5 ; 4 ; 3 ; 2 ; 1 ; 0 ; P ;CLK;CLK; 4 ; 3 ; 2 ; 1 ;GO ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;DT:                    ; DRIVE     ; 7 ;   ;SLV;                                     ;
                        ; TYPE      ;CH ;   ;PR ;                                     ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;SN:                    ; SERIAL                                                        ;
                        ; NUMBER                                                        ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;TC:                    ;ACL;FCS;SAC;ENA;SPR;DEN;DEN;DEN;FMT;FMT;FMT;FMT;EVN;SSN;SSN;SSN;
                        ;   ;   ;   ;DT ;   ; 4 ; 2 ; 1 ; 8 ; 4 ; 2 ; 1 ;PAR; 4 ; 2 ; 1 ;
                        ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

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

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


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

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

            000060                      .=60
000060      016664                      .WORD   TTINT   ;TTY INTERRUPT HEADER ADDRESS
000062      000340                      .WORD   340     ;PRIORITY LEVEL 7

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

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

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

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

            000224                      .=224
000224      016654                      MTINT           ;TAPE INTERRUPT HANDLER ADDRESS
000226      000340                      340
```

```
1292
1293            000510                          .=510
1294                                            ;MASS BUS REGISTER EQUIVS**********
1295
1296    000510  172440                 C1:     172440
1297    000512  172442                 WC:     172442
1298    000514  172444                 BA:     172444
1299    000516  172446                 FC:     172446
1300    000520  172450                 CS:     172450
1301    000522  172452                 DS:     172452
1302    000524  172454                 ER:     172454
1303    000526  172456                 AS:     172456
1304    000530  172460                 CC:     172460
1305    000532  172462                 DB:     172462
1306    000534  172464                 MR:     172464
1307    000536  172466                 DT:     172466
1308    000540  172470                 SN:     172470
1309    000542  172472                 TC:     172472
1310
1311                                            ;ILLEGAL FUNCTION CODES
1312
1313    000544  005405                 ILFT:   5405
1314    000546  007415                         7415
1315    000550  016423                         16423
1316    000552  020437                         20437
1317    000554  022443                         22443
1318    000556  025447                         25447
1319    000560  031455                         31455
1320    000562  033465                         33465
1321    000564  036473                         36473
1322
1323                                            ;CONSTANTS**********
1324
1325    000566  177776                 PSW:    177776          ;PROCESSOR STATUS
1326    000570  177570                 SWR:    177570          ;SWITCH REGISTER
1327    000572  177560                 TKS:    177560          ;TTY READER STATUS
1328    000574  177562                 TKB:    177562          ;TTY READ BUFFER
1329    000576  177564                 TPS:    177564          ;TTY PUNCH STATUS
1330    000600  177566                 TPB:    177566          ;TTY PUNCH BUFFER
1331    000602  177777                 SERNUM: 177777          ;SERIAL NUMBER
1332    000604  000011                 DRVTP:  011             ;DRIVE TYPE
1333    000606  000020                 ITAMT:  20              ;ITERATION AMOUNT
1334    000610  000224                 VECT:   224             ;INTERRUPT VECTOR(RH)
1335    000612  172440                 REGS:   172440          ;STARTING REGISTER ADDRESS
```

```
 1336                                   ;FLAGS AND COUNTERS**********
 1337
 1338  000614  000000                   TOB:    0
 1339  000616  000000                   TIB:    0
 1340  000620  000000                   HDRFL:  0
 1341  000622  000000                   EMADDR: 0
 1342  000624  000000                   DRVN:   0
 1343  000626  000000                   TR00:   0
 1344  000630  000000                   TR01:   0
 1345  000632  000000                   TR02:   0
 1346  000634  000000                   TR03:   0
 1347  000636  000000                   TR04:   0
 1348  000640  000000                   TR05:   0
 1349  000642  000000                   TR06:   0
 1350  000644  000000                   TR07:   0
 1351  000646  000000                   TR10:   0
 1352  000650  000000                   TR11:   0
 1353  000652  000000                   TR12:   0
 1354  000654  000000                   TR13:   0
 1355  000656  000000                   TR14:   0
 1356  000660  000000                   TR15:   0
 1357  000662  000000                   NRZOF:  0
 1358  000664  000000                   SLVN:   0
 1359  000666  000000                   PFLG:   0
 1360  000670  000000                   RTRN:   0
 1361  000672  000000                   ERADD:  0
 1362  000674  000000                   TEMP1:  0
 1363  000676  000000                   TEMP2:  0
 1364  000700  000000                   TEMP3:  0
 1365  000702  000000                   ITCNT:  0
 1366  000704  000000                   SAV1:   0
 1367  000706  000000                   SAV2:   0
 1368  000710  000000                   SAV3:   0
 1369  000712  000000                   SCOLP:  0
 1370  000714  000000                   ITRLP:  0
 1371  000716  000000                   EXFL:   0
 1372  000720  000000                   ATAF:   0
 1373  000722  000000                   SLAF:   0
 1374  000724  000000                   SSCF:   0
 1375  000726  000000                   ERRF:   0
 1376  000730  000000                   ASF:    0
 1377  000732  000000                   SCF:    0
 1378  000734  000000                   TREF:   0
 1379  000736  000000                   PEXFL:  0
 1380  000740  000000                   STFLG:  0
 1381  000742  000000                   LTADD:  0
 1382  000744  000000                   T24FL:  0
 1383  000746  000000                   ADDFL:  0
 1384  000750  000000                   WAM:    0
 1385  000752  000000                   FUN:    0
 1386  000754  000000                   DATC:   0
 1387  000756  000000                   WTAD:   0
 1388  000760  000000                   DATAD:  0
 1389  000762  000000                   RDAD:   0
 1390  000764  000000                   W2FLG:  0
 1391  000766  000000                   DERFL:  0
```

```
1392  000770  000000                  PREFL:  0
1393  000772  000000                  SERFL:  0
1394  000774  000000                  CRCNT:  0
1395  000776  000000                  UDES:   0
1396  001000  000000                  WPGFL:  0
1397  001002  000000                  PATRN:  0
1398  001004  000000                  STATF:  0
1399  001006  000000                  RDRVF:  0
1400  001010  000000                  RCDP:   0
1401  001012  000000                  STATC:  0
1402  001014  000000                  SKAT:   0
1403  001016  000000                  PCNTR:  0               ;PASS COUNTER
1404
1405                                  ;EXPT WRAP STATUS*******
1406
1407  001020  000000                  WCS1:   0
1408  001022  000000                  WCS2:   0
1409  001024  000000                  WDS:    0
1410  001026  000000                  WER:    0
1411
1412                                  ;CORE DUMP PATTERNS*******
1413
1414  001030  000005                  WCDP2:  5
1415  001032  000005                          5
1416  001034  000012                          12
1417  001036  000012                          12
1418  001040  000000                          0
1419  001042  000017                  WCDP0:  17
1420  001044  000017                          17
1421  001046  000017                          17
1422  001050  000017                          17
1423  001052  000000                          0
```

```
1424
1425                                    ;LOGIC TEST ENTRY TABLE**********
1426
1427  001054  000000                TSTTBL: 0
1428  001056  000000                        0
1429  001060  002552                        LT1
1430  001062  002552                        LT1
1431  001064  003026                        LT2
1432  001066  003026                        LT2
1433  001070  003232                        LT3
1434  001072  003234                        LT3IT
1435  001074  003414                        LT4
1436  001076  003414                        LT4
1437  001100  003714                        LT5
1438  001102  003722                        LT5IT
1439  001104  004104                        LT6
1440  001106  004106                        LT6IT
1441  001110  004220                        LT7
1442  001112  004222                        LT7IT
1443  001114  004334                        LT10
1444  001116  004336                        LT10IT
1445  001120  004460                        LT11
1446  001122  004462                        LT11IT
1447  001124  004704                        LT12
1448  001126  004706                        LT12IT
1449  001130  005100                        LT13
1450  001132  005110                        LT13IT
1451  001134  005202                        LT14
1452  001136  005242                        LT14IT
1453  001140  005316                        LT15
1454  001142  005356                        LT15IT
1455  001144  005432                        LT16
1456  001146  005472                        LT16IT
1457  001150  005546                        LT17
1458  001152  005606                        LT17IT
1459  001154  005662                        LT20
1460  001156  005676                        LT20IT
1461  001160  006024                        LT21
1462  001162  006040                        LT21IT
1463  001164  006170                        LT22
1464  001166  006204                        LT22IT
1465  001170  006314                        LT23
1466  001172  006330                        LT23IT
1467  001174  006444                        LT24
1468  001176  006460                        LT24IT
1469  001200  006770                        LT25
1470  001202  006776                        LT25IT
1471  001204  007122                        LT26
1472  001206  007130                        LT26IT
1473  001210  007336                        LT27
1474  001212  007362                        LT27IT
1475  001214  007454                        LT30
1476  001216  007476                        LT30IT
1477  001220  007772                        LT31
1478  001222  010000                        LT31IT
1479  001224  010604                        LT32
```

```
 1480  001226  010620                    LT32IT
 1481  001230  010746                    LT33
 1482  001232  010762                    LT33IT
 1483  001234  011050                    LT34
 1484  001236  011064                    LT34IT
 1485  001240  011204                    LT35
 1486  001242  011220                    LT35IT
 1487  001244  011356                    LT36
 1488  001246  011372                    LT36IT
 1489  001250  011476                    LT37
 1490  001252  011512                    LT37IT
 1491  001254  011646                    LT40
 1492  001256  011662                    LT40IT
 1493  001260  011766                    LT41
 1494  001262  012002                    LT41IT
 1495  001264  012230                    LT42
 1496  001266  012266                    LT42IT
 1497  001270  012560                    LT43
 1498  001272  012606                    LT43IT
 1499  001274  013010                    LT44
 1500  001276  013036                    LT44IT
 1501  001300  013256                    LT45
 1502  001302  013304                    LT45IT
 1503  001304  013522                    LT46
 1504  001306  013550                    LT46IT
 1505  001310  013760                    LT47
 1506  001312  013774                    LT47IT
 1507  001314  014150                    LT50
 1508  001316  014164                    LT50IT
 1509  001320  014324                    LT51
 1510  001322  014340                    LT51IT
 1511  001324  002436          TADX:     .WORD   TEND
 1512  001326  000051          TLAST:    .WORD   51              ;CONTAINS # OF TESTS
```

```
1513                                            .EVEN
1514                                            ;PROGRAM START AND HOUSEKEEPING**********
1515
1516                                    ;NOTE: PROGRAM STARTS HERE ON START AT 200
1517  001330  012706  000500          START:  MOV     #500,SP         ;SET STACK POINTER
1518  001334  013746  000004                  MOV     @#4,-(SP)       ;SAVE ERROR TRAP VECTOR
1519  001340  013746  000006                  MOV     @#6,-(SP)       ;AND VECTOR +2
1520  001344  012737  001370  000004          MOV     #1$,@#4         ;SET NEW VECTOR
1521  001352  005037  000006                  CLR     @#6             ;AND PSW
1522  001356  022777  177777  177204          CMP     #-1,@SWR        ;USE SOFTWARE SWITCH IF HARDWARE
1523  001364  001402                          BEQ     2$              ;IS = 177777
1524  001366  000404                          BR      3$              ;OTHERWISE USE HARDWARE SWR
1525  001370  022626                  1$:     CMP     (SP)+,(SP)+     ;RESET STACK PTR
1526  001372  012737  000176  000570  2$:     MOV     #SWREG,SWR      ;SET SOFTWARE SWITCH REGISTER
1527  001400  012637  000006          3$:     MOV     (SP)+,@#6       ;RESTORE ERROR TRAP VECTORS
1528  001404  012637  000004                  MOV     (SP)+,@#4
1529  001410  005037  001014                  CLR     SKAT            ;CLEAR SKIP ADDRESS TEST FLAG
1530  001414  005027                          CLR     (PC)+           ;;CLEAR CHAIN INDICATOR
1531  001416  000000                  CHNFLG: .WORD   0               ;;CHAIN MODE INDICATOR
1532                                                                  ;;1/0 = CHAIN/NOT CHAIN MODE
1533  001420  022737  002502  000042          CMP     #$ENDAD,@#42    ;;BRANCH IF LOADED VIA ACT11 CHAIN MODE
1534  001426  001404                          BEQ     50$
1535  001430  005737  000042                  TST     @#42            ;;BRANCH IF IN DUMP MODE
1536  001434  001413                          BEQ     52$
1537  001436  000406                          BR      51$
1538  001440  012737  000176  000570  50$:    MOV     #SWREG,SWR      ;;INVOKE SOFTWARE SWR
1539  001446  012777  100000  177114          MOV     #100000,@SWR    ;;WITH HALT ON ERROR SET
1540  001454  005237  001416          51$:    INC     CHNFLG          ;;SET CHNFLG = CHAIN MODE
1541  001460  000137  002106                  JMP     TSCD            ;;GO TO CHAIN ADDRESS
1542  001464                          52$:
1543  001464  122737  000006  000041  4$:     CMPB    #6,@#41         ;BRANCH IF NOT LOADED VIA TMDP
1544  001472  001004                          BNE     5$
1545  001474  012704  022636                  MOV     #MSG62,R4       ;ADVISE USER TO REMOVE TMDP FROM
1546  001500  004737  017334                  JSR     PC,TTOUT        ;UNIT UNDER TEST
1547  001504  012704  020252          5$:     MOV     #MSG1,R4
1548  001510  004737  017334                  JSR     PC,TTOUT        ;PRINT TITLE
1549  001514  112737  000043  020252          MOVB    #'#,MSG1        ;DO NOT PRINT TITLE ON RESTART
1550  001522  012704  022206                  MOV     #MSG44,R4
1551  001526  004737  017334                  JSR     PC,TTOUT        ;REQUEST REGISTER ADDRESS
1552  001532  013703  000612                  MOV     REGS,R3
1553  001536  004737  017462                  JSR     PC,OCTP         ;PRINT CURRENT ADDRESS
1554  001542  012705  000612                  MOV     #REGS,R5        ;SET ADDRESS SAVE LOC
1555  001546  012701  000007                  MOV     #7,R1           ;SET SIZE OF RESPONSE
1556  001552  012702  176400                  MOV     #176400,R2      ;SET UPPER LIMIT
1557  001556  012703  172300                  MOV     #172300,R3      ;SET LOWER LIMIT
1558  001562  004737  017012                  JSR     PC,TTR          ;GO GET RESPONSE
1559  001566  012704  02223C                  MOV     #MSG45,R4
1560  001572  004737  017334                  JSR     PC,TTOUT        ;REQUEST VECTOR
1561  001576  013703  000610                  MOV     VECT,R3
1562  001602  004737  017462                  JSR     PC,OCTP         ;PRINT CURRENT VECTOR
1563  001606  012705  000610                  MOV     #VECT,R5        ;SET ADDRESS SAVE LOC
1564  001612  012701  000004                  MOV     #4,R1           ;SET SIZE OF RESPONSE
1565  001616  012702  000224                  MOV     #224,R2         ;SET UPPER LIMIT
1566  001622  012703  000150                  MOV     #150,R3         ;SET LOWER LIMIT
1567  001626  004737  017012                  JSR     PC,TTR          ;GO GET RESPONSE
1568  001632  013700  000610                  MOV     VECT,R0         ;GET VECTOR
```

```
 1569  001636  012720  016654                MOV     #MTINT,(R0)+    ;LOAD INTERRUPT ADDRESS IN VECTOR
 1570  001642  012710  000340                MOV     #340,(R0)       ;LOAD PRIORITY
 1571  001646  013700  000612                MOV     REGS,R0         ;GET START OF REGS
 1572  001652  012701  000016                MOV     #16,R1          ;SET NUMBER OF REGS
 1573  001656  012702  000510                MOV     #C1,R2          ;GET START OF TABLE
 1574  001662  010022                ST0:    MOV     R0,(R2)+        ;BUILD TABLE
 1575  001664  062700  000002                ADD     #2,R0           ;BUMP ADDRESS
 1576  001670  005301                        DEC     R1              ;SEE IF DONE
 1577  001672  001373                        BNE     ST0             ;IF NOT: BR
 1578  001674  012702  000614                MOV     #TOB,R2
 1579  001700  012700  000077                MOV     #77,R0
 1580  001704  005022                ST1:    CLR     (R2)+           ;CLEAR FLAGS + COUNTERS
 1581  001706  005300                        DEC     R0
 1582  001710  001375                        BNE     ST1
 1583  001712  012704  022576                MOV     #MSG57,R4       ;REQUEST TM03 DRIVE #
 1584  001716  004737  017334                JSR     PC,TTOUT
 1585  001722  013703  000624                MOV     DRVN,R3         ;GET CURRENT DRIVE #
 1586  001726  004737  017462                JSR     PC,OCTP         ;PRINT IT
 1587  001732  012705  000624                MOV     #DRVN,R5        ;TTR ROUTINE RETURNS USER VALUE TO (R5)
 1588  001736  012701  000002                MOV     #2,R1           ;LIMIT RESPONSE
 1589  001742  012702  000007                MOV     #7,R2           ;LIMIT RANGE TO 0-7
 1590  001746  012703  000000                MOV     #0,R3
 1591  001752  004737  017012                JSR     PC,TTR          ;GET USER RESPONSE
 1592  001756  012704  022614                MOV     #MSG58,R4       ;REQUEST TU45 SLAVE #
 1593  001762  004737  017334                JSR     PC,TTOUT
 1594  001766  013703  000664                MOV     SLVN,R3         ;GET CURRENT SLAVE #
 1595  001772  004737  017462                JSR     PC,OCTP         ;AND PRINT IT
 1596  001776  012705  000664                MOV     #SLVN,R5        ;TTR ROUTINE RETURNS RESPONSE TO (R5)
 1597  002002  012701  000002                MOV     #2,R1           ;LIMIT RESONSE TO 1 CHARACTER
 1598  002006  012702  000007                MOV     #7,R2           ;BETWEEN 0 AND 7
 1599  002012  012703  000000                MOV     #0,R3
 1600  002016  004737  017012                JSR     PC,TTR          ;GET USER RESPONSE
 1601  002022  012704  022551                MOV     #MSG56,R4
 1602  002026  004737  017334                JSR     PC,TTOUT        ;REQUEST STATIC ONLY
 1603  002032  013703  001012                MOV     STATC,R3        ;GET CURRENT VALUE
 1604  002036  004737  017462                JSR     PC,OCTP         ;AND TYPE IT
 1605  002042  012705  001012                MOV     #STATC,R5       ;SET ADDRESS OF STATIC FLAG
 1606  002046  012701  000002                MOV     #2,R1           ;SET SIZE OF RESPONSE
 1607  002052  012702  000001                MOV     #1,R2           ;SET UPPER LIMIT
 1608  002056  012703  000000                MOV     #0,R3           ;SET LOWER LIMIT
 1609  002062  004737  017012                JSR     PC,TTR          ;GET RESPONSE
 1610
 1611                          ;START 210
 1612  002066  012706  000500        ST2:    MOV     #500,SP         ;SET STACK PTR
 1613  002072  005037  001006                CLR     RDRVF           ;CLEAR REVERSE FLAG
 1614  002076  005037  001016                CLR     PCNTR           ;CLEAR PASS COUNTER
 1615  002102  004737  020110                JSR     PC,GTSWR        ;GET SWITCHES
```

```
 1616
 1617                                        ;TEST SCHEDULAR**********
 1618
 1619  002106  052777  000100  176456  TSCD:   BIS     #100,@TKS       ;SET KEYBOARD IE BIT
 1620  002114  005037  001000                  CLR     WPGFL           ;CLEAR WRAP PATRN FLAG
 1621  002120  005037  000740                  CLR     STFLG           ;CLEAR SINGLE TEST FLAG
 1622  002124  017700  176440                  MOV     @SWR,R0
 1623  002130  042700  177700                  BIC     #177700,R0      ;BRANCH IF SINGLE
 1624  002134  001122                          BNE     STSCD           ;TEST SELECTED
 1625  002136  005737  001416                  TST     CHNFLG          ;;BRANCH IF NOT IN CHAIN MODE
 1626  002142  001457                          BEQ     TSCDA
 1627  002144  012737  177777  000624          MOV     #-1,DRVN        ;;INITIALIZE DRIVE #
 1628  002152  012737  177777  000664  NXTDRV: MOV     #-1,SLVN        ;;INITIALIZE SLAVE #
 1629  002160  012777  000040  176332  1$:     MOV     #40,@CS         ;;INIT CONTROLLER
 1630  002166  005237  000624                  INC     DRVN            ;;STEP DRIVE #
 1631  002172  022737  000010  000624          CMP     #10,DRVN        ;;EXIT IF ALL DRIVES TESTED
 1632  002200  001521                          BEQ     $DONE           ;;FOR AVAILABILITY
 1633  002202  013777  000624  176310          MOV     DRVN,@CS        ;;LOAD DRIVE #
 1634  002210  005777  176274                  TST     @C1             ;;ACCESS DRIVE
 1635  002214  032777  010000  176276          BIT     #10000,@CS      ;;BRANCH IF DRIVE NON EXISTANT
 1636  002222  001356                          BNE     1$              ;;(NED = 1)
 1637  002224  005237  000664          NXTSLV: INC     SLVN            ;;STEP SLAVE # AND BRANCH
 1638  002230  001011                          BNE     1$              ;;IF NOT SLAVE 0
 1639  002232  005737  000624                  TST     DRVN            ;;BRANCH IF NOT DRIVE # 0
 1640  002236  001006                          BNE     1$
 1641  002240  122737  000006  000041          CMPB    #6,@#41         ;;BRANCH IF NOT TMDP
 1642  002246  001002                          BNE     1$
 1643  002250  005237  000664                  INC     SLVN            ;;STEP TO SLAVE # 1
 1644  002254  022737  000010  000664  1$:     CMP     #10,SLVN        ;;BRANCH IF ALL SLAVES TESTED
 1645  002262  001733                          BEQ     NXTDRV          ;;FOR AVAILABILITY
 1646  002264  013777  000664  176250          MOV     SLVN,@TC        ;;LOAD SLAVE UNIT #
 1647  002272  032777  002000  176236          BIT     #2000,@DT       ;;BRANCH IF SLAVE NOT
 1648  002300  001751                          BEQ     NXTSLV          ;;PRESENT (SPR = 0)
 1649  002302  012737  001054  000742  TSCDA:  MOV     #TSTTBL,LTADD
 1650  002310  062737  000004  000742  TSCD0:  ADD     #4,LTADD
 1651  002316  013737  000742  000714  TSCD1:  MOV     LTADD,ITRLP
 1652  002324  062737  000002  000714          ADD     #2,ITRLP        ;SET ITERATION ADDRESS
 1653  002332  005037  000620                  CLR     HDRFL           ;CLEAR PRINT HEADER FLAG
 1654  002336  017700  176400                  MOV     @LTADD,R0       ;SET POINTER TO TEST
*1655  002342  000110                          JMP     (R0)            ;GO TO TEST
 1656  002344  032777  002000  176216  TSCD2:  BIT     #2000,@SWR      ;SEE IF HALT ON TEST
 1657  002352  001403                          BEQ     TSCD3           ;IF NOT: BR
 1658  002354  000000                          HALT
 1659  002356  005037  001000                  CLR     WPGFL           ;CLEAR WRAP DATA GENERATOR FLAG
 1660  002362  005737  000740          TSCD3:  TST     STFLG           ;SE IF SINGLE TEST
 1661  002366  001750                          BEQ     TSCD0           ;IF NOT: BR
 1662  002370  017700  176174                  MOV     @SWR,R0
 1663  002374  042700  177700                  BIC     #177700,R0      ;BRANCH IF ALL TESTS DESIRED
 1664  002400  001642                          BEQ     TSCD            ;IF SO: BR
 1665  002402  012737  000001  000740  STSCD:  MOV     #1,STFLG        ;SET SINGLE TEST FLAG
 1666  002410  023700  001326                  CMP     TLAST,R0        ;SEE IF EXCEEDED TESTS
 1667  002414  002410                          BLT     TEND            ;IF SO: BR
 1668  002416  006300                          ASL     R0
 1669  002420  006100                          ROL     R0              ;SET TABLE MODIFIER
 1670  002422  012737  001054  000742          MOV     #TSTTBL,LTADD
 1671  002430  060037  000742                  ADD     R0,LTADD        ;SET TEST POINTER
```

```
 1672  002434  000730                        BR      TSCD1
 1673  002436  005737  001416          TEND: TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
 1674  002442  001270                        BNE     NXTSLV          ;STEP TO NEXT SLAVE
 1675  002444  012704  022046          $DONE: MOV    #MSG41,R4
 1676  002450  004737  017334                JSR     PC,TTOUT        ;PRINT END OF PASS
 1677  002454  013703  001016                MOV     PCNTR,R3
 1678  002460  004737  017462                JSR     PC,OCTP         ;PRINT PASS NUMBER
 1679  002464  005000                        CLR     R0              ;DELAY WAITING FOR
 1680  002466  005300                1$:     DEC     R0
 1681  002470  001376                        BNE     1$
 1682  002472  013700  000042                MOV     @#42,R0         ;GET ACT11 RETURN ADDRESS
 1683  002476  001405                        BEQ     HERE            ;BRANCH IF NOT ACT11
 1684  002500  000005                        RESET
 1685  002502  004710                $ENDAD: JSR     PC,(R0)
 1686  002504  000240                        NOP
 1687  002506  000240                        NOP
 1688  002510  000240                        NOP
 1689  002512  000240                HERE:   NOP
 1690  002514  005737  001416                TST     CHNFLG          ;BRANCH IF IN CHAIN MODE
 1691  002520  001005                        BNE     TENDX
 1692  002522  032777  010000  176040        BIT     #10000,@SWR     ;SEE IF HALT ON PASS
 1693  002530  001001                        BNE     TENDX           ;IF NOT: BR
 1694  002532  000000                        HALT
 1695  002534  012737  000001  001014  TENDX: MOV    #1,SKAT         ;SET SKIP ADDRESS TEST FLAG
 1696  002542  005237  001016                INC     PCNTR           ;BUMP PASS COUNTER
 1697  002546  000137  002106                JMP     TSCD            ;RESTART
```

```
1698                                            ;LOGIC TEST 1: DRIVE ADDRESSING**********
1699
1700  002552  013737  000624  000700  LT1:      MOV     DRVN,TEMP3      ;GET DRIVE # TO BE TESTED
1701  002560  013701  000624                    MOV     DRVN,R1
1702  002564  005737  001014                    TST     SKAT            ;SEE IF SKIP ADDRESS TESTS
1703  002570  001403                            BEQ     1$              ;IF NOT: BR
1704  002572  005737  000740                    TST     STFLG           ;SEE IF SINGLE TEST
1705  002576  001511                            BEQ     LT1X            ;IF NOT: BR
1706  002600  032777  001000  175762  1$:       BIT     #1000,@SWR      ;BRANCH IF MAN INTERVENTION
1707  002606  001433                            BEQ     LT1A            ;NOT SELECTED
1708  002610  012704  020511          LT1G0:    MOV     #MSG2A,R4
1709  002614  004737  016606                    JSR     PC,INST         ;PRINT TEST INSTRUCTIONS
1710  002620  012737  022732  000622  LT1G:     MOV     #MSLT1,EMADDR   ;SET HEADER ADDRESS
1711  002626  012704  020450                    MOV     #MSG2,R4
1712  002632  004737  017334                    JSR     PC,TTOUT        ;REQUEST DRIVE NUMBER
1713  002636  012705  000700                    MOV     #TEMP3,R5       ;TTR ROUTINE RETURNS RESPONSE TO (R5)
1714  002642  012701  000002                    MOV     #2,R1
1715  002646  012702  000007                    MOV     #7,R2
1716  002652  012703  000000                    MOV     #0,R3
1717  002656  004737  017012                    JSR     PC,TTR          ;GET DRIVE NUMBER
1718  002662  005737  000674                    TST     TEMP1           ;SEE IF ANOTHER DRIVE
1719  002666  001455                            BEQ     LT1X            ;IF NOT: BR
1720  002670  005001                            CLR     R1              ;SELECT DRIVE 0
1721  002672  012700  000010                    MOV     #10,R0          ;SET NUMBER OF DRIVES
1722  002676  012777  000040  175614  LT1A:     MOV     #40,@CS         ;INIT
1723  002704  010177  175610                    MOV     R1,@CS          ;SELECT DRIVE
1724  002710  005777  175574                    TST     @C1             ;ACCESS DRIVE
1725  002714  032777  010000  175576            BIT     #10000,@CS      ;SEE IF NED
1726  002722  001010                            BNE     LT1B            ;IF SO: BR
1727  002724  032777  001000  175636            BIT     #1000,@SWR      ;BRANCH IF NOT MANUAL INTERVENTION
1728  002732  001433                            BEQ     LT1X
1729  002734  023701  000700                    CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1730  002740  001404                            BEQ     LT1C            ;IF NOT: BR
1731  002742  000407                            BR      LT1ER           ;ELSE GO TO ERROR
1732  002744  023701  000700          LT1B:     CMP     TEMP3,R1        ;SEE IF SHOULD BE NED
1733  002750  001410                            BEQ     LT1ER1
1734  002752  005300                  LT1C:     DEC     R0
1735  002754  001721                            BEQ     LT1G            ;IF DONE ALL: BR
1736  002756  005201                            INC     R1              ;SELECT NEXT DRIVE
1737  002760  000746                            BR      LT1A            ;CONTINUE
1738  002762  012737  000001  000716  LT1ER:     MOV     #1,EXFL         ;FLAG EXPT
1739  002770  000403                            BR      LT1ER2
1740  002772  012737  000002  000716  LT1ER1:    MOV     #2,EXFL         ;FLAG NOT EXPT
1741  003000  012737  020640  000672  LT1ER2:    MOV     #MSG3,ERADD     ;FLAG CONDITION
1742  003006  012737  002676  000712            MOV     #LT1A,SCOLP     ;SET SCOPE ADDRESS
1743  003014  004737  014704                    JSR     PC,LTGER        ;GO PRINT LOGIC TEST ERROR
1744  003020  000754                            BR      LT1C            ;CONTINUE TEST
1745  003022  000137  002344          LT1X:     JMP     TSCD2           ;RETURN TO SCHED
1746
```

```
1747                                        ;LOGIC TEST 2: REGISTER ADDRESSING**********
1748
1749  003026  000240                LT2:    NOP
1750  003030  012777  000040  175462 LT2IT:  MOV     #40,@CS         ;INIT
1751  003036  013777  000624  175454        MOV     DRVN,@CS        ;SELECT DRIVE
1752  003044  012737  023006  000622        MOV     #MSLT2,EMADDR   ;SAVE LT2 HEADER ADDRESS
1753  003052  012705  000510               MOV     #C1,R5          ;SET ADDRESS OF FIRST REGISTER
1754  003056  012700  000016               MOV     #16,R0          ;SET NUMBER OF REGISTERS
1755  003062  012702  000626               MOV     #TR00,R2        ;SET START OF REGISTER BUFFER
1756  003066  011501                LT2A:   MOV     (R5),R1
1757  003070  011112                        MOV     (R1),(R2)       ;READ REGISTER
1758  003072  032777  020000  175410        BIT     #20000,@C1      ;SEE IF ERROR
1759  003100  001402                        BEQ     LT2B            ;IF NOT: BR
1760  003102  004737  003132               JSR     PC,LT2ER1       ;ELSE GO TO ERROR 1
1761  003106  032777  000002  175410 LT2B:  BIT     #2,@ER          ;SEE IF ILR
1762  003114  001402                        BEQ     LT2C            ;IF NOT: BR
1763  003116  004737  003150               JSR     PC,LT2ER2       ;ELSE GO TO ERROR 2
1764  003122  022225                LT2C:   CMP     (R2)+,(R5)+     ;BUMP ADDRESS
1765  003124  005300                        DEC     R0
1766  003126  001357                        BNE     LT2A            ;CONTINUE FOR ALL REGISTERS
1767  003130  000434                        BR      LT2X
1768  003132  012737  000002  000716 LT2ER1: MOV     #2,EXFL         ;FLAG NOT EXPECTED
1769  003140  012737  020662  000672        MOV     #MSG4,ERADD     ;POINT TO CONTROLLER ERROR
1770  003146  000415                        BR      LT2ERG          ;GO TO ERROR
1771  003150  012737  000002  000716 LT2ER2: MOV     #2,EXFL         ;FLAG NOT EXPECTED
1772  003156  012737  020700  000672        MOV     #MSG5,ERADD     ;POINT TO DRIVE ERROR
1773  003164  000406                        BR      LT2ERG          ;GO TO ERROR
1774  003166  012737  000001  000716 LT2ER3: MOV     #1,EXFL         ;FLAG EXPECTED
1775  003174  012737  020662  000672        MOV     #MSG4,ERADD     ;POINT TO DRIVE
1776  003202  012737  003216  000712 LT2ERG: MOV     #LT2LP,SCOLP    ;SET SCOPE ADDRESS
1777  003210  004737  014704               JSR     PC,LTGER        ;GO PRINT
1778  003214  000207                        RTS     PC              ;ELSE CONTINUE
1779  003216  005726                LT2LP:  TST     (SP)+           ;RESET STACK
1780  003220  000722                        BR      LT2A            ;LOOP
1781  003222  004737  016254        LT2X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
1782  003226  000137  002344                JMP     TSCD2           ;RETURN TO SCHED
```

```
1783                                        ;LOGIC TEST 3: CONTROL BUS**********
1784
1785  003232  000240                LT3:    NOP
1786  003234  012737  023065  000622 LT3IT:  MOV     #MSLT3,EMADDR   ;SET TEST HEADER
1787  003242  012701  000001                MOV     #1,R1           ;PRESET PATTERN 1
1788  003246  012700  000020                MOV     #20,R0          ;SET PATTERN CHANGE NUMBER
1789  003252  004737  016344        LT3A:   JSR     PC,INIT1        ;GO INIT
1790  003256  010177  175234                MOV     R1,@FC          ;WRITE TO FC
1791  003262  032777  000010  175234        BIT     #10,@ER         ;SEE IF CPAR (TM03)
1792  003270  001012                        BNE     LT3ER1          ;IF SO: BR
1793  003272  017702  175220        LT3B:   MOV     @FC,R2          ;READ FC
1794  003276  032777  020000  175204        BIT     #20000,@C1      ;SEE IF MCPE (RH)
1795  003304  001017                        BNE     LT3ER2          ;IF SO: BR
1796  003306  005300                LT3C:   DEC     R0              ;SEE IF DONE PATTERN CHANGES
1797  003310  001426                        BEQ     LT3X            ;IF SO: BR
1798  003312  006301                        ASL     R1              ;CHANGE PATTERN
1799  003314  000756                        BR      LT3A            ;CONTINUE
1800  003316  012737  021213  000672 LT3ER1: MOV     #MSG11,ERADD    ;SET ERROR CODE
1801  003324  012737  003252  000712        MOV     #LT3A,SCOLP     ;SET SCOPE ADDRESS
1802  003332  017702  175160                MOV     @FC,R2          ;GET DATA
1803  003336  004737  016012                JSR     PC,LTGER1       ;GO DO ERROR
1804  003342  000753                        BR      LT3B
1805  003344  012737  021167  000672 LT3ER2: MOV     #MSG10,ERADD    ;SET ERROR CODE
1806  003352  012737  003272  000712        MOV     #LT3B,SCOLP     ;SET SCOPE ADDRESS
1807  003360  004737  016012                JSR     PC,LTGER1       ;GO DO ERROR
1808  003364  000750                        BR      LT3C
1809  003366  105701                LT3X:   TSTB    R1              ;SEE IF DONE PATTERN 2
1810  003370  100405                        BMI     LT3XX           ;IF SO: BR
1811  003372  012701  000401                MOV     #401,R1         ;SET PATTERN 2
1812  003376  012700  000010                MOV     #10,R0          ;SET PATTERN CHANGE NUMBER
1813  003402  000723                        BR      LT3A            ;DO PATTERN 2
1814  003404  004737  016254        LT3XX:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1815  003410  000137  002344                JMP     TSCD2           ;RETURN TO SCHEDULAR
```

```
1816
1817                                        ;LOGIC TEST 4: SLAVE ADDRESSING**********
1818
1819  003414  013737  000664  000700  LT4:    MOV     SLVN,TEMP3
1820  003422  013701  000664                  MOV     SLVN,R1
1821  003426  005737  001014                  TST     SKAT            ;SEE IF SKIP ADDRESS TESTS
1822  003432  001403                          BEQ     1$              ;IF NOT: BR
1823  003434  005737  000740                  TST     STFLG           ;SEE IF SINGLE TEST
1824  003440  001523                          BEQ     LT4X            ;IF NOT: BR
1825  003442  032777  001000  175120  1$:     BIT     #1000,@SWR      ;BRANCH IF MAN INTERVETION
1826  003450  001430                          BEQ     LT4A            ;NOT SELECTED
1827  003452  012704  021016          LT4G0:  MOV     #MSG8A,R4
1828  003456  004737  016606                  JSR     PC,INST         ;PRINT TEST INSTRUCTIONS
1829  003462  012704  020755          LT4G:   MOV     #MSG8,R4
1830  003466  004737  017334                  JSR     PC,TTOUT        ;REQUEST SLAVE
1831  003472  012705  000700                  MOV     #TEMP3,R5
1832  003476  012701  000002                  MOV     #2,R1
1833  003502  012702  000007                  MOV     #7,R2
1834  003506  012703  000000                  MOV     #0,R3
1835  003512  004737  017012                  JSR     PC,TTR          ;GET SLAVE NUMBER
1836  003516  005737  000674                  TST     TEMP1           ;SEE IF SLAVE
1837  003522  001472                          BEQ     LT4X            ;IF NOT: BR
1838  003524  005001                          CLR     R1              ;SELECT SLAVE 0
1839  003526  012700  000010                  MOV     #10,R0          ;SET NUMBER OF SLAVES
1840  003532  012777  000040  174760  LT4A:   MOV     #40,@CS         ;INIT
1841  003540  013777  000624  174752          MOV     DRVN,@CS        ;SELECT DRIVE
1842  003546  010177  174770                  MOV     R1,@TC          ;SELECT SLAVE
1843  003552  017703  174760                  MOV     @DT,R3          ;GET DT
1844  003556  020137  000700                  CMP     R1,TEMP3        ;SEE IF SHOULD HAVE SPR
1845  003562  001404                          BEQ     LT4B            ;IF SO: BR
1846  003564  032703  002000                  BIT     #2000,R3        ;SEE IF SPR
1847  003570  001420                          BEQ     LT4D            ;IF NOT: BR
1848  003572  000423                          BR      LT4ER1          ;GO TO ERROR 1
1849  003574  032703  002000          LT4B:   BIT     #2000,R3        ;SEE IF NO SLAVE PRESENT
1850  003600  001424                          BEQ     LT4ER2          ;(SPR=0)
1851  003602  012704  021670          LT4C:   MOV     #MSG30,R4
1852  003606  004737  017334                  JSR     PC,TTOUT        ;PRINT SERIAL NUMBER TAG
1853  003612  017703  174722                  MOV     @SN,R3
1854  003616  004737  020006                  JSR     PC,SNPT         ;PRINT SERIAL NUMBER
1855  003622  032777  001000  174740          BIT     #1000,@SWR      ;BRANCH IF NOT MANUAL INTERVENTION
1856  003630  001427                          BEQ     LT4X
1857  003632  005300                  LT4D:   DEC     R0
1858  003634  001712                          BEQ     LT4G            ;IF DONE ALL: BR
1859  003636  005201                          INC     R1              ;BUMP SLAVE
1860  003640  000734                          BR      LT4A            ;CONTINUE
1861  003642  012737  000001  000716  LT4ER1: MOV     #1,EXFL         ;FLAG EXPT: NOT RECEIVED
1862  003650  000403                          BR      LT4ERG
1863  003652  012737  000002  000716  LT4ER2: MOV     #2,EXFL         ;FLAG RECVD: NOT EXPT
1864  003660  012737  023152  000622  LT4ERG: MOV     #MSLT4,EMADDR   ;SET  LT4 HEADER
1865  003666  012737  021145  000672          MOV     #MSG9,ERADD     ;SET ERROR CONDITION
1866  003674  012737  003532  000712          MOV     #LT4A,SCOLP     ;SET SCOPE ADDRESS
1867  003702  004737  014704                  JSR     PC,LTGER        ;GO TO ERROR
1868  003706  000751                          BR      LT4D            ;IF NO SCOPE: BR
1869  003710  000137  002344          LT4X:   JMP     TSCD2           ;RETURN TO SCHED
1870
```

```
1871                                            ;LOGIC TEST 5: MAINTENANCE REGISTER BIT TEST**********
1872
1873  003714  012737  023234  000622  LT5:    MOV     #MSLT5,EMADDR   ;SET TEST HEADER
1874  003722  004737  016344          LT5IT:  JSR     PC,INIT1        ;GO INIT
1875  003726  012700  000032                  MOV     #32,R0          ;SET LOOP FOR BITS 4-0
1876  003732  005001                          CLR     R1              ;SET TEST WORD
1877  003734  010177  174574          LT5A:   MOV     R1,@MR          ;SEND TEST WORD TO MR
1878  003740  017702  174570                  MOV     @MR,R2          ;READ MR
1879  003744  042702  177740                  BIC     #177740,R2      ;MASK BITS 4-0
1880  003750  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
1881  003752  001026                          BNE     LT5ER1
1882  003754  005300          LT5B:   DEC     R0
1883  003756  001402                          BEQ     LT5C            ;IF DONE LOOP: BR
1884  003760  005201                          INC     R1              ;BUMP TEST WORD
1885  003762  000764                          BR      LT5A            ;CONTINUE LOOP
1886  003764  012701  000015          LT5C:   MOV     #15,R1          ;SET TEST WORD + WAM 3
1887  003770  012700  001000                  MOV     #1000,R0        ;SET LOOP FOR BITS 15-7
1888  003774  010177  174534          LT5D:   MOV     R1,@MR          ;LOAD MR
1889  004000  017702  174530                  MOV     @MR,R2          ;READ MR
1890  004004  042702  000140                  BIC     #140,R2         ;MASK OUT BITS 5,6
1891  004010  020102                          CMP     R1,R2           ;SEE IF EXPT = RECVD
1892  004012  001401                          BEQ     LT5E            ;IF SO: BR
1893  004014  000416                          BR      LT5ER2          ;ELSE GO TO ERR 2
1894  004016  005300          LT5E:   DEC     R0
1895  004020  001425                          BEQ     LT5X            ;IF DONE LOOP: BR
1896  004022  062701  000200                  ADD     #200,R1         ;BUMP TEST WORD
1897  004026  000762                          BR      LT5D            ;CONTINUE LOOP
1898  004030  012737  021256  000672  LT5ER1: MOV     #MSG14,ERADD    ;SET ERROR CODE
1899  004036  012737  003734  000712          MOV     #LT5A,SCOLP     ;SET SCOPE ADDRESS
1900  004044  004737  016012                  JSR     PC,LTGER1       ;GO TO ERROR
1901  004050  000741                          BR      LT5B            ;CONTINUE
1902  004052  012737  021273  000672  LT5ER2: MOV     #MSG15,ERADD    ;SET ERROR CODE
1903  004060  012737  003774  000712          MOV     #LT5D,SCOLP     ;SET SCOPE ADDRESS
1904  004066  004737  016012                  JSR     PC,LTGER1       ;GO TO ERROR
1905  004072  000751                          BR      LT5E            ;CONTINUE
1906  004074  004737  016254          LT5X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
1907  004100  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
1908
```

```
1909                                        ;LOGIC TEST 6: TC REGISTER BIT TEST**********
1910
1911  004104  000240                LT6:    NOP
1912  004106  012737  023303  000622 LT6IT:  MOV     #MSLT6,EMADDR   ;POINT TO LT6 HEADER
1913  004114  012700  000003                MOV     #3,R0           ;SET NUMBER OF TESTS
1914  004120  005001                LT6A1:  CLR     R1
1915  004122  004737  016344        LT6A:   JSR     PC,INIT1        ;GO INIT
1916  004126  010177  174410        LT6B:   MOV     R1,@TC          ;WRITE TC
1917  004132  017702  174404                MOV     @TC,R2          ;READ TC
1918  004136  042702  160000                BIC     #160000,R2      ;MASK OUT SAC
1919  004142  020102                        CMP     R1,R2           ;SEE IF EXPT = RECVD
1920  004144  001010                        BNE     LT6ER1          ;IF NOT: BR
1921  004146  005300                LT6D:   DEC     R0
1922  004150  001417                        BEQ     LT6X            ;IF DONE ALL: BR
1923  004152  022700  000001                CMP     #1,R0           ;SEE IF RESET TEST
1924  004156  001760                        BEQ     LT6A1           ;IF SO: BR
1925  004160  012701  017777                MOV     #17777,R1       ;SET TEST WORD
1926  004164  000756                        BR      LT6A            ;DO SET TEST
1927  004166  012737  021321  000672 LT6ER1: MOV     #MSG18,ERADD    ;SET ERROR CODE
1928  004174  012737  004126  000712        MOV     #LT6B,SCOLP     ;SET SCOPE ADDRES
1929  004202  004737  016012                JSR     PC,LTGER1       ;GO TO ERROR
1930  004206  000757                        BR      LT6D            ;CONTINUE
1931  004210  004737  016254        LT6X:   JSR     PC,ITER         ;GO SEE IF ITERATIONS
1932  004214  000137  002344                JMP     TSCD2           ;RETURN TO SCHED
1933
```

```
1934                                        ;LOGIC TEST 7: FRAME COUNT BIT TEST**********
1935
1936  004220  000240                LT7:    NOP
1937  004222  012700  000003        LT7IT:  MOV     #3,R0           ;SET TEST NUMBER
1938  004226  012737  023352  000622 LT7C:   MOV     #MSLT7,EMADDR   ;SET TEST HEADER
1939  004234  005001                        CLR     R1              ;SET TEST WORD
1940  004236  004737  016344        LT7A:   JSR     PC,INIT1        ;GO INIT
1941  004242  010177  174250                MOV     R1,@FC          ;CLEAR FRAME COUNT
1942  004246  017702  174244                MOV     @FC,R2          ;READ FC
1943  004252  020102                        CMP     R1,R2           ;SEE IF EXPT = RECVD
1944  004254  001010                        BNE     LT7ER1
1945  004256  005300                LT7B:   DEC     R0              ;SEE IF DONE ALL
1946  004260  001417                        BEQ     LT7X            ;IF SO: BR
1947  004262  022700  000001                CMP     #1,R0           ;SEE IF RESET TEST
1948  004266  001757                        BEQ     LT7C            ;IF SO: BR
1949  004270  012701  177777                MOV     #-1,R1          ;SET TEST WORD TO -1
1950  004274  000760                        BR      LT7A            ;CONTINUE
1951  004276  012737  021340  000672 LT7ER1: MOV     #MSG19,ERADD    ;SET ERROR CODE
1952  004304  012737  004236  000712        MOV     #LT7A,SCOLP     ;SET SCOPE ADDRESS
1953  004312  004737  016012                JSR     PC,LTGER1       ;GO PRINT ERROR
1954  004316  000757                        BR      LT7B            ;ELSE CONTINUE
1955  004320  012700  000003        LT7X:   MOV     #3,R0           ;RESET TEST AMT
1956  004324  004737  016254                JSR     PC,ITER         ;GO SEE IF ITERATIONS
1957  004330  000137  002344                JMP     TSCD2           ;RETURN TO SCHED
1958
```

```
1959                                        ;LOGIC TEST 10: FUNCTION CODE BIT TEST**********
1960
1961  004334  000240                LT10:   NOP
1962  004336  012737  023421  000622 LT10IT: MOV     #MSLT10,EMADDR  ;SET TEST HEADER
1963  004344  012700  000003                MOV     #3,R0           ;SET NUMBER OF TESTS
1964  004350  005001                LT10A1: CLR     R1              ;SET TEST WORRD
1965  004352  012777  000040  174140 LT10A:  MOV     #40,@CS         ;INIT
1966  004360  013777  000624  174132        MOV     DRVN,@CS        ;SELECT DRIVE
1967  004366  010177  174116                MOV     R1,@C1          ;WRITE C1
1968  004372  017702  174112                MOV     @C1,R2          ;READ C1
1969  004376  042702  177701                BIC     #177701,R2      ;MASK FUNCTION CODE
1970  004402  020102                        CMP     R1,R2           ;SEE IF EXPT = RECVD
1971  004404  001010                        BNE     LT10E1
1972  004406  005300                LT10B:  DEC     R0
1973  004410  001417                        BEQ     LT10X           ;IF DONE ALL: BR
1974  004412  022700  000001                CMP     #1,R0           ;SEE IF RESET TEST
1975  004416  001754                        BEQ     LT10A1          ;IF SO: BR
1976  004420  012701  000076                MOV     #76,R1          ;SET TEST WORD
1977  004424  000752                        BR      LT10A           ;DO SET TEST
1978  004426  012737  021357  000672 LT10E1: MOV     #MSG20,ERADD    ;SET ERROR CODE
1979  004434  012737  004352  000712        MOV     #LT10A,SCOLP    ;SET SCOPE ADDRESS
1980  004442  004737  016012                JSR     PC,LTGER1       ;GO PRINT ERROR
1981  004446  000757                        BR      LT10B           ;ELSE CONTINUE
1982  004450  004737  016254        LT10X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
1983  004454  000137  002344                JMP     TSCD2           ;RETURN TO SCHED
```

```
 1984
 1985                                   ;LOGIC TEST 11: GO BIT SET RESET**********
 1986
 1987  004460  000240                LT11:   NOP
 1988  004462  012737  023477  000622 LT11IT: MOV    #MSLT11,EMADDR  ;SET TEST HEADER
 1989  004470  004737  016344               JSR     PC,INIT1        ;GO INIT
 1990  004474  017702  174010               MOV     @C1,R2          ;READ C1
 1991  004500  032702  000001               BIT     #1,R2           ;SEE IF GO=0
 1992  004504  001030                       BNE     LT11E1
 1993  004506  012777  000015  174020 LT11B:  MOV    #15,@MR         ;SELECT WAM 3
 1994  004514  005077  173776               CLR     @FC             ;ASSURE FCS = 1
 1995  004520  052777  001700  174014       BIS     #1700,@TC       ;ASSURE FMT OK
 1996  004526  012777  000071  173754       MOV     #71,@C1         ;SET READ+GO
 1997  004534  017702  173750               MOV     @C1,R2          ;READ C1
 1998  004540  032702  000001               BIT     #1,R2           ;SEE IF GO =1
 1999  004544  001424                       BEQ     LT11E2
 2000  004546  004737  016344        LT11C:  JSR     PC,INIT1        ;GO INIT
 2001  004552  017702  173732               MOV     @C1,R2          ;READ C1
 2002  004556  032702  000001               BIT     #1,R2           ;SEE IF GO=0
 2003  004562  001444                       BEQ     LT11X           ;IF SO:BR
 2004  004564  000430                       BR      LT11E3          ;ELSE GO TO ERROR 3
 2005  004566  012737  021411  000672 LT11E1: MOV    #MSG21,ERADD    ;SET ERROR CODE
 2006  004574  012702  000001               MOV     #1,R2           ;SET REVD
 2007  004600  005001                       CLR     R1              ;SET EXPT
 2008  004602  012737  004462  000712       MOV     #LT11IT,SCOLP           ;SET SCOPE ADDRESS
 2009  004610  004737  016012               JSR     PC,LTGER1       ;GO PRINT ERROR
 2010  004614  000734                       BR      LT11B           ;ELSE CONTINUE
 2011  004616  012737  021447  000672 LT11E2: MOV    #MSG22,ERADD    ;SET ERROR CODE
 2012  004624  005002                       CLR     R2              ;SET RCVD
 2013  004626  012701  000001               MOV     #1,R1           ;SET EXPT
 2014  004632  012737  004506  000712       MOV     #LT11B,SCOLP    ;SET SCOPE ADDRESS
 2015  004640  004737  016012               JSR     PC,LTGER1       ;GO PRINT ERROR
 2016  004644  000740                       BR      LT11C           ;ELSE CONTINUE
 2017  004646  012737  021470  000672 LT11E3: MOV    #MSG23,ERADD    ;SET ERROR CODE
 2018  004654  005001                       CLR     R1              ;SET EXPT
 2019  004656  012702  000001               MOV     #1,R2           ;SET RCVD
 2020  004662  012737  004546  000712       MOV     #LT11C,SCOLP    ;SET SCOPE ADDRESS
 2021  004670  004737  016012               JSR     PC,LTGER1       ;GO PRINT ERROR
 2022  004674  004737  016254        LT11X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
 2023  004700  000137  002344               JMP     TSCD2           ;RETURN TO SCHED
```

```
2024
2025                                    ;LOGIC TEST 12: DRIVE READY BIT**********
2026
2027  004704  000240                    LT12:   NOP
2028  004706  012737  023544  000622    LT12IT: MOV     #MSLT12,EMADDR  ;SET TEST HEADER
2029  004714  004737  016344            JSR     PC,INIT1        ;GO INIT
2030  004720  032777  000200  173574    BIT     #200,@DS        ;SEE IF DRY=1
2031  004726  001426                    BEQ     LT12E1
2032  004730  012777  000015  173576    LT12B:  MOV     #15,@MR         ;SET WAM3
2033  004736  005077  173554            CLR     @FC             ;ASSURE FCS = 1
2034  004742  052777  001700  173572    BIS     #1700,@TC       ;ASSURE FMT OK
2035  004750  012777  000071  173532    MOV     #71,@C1         ;SET READ+GO
2036  004756  032777  000200  173536    BIT     #200,@DS        ;SEE IF DRY=0
2037  004764  001020                    BNE     LT12E2
2038  004766  004737  016344            LT12C:  JSR     PC,INIT1        ;GO INIT
2039  004772  032777  000200  173522    BIT     #200,@DS        ;SEE IF DRY=1
2040  005000  001033                    BNE     LT12X           ;IF SO: BR
2041  005002  000422                    BR      LT12E3          ;ELSE GO TO ERROR 3
2042  005004  012737  021523  000672    LT12E1: MOV     #MSG24,ERADD    ;SET ERROR CODE
2043  005012  012737  004706  000712    MOV     #LT12IT,SCOLP   ;SET SCOPE ADDRESS
2044  005020  004737  016004            JSR     PC,LTGER2       ;GO TO ERROR
2045  005024  000741                    BR      LT12B           ;CONTINUE
2046  005026  012737  021551  000672    LT12E2: MOV     #MSG25,ERADD    ;SET ERROR CODE
2047  005034  012737  004730  000712    MOV     #LT12B,SCOLP    ;SET LOOP ADDRESS
2048  005042  004737  016004            JSR     PC,LTGER2       ;GO PRINT ERROR
2049  005046  000747                    BR      LT12C           ;CONTINUE
2050  005050  012737  021600  000672    LT12E3: MOV     #MSG25A,ERADD   ;SET ERROR CODE
2051  005056  012737  004766  000712    MOV     #LT12C,SCOLP    ;SET ERROR LOOP
2052  005064  004737  016004            JSR     PC,LTGER2       ;GET PRINT ERROR
2053  005070  004737  016254    LT12X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
2054  005074  000137  002344            JMP     TSCD2           ;RETURN TO SCHED
```

```
2055
2056                                    ;LOGIC TEST 13: INTERRUPT TEST**********
2057
2058  005100  005000                LT13:   CLR     R0
2059  005102  012737  023615  000622        MOV     #MSLT13,EMADDR  ;SET TEST HEADER
2060  005110  004737  016344        LT13IT: JSR     PC,INIT1        ;GO INIT,SELECT DRIVE, SELECT ABOVE
2061  005114  012737  005172  000670        MOV     #LT13X,RTRN     ;SET RETURN ADDRESS
2062  005122  005077  173362                CLR     @C1             ;CLEAR CS1
2063  005126  005077  173434                CLR     @PSW            ;SET PRIORITY
2064  005132  052777  000100  173350        BIS     #100,@C1        ;BIT SET IE
2065  005140  005300                LT13A:  DEC     R0
2066  005142  001376                        BNE     LT13A           ;AWAIT INTERRUPT
2067  005144  012777  000340  173414  LT13E1: MOV    #340,@PSW       ;RESET PRIORITY
2068  005152  012737  021625  000672        MOV     #MSG26,ERADD    ;SET ERROR CODE
2069  005160  012737  005110  000712        MOV     #LT13IT,SCOLP   ;SET LOOP ADDRESS
2070  005166  004737  016004                JSR     PC,LTGER2       ;GO PRINT ERROR
2071  005172  004737  016254        LT13X:  JSR     PC,ITER         ;GO TO ITERATION SUBROUTINE
2072  005176  000137  002344                JMP     TSCD2           ;RETURN TO SCHED
```

```
2073
2074                                    ;THE NEXT 4 TESTS ARE MANUAL INTERVENTION STATUS TESTS.
2075                                    ;THE OPERATOR WILL BE REQUIRED TO MANIPULATE THE TU45
2076                                    ;CONTROL PANEL IN ACCORDANCE WITH TTY INSTRUCTIONS.
2077
2078                                    ;LOGIC TEST 14: STATUS AT BOT ON LINE, LOADED, NO WRITE RING**********
2079
2080  005202  032777  001000  173360  LT14:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2081  005210  001005                          BNE     LT14A           ;IF NOT: BR
2082  005212  005737  000740                  TST     STFLG           ;SEE IF SINGLE TEST
2083  005216  001435                          BEQ     LT14XX          ;IF NOT: BR
2084  005220  000137  016324                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2085  005224  012737  023662  000622  LT14A:  MOV     #MSLT14,EMADDR  ;SET TEST HEADER
2086  005232  012704  025750                  MOV     #MMSG1,R4       ;SET INSTRUCTION ONE
2087  005236  004737  016606                  JSR     PC,INST         ;GO DO INSTRUCTION
2088  005242  004737  016344          LT14IT: JSR     PC,INIT1        ;INIT, SELECT DRIVE + SLAVE
2089  005246  012701  014602                  MOV     #14602,R1       ;SET TEST WORD
2090  005252  017702  173244                  MOV     @DS,R2          ;ASSURE MOL,WRL,DPR,DRY,BOT
2091  005256  042702  000040                  BIC     #40,R2          ;CLEAR PE, IF UP
2092  005262  020102                          CMP     R1,R2
2093  005264  001410                          BEQ     LT14X           ;IF SO: BR
2094  005266  012737  005242  000712          MOV     #LT14IT,SCOLP   ;SET LOOP ADDRESS
2095  005274  012737  021654  000672          MOV     #MSG27,ERADD    ;SET ERROR CODE
2096  005302  004737  016012                  JSR     PC,LTGER1       ;GO PRINT ERROR
2097  005306  004737  016254          LT14X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2098  005312  000137  002344          LT14XX: JMP     TSCD2           ;RETURN TO SCHED
2099
```

```
2100
2101                                        ;LOGIC TEST 15: STATUS AT BOT, OFFLINE, LOADED, NO WRITE RING**********
2102
2103  005316  032777  001000  173244  LT15:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2104  005324  001005                          BNE     LT15A           ;IF NOT: BR
2105  005326  005737  000740                  TST     STFLG           ;SEE IF SINGLE TEST
2106  005332  001435                          BEQ     LT15XX          ;IF NOT: BR
2107  005334  000137  016324                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2108  005340  012737  023730  000622  LT15A:  MOV     #MSLT15,EMADDR  ;SET TEST HEADER
2109  005346  012704  026046                  MOV     #MMSG2,R4
2110  005352  004737  016606                  JSR     PC,INST         ;PRINT INSTRUCTION
2111  005356  004737  016354          LT15IT: JSR     PC,INIT2        ;GO INIT, SELECT DRIVE, SLAV
2112  005362  012701  100700                  MOV     #100700,R1      ;SET TEST WORD
2113  005366  017702  173130                  MOV     @DS,R2          ;READ STATUS
2114  005372  042702  000040                  BIC     #40,R2          ;CLEAR PE, IF UP
2115  005376  020102                          CMP     R1,R2           ;SEE OF EXPT=RCVD
2116  005400  001410                          BEQ     LT15X
2117  005402  012737  005356  000712          MOV     #LT15IT,SCOLP   ;SET LOOP ADDRESS
2118  005410  012737  021654  000672          MOV     #MSG27,ERADD    ;SET ERROR CODE
2119  005416  004737  016012                  JSR     PC,LTGER1       ;GO PRINT ERROR
2120  005422  004737  016254          LT15X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2121  005426  000137  002344          LT15XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2122
2123                                        ;LOGIC TEST 16: STATUS AT EOT, OFFLINE LOADED, NO WRITE RING**********
2124
2125  005432  032777  001000  173130  LT16:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2126  005440  001005                          BNE     LT16A           ;IF NOT: BR
2127  005442  005737  000740                  TST     STFLG           ;SEE IF SINGLE TEST
2128  005446  001435                          BEQ     LT16XX          ;IF NOT: BR
2129  005450  000137  016324                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2130  005454  012737  023776  000622  LT16A:  MOV     #MSLT16,EMADDR  ;SET TEST HEADER
2131  005462  012704  026067                  MOV     #MMSG3,R4
2132  005466  004737  016606                  JSR     PC,INST         ;GO PRINT INSTRUCTION
2133  005472  004737  016354          LT16IT: JSR     PC,INIT2        ;SELECT DRIVE,SLAVE
2134  005476  012701  116701                  MOV     #116701,R1      ;SET TEST WORD
2135  005502  017702  173014                  MOV     @DS,R2          ;READ STATUS
2136  005506  042702  000040                  BIC     #40,R2          ;CLEAR PE, IF UP
2137  005512  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2138  005514  001410                          BEQ     LT16X           ;IF SO: BR
2139  005516  012737  005472  000712          MOV     #LT16IT,SCOLP   ;SET LOOP ADDRESS
2140  005524  012737  021654  000672          MOV     #MSG27,ERADD    ;SET ERROR CODE
2141  005532  004737  016012                  JSR     PC,LTGER1       ;GO PRINT ERROR
2142  005536  004737  016254          LT16X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2143  005542  000137  002344          LT16XX: JMP     TSCD2           ;RETURN TO SCHED
2144
```

```
2145
2146                                        ;LOGIC TEST 17: STATUS AT ON LINE, LOADED**********
2147
2148  005546  032777  001000  173014  LT17:   BIT     #1000,@SWR      ;SEE IF INHIB MAN TST
2149  005554  001005                          BNE     LT17A           ;IF NOT: BR
2150  005556  005737  000740                  TST     STFLG           ;SEE IF SINGLE TEST
2151  005562  001435                          BEQ     LT17XX          ;IF NOT: BR
2152  005564  000137  016324                  JMP     INMT            ;ELSE GO PRINT INHIB MSG
2153  005570  012737  024044  000622  LT17A:  MOV     #MSLT17,EMADDR  ;SET TEST HEADER
2154  005576  012704  026174                  MOV     #MMSG4,R4
2155  005602  004737  016606                  JSR     PC,INST         ;GO PRINT INSTRUCTION
2156  005606  004737  016354          LT17IT: JSR     PC,INIT2        ;SELECT DRIVE, SLAVE
2157  005612  012701  110701                  MOV     #110701,R1      ;SET TEST WORD
2158  005616  017702  172700                  MOV     @DS,R2          ;READ STATUS
2159  005622  042702  000040                  BIC     #40,R2          ;CLEAR PE, IF UP
2160  005626  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2161  005630  001410                          BEQ     LT17X           ;IF SO: BR
2162  005632  012737  005606  000712          MOV     #LT17IT,SCOLP   ;SET LOOP ADDRESS
2163  005640  012737  021654  000672          MOV     #MSG27,ERADD    ;SET ERROR CODE
2164  005646  004737  016012                  JSR     PC,LTGER1       ;YES PRINT ERROR
2165  005652  004737  016254          LT17X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2166  005656  000137  002344          LT17XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
 2167                                        ;THE FOLLOWING 11 TESTS WILL TEST ALL POSSIBLE ERROR BITS
 2168                                        ;BY FORCING THEIR CONDITIONS THROUGH VARIOUS ILLEGAL PROGRAMMING
 2169                                        ;SEQUENCES AND USING THE MAINTENANCE WILL MODES AVAILABLE WITH TM03
 2170                                        ;FOR EACH ERROR CONDITION SET THE APPROPRIATE STATUS WILL BE
 2171                                        ;CHECKED. IE: ERR, ATA, SLA, SC ETC.
 2172
 2173                                        ;LOGIC TEST 20: ILLEGAL FUNCTION (ILF)**********
 2174
 2175  005662  012737  024112  000622  LT20:   MOV     #MSLT20,EMADDR  ;SET TEST HEADER
 2176  005670  012737  005710  000712          MOV     #LT20A,SCOLP    ;SET LOOP ADDRESS
 2177  005676  012700  000022          LT20IT: MOV     #22,R0          ;SET NUMBER OF ILL CODES
 2178  005702  012737  000544  000674          MOV     #ILFT,TEMP1     ;POINT TO START IF TABLE
 2179  005710  004737  016344          LT20A:  JSR     PC,INIT1        ;GO INIT, SELECT SLAVE + DRIVE
 2180  005714  012777  177777  172570          MOV     #-1,@WC         ;SET WC= -1
 2181  005722  012701  000001                  MOV     #1,R1           ;SET TEST WORD
 2182  005726  117777  172742  172554          MOVB    @TEMP1,@C1      ;SET ILL CODE
 2183  005734  017702  172564                  MOV     @ER,R2          ;READ ER
 2184  005740  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
 2185  005742  001011                          BNE     LT20B           ;IF SO: BR
 2186  005744  012737  026611  000672          MOV     #TMS17,ERADD    ;SET ERROR CODE
 2187  005752  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
 2188  005760  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
 2189  005764  000404                          BR      LT20C
 2190  005766  020102                  LT20B:  CMP     R1,R2           ;SEE UNEXPECTED ERRORS
 2191  005770  001402                          BEQ     LT20C           ;IF NOT: BR
 2192  005772  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
 2193  005776  005300                  LT20C:  DEC     R0              ;SEE IF DONE ALL ILL CODES
 2194  006000  001403                          BEQ     LT20X           ;IF SO: BR
 2195  006002  005237  000674                  INC     TEMP1           ;BUMP ADDRESS
 2196  006006  000740                          BR      LT20A           ;CONTINUE
 2197  006010  004737  016254  LT20X:  JSR     PC,ITER         ;GO SEE IF ITERATION
 2198  006014  004737  015344          JSR     PC,DRVCLR
 2199  006020  000137  002344          JMP     TSCD2           ;RETURN TO SCHED
```

```
2200
2201                                        ;LOGIC TEST 21: REGISTER MODIFICATION REFUSED(RMR)**********
2202
2203  006024  012737  024171  000622  LT21:   MOV     #MSLT21,EMADDR  ;SET TEST HEADER
2204  006032  012737  006040  000712          MOV     #LT21IT,SCOLP   ;SET SCOPE LOOP ADDRESS
2205  006040  004737  016344          LT21IT: JSR     PC,INIT1        ;GO INIT, SELECT SLAVE, DRIVE
2206  006044  052777  000300  172470          BIS     #300,@TC        ;SET FORMAT
2207  006052  012777  000015  172454          MOV     #15,@MR         ;SET WAM3
2208  006060  012777  000071  172422          MOV     #71,@C1         ;SET READ+GO
2209  006066  005077  172424                  CLR     @FC             ;ATTEMPT WRITE TO FC
2210  006072  012701  000004                  MOV     #4,R1           ;SET TEST WORD
2211  006076  017702  172422                  MOV     @ER,R2          ;GET ER
2212  006102  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2213  006104  001011                          BNE     LT21A           ;IF SO: BR
2214  006106  012737  026625  000672          MOV     #TMS19,ERADD    ;SET ERROR CODE
2215  006114  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2216  006122  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2217  006126  000404                          BR      LT21B
2218  006130  020102                  LT21A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2219  006132  001402                          BEQ     LT21B           ;IF NOT: BR
2220  006134  004737  014664                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2221  006140  004737  016254          LT21B:  JSR     PC,ITER         ;GO SEE IF ITERATION
2222  006144  012703  040000                  MOV     #40000,R3
2223  006150  005303                  LT21XA: DEC     R3              ;DELAY FOR ALPHA
2224  006152  001376                          BNE     LT21XA
2225  006154  004737  014530                  JSR     PC,EORPA        ;GO DO EOR CLEAR
2226  006160  004737  015344                  JSR     PC,DRVCLR
2227  006164  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
                                                ;LOGIC TEST 22: CONTROL BUS PARITY (CPAR)**********
                                                
006170  012737  024225  000622  LT22:   MOV     #MSLT22,EMADDR  ;SET TEST HEADER
006176  012737  006204  000712          MOV     #LT22IT,SCOLP   ;SET SCOPE LOOP ADDRESS
006204  004737  016344          LT22IT: JSR     PC,INIT1        ;INIT, SELECT SLAVE+DRIVE
006210  052777  000020  172302          BIS     #20,@CS         ;ENABLE EVEN PARITY ON MB
006216  012777  177777  172272          MOV     #-1,@FC         ;WRITE TO FC
006224  012701  000010                  MOV     #10,R1          ;SET TEST WORD
006230  042777  000020  172262          BIC     #20,@CS         ;RESET PARITY TO ODD
006236  017702  172262                  MOV     @ER,R2          ;GET ER
006242  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
006244  001011                          BNE     LT22A           ;IF SO: BR
006246  012737  026633  000672          MOV     #TMS20,ERADD    ;SET ERROR CODE
006254  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
006262  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
006266  000404                          BR      LT22X
006270  020102          LT22A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
006272  001402                          BEQ     LT22X           ;IF NOT: BR
006274  004737  014664                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
006300  004737  016254          LT22X:  JSR     PC,ITER         ;GO SEE IF ITERATION
006304  004737  015344                  JSR     PC,DRVCLR
006310  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2251
2252                                        ;LOGIC TEST 23: FORMAT ERROR(FMT)**********
2253
2254  006314  012737  024262  000622  LT23:   MOV     #MSLT23,EMADDR  ;SET TEST HEADER
2255  006322  012737  006330  000712          MOV     #LT23IT,SCOLP   ;SET SCOPE ADDRESS
2256  006330  004737  016344          LT23IT: JSR     PC,INIT1        ;GO INIT SELECT DRIVE+SLAVE
2257  006334  042777  000360  172200          BIC     #360,@TC        ;SET ILLEGAL FORMAT
2258  006342  012701  000020                  MOV     #20,R1          ;SET TEST WORD
2259  006346  012777  000015  172160          MOV     #15,@MR         ;SET WAM 3
2260  006354  012777  000071  172126          MOV     #71,@C1         ;SET READ+GO
2261  006362  017702  172136                  MOV     @ER,R2          ;READ ER
2262  006366  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2263  006370  001011                          BNE     LT23A           ;IF SO: BR
2264  006372  012737  026642  000672          MOV     #TMS21,ERADD    ;SET ERROR CODE
2265  006400  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2266  006406  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2267  006412  000404                          BR      LT23X
2268  006414  020102                  LT23A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2269  006416  001402                          BEQ     LT23X           ;IF NOT: BR
2270  006420  004737  014664                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2271  006424  004737  016254          LT23X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2272  006430  004737  014530                  JSR     PC,EORPA
2273  006434  004737  015344                  JSR     PC,DRVCLR
2274  006440  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2275                                        ;LOGIC TEST 24: DATA BUS PARITY ERROR(DPAR)**********
2276
2277  006444  012737  024327  000622  LT24:   MOV     #MSLT24,EMADDR  ;SET TEST HEADER
2278  006452  012737  006460  000712          MOV     #LT24IT,SCOLP   ;SET SCOPE ADDRESS
2279  006460  012737  000005  000606  LT24IT: MOV     #5,ITAMT
2280  006466  004737  016372                  JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2281  006472  052777  000300  172042          BIS     #300,@TC        ;SET NORMAL FORMAT
2282  006500  012777  027142  172006          MOV     #WDATA,@BA      ;SET BA
2283  006506  012777  177760  172002          MOV     #-20,@FC        ;SET FC
2284  006514  012777  177770  171770          MOV     #-10,@WC        ;SET WC
2285  006522  012777  000013  172004          MOV     #13,@MR         ;SELECT WAM 2
2286  006530  012777  000061  171752          MOV     #61,@C1         ;SET WRITE+GO
2287  006536  052777  000020  171754          BIS     #20,@CS         ;FORCE EVEN PARITY
2288  006544  012701  000040                  MOV     #40,R1          ;SET TEST WORD
2289  006550  012703  000004                  MOV     #4,R3
2290  006554  005000                          CLR     R0
2291  006556  005300                  1$:     DEC     R0
2292  006560  001376                          BNE     1$              ;DELAY
2293  006562  005303                          DEC     R3
2294  006564  001374                          BNE     1$
2295  006566  012700  000004                  MOV     #4,R0
2296  006572  012777  000013  171734  LT24B:  MOV     #13,@MR         ;CLOCK MR 4 TIMES
2297  006600  005300                          DEC     R0
2298  006602  022700  000002                  CMP     #2,R0           ;SEE IF DONE 1 BYTE
2299  006606  001002                          BNE     LT24B0          ;IF NOT: BR
2300  006610  017701  171720                  MOV     @MR,R1          ;ELSE GET BYTE 1
2301  006614  005700                  LT24B0: TST     R0              ;SEE IF BYTE 2
2302  006616  001365                          BNE     LT24B           ;IF NOT: BR
2303  006620  017704  171710                  MOV     @MR,R4          ;GET BYTE 2
2304  006624  005000                          CLR     R0
2305  006626  005300                  LT24C:  DEC     R0
2306  006630  001376                          BNE     LT24C           ;DELAY
2307  006632  032777  000040  171664          BIT     #40,@ER         ;SEE IF DPAR IS SET
2308  006640  001023                          BNE     LT24D           ;IF SO: BR
2309  006642  000301                          SWAB    R1
2310  006644  042701  177400                  BIC     #177400,R1      ;GET LOW BYTE
2311  006650  042704  000377                  BIC     #377,R4
2312  006654  050401                          BIS     R4,R1           ;GET HIGH BYTE
2313  006656  005237  000744                  INC     T24FL           ;SET T24 FLAG
2314  006662  012737  026650  000672          MOV     #TMS22,ERADD    ;SET ERROR CODE
2315  006670  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2316  006676  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2317  006702  005037  000744                  CLR     T24FL           ;CLEAR FLAG
2318  006706  000412                          BR      LT24X
2319  006710  012701  000050          LT24D:  MOV     #50,R1
2320  006714  017702  171604                  MOV     @ER,R2          ;GET ERROR REGISTER
2321  006720  042702  020000                  BIC     #20000,R2       ;MASK OPI
2322  006724  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2323  006726  001402                          BEQ     LT24X           ;IF NOT: BR
2324  006730  004737  014664                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2325  006734  042777  000020  171556  LT24X:  BIC     #20,@CS         ;RESET EVEN PARITY
2326  006742  004737  014530                  JSR     PC,EORPA        ;GO DO EOR CLEAR
2327  006746  004737  015344                  JSR     PC,DRVCLR       ;GO SEE IF DRIVE CLEAR OK
2328  006752  004737  016254                  JSR     PC,ITER         ;GO SEE IF ITERATION
2329  006756  012737  000020  000606          MOV     #20,ITAMT
2330  006764  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2331
2332                                        ;LOGIC TEST 25: NON-EXECUTABLE FUNCTION(NEF)**********
2333
2334  006770  012737  024367  000622  LT25:   MOV     #MSLT25,EMADDR  ;SET TEST HEADER
2335  006776  004737  016372          LT25IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2336  007002  052777  000300  171532          BIS     #300,@TC        ;SET NORMAL FORMAT
2337  007010  012777  177777  171500          MOV     #-1,@FC         ;SET ITLLEGAL FC
2338  007016  012777  000013  171510          MOV     #13,@MR         ;SET WAM 2
2339  007024  012777  000061  171456          MOV     #61,@C1         ;LOAD WRITE+GO
2340  007032  012701  004000                  MOV     #4000,R1        ;SET TEST WORD
2341  007036  017702  171462                  MOV     @ER,R2          ;GET ER
2342  007042  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2343  007044  001014                          BNE     LT25A           ;IF SO: BR
2344  007046  012737  006776  000712          MOV     #LT25IT,SCOLP   ;SET LOOP ADDRESS
2345  007054  012737  026736  000672          MOV     #TMS31,ERADD    ;SET ERROR CODE
2346  007062  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
2347  007070  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2348  007074  000404                          BR      LT25X
2349  007076  020102                  LT25A:  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2350  007100  001402                          BEQ     LT25X           ;IF NOT: BR
2351  007102  004737  014664                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2352  007106  004737  016254          LT25X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2353  007112  004737  015344                  JSR     PC,DRVCLR
2354  007116  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2355
2356                                        ;LOGIC TEST 26: FRAME COUNT ERROR(FCE)**********
2357
2358  007122  012737  024423  000622  LT26:   MOV     #MSLT26,EMADDR  ;SET TEST HEADER
2359  007130  004737  016372          LT26IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2360  007134  005000                          CLR     R0
2361  007136  005300                  1$:     DEC     R0
2362  007140  001376                          BNE     1$              ;AWAIT OPI RESET
2363  007142  052777  000300  171372          BIS     #300,@TC        ;SET NORMAL FORMAT
2364  007150  012777  177770  171334          MOV     #-10,@WC        ;SET WC=-10
2365  007156  012777  177760  171332          MOV     #-20,@FC        ;SET FC=-20
2366  007164  012777  000013  171342          MOV     #13,@MR         ;SET WAM 3
2367  007172  012777  000061  171310          MOV     #61,@C1         ;LOAD WRITE+GO
2368  007200  012701  001000                  MOV     #1000,R1        ;SET TEST WORD
2369  007204  005000                          CLR     R0
2370  007206  005300                  2$:     DEC     R0
2371  007210  001376                          BNE     2$              ;DELAY
2372  007212  012777  000025  171314          MOV     #25,@MR         ;LOAD MM EOR CLEAR
2373  007220  105077  171310                  CLRB    @MR             ;RESET MR
2374  007224  012703  000004                  MOV     #4,R3
2375  007230  005000                          CLR     R0
2376  007232  032777  001000  171264  3$:     BIT     #1000,@ER       ;SEE IF FCE SET
2377  007240  001022                          BNE     4$              ;IF SO: BR
2378  007242  005300                          DEC     R0
2379  007244  001372                          BNE     3$              ;DELAY
2380  007246  005303                          DEC     R3
2381  007250  001370                          BNE     3$
2382  007252  017702  171246                  MOV     @ER,R2          ;GET ER
2383  007256  012737  007130  000712          MOV     #LT26IT,SCOLP   ;SET SCOPE ADDRESS
2384  007264  012737  026715  000672          MOV     #TMS28,ERADD
2385  007272  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2386  007300  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2387  007304  000406                          BR      LT26X
2388  007306  017702  171212          4$:     MOV     @ER,R2          ;GET ERROR REGISTER
2389  007312  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2390  007314  001402                          BEQ     LT26X           ;IF NOT: BR
2391  007316  004737  014664                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2392  007322  004737  016254          LT26X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2393  007326  004737  015344                  JSR     PC,DRVCLR
2394  007332  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2395
2396                                        ;LOGIC TEST 27: ILLEGAL REGISTER(ILR)**********
2397
2398  007336  022737  172400  000510  LT27:   CMP     #172400,C1      ;SEE IF ADDRESSES OPEN
2399  007344  001041                          BNE     LT27XX          ;IF NOT: BR
2400  007346  012737  007372  000712          MOV     #LT27A,SCOLP    ;SET SCOPE ADDRESS
2401  007354  012737  024457  000622          MOV     #MSLT27,EMADDR  ;SET TEST HEADER
2402  007362  012700  000020          LT27IT: MOV     #20,R0          ;SET NUMBER OF ILR TESTS
2403  007366  012701  172434                  MOV     #172434,R1      ;SET FIRST ILR ADDRESS
2404  007372  004737  016372          LT27A:  JSR     PC,INIT3        ;GO INIT, SELECT DRIVE+SLAVE
2405  007376  011103                          MOV     (R1),R3         ;ATTEMPT ILR READ
2406  007400  032777  000002  171116          BIT     #2,@ER          ;SEE IF ILR=1
2407  007406  001010                          BNE     LT27B           ;IF SO: BR
2408  007410  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT-NOT RCVD FLAG
2409  007416  012737  026537  000672          MOV     #TMS10,ERADD    ;SET ERROR CODE
2410  007424  004737  014704                  JSR     PC,LTGER        ;GO PRINT ERROR
2411  007430  005300                  LT27B:  DEC     R0              ;SEE IF DONE ALL
2412  007432  001402                          BEQ     LT27X           ;IF SO: BR
2413  007434  005721                          TST     (R1)+           ;BUMP ADDRESS
2414  007436  000755                          BR      LT27A           ;CONTINUE TESTS
2415  007440  004737  016254          LT27X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2416  007444  004737  015344                  JSR     PC,DRVCLR
2417  007450  000137  002344          LT27XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
2418
2419                                        ;LOGIC TEST 30: DRIVE TIMING ERROR**********
2420
2421  007454  012737  026744  000672  LT30:   MOV     #TMS32,ERADD    ;SET ERROR CODE
2422  007462  012737  024513  000622          MOV     #MSLT30,EMADDR  ;SET TEST HEADER
2423  007470  012737  007476  000712          MOV     #LT30IT,SCOLP   ;SET SCOPE ADDRESS
2424  007476  004737  016372          LT30IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE + SLAVE
2425  007502  052777  000300  171032          BIS     #300,@TC        ;SET NORMAL FORMAT
2426  007510  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2427  007514  012777  000017  171012          MOV     #17,@MR         ;CRIPPLE OCCUPIED
2428  007522  005077  170770                  CLR     @FC             ;SET FC3
2429  007526  012777  000061  170754          MOV     #61,@C1         ;LOAD WRITE+GO
2430  007534  032777  010000  170762          BIT     #10000,@ER      ;SEE IF DTE SET
2431  007542  001005                          BNE     LT30A           ;IF SO: BR
2432  007544  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2433  007552  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2434  007556  004737  016372          LT30A:  JSR     PC,INIT3        ;GO INIT SELECT DRIVE,SLAVE
2435  007562  052777  000300  170752          BIS     #300,@TC        ;SET FORMAT
2436  007570  012701  010000                  MOV     #10000,R1       ;SET TEST WORD
2437  007574  005077  170716                  CLR     @FC             ;SET FCS
2438  007600  012777  000015  170726          MOV     #15,@MR         ;SET WRAP 3
2439  007606  012777  000061  170674          MOV     #61,@C1         ;LOAD WRITE+GO
2440  007614  012704  040000                  MOV     #40000,R4
2441  007620  005777  170716          LT30B:  TST     @TC             ;SEE IF ALPHA
2442  007624  100015                          BPL     LT30C           ;AWAIT ALPHA
2443  007626  005300                          DEC     R0
2444  007630  001373                          BNE     LT30B
2445  007632  013704  000622                  MOV     EMADDR,R4
2446  007636  004737  017334                  JSR     PC,TTOUT        ;PRINT HEADER
2447  007642  012704  022410                  MOV     #MSG50,R4
2448  007646  004737  017334                  JSR     PC,TTOUT        ;PRINT ALPHA ERROR
2449  007652  004737  016224                  JSR     PC,SCOPE
2450  007656  000435                          BR      LT30X
2451  007660  012777  000015  170646  LT30C:  MOV     #15,@MR         ;CLOCK MR
2452  007666  012777  000015  170640          MOV     #15,@MR         ;CLOCK MR
2453  007674  005000                          CLR     R0
2454  007676  005300                  LT30D:  DEC     R0
2455  007700  001376                          BNE     LT30D           ;DELAY
2456  007702  032777  010000  170614          BIT     #10000,@ER      ;SEE IF DTE SET
2457  007710  001006                          BNE     LT30E           ;IF SO: BR
2458  007712  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLG
2459  007720  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2460  007724  000412                          BR      LT30X
2461  007726  012701  010000          LT30E:  MOV     #10000,R1       ;SET TEST WORD
2462  007732  017702  170566                  MOV     @ER,R2          ;GET ERROR REGISTER
2463  007736  042702  020100                  BIC     #20100,R2       ;MASK OPI AND VPE
2464  007742  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2465  007744  001402                          BEQ     LT30X           ;IF NOT: BR
2466  007746  004737  014664                  JSR     PC,LTGER3       ;ELSE GO PRINT ERROR
2467  007752  004737  016254          LT30X:  JSR     PC,ITER         ;GO SEE IF ITERATION
2468  007756  004737  014530                  JSR     PC,EORPA        ;GO CLEAR GO BIT
2469  007762  004737  015344                  JSR     PC,DRVCLR
2470  007766  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
2471
```

```
2472
2473                                        ;LOGIC TEST 31: OPERATION INCOMPLETE(OPI)**********
2474
2475  007772  012737  024551  000622  LT31:   MOV     #MSLT31,EMADDR  ;SET TEST HEADER
2476  010000  012737  010000  000712  LT31IT: MOV     #LT31IT,SCOLP   ;SET SCOPE ADDRESS
2477  010006  012737  026760  000672          MOV     #TMS33A,ERADD   ;SET ERROR MSG HDR
2478  010014  012737  000002  000606          MOV     #2,ITAMT        ;SET REDUCED ITER COUNT
2479  010022  004737  016372                  JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2480  010026  005000                          CLR     R0
2481  010030  005300                  1$:     DEC     R0
2482  010032  001376                          BNE     1$              ;AWAIT OPI RESET
2483  010034  052777  000300  170500          BIS     #300,@TC        ;SET FORMAT
2484  010042  012777  000013  170464          MOV     #13,@MR         ;SET WAM 2
2485  010050  005077  170442                  CLR     @FC             ;SET FRAME COUNT
2486  010054  012705  020000                  MOV     #20000,R5       ;SET TEST BIT (OPI)
2487  010060  012702  010076                  MOV     #2$,R2          ;SET RETURN ADDRESS FROM TIMER
2488  010064  004737  010276                  JSR     PC,TIMON        ;START TIMER
2489  010070  012777  000061  170412          MOV     #61,@C1         ;LOAD WRITE+GO
2490  010076  030577  170422          2$:     BIT     R5,@ER          ;BRANCH WHEN OPI SETS
2491  010102  001002                          BNE     3$
2492  010104  000163  010370                  JMP     TIMER(R3)       ;GO TO TIMER & RETURN TO 2$ ABOVE
2493  010110  017702  170410          3$:     MOV     @ER,R2          ;GET ERROR REGISTER
2494  010114  020502                          CMP     R5,R2           ;SEE IF UNEXPECTED ERRORS
2495  010116  001403                          BEQ     4$              ;IF NOT: BR
2496  010120  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2497  010124  000453                          BR      LT31X
2498  010126  004737  010462          4$:     JSR     PC,TIMOK        ;GO CHECK TIME FOR OPI TO SET
2499  010132  102450                          BVS     LT31X           ;BRANCH IF TIME WAS INCORRECT
2500
2501  010134  012737  010150  000712          MOV     #LT31A,SCOLP    ;SET SCOPE LOOP
2502  010142  012737  026774  000672          MOV     #TMS33B,ERADD   ;SET ERROR MSG HEADER
2503  010150  004737  016372          LT31A:  JSR     PC,INIT3        ;GO INIT
2504  010154  005000                          CLR     R0
2505  010156  005300                  1$:     DEC     R0              ;WAIT FOR OPI TO CLEAR
2506  010160  001376                          BNE     1$
2507  010162  052777  000300  170352          BIS     #300,@TC        ;SET FORMAT
2508  010170  012777  000015  170336          MOV     #15,@MR         ;SET WRAP 3
2509  010176  012702  010220                  MOV     #2$,R2          ;SET RETURN ADDRESS FROM TIMER
2510  010202  012705  020000                  MOV     #20000,R5       ;SET TEST WORD
2511  010206  004737  010276                  JSR     PC,TIMON        ;START TIMER
2512  010212  012777  000071  170270          MOV     #71,@C1         ;LOAD READ+GO
2513  010220  030577  170300          2$:     BIT     R5,@ER          ;BRANCH WHEN OPI SETS
2514  010224  001002                          BNE     3$
2515  010226  000163  010370                  JMP     TIMER(R3)       ;GO TO TIMER
2516  010232  017702  170266          3$:     MOV     @ER,R2          ;GET ERROR REGISTER
2517  010236  020502                          CMP     R5,R2           ;SEE IF UNEXPECTED ERRORS
2518  010240  001403                          BEQ     4$
2519  010242  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2520  010246  000402                          BR      LT31X           ;EXIT TEST
2521  010250  004737  010462          4$:     JSR     PC,TIMOK        ;GO CHECK TIME
2522  010254  004737  016254          LT31X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2523  010260  004737  015344                  JSR     PC,DRVCLR
2524  010264  012737  000020  000606          MOV     #20,ITAMT
2525  010272  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
2526
2527                                        ;ROUTINE TO START THE TIMER. THE TIMER IS AN OSCILLATOR IN THE MAINT-
```

```
                                        ;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.
010276  005000                  TIMON:  CLR     R0              ;CLEAR TICK COUNT
010300  005001                          CLR     R1
010302  012703  000024                  MOV     #24,R3          ;PRESET INDEX TO TIMER
010306  032777  000100  170220          BIT     #100,@MR        ;BRANCH IF OSC CLEAR
010314  001405                          BEQ     2$
010316  032777  000100  170210  1$:     BIT     #100,@MR        ;WAIT FOR OSC TO CLEAR
010324  001374                          BNE     1$
010326  000405                          BR      4$              ;EXIT

010330  005403                  2$:     NEG     R3              ;SET INDEX TO TIMER
010332  032777  000100  170174  3$:     BIT     #100,@MR        ;WAIT FOR OSC TO SET
010340  001774                          BEQ     3$
010342  000207                  4$:     RTS     PC              ;RETURN

                                        ;THIS ROUTINE TIMES AN EVENT. EACH TIME THE OSCILLATOR BIT CHANGES
                                        ;STATE THE TICK COUNT IN R1 & R0 IS INCREMENTED. THE ROUTINE IS CALLED
                                        ;USING R3 AS AN INDEX TO INDICATE THE OSCILLATORS PAST STATE. WHEN
                                        ;THE OSC BIT CHANGES STATE R3 IS NEGATED.
010344  032777  000100  170162  TIMER1: BIT     #100,@MR        ;BRANCH IF OSC HAS CHANGED STATE
010352  001406                          BEQ     TIMER
010354  000112                          JMP     (R2)            ;RETURN
        010370                          .=TIMER1+24
010370  005403                  TIMER:  NEG     R3              ;SET INDEX TO OTHER STATE
010372  062700  000001                  ADD     #1,R0           ;INCREMENT TICK COUNT
010376  005501                          ADC     R1
010400  022701  000003                  CMP     #3,R1           ;BRANCH IF TIMER OVERFLOWS
010404  001410                          BEQ     TIMOVF
010406  000112                          JMP     (R2)            ;RETURN
        010414                          .=TIMER +24
010414  032777  000100  170112  TIMER0: BIT     #100,@MR        ;BRANCH IF OSC SET
010422  001362                          BNE     TIMER
010424  000112                          JMP     (R2)            ;RETURN

010426  013704  000622          TIMOVF: MOV     EMADDR,R4       ;TYPE TEST HEADER
010432  004737  017334                  JSR     PC,TTOUT
010436  013704  000672                  MOV     ERADD,R4        ;GET ERROR MSG ADDRESS
010442  004737  017334                  JSR     PC,TTOUT        ;AND TYPE IT
010446  012704  027054                  MOV     #TMS33E,R4      ;TYPE
010452  004737  017334                  JSR     PC,TTOUT        ;'TIMER OVERFLOWED'
010456  000137  010254                  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.
010462  000240                  TIMOK:  NOP
010464  006201                          ASR     R1              ;DIVIDE COUNT BY 8
010466  006000                          ROR     R0
010470  006201                          ASR     R1
010472  006000                          ROR     R0
010474  006201                          ASR     R1
010476  006000                          ROR     R0
```

```
 2584  010500  020027  027764            CMP     R0,#12276.      ;BRANCH IF GREATER THAN LOWER LIMIT
 2585  010504  101016                    BHI     1$
 2586  010506  013704  000622            MOV     EMADDR,R4       ;GET ERROR MSG HEADER
 2587  010512  004737  017334            JSR     PC,TTOUT        ;TYPE ERROR MSG HEADER
 2588  010516  013704  000672            MOV     ERADD,R4        ;GET ERROR DESCRIPTOR MSG
 2589  010522  004737  017334            JSR     PC,TTOUT
 2590  010526  012704  027007            MOV     #TMS33C,R4      ;TYPE 'OCCURED TOO SOON'
 2591  010532  004737  017334            JSR     PC,TTOUT
 2592  010536  000262                    SEV                     ;SET 'V' TO INDICATE ERROR
 2593  010540  000420                    BR      2$
 2594
 2595  010542  020027  045035    1$:     CMP     R0,#18973.      ;BRANCH IF LESS THAN UPPER LIMIT
 2596  010546  003415                    BLE     2$
 2597  010550  013704  000622            MOV     EMADDR,R4       ;GET ERROR MSG HEADER
 2598  010554  004737  017334            JSR     PC,TTOUT
 2599  010560  013704  000672            MOV     ERADD,R4
 2600  010564  004737  017334            JSR     PC,TTOUT        ;TYPE ERROR MSG HEADER
 2601  010570  012704  027031            MOV     #TMS33D,R4      ;TYPE 'OCCURED TOO LATE'
 2602  010574  004737  017334            JSR     PC,TTOUT
 2603  010600  000262                    SEV
 2604  010602  000207            2$:     RTS     PC
```

```
2605
2606
2607
2608                                        ;LOGIC TEST 32: UNSAFE(UNS)**********
2609
2610  010604  012737  024605  000622  LT32:   MOV     #MSLT32,EMADDR  ;SET TEST HEADER
2611  010612  012737  010620  000712          MOV     #LT32IT,SCOLP   ;SET SCOPE ADDRESS
2612  010620  004737  016372          LT32IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE +SLAVE
2613  010624  013700  000664                  MOV     SLVN,R0         ;GET SLAVE NUMBER
2614  010630  005100                          COM     R0              ;SET NONEXISTANT SLAVE
2615  010632  042700  177770                  BIC     #177770,R0      ;MASK SLAVE NUMBER
2616  010636  052700  000300                  BIS     #300,R0         ;SET FORMAT
2617  010642  010077  167674                  MOV     R0,@TC          ;SELECT ILLEGAL SLAVE
2618  010646  032777  002000  167662          BIT     #2000,@DT       ;EXIT TEST IF SALVE AVAILABLE
2619  010654  001032                          BNE     LT32XX
2620  010656  012777  000071  167624          MOV     #71,@C1         ;LOAD READ+GO
2621  010664  012701  044000                  MOV     #44000,R1       ;SET TEST WORD
2622  010670  017702  167630                  MOV     @ER,R2          ;READ ER
2623  010674  030102                          BIT     R1,R2           ;SEE IF EXPT=RCVD
2624  010676  001011                          BNE     1$              ;IF SO: BR
2625  010700  012737  027074  000672          MOV     #TMS34,ERADD    ;SET ERROR CODE
2626  010706  012737  000001  000716          MOV     #1,EXFL         ;SET ERROR CODE
2627  010714  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2628  010720  000404                          BR      LT32X
2629  010722  020102                  1$:     CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2630  010724  001402                          BEQ     LT32X           ;IF NOT: BR
2631  010726  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2632  010732  004737  016254          LT32X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2633  010736  004737  015344                  JSR     PC,DRVCLR
2634  010742  000137  002344          LT32XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
 2635
 2636                                        ;THE FOLLOWING 6 TESTS WILL LOOK AT VARIOUS BITS IN THE
 2637                                        ;DRIVE STATUS(DS) AND TAPE CONTROL(TC)
 2638                                        ;REGISTERS BY FORCING CERTAIN CONDITONS WHICH DO NOT
 2639                                        ;REQUIRE TAPE MOVEMENT.
 2640
 2641                                        ;LOGIC TEST 33: POSITIONING IN PROGRESS(PIP)**********
 2642
 2643  010746  012737  024641  000622  LT33:   MOV     #MSLT33,EMADDR  ;SET TEST HEADER
 2644  010754  012737  010762  000712          MOV     #LT33IT,SCOLP   ;SET SCOPE ADDRESS
 2645  010762  004737  016372          LT33IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
 2646  010766  012777  000013  167540          MOV     #13,@MR         ;SET WAM 2
 2647  010774  012777  177777  167514          MOV     #-1,@FC         ;SET FCS
 2648  011002  012777  000031  167500          MOV     #31,@C1         ;LOAD SPACE FORWARD+GO
 2649  011010  032777  020000  167504          BIT     #20000,@DS      ;SEE IF PIP=1
 2650  011016  001010                          BNE     LT33X           ;IF SO: BR
 2651  011020  012737  026567  000672          MOV     #TMS14,ERADD    ;SET ERROR CODE
 2652  011026  012737  000001  000716          MOV     #1,EXFL         ;SET ERROR CODE
 2653  011034  004737  014676                  JSR     PC,LTGERO       ;GO PRINT ERROR
 2654  011040  004737  016254          LT33X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
 2655  011044  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2656
2657                                        ;LOGIC TEST 34: PHASE ENCODED STATUS(PES)********
2658
2659  011050  012737  026507  000672  LT34:   MOV     #TMS6,ERADD      ;SET ERROR CODE
2660  011056  012737  024675  000622          MOV     #MSLT34,EMADDR   ;SET TEST HEADER
2661  011064  012700  000004          LT34IT: MOV     #4,R0
2662  011070  004737  016372          LT34A1: JSR     PC,INIT3         ;GO INIT, SELECT DRIVE+SLAVE
2663  011074  042777  003400  167440          BIC     #3400,@TC        ;SELECT NRZI
2664  011102  052777  001400  167432          BIS     #1400,@TC
2665  011110  032777  000040  167404  LT34A:  BIT     #40,@DS          ;SEE IF PES=0
2666  011116  001410                          BEQ     LT34B            ;IF SO: BR
2667  011120  012737  000002  000716          MOV     #2,EXFL          ;SET RCVD-NOT EXPT
2668  011126  012737  011070  000712          MOV     #LT34A1,SCOLP    ;SET SCOPE ADDRESS
2669  011134  004737  014676                  JSR     PC,LTGER0        ;GO PRINT ERROR
2670  011140  004737  016406          LT34B:  JSR     PC,INIT4
2671  011144  032777  000040  167350  LT34C:  BIT     #40,@DS          ;SEE IF PES=1
2672  011152  001010                          BNE     LT34X            ;IF SO: BR
2673  011154  012737  011144  000712          MOV     #LT34C,SCOLP     ;SET SCOPE ADDRESS
2674  011162  012737  000001  000716          MOV     #1,EXFL          ;SET EXPT-NOT RCVD FLAG
2675  011170  004737  014676                  JSR     PC,LTGER0        ;GO PRINT ERROR
2676  011174  004737  016254          LT34X:  JSR     PC,ITER          ;GO SEE IF ITERATION
2677  011200  000137  002344          LT34XX: JMP     TSCD2            ;RETURN TO SCHED
```

```
2678
2679                                    ;LOGIC TEST 35: SLAVE ADDRESS CHANGE (SAC)**********
2680
2681  011204  012737  027117  000672  LT35:   MOV     #TMS37,ERADD
2682  011212  012737  024731  000622          MOV     #MSLT35,EMADDR
2683  011220  004737  016372          LT35IT: JSR     PC,INIT3        ;INIT SELECT DRIVE, SLAVE
2684  011224  032777  000020  167270  1$:     BIT     #20,@DS         ;SEE IF SDWN IS RESET
2685  011232  001374                          BNE     1$              ;IF NOT: BR
2686  011234  052777  000300  167300          BIS     #300,@TC        ;SET FORMAT
2687  011242  012777  000015  167264          MOV     #15,@MR         ;SET WAM 3
2688  011250  012777  000071  167232          MOV     #71,@C1         ;LOAD READ+GO
2689  011256  032777  020000  167256          BIT     #20000,@TC      ;SEE IF SAC=0
2690  011264  001410                          BEQ     LT35A           ;IF SO: BR
2691  011266  012737  000002  000716          MOV     #2,EXFL         ;SET RCV-NOT EXPT FLAG
2692  011274  012737  011220  000712          MOV     #LT35IT,SCOLP   ;SET SCOPE ADDRESS
2693  011302  004737  014676                  JSR     PC,LTGERO       ;GO PRINT ERROR
2694  011306  004737  016372          LT35A:  JSR     PC,INIT3        ;INIT
2695  011312  005277  167224                  INC     @TC             ;BUMP SLAVE ADDRESS
2696  011316  032777  020000  167216          BIT     #20000,@TC      ;SEE IF SAC=1
2697  011324  001010                          BNE     LT35X           ;IF SO: BR
2698  011326  012737  011306  000712          MOV     #LT35A,SCOLP    ;SET SCOPE ADDRESS
2699  011334  012737  000001  000716          MOV     #1,EXFL         ;SE EXPT-NOT RCVD FLAG
2700  011342  004737  014676                  JSR     PC,LTGERO       ;GO PRINT ERROR
2701  011346  004737  016254          LT35X:  JSR     PC,ITER
2702  011352  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
 2703
 2704                                          ;LOGIC TEST 36: FRAME COUNTER STATUS(FCS)**********
 2705
 2706  011356  012737  024776  000622  LT36:   MOV     #MSLT36,EMADDR
 2707  011364  012737  027125  000672          MOV     #TMS38,ERADD    ;SET ERROR CODE
 2708  011372  004737  016372          LT36IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
 2709  011376  032777  040000  167136          BIT     #40000,@TC      ;SEE IF FCS=0
 2710  011404  001410                          BEQ     1$              ;IF SO: BR
 2711  011406  012737  011372  000712          MOV     #LT36IT,SCOLP   ;SET SCOPE ADDRESS
 2712  011414  012737  000002  000716          MOV     #2,EXFL         ;SET RCVD-NOT EXPT
 2713  011422  004737  014676                  JSR     PC,LTGERO       ;GO PRINT ERROR
 2714  011426  004737  016372          1$:     JSR     PC,INIT3        ;INIT
 2715  011432  005077  167060                  CLR     @FC             ;WRITE TO FC
 2716  011436  032777  040000  167076          BIT     #40000,@TC      ;SEE IF FCS=1
 2717  011444  001010                          BNE     LT36X           ;IF SO: BR
 2718  011446  012737  011426  000712          MOV     #1$,SCOLP       ;SET SCOPE ADDRESS
 2719  011454  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT-NOT RCVD
 2720  011462  004737  014676                  JSR     PC,LTGERO       ;GO PRINT ERROR
 2721  011466  004737  016254          LT36X:  JSR     PC,ITER
 2722  011472  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2723
2724                                        ;LOGIC TEST 37: ACCELERATION(ACCL)**********
2725
2726  011476  012737  025043  000622  LT37:   MOV     #MSLT37,EMADDR
2727  011504  012737  027133  000672          MOV     #TMS39,ERADD    ;SET ERROR CODE
2728  011512  004737  016372          LT37IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE+SLAVE
2729  011516  052777  000300  167016          BIS     #300,@TC        ;SET FORMAT
2730  011524  005777  167012                  TST     @TC             ;SEE IF ACCL=1
2731  011530  100410                          BMI     LT37A           ;IF SO: BR
2732  011532  012737  000001  000716          MOV     #1,EXFL
2733  011540  012737  011512  000712          MOV     #LT37IT,SCOLP   ;SET SCOPE ADDRESS
2734  011546  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2735  011552  004737  016372          LT37A:  JSR     PC,INIT3        ;INIT
2736  011556  052777  000300  166756          BIS     #300,@TC        ;SET FORMAT
2737  011564  012777  000015  166742          MOV     #15,@MR         ;SET WAM 3
2738  011572  012777  000071  166710          MOV     #71,@C1         ;LOAD READ+GO
2739  011600  012700  100000                  MOV     #100000,R0      ;SET ACCL DELAY
2740  011604  005777  166732          LT37B:  TST     @TC             ;SEE IF ACCL=0
2741  011610  100012                          BPL     LT37X           ;IF SO: BR
2742  011612  005300                          DEC     R0
2743  011614  001373                          BNE     LT37B           ;DELAY
2744  011616  012737  011552  000712          MOV     #LT37A,SCOLP    ;SET SCOPE ADDRESS
2745  011624  012737  000002  000716          MOV     #2,EXFL
2746  011632  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2747  011636  004737  016254          LT37X:  JSR     PC,ITER
2748  011642  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

''

```
2749
2750                                        ;LOGIC TEST 40: PE TAPE MARK (TM)**********
2751
2752  011646  012737  011662  000712  LT40:   MOV     #LT40IT,SCOLP   ;SET SCOPE ADDRESS
2753  011654  012737  025111  000622          MOV     #MSLT40,EMADDR
2754  011662  004737  016406          LT40IT: JSR     PC,INIT4        ;INIT, SELECT DRIVE+SLAVE
2755  011666  005000                  CLR     R0
2756  011670  005300                  1$:     DEC     R0
2757  011672  001376                  BNE     1$              ;DELAY FOR OPI RESET
2758  011674  052777  002300  166640          BIS     #2300,@TC
2759  011702  012777  000007  166624          MOV     #7,@MR          ;SET WAM 0
2760  011710  012777  000027  166572          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
2761  011716  012700  100000                  MOV     #100000,R0      ;SET DELAY
2762  011722  032777  000004  166572  2$:     BIT     #4,@DS          ;SEE IF TM=1
2763  011730  001012                  BNE     LT40X           ;IF SO: BR
2764  011732  005300                  DEC     R0
2765  011734  001372                  BNE     2$              ;DELAY
2766  011736  012737  026465  000672          MOV     #TMS3,ERADD
2767  011744  012737  000001  000716          MOV     #1,EXFL
2768  011752  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2769  011756  004737  016254          LT40X:  JSR     PC,ITER
2770  011762  000137  002344          LT40XX: JMP     TSCD2           ;RETURN TO SCHED
```

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

011766  012737  012002  000712  LT41:   MOV     #LT41IT,SCOLP   ;SET SCOPE ADDRESS
011774  012737  025156  000622          MOV     #MSLT41,EMADDR
012002  004737  016372          LT41IT: JSR     PC,INIT3        ;INIT, SELECT DRIVE,SLAVE
012006  052777  001700  166526          BIS     #1700,@TC       ;SET NRZ+NORMAL FORMAT
012014  012777  177760  166474          MOV     #-20,@FC        ;SET FCS
012022  012777  000007  166504          MOV     #7,@MR          ;SET WAM 0
012030  012777  000027  166452          MOV     #27,@C1         ;LOAD WRITE TAPE MARK+GO
012036  005000                          CLR     R0
012040  032777  000004  166454  1$:     BIT     #4,@DS          ;SEE IF TM=1
012046  001012                          BNE     2$              ;IF SO: BR
012050  005300                          DEC     R0
012052  001372                          BNE     1$              ;DELAY
012054  012737  026465  000672          MOV     #TMS3,ERADD     ;SET ERROR CODE
012062  012737  000001  000716          MOV     #1,EXFL
012070  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
012074  032777  002000  166422  2$:     BIT     #2000,@ER       ;SEE IF ITM=1
012102  001010                          BNE     3$              ;IF SO: BR
012104  012737  026730  000672          MOV     #TMS30,ERADD    ;SET ERROR CODE
012112  012737  000001  000716          MOV     #1,EXFL
012120  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
012124  032777  000100  166372  3$:     BIT     #100,@ER        ;SEE IF VPE=1
012132  001011                          BNE     4$              ;IF SO: BR
012134  012737  026715  000672          MOV     #TMS28,ERADD    ;SET ERROR CODE
012142  012737  000001  000716          MOV     #1,EXFL
012150  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
012154  000410                          BR      LT41X
012156  012701  002100          4$:     MOV     #2100,R1        ;SET EXPT ERROR BITS
012162  017702  166336                  MOV     @ER,R2          ;GET ERROR REGISTER
012166  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
012170  001402                          BEQ     LT41X           ;IF NOT: BR
012172  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
012176  005002                  LT41X:  CLR     R2              ;SET TIMER
012200  032777  000200  166314  1$:     BIT     #200,@DS        ;SEE IF DRY SET
012206  001002                          BNE     2$              ;IF SO: BR
012210  005302                          DEC     R2              ;AWAIT DRY
012212  001372                          BNE     1$              ;DELAY
012214  004737  016254          2$:     JSR     PC,ITER         ;GO SEE IF ITERATIONS
012220  004737  015344                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
012224  000137  002344                  JMP     TSCD2           ;RETURN TO SCHED
```

```
2813
2814                                    ;THE FOLLOWING SIX(6) TEST WILL REQUIRE TAPE MOVEMENT.  EACH
2815                                    ;TEST WILL PERFORM A TAPE WRITE WHILE IN A PARITICULAR MAINTENANCE
2816                                    ;MODE IN ORDER TO FORCE THE REMAINING ERROR CONDITIONS.
2817
2818                                    ;LOGIC TEST 42: CYCLIC REDUNDANCY ERROR(CRC)**********
2819
2820  012230  012737  001700  000776  LT42:   MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
2821  012236  004737  014500          JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2822  012242  012700  001000          MOV     #1000,R0
2823  012246  005300                  1$:     DEC     R0
2824  012250  001376                  BNE     1$              ;PAUSE
2825  012252  012737  025225  000622          MOV     #MSLT42,EMADDR
2826  012260  012737  012266  000712          MOV     #LT42IT,SCOLP   ;SET SCOPE ADDRESS
2827  012266  004737  016422          LT42IT: JSR     PC,INIT         ;INIT SELECT DRIVE+SLAVE
2828  012272  012777  177770  166212          MOV     #-10,@WC
2829  012300  012777  177760  166210          MOV     #-20,@FC        ;SET FC=20
2830  012306  012777  027142  166200          MOV     #WDATA,@BA      ;SET BUS ADDRESS
2831  012314  012777  000007  166212          MOV     #7,@MR          ;SET MM CODE
2832  012322  012777  000061  166160          MOV     #61,@C1         ;LOAD WRITE+GO
2833  012330  005000                  CLR     R0
2834  012332  032777  000200  166162  LT42A:  BIT     #200,@DS        ;SEE IF DRY=1
2835  012340  001002                  BNE     LT42B           ;IF SO: BR
2836  012342  005300                  DEC     R0
2837  012344  001372                  BNE     LT42A           ;DELAY
2838  012346  022777  000200  166150  LT42B:  CMP     #200,@ER        ;SEE IF LRC ERROR ONLY
2839  012354  001007                  BNE     LT42B1          ;IF NOT: BR
2840  012356  017702  166146          MOV     @CC,R2          ;GET CHECK CHAR
2841  012362  042702  177000          BIC     #177000,R2      ;MASK CRC
2842  012366  022702  000777          CMP     #777,R2         ;SEE IF SETUP CRC IS CORRECT
2843  012372  001410                  BEQ     LT42B2          ;IF SO: BR
2844  012374  004737  014664  LT42B1: JSR     PC,LTGER3       ;ELSE PRINT ERROR SETUP
2845  012400  012704  022501          MOV     #MSG55,R4
2846  012404  004737  017334          JSR     PC,TTOUT        ;PRINT SETUP ERROR MSG
2847  012410  000137  002344          JMP     TSCD2           ;RETURN TO SCHED
2848  012414  004737  016422  LT42B2: JSR     PC,INIT         ;GO INIT
2849  012420  012777  177770  166064          MOV     #-10,@WC        ;SET WC
2850  012426  012777  177760  166062          MOV     #-20,@FC        ;SET FC
2851  012434  012777  027142  166052          MOV     #WDATA,@BA      ;SET BA
2852  012442  012777  000021  166064          MOV     #21,@MR         ;SET MM
2853  012450  012777  000061  166032          MOV     #61,@C1         ;LOAD WRITE+GO
2854  012456  005000                  CLR     R0
2855  012460  032777  000200  166034  LT42C:  BIT     #200,@DS        ;SEE IF DRY
2856  012466  001002                  BNE     LT42D           ;IF SO: BR
2857  012470  005300                  DEC     R0
2858  012472  001372                  BNE     LT42C           ;AWAIT DRY
2859  012474  005777  166024  LT42D:  TST     @ER             ;SEE IF CRC=1
2860  012500  100411                  BMI     LT42E           ;IF SO: BR
2861  012502  012737  027111  000672          MOV     #TMS36,ERADD    ;SET ERROR CODE
2862  012510  012737  000001  000716          MOV     #1,EXFL
2863  012516  004737  014676          JSR     PC,LTGER0       ;GO PRINT ERROR
2864  012522  000410                  BR      LT42X
2865  012524  012701  100200  LT42E:  MOV     #100200,R1      ;SET EXPT ERROR BITS
2866  012530  017702  165770          MOV     @ER,R2          ;GET ERROR REGISTER
2867  012534  020102                  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2868  012536  001402                  BEQ     LT42X           ;IF NOT: BR
```

```
2869  012540  004737  014664                LT42X:  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2870  012544  004737  016254                        JSR     PC,ITER         ;DO ITERATIONS
2871  012550  004737  015344                        JSR     PC,DRVCLR
2872  012554  000137  002344                        JMP     TSCD2           ;RETURN TO SCHED
2873
2874                                              ;LOGIC TEST 43: LONGITUALINAL REDUNDANCY(LRC)**********
2875
2876  012560  012737  001700  000776        LT43:   MOV     #1700,UDES      ;SET UNIT DESCRIPTION = NRZ
2877  012566  004737  014500                        JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2878  012572  012737  012606  000712                MOV     #LT43IT,SCOLP   ;SET SCOPE ADDRESS
2879  012600  012737  025261  000622                MOV     #MSLT43,EMADDR
2880  012606  004737  016422              LT43IT:   JSR     PC,INIT         ;INIT, SELECT DRIVE+SLAVE
2881  012612  005001                                CLR     R1
2882  012614  005301                        1$:     DEC     R1              ;DELAY
2883  012616  001376                                BNE     1$
2884  012620  012777  000023  165706                MOV     #23,@MR         ;SET MM
2885  012626  012777  177770  165656                MOV     #-10,@WC        ;SET WC
2886  012634  012777  177760  165654                MOV     #-20,@FC        ;SET FC
2887  012642  012777  027142  165644                MOV     #WDATA,@BA      ;SET BA
2888  012650  012777  000061  165632                MOV     #61,@C1         ;LOAD WRITE+GO
2889  012656  005000                                CLR     R0
2890  012660  032777  000200  165634        LT43C:  BIT     #200,@DS        ;SEE IF DRY
2891  012666  001002                                BNE     LT43D           ;IF SO: BR
2892  012670  005300                                DEC     R0
2893  012672  001372                                BNE     LT43C           ;AWAIT DRY
2894  012674  032777  000200  165622        LT43D:  BIT     #200,@ER        ;SEE IF LRC=1
2895  012702  001411                                BEQ     LT43E           ;IF NOT: BR
2896  012704  012737  026701  000672                MOV     #TMS26,ERADD    ;SET ERROR CODE
2897  012712  012737  000002  000716                MOV     #2,EXFL
2898  012720  004737  014676                        JSR     PC,LTGER0       ;GO PRINT
2899  012724  000425                                BR      LT43X
2900  012726  017702  165602              LT43E:    MOV     @MR,R2
2901  012732  042702  000177                        BIC     #177,R2         ;MASK LRC
2902  012736  012701  177600                        MOV     #177600,R1      ;SET EXPT LRC
2903  012742  020102                                CMP     R1,R2           ;SEE IF EXPT = RCVD
2904  012744  001405                                BEQ     LT43F           ;IF SO: BR
2905  012746  012737  022456  000672                MOV     #MSG53,ERADD    ;SET ERROR CODE
2906  012754  004737  016012                        JSR     PC,LTGER1       ;PRINT ERROR
2907  012760  017702  165540              LT43F:    MOV     @ER,R2          ;GET ERROR REGISTER
2908  012764  012701  000000                        MOV     #0,R1           ;SET EXPT ERROR BITS
2909  012770  020102                                CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
2910  012772  001402                                BEQ     LT43X           ;IF NOT: BR
2911  012774  004737  014664                        JSR     PC,LTGER3       ;ELSE PRINT ERROR
2912  013000  004737  016254              LT43X:    JSR     PC,ITER
2913  013004  000137  002344                        JMP     TSCD2           ;RETURN TO SCHED
```

```
2914                                        ;LOGIC TEST 44: PE CORRECTABLE DATA (CORR)**********
2915
2916  013010  012737  002300  000776  LT44:   MOV     #2300,UDES      ;SET UNIT DESRIPTION = PE
2917  013016  004737  014500                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
2918  013022  012737  025315  000622          MOV     #MSLT44,EMADDR  ;SET HEADER
2919  013030  012737  013036  000712          MOV     #LT44IT,SCOLP   ;SET SCOP
2920  013036  004737  016422          LT44IT: JSR     PC,INIT         ;GO INITIALIZE
2921  013042  012777  177600  165442          MOV     #-200,@WC       ;SET WC=200
2922  013050  012777  177400  165440          MOV     #-400,@FC       ;SET FC=400
2923  013056  012777  027142  165430          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
2924  013064  012777  000061  165416          MOV     #61,@C1         ;LOAD WRITE AND GO
2925  013072  005000                          CLR     R0
2926  013074  005777  165416          LT44A:  TST     @FC             ;SEE IF FC=0
2927  013100  001402                          BEQ     LT44A1          ;IF SO:BR
2928  013102  005300                          DEC     R0
2929  013104  001373                          BNE     LT44A           ;AWAIT FC=0
2930  013106  012777  000021  165420  LT44A1: MOV     #21,@MR         ;SET MAINT MODE
2931  013114  005000                          CLR     R0
2932  013116  032777  000200  165376  LT44B:  BIT     #200,@DS        ;SEE IF DRY
2933  013124  001002                          BNE     LT44C           ;IF SO :BR
2934  013126  005300                          DEC     R0
2935  013130  001372                          BNE     LT44B           ;AWAIT DRY
2936  013132  005777  165366          LT44C:  TST     @ER             ;SEE IF CORR=1
2937  013136  100410                          BMI     LT44D           ;IF SO: BR
2938  013140  012737  027102  000672          MOV     #TMS35,ERADD    ;ELSE SET ERROR CODE
2939  013146  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
2940  013154  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
2941  013160  000240          LT44D:  NOP
2942  013162  122777  000002  165340  LT44E:  CMPB    #2,@CC          ;SEE IF DEAD TRACK BIT 1
2943  013170  001414                          BEQ     LT44F           ;IF SO: BR
2944  013172  117702  165332                  MOVB    @CC,R2          ;ELSE SAVE RECVD
2945  013176  042702  177000                  BIC     #177000,R2      ;MASK OUT CRC
2946  013202  112701  000002                  MOVB    #2,R1           ;SAVE EXPT
2947  013206  012737  022065  000672          MOV     #MSG42,ERADD    ;SET ERROR CODE
2948  013214  004737  016012                  JSR     PC,LTGER1       ;GO PRINT ERROR
2949  013220  000410                          BR      LT44X
2950  013222  017702  165276          LT44F:  MOV     @ER,R2          ;GET ERROR REGISTER
2951  013226  012701  100000                  MOV     #100000,R1      ;SET EXPT ERROR BITS
2952  013232  020102                          CMP     R1,R2           ;SEE IF EXPT=RCVD
2953  013234  001402                          BEQ     LT44X           ;IF SO: BR
2954  013236  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
2955  013242  004737  016254          LT44X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
2956  013246  004737  015344                  JSR     PC,DRVCLR       ;GO DO DRIVE CLEAR
2957  013252  000137  002344          LT44XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
                                        ;LOGIC TEST 45: PE INCORRECTABLE DATA(INC)**********

013256  012737  002300  000776  LT45:   MOV     #2300,UDES      ;SET UNIT DESCRIPTION = PE
013264  004737  014500                  JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
013270  012737  025375  000622          MOV     #MSLT45,EMADDR
013276  012737  013304  000712          MOV     #LT45IT,SCOLP
013304  004737  016422          LT45IT: JSR     PC,INIT         ;INIT SELECT DRIVE SLAVE
013310  012777  177600  165174          MOV     #-200,@WC       ;SET WC=200
013316  012777  177400  165172          MOV     #-400,@FC       ;SET FC=400
013324  012777  027142  165162          MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
013332  012777  000061  165150          MOV     #61,@C1         ;LOAD WRITE+GO
013340  005000                          CLR     R0
013342  005777  165150          LT45E:  TST     @FC             ;AWAIT FC=0
013346  001402                          BEQ     LT45E1
013350  005300                          DEC     R0
013352  001373                          BNE     LT45E           ;AWAIT FC=0
013354  012777  000023  165152  LT45E1: MOV     #23,@MR         ;SET MAINT CODE
013362  005000                          CLR     R0
013364  032777  000200  165130  LT45A:  BIT     #200,@DS        ;SEE IF DRY IS SET
013372  001002                          BNE     LT45B           ;IF SO: BR
013374  005300                          DEC     R0
013376  001372                          BNE     LT45A           ;AWAIT DRY
013400  032777  000100  165116  LT45B:  BIT     #100,@ER        ;SEE IF INC=1
013406  001010                          BNE     LT45D           ;IF SO:BR
013410  012737  026657  000672          MOV     #TMS23,ERADD    ;SET ERROR CODE
013416  012737  000001  000716          MOV     #1,EXFL
013424  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
013430  017702  165074          LT45D:  MOV     @CC,R2          ;GET CHECK CHAR
013434  042702  177000                  BIC     #177000,R2      ;MASK CHECK CHAR
013440  012701  000046                  MOV     #46,R1          ;SET EXPT CK
013444  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
013446  001405                          BEQ     LT45F           ;IF SO: BR
013450  012737  022470  000672          MOV     #MSG54,ERADD
013456  004737  016012                  JSR     PC,LTGER1       ;ELSE GO PRINT ERROR
013462  017702  165036          LT45F:  MOV     @ER,R2
013466  042702  120600                  BIC     #120600,R2      ;MASK OPI ,NSG, CORR, AND PEF
013472  012701  000100                  MOV     #100,R1         ;SET EXPT ERROR BITS
013476  020102                          CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
013500  001402                          BEQ     LT45X           ;IF NOT: BR
013502  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
013506  004737  016254          LT45X:  JSR     PC,ITER
013512  004737  015344                  JSR     PC,DRVCLR
013516  000137  002344          LT45XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
3002
3003                                    ;LOGIC TEST 46: PE FORMAT ERROR(PEF)**********
3004
3005  013522  012737  002300  000776  LT46:   MOV     #2300,UDES      ;SET UNIT DESCRIPTION = PE
3006  013530  004737  014500          JSR     PC,STATIC       ;GO SEE IF STATIC ONLY
3007  013534  012737  025457  000622  MOV     #MSLT46,EMADDR  ;SET HEADER
3008  013542  012737  013550  000712  MOV     #LT46IT,SCOLP   ;SET SCOPE ADDRESS
3009  013550  004737  016422          LT46IT: JSR     PC,INIT         ;INITIALIZE
3010  013554  012777  177770  164730  MOV     #-10,@WC        ;SET WC=10
3011  013562  012777  177760  164726  MOV     #-20,@FC        ;SET FC=20
3012  013570  012777  027142  164716  MOV     #WDATA,@BA      ;SET BA=START OF WRITE BUFFER
3013  013576  012777  000061  164704  MOV     #61,@C1         ;LOAD WRITE+GO
3014  013604  005777  164706          LT46A:  TST     @FC
3015  013610  001375                  BNE     LT46A           ;AWAIT FC=0
3016  013612  032777  000100  164714  1$:     BIT     #100,@MR
3017  013620  001774                  BEQ     1$              ;DELAY
3018  013622  032777  000100  164704  2$:     BIT     #100,@MR
3019  013630  001374                  BNE     2$
3020  013632  032777  000100  164674  3$:     BIT     #100,@MR
3021  013640  001774                  BEQ     3$
3022  013642  012777  000027  164664          MOV     #27,@MR         ;SET MM CODE TO KILL PEF
3023  013650  005000                  CLR     R0              ;INIT TIMING LOOP
3024  013652  032777  000200  164642  LT46B:  BIT     #200,@DS        ;SEE IF DRY SET
3025  013660  001002                  BNE     LT46C           ;IF SO: BR
3026  013662  005300                  DEC     R0
3027  013664  001372                  BNE     LT46B           ;AWAIT DRY
3028  013666  032777  000200  164630  LT46C:  BIT     #200,@ER        ;SEE IF PEF SET
3029  013674  001011                  BNE     LT46D           ;IF SO: BR
3030  013676  012737  026673  000672          MOV     #TMS25,ERADD    ;SET ERROR TAG
3031  013704  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
3032  013712  004737  014676                  JSR     PC,LTGER0       ;GO PRINT ERROR
3033  013716  000412                  BR      LT46X
3034  013720  017702  164600          LT46D:  MOV     @ER,R2          ;GET ERROR REGISTER
3035  013724  042702  020500                  BIC     #20500,R2       ;CLEAR OPI BIT (MAY OR MAY NOT SET)
3036  013730  012701  000200                  MOV     #200,R1         ;SET EXPT ERROR BITS
3037  013734  020102                  CMP     R1,R2           ;SEE IF UNEXPECTED ERRORS
3038  013736  001402                  BEQ     LT46X           ;IF NOT: BR
3039  013740  004737  014664                  JSR     PC,LTGER3       ;ELSE PRINT ERROR
3040  013744  004737  016254          LT46X:  JSR     PC,ITER
3041  013750  004737  015344                  JSR     PC,DRVCLR
3042  013754  000137  002344          LT46XX: JMP     TSCD2           ;RETURN TO SCHED
```

```
3043                                        ;LOGIC TEST 47: FRAME COUNT OVERFLOW(M8905-YB)*******
3044
3045  013760  012737  025513  000622  LT47:   MOV     #MSLT47,EMADDR  ;SET TEST HEADER
3046  013766  012737  013774  000712          MOV     #LT47IT,SCOLP   ;SET SCOPE ADDRESS
3047  013774  004737  016372          LT47IT: JSR     PC,INIT3        ;GO INIT
3048  014000  012777  177770  164504          MOV     #-10,@WC        ;SET WC = 10
3049  014006  012777  177760  164502          MOV     #-20,@FC        ;SET FC = 20
3050  014014  052777  001700  164520          BIS     #1700,@TC       ;SET TO NRZ, NORMAL, ODD
3051  014022  012777  027142  164464          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3052  014030  012777  000013  164476          MOV     #13,@MR         ;SET WRAP 2
3053  014036  012777  000061  164444          MOV     #61,@C1         ;LOAD WRITE+GO
3054  014044  012700  040000                  MOV     #40000,R0
3055  014050  005777  164466          LT47A:  TST     @TC             ;SEE IF ALPHA
3056  014054  100002                          BPL     LT47B           ;IF SO: BR
3057  014056  005300                          DEC     R0
3058  014060  001373                          BNE     LT47A           ;AWAIT ALPHA
3059  014062  012700  000020          LT47B:  MOV     #20,R0          ;SET CLK CNT
3060  014066  052777  000040  164440  LT47C:  BIS     #40,@MR
3061  014074  042777  000040  164432          BIC     #40,@MR         ;CLOCK MR
3062  014102  005300                          DEC     R0
3063  014104  001370                          BNE     LT47C           ;IF NOT DONE ALL: BR
3064  014106  017702  164404                  MOV     @FC,R2
3065  014112  005001                          CLR     R1              ;SET TEST WORD
3066  014114  020102                          CMP     R1,R2           ;SEE IF EXPT = RCVD
3067  014116  001410                          BEQ     LT47X           ;IF SO: BR
3068  014120  012737  021340  000672          MOV     #MSG19,ERADD    ;SET ERROR CODE
3069  014126  012737  000001  000716          MOV     #1,EXFL         ;SET EXPT FLAG
3070  014134  004737  016012                  JSR     PC,LTGER1       ;GO PRINT ERROR
3071  014140  004737  016254          LT47X:  JSR     PC,ITER         ;GO SEE IF ITERATIONS
3072  014144  000137  002344                  JMP     TSCD2           ;RETURN TO SCHEDULAR
3073
```

```
3074
3075                                        ;LOGIC TEST 50: NEF WHEN WRITING PE ON NRZ SELECTED SLAVE
3076
3077  014150  012737  025563  000622  LT50:   MOV     #MSLT50,EMADDR  ;SET TEST HEADER
3078  014156  012737  014164  000712          MOV     #LT50IT,SCOLP
3079  014164  004737  016372          LT50IT: JSR     PC,INIT3        ;SET SLAVE = NRZ
3080  014170  042777  003400  164344          BIC     #3400,@TC       ;CLEAR DENSITY BITS
3081  014176  052777  002300  164336          BIS     #2300,@TC       ;SET DENSITY = PE
3082  014204  012777  177770  164300          MOV     #-10,@WC        ;SET WORD COUNT
3083  014212  012777  177760  164276          MOV     #-20,@FC        ;SET FRAME COUNT
3084  014220  012777  027142  164266          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3085  014226  012777  000013  164300          MOV     #13,@MR         ;SET WRAP 2
3086  014234  012777  000061  164246          MOV     #61,@C1         ;LOAD WRITE COMMAND
3087  014242  000240                          NOP
3088  014244  000240                          NOP
3089  014246  000240                          NOP
3090  014250  012701  004000                  MOV     #4000,R1        ;SET EXPECTED RESULT
3091  014254  017702  164244                  MOV     @ER,R2          ;GET ERROR REGISTER
3092  014260  030102                          BIT     R1,R2           ;BRANCH IF NEF BIT SET
3093  014262  001006                          BNE     1$
3094  014264  012737  000001  000716          MOV     #1,EXFL         ;SET EXPECTED FLAG
3095  014272  004737  014676                  JSR     PC,LTGER0       ;PRINT ERROR
3096  014276  000404                          BR      LT50X
3097  014300  020102                  1$:     CMP     R1,R2           ;BRANCH IF NO UNEXPECTED ERROR
3098  014302  001402                          BEQ     LT50X           ;BITS WERE SET
3099  014304  004737  014664                  JSR     PC,LTGER3       ;PRINT ERROR MSG
3100  014310  004737  016254          LT50X:  JSR     PC,ITER         ;ITERATE TEST
3101  014314  004737  015344                  JSR     PC,DRVCLR       ;RESET DRIVE
3102  014320  000137  002344                  JMP     TSCD2
3103                                        ;LOGIC TEST 51: NEF WHEN WRITING NRZ ON PE SELECTED SLAVE
3104
3105  014324  012737  025643  000622  LT51:   MOV     #MSLT51,EMADDR  ;SET ERROR MSG HEADER
3106  014332  012737  014340  000712          MOV     #LT51IT,SCOLP   ;SET SCOPE LOOP ADDRESS
3107  014340  004737  016406          LT51IT: JSR     PC,INIT4        ;SET SLAVE = PE
3108  014344  042777  002300  164170          BIC     #2300,@TC       ;CLEAR DENSITY BITS
3109  014352  052777  001300  164162          BIS     #1300,@TC       ;SET DENSITY = NRZ
3110  014360  012777  177770  164124          MOV     #-10,@WC        ;SET WORD COUNT
3111  014366  012777  177760  164122          MOV     #-20,@FC        ;SET FRAME COUNT
3112  014374  012777  027142  164112          MOV     #WDATA,@BA      ;SET BUS ADDRESS
3113  014402  012777  000013  164124          MOV     #13,@MR         ;SET WRAP 2
3114  014410  012777  000061  164072          MOV     #61,@C1         ;SET WRITE COMMAND AND GO
3115  014416  000240                          NOP
3116  014420  000240                          NOP
3117  014422  000240                          NOP
3118  014424  012701  004000                  MOV     #4000,R1        ;SET EXPECTED RESULT
3119  014430  017702  164070                  MOV     @ER,R2          ;GET ERROR REGISTER
3120  014434  030102                          BIT     R1,R2           ;BRANCH IF NEF SET
3121  014436  001006                          BNE     1$
3122  014440  012737  000001  000716          MOV     #1,EXFL         ;SET EXPECTED FLAG
3123  014446  004737  014676                  JSR     PC,LTGER0       ;PRINT ERROR MSG
3124  014452  000404                          BR      LT51X
3125  014454  020102                  1$:     CMP     R1,R2           ;BRANCH IF NO UNEXPECTED
3126  014456  001402                          BEQ     LT51X           ;ERROR BITS WERE SET
3127  014460  004737  014664                  JSR     PC,LTGER3
3128  014464  004737  016254          LT51X:  JSR     PC,ITER         ;ITERATE TEST
3129  014470  004737  015344                  JSR     PC,DRVCLR       ;CLEAR DRIVE
```

```
  3130  014474  000137  002344                    JMP     TSCD2           ;RETURN TO SCHEDULER
```

```
3131                                    ;STATIC TESTS ONLY SUBROUTINE*********
3132
3133  014500  005737  000740  STATIC: TST     STFLG           ;SEE IF SINGLE TEST ONLY
3134  014504  001006                  BNE     1$              ;IF SO: BR
3135  014506  005737  001012          TST     STATC           ;SEE IF STATIC ONLY
3136  014512  001403                  BEQ     1$              ;IF NOT: BR
3137  014514  005726                  TST     (SP)+           ;RESET STACK
3138  014516  000137  002344          JMP     TSCD2           ;RETURN TO SCHEDULAR
3139  014522  005037  001006  1$:     CLR     RDRVF
3140  014526  000207                  RTS     PC              ;RETURN TO TEST
3141
```

```
3142
3143                                        ;END OF RECORD FORCE SUBROUTINE*****
3144
3145  014530  017700  164000          EORPA:  MOV     @MR,R0          ;GET MAINT REG
3146  014534  042700  000036                  BIC     #36,R0          ;CLEAR CURRENT OP CODE
3147  014540  052700  000024                  BIS     #24,R0          ;SET EOR CLEAR OP CODE
3148  014544  010077  163764                  MOV     R0,@MR          ;DO EOR
3149  014550  042777  000037  163756          BIC     #37,@MR         ;CLEAR EOR AND MM
3150  014556  005000                          CLR     R0
3151  014560  012701  000002                  MOV     #2,R1
3152  014564  032777  000001  163716  EORP1:  BIT     #1,@C1          ;SEE IF GO GONE
3153  014572  001430                          BEQ     EORP2           ;IF SO: BR
3154  014574  005300                          DEC     R0
3155  014576  001372                          BNE     EORP1           ;AWAIT GO RESET
3156  014600  005301                          DEC     R1
3157  014602  001370                          BNE     EORP1
3158  014604  032777  020000  163756          BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3159  014612  001020                          BNE     EORP2           ;IF SO: BR
3160  014614  005737  000620                  TST     HDRFL           ;SEE IF DONE HEADER
3161  014620  001004                          BNE     EORP1A          ;IF SO: BR
3162  014622  013704  000622                  MOV     EMADDR,R4
3163  014626  004737  017334                  JSR     PC,TTOUT        ;PRINT HEADER
3164  014632  012704  027555          EORP1A: MOV     #WMSG31,R4
3165  014636  004737  017334                  JSR     PC,TTOUT        ;PRINT EOR GO BIT ERROR
3166  014642  032777  100000  163720          BIT     #100000,@SWR    ;SEE IF HALT ON ERROR
3167  014650  001401                          BEQ     EORP2           ;IF NOT: BR
3168  014652  000000                          HALT
3169  014654  000240          EORP2:  NOP
3170  014656  005037  000676          EORPX:  CLR     TEMP2           ;CLEAR FLAG
3171  014662  000207                          RTS     PC              ;RETURN
3172
```

```
3173                                    ;LOGIC TEST ADDRESSING ERROR SUBROUTINE**********
3174
3175  014664  005037  000716          LTGER3: CLR     EXFL
3176  014670  012737  022427  000672          MOV     #MSG51,ERADD
3177  014676  012737  000001  000746  LTGER0: MOV     #1,ADDFL        ;SET NO ADDRESS FLAG
3178  014704  000240                  LTGER:  NOP
3179  014706  005037  000666                  CLR     PFLG            ;CLEAR PRINT FLAG
3180  014712  032777  020000  163650          BIT     #20000,@SWR     ;SEE IF SHOULD PRINT
3181  014720  001112                          BNE     LTGX            ;IF NOT: BR
3182  014722  005737  000620          LTGA:   TST     HDRFL           ;SEE IF PRINTED HEADER
3183  014726  001004                          BNE     LTGA1           ;IF SO: BR
3184  014730  013704  000622                  MOV     EMADDR,R4
3185  014734  004737  017334                  JSR     PC,TTOUT        ;PRINT TEST HEADER
3186  014740  012737  000001  000620  LTGA1:  MOV     #1,HDRFL        ;SET HEADER FLAG
3187  014746  013704  000672                  MOV     ERADD,R4
3188  014752  004737  017334                  JSR     PC,TTOUT        ;PRINT CONDITION ERROR
3189  014756  005737  000746                  TST     ADDFL
3190  014762  001003                          BNE     LTGA2
3191  014764  010103                          MOV     R1,R3
3192  014766  004737  017462                  JSR     PC,OCTP         ;PRINT ADDRESS
3193  014772  005737  000716          LTGA2:  TST     EXFL
3194  014776  001412                          BEQ     LTGC            ;IF NO STATUS: BR
3195  015000  012704  020720                  MOV     #MSG6,R4
3196  015004  022737  000001  000716          CMP     #1,EXFL         ;EXPT-NOT RCVD
3197  015012  001402                          BEQ     LTGB
3198  015014  012704  020737                  MOV     #MSG7,R4        ;RCVD-NOT EXPT
3199  015020  004737  017334          LTGB:   JSR     PC,TTOUT        ;PRINT STATUS
3200  015024  005237  000666          LTGC:   INC     PFLG
3201  015030  005737  000746                  TST     ADDFL           ;SEE IF ADD TST
3202  015034  001430                          BEQ     LTGD            ;IF SO: BR
3203  015036  005737  000744                  TST     T24FL           ;SEE IF TEST 24
3204  015042  001423                          BEQ     LTGC0           ;IF NOT: BR
3205  015044  012704  027542                  MOV     #WMSG27,R4
3206  015050  004737  017334                  JSR     PC,TTOUT        ;PRINT DATA TAG
3207  015054  012704  021240                  MOV     #MSG12,R4
3208  015060  004737  017334                  JSR     PC,TTOUT        ;PRINT EXPT TAG
3209  015064  012703  177777                  MOV     #-1,R3
3210  015070  004737  017452                  JSR     PC,OCTPE        ;PRINT EXPT
3211  015074  012704  021247                  MOV     #MSG13,R4
3212  015100  004737  017334                  JSR     PC,TTOUT        ;PRINT RCVD TAG
3213  015104  010103                          MOV     R1,R3           ;GET RCVD
3214  015106  004737  017452                  JSR     PC,OCTPE        ;PRINT RCVD
3215  015112  004737  015210          LTGC0:  JSR     PC,REGP         ;PRINT REGISTERS
3216  015116  032777  004000  163444  LTGD:   BIT     #4000,@SWR
3217  015124  001010                          BNE     LTGX
3218  015126  012704  021311                  MOV     #MSG16,R4
3219  015132  004737  017334                  JSR     PC,TTOUT
3220  015136  013703  000702                  MOV     ITCNT,R3        ;PRINT ITERATION
3221  015142  004737  017462                  JSR     PC,OCTP
3222  015146  005777  163416          LTGX:   TST     @SWR
3223  015152  100001                          BPL     LTGXA           ;IF NOT STOP ON ERROR: BR
3224  015154  000000                          HALT
3225  015156  005737  000666          LTGXA:  TST     PFLG
3226  015162  001004                          BNE     LTGXX           ;IF PRINTED: BR
3227  015164  032777  020000  163376          BIT     #20000,@SWR
3228  015172  001653                          BEQ     LTGA
```

```
 3229  015174  005037  000746          LTGXX:  CLR     ADDFL           ;CLEAR ADDRESS FLAG
 3230  015200  005037  000716                  CLR     EXFL
 3231  015204  000137  016224                  JMP     SCOPE
 3232
 3233                                          ;SUBROUTINE TO PRINT MAJOR REGISTERS**********
 3234
 3235  015210  000240                  REGP:   NOP
 3236  015212  012704  022252                  MOV     #MSG46,R4
 3237  015216  004737  017334                  JSR     PC,TTOUT        ;PRINT REGISTER HEADER
 3238  015222  017703  163262                  MOV     @C1,R3
 3239  015226  004737  017452                  JSR     PC,OCTPE
 3240  015232  017703  163254                  MOV     @WC,R3
 3241  015236  004737  017452                  JSR     PC,OCTPE
 3242  015242  017703  163246                  MOV     @BA,R3
 3243  015246  004737  017452                  JSR     PC,OCTPE
 3244  015252  017703  163240                  MOV     @FC,R3
 3245  015256  004737  017452                  JSR     PC,OCTPE
 3246  015262  017703  163232                  MOV     @CS,R3
 3247  015266  004737  017452                  JSR     PC,OCTPE
 3248  015272  017703  163224                  MOV     @DS,R3
 3249  015276  004737  017452                  JSR     PC,OCTPE                ;PRINT REGISTERS
 3250  015302  017703  163216                  MOV     @ER,R3
 3251  015306  004737  017452                  JSR     PC,OCTPE
 3252  015312  017703  163210                  MOV     @AS,R3
 3253  015316  004737  017452                  JSR     PC,OCTPE
 3254  015322  017703  163206                  MOV     @MR,R3
 3255  015326  004737  017452                  JSR     PC,OCTPE
 3256  015332  017703  163204                  MOV     @TC,R3
 3257  015336  004737  017452                  JSR     PC,OCTPE
 3258  015342  000207                  RTS     PC
 3259
 3260
```

```
3261                                        ;DRIVE CLEAR SUBROUTINE*********
3262
3263  015344  000240                DRVCLR: NOP
3264  015346  012704  040000                MOV     #40000,R4
3265  015352  005304                DCD:    DEC     R4
3266  015354  001376                        BNE     DCD             ;DELAY
3267  015356  005037  000666                CLR     PFLG
3268  015362  004737  015600                JSR     PC,ATTN         ;GO SEE OF ATTN SET
3269  015366  012777  000011  163114        MOV     #11,@C1         ;ISSUE DRIVE CLEAR
3270  015374  005000                        CLR     R0
3271  015376  032777  000200  163116  DCA:  BIT     #200,@DS        ;SEE IF DRY
3272  015404  001002                        BNE     DCA0
3273  015406  005300                        DEC     R0
3274  015410  001372                        BNE     DCA             ;WAIT FOR DRY
3275  015412  032777  040000  163102  DCA0: BIT     #40000,@DS      ;SEE IF ERR RESET
3276  015420  001022                        BNE     DCE             ;IF NOT: BR
3277  015422  005777  163076                TST     @ER             ;SEE IF ERROR REGISTER RESET
3278  015426  001017                        BNE     DCE             ;IF NOT: BR
3279  015430  005777  163066                TST     @DS             ;SEE IF ATA RESET
3280  015434  100414                        BMI     DCE             ;IF NOT: BR
3281  015436  012703  000001                MOV     #1,R3           ;SET TEST BIT
3282  015442  013704  000624                MOV     DRVN,R4         ;GET DRIVE NUMBER & BRANCH
3283  015446  001403                        BEQ     DCC             ;IF DRIVE 0
3284  015450  006303                DCB:    ASL     R3              ;POSITION TEST BIT PER DRIVE NUMBER
3285  015452  005304                        DEC     R4              ;SEE IF DONE
3286  015454  001375                        BNE     DCB             ;IF NOT: BR
3287  015456  030377  163044        DCC:    BIT     R3,@AS          ;SEE IF ATTEN IS RESET
3288  015462  001001                        BNE     DCE             ;IF NOT: BR
3289  015464  000207                        RTS     PC              ;RETURN
3290
3291  015466  000240                DCE:    NOP
3292  015470  032777  020000  163072        BIT     #20000,@SWR     ;SEE IF ERROR PRINT INHIBIT
3293  015476  001017                        BNE     DCEX            ;IF SO: BR
3294  015500  005737  000620                TST     HDRFL           ;SEE IF PRINT HEADER
3295  015504  001004                        BNE     DCEA            ;IF NOT: BR
3296  015506  013704  000622                MOV     EMADDR,R4
3297  015512  004737  017334                JSR     PC,TTOUT        ;PRINT HEADER
3298  015516  012704  022356        DCEA:   MOV     #MSG47,R4
3299  015522  004737  017334                JSR     PC,TTOUT        ;PRINT DRIVE CLEAR ERROR
3300  015526  004737  015210                JSR     PC,REGP         ;PRINT REGISTERS
3301  015532  005237  000666                INC     PFLG            ;SET PRINTED FLAG
3302  015536  005777  163026        DCEX:   TST     @SWR            ;SEE IF HALT ON ERROR
3303  015542  100001                        BPL     DCEXA           ;IF NOT: BR
3304  015544  000000                        HALT
3305  015546  005737  000666        DCEXA:  TST     PFLG            ;SEE IF HAVE PRINTED
3306  015552  001004                        BNE     DCEXX           ;IF SO: BR
3307  015554  032777  020000  163006        BIT     #20000,@SWR     ;BRANCH IF ERROR
3308  015562  001741                        BEQ     DCE             ;PRINTOUT DESIRED
3309  015564  000240                DCEXX:  NOP
3310  015566  012737  015344  000712        MOV     #DRVCLR,SCOLP   ;SET SCOPE LOOP ADDRESS
3311  015574  000137  016224                JMP     SCOPE           ;GO DO SCOPE LOOP
```

```
3312                                        ;COMPOSITE ERROR CHECK SUBROUTINE**********
3313
3314  015600  000240                ATTN:   NOP
3315  015602  005777  162714                TST     @DS             ;SEE IF ATA SET
3316  015606  001004                        BNE     ATTA            ;IF SO: BR
3317  015610  012737  021714  000700        MOV     #MSG32,TEMP3
3318  015616  000427                        BR      ATTP            ;ELSE PRINT ERROR
3319  015620  032777  040000  162674  ATTA:   BIT     #40000,@DS      ;SEE IF COMPOSITE ERROR SET
3320  015626  001004                        BNE     ATTB            ;IF SO: BR
3321  015630  012737  021676  000700        MOV     #MSG31,TEMP3
3322  015636  000417                        BR      ATTP            ;ELSE PRINT ERROR
3323  015640  012703  000001          ATTB:   MOV     #1,R3           ;SET TEST BIT
3324  015644  012737  021732  000700        MOV     #MSG33,TEMP3
3325  015652  013704  000624                MOV     DRVN,R4         ;GET DRIVE NUMBER & BRANCH
3326  015656  001403                        BEQ     ATTD            ;IF DRIVE 0
3327  015660  006303                ATTC:   ASL     R3              ;POSITION TEST BIT
3328  015662  005304                        DEC     R4              ;SEE IF DONE
3329  015664  001375                        BNE     ATTC            ;IF NOT: BR
3330  015666  030377  162634          ATTD:   BIT     R3,@AS          ;SEE IF ATTEN SUMMARY SET
3331  015672  001401                        BEQ     ATTP            ;IF NOT: BR
3332  015674  000207                        RTS     PC              ;ELSE RETURN
3333  015676  032777  020000  162664  ATTP:   BIT     #20000,@SWR     ;SEE IF PRINT INHIBIT
3334  015704  001021                        BNE     ATTX            ;IF SO: BR
3335  015706  005737  000620                TST     HDRFL           ;SEE IF DONE HEADER
3336  015712  001004                        BNE     ATTPA           ;IF SO: BR
3337  015714  013704  000622                MOV     EMADDR,R4
3338  015720  004737  017334                JSR     PC,TTOUT        ;PRINT HEADER
3339  015724  013704  000700          ATTPA:  MOV     TEMP3,R4
3340  015730  004737  017334                JSR     PC,TTOUT        ;PRINT ERROR TYPE
3341  015734  004737  015210                JSR     PC,REGP         ;PRINT REGISTERS
3342  015740  005237  000666                INC     PFLG            ;SET PRINT FLAG
3343  015744  005237  000620                INC     HDRFL           ;SET HEADER FLAG
3344  015750  005777  162614          ATTX:   TST     @SWR            ;SEE IF HALT ON ERROR
3345  015754  100001                        BPL     ATTXA           ;IF NOT: BR
3346  015756  000000                        HALT
3347  015760  005737  000666          ATTXA:  TST     PFLG            ;SEE IF DONE PRINT
3348  015764  001004                        BNE     ATTXX           ;IF SO: BR
3349  015766  032777  020000  162574        BIT     #20000,@SWR     ;BRANCH IF NO ERROR
3350  015774  001740                        BEQ     ATTP            ;PRINTOUT DESIRED
3351  015776  005037  000666          ATTXX:  CLR     PFLG            ;CLEAR PRINT FLAG
3352  016002  000207                        RTS     PC              ;RETURN
```

```
                                                ;LOGIC TEST REGISTER BIT ERROR SUBROUTIINE**********

016004  012737  000001  000736  LTGER2: MOV     #1,PEXFL        ;SET FLAG
016012  000240                  LTGER1: NOP
016014  005037  000666                  CLR     PFLG            ;CLEAR PRINT FLAG
016020  032777  020000  162542          BIT     #20000,@SWR     ;BRANCH IF ERROR
016026  001055                          BNE     LTG1X           ;PRINTOUT DESIRED
016030  005737  000620          LTG1A:  TST     HDRFL           ;SEE IF PRINT HEADER
016034  001004                          BNE     LTG1B           ;IF NOT: BR
016036  013704  000622                  MOV     EMADDR,R4
016042  004737  017334                  JSR     PC,TTOUT        ;PRINT HEADER
016046  012737  000001  000620  LTG1B:  MOV     #1,HDRFL        ;SET FLAG
016054  013704  000672                  MOV     ERADD,R4
016060  004737  017334                  JSR     PC,TTOUT        ;PRINT ERROR CODE
016064  005737  000736                  TST     PEXFL           ;SEE IF PRINT EXPT-RCVD
016070  001016                          BNE     LTG1T           ;IF NOT: BR
016072  012704  021240                  MOV     #MSG12,R4
016076  004737  017334                  JSR     PC,TTOUT        ;PRINT EXPT TAG
016102  010103                          MOV     R1,R3
016104  004737  017462                  JSR     PC,OCTP         ;PRINT EXPT
016110  012704  021247                  MOV     #MSG13,R4
016114  004737  017334                  JSR     PC,TTOUT        ;PRINT RCVD TAG
016120  010203                          MOV     R2,R3
016122  004737  017462                  JSR     PC,OCTP         ;PRINT RCVD
016126  032777  004000  162434  LTG1T:  BIT     #4000,@SWR
016134  001010                          BNE     LTG1C
016136  012704  021311                  MOV     #MSG16,R4
016142  004737  017334                  JSR     PC,TTOUT
016146  013703  000702                  MOV     ITCNT,R3
016152  004737  017462                  JSR     PC,OCTP         ;PRINT ITERATION
016156  005237  000666          LTG1C:  INC     PFLG
016162  000240                  LTG1X:  NOP
016164  005777  162400                  TST     @SWR
016170  100001                          BPL     LTG1X1          ;IF NOT STOP ON ERROR: BR
016172  000000                          HALT
016174  005737  000666          LTG1X1: TST     PFLG
016200  001004                          BNE     LTG1XX          ;IF HAVE PRINTED: BR
016202  032777  020000  162360          BIT     #20000,@SWR
016210  001707                          BEQ     LTG1A
016212  000240                  LTG1XX: NOP
016214  005037  000736                  CLR     PEXFL           ;CLEAR EXPT-RCVD FLAG
016220  000137  016224                  JMP     SCOPE           ;GO TO SCOPE


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

016224  000240          SCOPE:  NOP
016226  032777  040000  162334          BIT     #40000,@SWR     ;SEE IF LOOP ON ERROR
016234  001001                          BNE     1$              ;IF SO: BR
016236  000207                          RTS     PC              ;ELSE EXIT
016240  000240          1$:     NOP
016242  005726                          TST     (SP)+           ;RESET STACK
016244  000240                          NOP
016246  000240                          NOP
016250  000177  162436                  JMP     @SCOLP          ;LOOP ON ERROR
```

```
3409                                            ;TEST ITERATION SUBROUTINE**********
3410
3411  016254  032777  004000  162306  ITER:     BIT     #4000,@SWR      ;SEE IF ITERATIONS
3412  016262  001403                            BEQ     2$              ;IF SO: BR
3413  016264  005037  000702          1$:       CLR     ITCNT           ;CLEAR ITERATION COUNTER
3414  016270  000207                            RTS     PC              ;ELSE EXIT
3415  016272  005737  001016          2$:       TST     PCNTR           ;NO SUBTEST ITERATIONS ON FIRST PASS
3416  016276  001772                            BEQ     1$
3417  016300  005237  000702                    INC     ITCNT           ;BUMP COUNTER
3418  016304  023737  000702  000606            CMP     ITCNT,ITAMT     ;SEE IF DONE ALL
3419  016312  001764                            BEQ     1$              ;IF SO: BR
3420  016314  005726                            TST     (SP)+           ;RESET STACK
3421  016316  017700  162372                    MOV     @ITRLP,R0       ;SET ITERATION POINTER
3422  016322  000110                            JMP     (R0)            ;GO ITERATE
3423
3424                                            ;MANUAL INTERVENTION INHIBIT*********
3425
3426  016324  000240          INMT:             NOP
3427  016326  012704  022102                    MOV     #MSG43,R4
3428  016332  004737  017334                    JSR     PC,TTOUT        ;GO PRINT INHIB MSG
3429  016336  000000                            HALT
3430  016340  000137  002344                    JMP     TSCD2           ;RETURN TO SCHED
3431
```

```
3432
3433                                        ;INITIALIZE SUBROUTINE**********
3434
3435  016344  000240                  INIT1:  NOP
3436  016346  012777  000040  162144          MOV     #40,@CS         ;INIT
3437  016354  013777  000624  162136  INIT2:  MOV     DRVN,@CS        ;SELECT DRIVE
3438  016362  013777  000664  162152          MOV     SLVN,@TC        ;SELECT SLAVE
3439  016370  000207                          RTS     PC              ;RETURN
3440
3441                                        ;ROUTINES TO INITIALIZE SLAVE. THESE ROUTINES PLACE THE SLAVE
3442                                        ;IN PROPER STATUS FOR THE CALLING TEST. INIT3 PLACES THE SLAVE IN
3443                                        ;NRZ MODE AND OFF BOT; INIT4 PLACES THE SLAVE IN PE MODE AND OFF
3444                                        ;BOT. IF THE SLAVE IS IN THE PROPER STATUS ON ENTRY NO ACTION IS TAKEN.
3445
3446                                        ;SET SLAVE IN NRZ OFF BOT
3447  016372  013746  000776          INIT3:  MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESCRIPTION
3448  016376  012737  001400  000776          MOV     #1400,UDES      ;SET UNIT DESCRIPTION = NRZ
3449  016404  000410                          BR      INIT5           ;GO TO INIT5 ROUTINE
3450
3451                                        ;SET SLAVE IN PE OFF BOT
3452  016406  013746  000776          INIT4:  MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESRIPTION
3453  016412  012737  002000  000776          MOV     #2000,UDES      ;SET UNIT DESRIPTION = PE
3454  016420  000402                          BR      INIT5           ;GO DO IT
3455
3456                                        ;THIS ROUTINE IS ENTERED AT INIT WHEN THE CALLER HAS SETUP UDES.
3457                                        ;IT IS ENTERED AT INIT5 WHEN EITHER INIT3 OR INIT4 HAS SET UP UDES.
3458  016422  013746  000776          INIT:   MOV     UDES,-(SP)      ;SAVE TEST'S UNIT DESRIPTION
3459  016426  012777  000040  162064  INIT5:  MOV     #40,@CS         ;INIT CONTROLLER
3460  016434  013777  000624  162056          MOV     DRVN,@CS        ;SELECT TM03 DRIVE
3461  016442  013777  000664  162072          MOV     SLVN,@TC        ;SELECT TU45 SLAVE
3462  016450  013746  000776                  MOV     UDES,-(SP)      ;GET SLAVE DESCRIPTION
3463  016454  042716  174377                  BIC     #174377,(SP)    ;CLEAR ALL BUT DENSITY SELECT BITS
3464  016460  022726  001400                  CMP     #1400,(SP)+     ;BRANCH IF REQUESTING PE MODE
3465  016464  001005                          BNE     1$
3466  016466  032777  000040  162026          BIT     #40,@DS         ;BRANCH IF SLAVE IS IN NRZ MODE
3467  016474  001420                          BEQ     4$              ;(PES = 0)
3468  016476  000404                          BR      2$
3469  016500  032777  000040  162014  1$:     BIT     #40,@DS         ;BRANCH IF SLAVE IS IN PE MODE
3470  016506  001013                          BNE     4$
3471  016510  012777  000007  161772  2$:     MOV     #7,@C1          ;REWIND SLAVE
3472  016516  032777  000200  161776  20$:    BIT     #200,@DS        ;WAIT FOR READY
3473  016524  001774                          BEQ     20$
3474  016526  032777  020000  161766  3$:     BIT     #20000,@DS      ;WAIT UNTIL PIP CLEARS
3475  016534  001374                          BNE     3$
3476  016536  053777  000776  161776  4$:     BIS     UDES,@TC        ;LOAD SLAVE DESRIPTION
3477  016544  032777  000002  161750          BIT     #2,@DS          ;BRANCH IF NOT AT BOT
3478  016552  001407                          BEQ     6$
3479  016554  012777  000025  161726          MOV     #25,@C1         ;ERASE TO GET OFF BOT
3480  016562  032777  000200  161732  5$:     BIT     #200,@DS        ;WAIT FOR READY
3481  016570  001774                          BEQ     5$
3482  016572  012777  000011  161710  6$:     MOV     #11,@C1         ;RESET DRIVE
3483  016600  012637  000776                  MOV     (SP)+,UDES      ;RESTORE UNIT DESRIPTION
3484  016604  000207                          RTS     PC              ;RETURN
3485
3486
3487
```

```
3488                                    ;MANUAL INSTRUCTION SUBROUTINE**********
3489
3490  016606  000240            INST:   NOP
3491  016610  004737  017334            JSR     PC,TTOUT        ;PRINT INSTRUCTION
3492  016614  012704  025723            MOV     #MMSG0,R4
3493  016620  004737  017334            JSR     PC,TTOUT        ;PRINT REPLY
3494  016624  012705  000700            MOV     #TEMP3,R5
3495  016630  012701  000001            MOV     #1,R1
3496  016634  012702  177777            MOV     #-1,R2
3497  016640  012703  000000            MOV     #0,R3
3498  016644  004737  017012            JSR     PC,TTR          ;AWAIT REPLY
3499  016650  000240                    NOP
3500  016652  000207                    RTS     PC              ;EXIT
3501
3502                                    ;MAG TAPE INTERRUPT HANDLER**********
3503
3504  016654  000240            MTINT:  NOP
3505  016656  013716  000670            MOV     RTRN,(SP)       ;SET RETURN FROM INTERUPT ADDRESS
3506  016662  000002                    RTI                     ;RETURN
3507
3508                                    ;TTY INTERRUPT HANDLER**********
3509
3510  016664  017746  161704    TTINT:  MOV     @TKB,-(SP)      ;GET CHARACTER
3511  016670  042716  000200            BIC     #200,(SP)       ;CLEAR PARITY BIT
3512  016674  122716  000003            CMPB    #3,(SP)         ;BRANCH IF NOT CONTROL C
3513  016700  001010                    BNE     1$
3514  016702  005737  001416            TST     CHNFLG          ;INHIBIT  C IF IN CHAIN MODE
3515  016706  001005                    BNE     1$
3516  016710  005077  161652            CLR     @PSW            ;CLEAR PSW
3517  016714  000005                    RESET
3518  016716  000137  000200            JMP     @#200           ;RESTART
3519  016722  122716  000001    1$:     CMPB    #1,(SP)         ;BRANCH IF NOT  A
3520  016726  001017                    BNE     2$
3521  016730  022737  000176  000570    CMP     #SWREG,SWR      ;BRANCH IF USING HARWARE SWR
3522  016736  001016                    BNE     3$
3523  016740  012737  177570  000570    MOV     #177570,SWR     ;INVOKE HARDWARE SWR
3524  016746  004737  020206            JSR     PC,.SAVE        ;SAVE REGISTERS ON THE STACK
3525  016752  012704  022704            MOV     #MSG63,R4       ;TYPE 'HARDWARE SWR IN USE'
3526  016756  004737  017334            JSR     PC,TTOUT
3527  016762  004737  020230            JSR     PC,.RESTORE
3528  016766  122716  000007    2$:     CMPB    #7,(SP)         ;BRANCH IF NOT  G
3529  016772  001005                    BNE     4$
3530  016774  012737  000176  000570    MOV     #SWREG,SWR      ;INVOKE SOFTWARE SWR
3531  017002  004737  020110    3$:     JSR     PC,GTSWR        ;GET SWITCHES
3532  017006  005726            4$:     TST     (SP)+           ;POP CHARACTER OFF STACK
3533  017010  000002                    RTI                     ;RETURN
3534
```

```
3535                                    ;*********************************************************
3536                                    ;TTY ENTRY SUBROUTINE:
3537                                    ;
3538                                    ;THIS SUBROUTINE IS USED BY THE TEST CONDITION
3539                                    ;ENTRY ROUTINE TO READ THE RESPONSE ENTERED
3540                                    ;AT THE TTY AND CHECK THEM FOR LEGALITY AND
3541                                    ;LIMITS.  ALL RESPONSE MUST BE TYPED IN OCTAL
3542                                    ;(0-7) AND MUST FALL WITHIN THE LIMITS SET BY
3543                                    ;THE CALLING ROUTINE.
3544                                    ;IF AN ENTRY IS ILLEGAL OR OUTSIDE THE LIMITS,
3545                                    ;A QUESTION MARK IS TYPED (?) AND THE RESPONSE
3546                                    ;MAY BE REENTERED.
3547                                    ;ENTRIES MAY NOT EXCEED SIX (6) CHARACTERS AND
3548                                    ;MAY BE TERMINATED AT LESS THAN SIX BY TYPING A
3549                                    ;CARRIAGE RETURN
3550                                    ;*********************************************************
3551
3552  017012  010146            TTR:   MOV     R1,-(SP)        ;SAVE CHARACTER COUNT
3553  017014  011601            10$:   MOV     (SP),R1         ;RESTORE CHARACTER COUNT (FOR  U)
3554  017016  005037  000674           CLR     TEMP1           ;CLEAR FIRST CHARACTER FLAG
3555  017022  005000                   CLR     R0
3556  017024  004737  017272    1$:    JSR     PC,TTIN         ;GO READ CHARACTER
3557  017030  122737  000003  000616   CMPB    #3,TIB          ;BRANCH IF NOT  C
3558  017036  001003                   BNE     11$
3559  017040  000005                   RESET                   ;RESET
3560  017042  000137  000200           JMP     @#200           ;RESTART PROGRAM
3561  017046  122737  000015  000616  11$:    CMPB    #15,TIB         ;SEE IF CR
3562  017054  001004                   BNE     2$              ;IF NOT: BR
3563  017056  005737  000674           TST     TEMP1           ;SEE IF FIRST CHARACTER
3564  017062  001471                   BEQ     9$              ;IF SO: BR
3565  017064  000457                   BR      6$
3566  017066  122737  000025  000616  2$:     CMPB    #25,TIB         ;BRANCH IF NOT CONTROL U
3567  017074  001005                   BNE     21$
3568  017076  012704  022632           MOV     #MSG59,R4       ;TYPE <CR><LF>
3569  017102  004737  017334           JSR     PC,TTOUT
3570  017106  000742                   BR      10$
3571  017110  122737  000177  000616  21$:    CMPB    #177,TIB        ;BRANCH IF NOT 'RUBOUT'
3572  017116  001012                   BNE     3$
3573  017120  000241                   CLC
3574  017122  006000                   ROR     R0              ;REMOVE LAST CHAR
3575  017124  006200                   ASR     R0
3576  017126  006200                   ASR     R0
3577  017130  012704  022634           MOV     #MSG60,R4       ;TYPE ' '
3578  017134  004737  017334           JSR     PC,TTOUT
3579  017140  005201                   INC     R1              ;DECREMENT CHARS RECEIVED COUNT
3580  017142  000730                   BR      1$
3581  017144  122737  000060  000616  3$:     CMPB    #60,TIB         ;SEE IF CHAR IS LESS THAN 0
3582  017152  101402                   BLOS    4$              ;IF NOT: BR
3583  017154  000137  017252           JMP     TINER           ;ELSE GO TO ERROR
3584  017160  122737  000070  000616  4$:     CMPB    #70,TIB         ;SEE IF CHAR IS GREATER THAN 7
3585  017166  101002                   BHI     5$              ;IF NOT: BR
3586  017170  000137  017252           JMP     TINER           ;ELSE GO TO ERROR
3587  017174  005237  000674           INC     TEMP1           ;SET FIRST CHARACTER FLAG
3588  017200  006300                   ASL     R0
3589  017202  006300                   ASL     R0              ;SHIFT 3 LEFT
3590  017204  006300                   ASL     R0
```

```
 3591  017206  042737  177770  000616          BIC     #177770,TIB     ;STRIP ASCII
 3592  017214  053700  000616                  BIS     TIB,R0          ;LOAD CHARACTER
 3593  017220  005301                          DEC     R1              ;SEE IF DONE
 3594  017222  001300                          BNE     1$              ;IF NOT: BR
 3595  017224  020002                  6$:     CMP     R0,R2           ;SEE IF EXCEEDED MAXIMUM LIMIT
 3596  017226  101402                          BLOS    7$              ;IF NOT: BR
 3597  017230  000137  017252                  JMP     TINER           ;ELSE GO TO ERROR
 3598  017234  020300                  7$:     CMP     R3,R0           ;SEE IF BELOW MINIMUM LIMIT
 3599  017236  101402                          BLOS    8$              ;IF NOT: BR
 3600  017240  000137  017252                  JMP     TINER           ;ELSE GO TO ERROR
 3601  017244  010015                  8$:     MOV     R0,(R5)         ;LOAD VALUE
 3602  017246  005726                  9$:     TST     (SP)+           ;POP CHAR COUNT OFF STACK
 3603  017250  000207                          RTS     PC              ;EXIT
 3604
```

```
3605
3606                                    ;TTY ENTRY ERROR SUBROUTINE**********
3607
3608  017252  012704  022042    TINER:  MOV     #MSG40,R4
3609  017256  004737  017334            JSR     PC,TTOUT        ;PRINT?
3610  017262  005726                    TST     (SP)+           ;POP CHAR COUNT OFF STACK
3611  017264  162716  000020            SUB     #20,(SP)        ;RESET SP TO START OF VALUE ROUTINE
3612  017270  000207                    RTS     PC              ;REDO VALUE ENTRY
3613
3614                                    ;TTY READ SUBROUTINE**********
3615
3616  017272  005277  161274    TTIN:   INC     @TKS
3617  017276  105777  161270    1$:     TSTB    @TKS
3618  017302  100375                    BPL     1$
3619  017304  017737  161264  000616    MOV     @TKB,TIB
3620  017312  042737  000200  000616    BIC     #200,TIB                ;STRIP PARITY BIT
3621  017320  013737  000616  000614    MOV     TIB,TOB                 ;MOVE CHAR TO TTY OUTPUT BFR
3622  017326  004737  017434            JSR     PC,TOG                  ;AND ECHO IT
3623  017332  000207                    RTS     PC
3624
3625                                    ;TTY OUTPUT SUBROUTINE***********
3626
3627  017334  112437  000614    TTOUT:  MOVB    (R4)+,TOB
3628  017340  122737  000043  000614    CMPB    #43,TOB
3629  017346  001440                    BEQ     TEX
3630  017350  122737  000045  000614    CMPB    #45,TOB
3631  017356  001403                    BEQ     1$
3632  017360  004737  017434            JSR     PC,TOG
3633  017364  000763                    BR      TTOUT
3634  017366  112737  000015  000614  1$:     MOVB    #15,TOB
3635  017374  004737  017434            JSR     PC,TOG
3636  017400  012703  000004            MOV     #4,R3
3637  017404  005037  000614    2$:     CLR     TOB
3638  017410  004737  017434            JSR     PC,TOG
3639  017414  005303                    DEC     R3
3640  017416  001372                    BNE     2$                      ;DO FILLERS
3641  017420  112737  000012  000614    MOVB    #12,TOB
3642  017426  004737  017434            JSR     PC,TOG
3643  017432  000740                    BR      TTOUT
3644  017434  105777  161136    TOG:    TSTB    @TPS
3645  017440  100375                    BPL     TOG
3646  017442  113777  000614  161130    MOVB    TOB,@TPB
3647  017450  000207            TEX:    RTS     PC
3648
3649
3650                                    ;OCTAL OUTPUT SUBROUTINE***********
3651
3652  017452  012737  000001  017702  OCTPE:  MOV     #1,OFL
3653  017460  000402                    BR      OCTPE1
3654  017462  005037  017702    OCTP:   CLR     OFL             ;CLEAR FLAG FOR LEADING ZERO
3655  017466  010304            OCTPE1: MOV     R3,R4           ;SEE IF NUMBER IS ZERO
3656  017470  001006                    BNE     OCTP0           ;IF NOT ZERO: BR
3657  017472  005737  017702            TST     OFL             ;SEE IF PRINT ALL 0
3658  017476  001003                    BNE     OCTP0           ;IF SO: BR
3659  017500  004737  017662            JSR     PC,OCTPG1       ;ELSE PRINT ZERO
3660  017504  000447                    BR      OCTP3           ;SPACE AND EXIT
```

```
 3661  017506  032704  100000          OCTP0:  BIT     #100000,R4      ;SEE IF MSD = 1
 3662  017512  001405                          BEQ     OCTP1           ;IF NOT: BR
 3663  017514  012704  000001                  MOV     #1,R4
 3664  017520  004737  017640                  JSR     PC,OCTPG        ;PRINT 1
 3665  017524  000403                          BR      OCTP2
 3666  017526  005004                  OCTP1:  CLR     R4
 3667  017530  004737  017640                  JSR     PC,OCTPG        ;PRINT 0
 3668  017534  010304                  OCTP2:  MOV     R3,R4
 3669  017536  006004                          ROR     R4
 3670  017540  006004                          ROR     R4
 3671  017542  006004                          ROR     R4              ;POSITION DIGIT
 3672  017544  006004                          ROR     R4
 3673  017546  000304                          SWAB    R4
 3674  017550  004737  017640                  JSR     PC,OCTPG        ;PRINT DIGIT 2
 3675  017554  010304                          MOV     R3,R4
 3676  017556  006004                          ROR     R4
 3677  017560  000304                          SWAB    R4
 3678  017562  004737  017640                  JSR     PC,OCTPG        ;PRINT DIGIT 3
 3679  017566  010304                          MOV     R3,R4
 3680  017570  006104                          ROL     R4
 3681  017572  006104                          ROL     R4
 3682  017574  000304                          SWAB    R4
 3683  017576  004737  017640                  JSR     PC,OCTPG        ;PRINT DIGIT 4
 3684  017602  010304                          MOV     R3,R4
 3685  017604  006004                          ROR     R4
 3686  017606  006004                          ROR     R4
 3687  017610  006004                          ROR     R4
 3688  017612  004737  017640                  JSR     PC,OCTPG
 3689  017616  010304                          MOV     R3,R4
 3690  017620  004737  017640                  JSR     PC,OCTPG        ;PRINT DIGIT 5
 3691  017624  012737  000240  000614  OCTP3:  MOV     #240,TOB
 3692  017632  004737  017434                  JSR     PC,TOG          ;PRINT SPACE
 3693  017636  000207                          RTS     PC              ;EXIT
 3694
 3695  017640  042704  177770          OCTPG:  BIC     #177770,R4
 3696  017644  001004                          BNE     OCTPG0
 3697  017646  005737  017702                  TST     OFL
 3698  017652  001001                          BNE     OCTPG0
 3699  017654  000207                          RTS     PC
 3700
 3701  017656  005237  017702          OCTPG0: INC     OFL
 3702  017662  052704  000260          OCTPG1: BIS     #260,R4
 3703  017666  010437  000614                  MOV     R4,TOB
 3704  017672  004737  017434                  JSR     PC,TOG
 3705  017676  010304                          MOV     R3,R4
 3706  017700  000207                          RTS     PC
 3707  017702  000000                  OFL:    0                       ;FIRST CHAR FLAG
 3708
```

```
3709
3710                                    ;DATA CHARACTER OUTPUT SUBROUTINE************
3711
3712  017704  012704  000010          DOUT:   MOV     #10,R4          ;SET NUMBER TO PRINT
3713  017710  110337  000614                  MOVB    R3,TOB
3714  017714  105777  160656          1$:     TSTB    @TPS
3715  017720  100375                          BPL     1$
3716  017722  132737  000200  000614        BITB    #200,TOB
3717  017730  001404                        BEQ     2$
3718  017732  012777  000061  160640        MOV     #061,@TPB
3719  017740  000403                        BR      3$
3720  017742  012777  000060  160630  2$:   MOV     #060,@TPB
3721  017750  006337  000614          3$:   ASL     TOB
3722  017754  005304                        DEC     R4
3723  017756  001356                        BNE     1$
3724  017760  000207                        RTS     PC
3725
3726  017762  013703  000700          DOUTD:  MOV     TEMP3,R3
3727  017766  000303                        SWAB    R3
3728  017770  004737  017704                JSR     PC,DOUT
3729  017774  013703  000700                MOV     TEMP3,R3
3730  020000  004737  017704                JSR     PC,DOUT
3731  020004  000207                        RTS     PC
3732
3733                                    ;TU45 SERIAL NUMBER PRINT SUBROUTINE**********
3734
3735  020006  010304          SNPT:   MOV     R3,R4
3736  020010  000304                  SWAB    R4
3737  020012  006004                  ROR     R4
3738  020014  006004                  ROR     R4
3739  020016  006004                  ROR     R4
3740  020020  006004                  ROR     R4              ;GET FIRST DIGIT
3741  020022  004737  020064          JSR     PC,SNPG         ;PRINT
3742  020026  010304                  MOV     R3,R4
3743  020030  000304                  SWAB    R4              ;GET SECOND DIGIT
3744  020032  004737  020064          JSR     PC,SNPG         ;PRINT
3745  020036  010304                  MOV     R3,R4
3746  020040  006004                  ROR     R4
3747  020042  006004                  ROR     R4
3748  020044  006004                  ROR     R4
3749  020046  006004                  ROR     R4
3750  020050  004737  020064          JSR     PC,SNPG         ;PRINT THIRD DIGIT
3751  020054  010304                  MOV     R3,R4
3752  020056  004737  020064          JSR     PC,SNPG         ;PRINT FOURTH DIGIT
3753  020062  000207                  RTS     PC              ;EXIT
3754  020064  012737  000260  000614  SNPG:  MOV     #260,TOB        ;SET BASE = 0
3755  020072  042704  177760          BIC     #177760,R4      ;MASK DIGIT
3756  020076  050437  000614          BIS     R4,TOB          ;SET ASCII
3757  020102  004737  017434          JSR     PC,TOG          ;TYPE DIGIT
3758  020106  000207                  RTS     PC              ;RETURN
```

```
3759
3760                                    ;ROUTINE TO LOAD CONTENTS OF SOFTWARE SWITCH REGISTER.
3761                                    ;IF A CONTROL G ( G) IS TYPED THE SOFTWARE SWITCH REGISTER IS LOADED
3762  020110  022737  000176  000570    GTSWR:  CMP     #SWREG,SWR      ;BRANCH IF SOFTWARE SWR
3763  020116  001032                            BNE     1$              ;NOT INVOKED
3764  020120  004737  020206                    JSR     PC,..SAVE       ;SAVE REGISTERS ON THE STACK
3765  020124  012704  026431                    MOV     #$MSWR,R4       ;TYPE 'SWR = '
3766  020130  004737  017334                    JSR     PC,TTOUT
3767  020134  017703  160430                    MOV     @SWR,R3         ;GET CURRENT VALUE
3768  020140  004737  017452                    JSR     PC,OCTPE        ;AND TYPE IT
3769  020144  012704  026441                    MOV     #$MNEW,R4       ;ASK FOR NEW VALUE
3770  020150  004737  017334                    JSR     PC,TTOUT
3771  020154  013705  000570                    MOV     SWR,R5          ;NEW VALUE WILL BE RETURNED IN (R5)
3772  020160  012701  000007                    MOV     #7,R1           ;LIMIT TO 7 CHARACTERS
3773  020164  012702  177777                    MOV     #177777,R2      ;LIMIT RESPONSE TO BETWEEN
3774  020170  012703  000000                    MOV     #0,R3           ;0 AND 177777
3775  020174  004737  017012                    JSR     PC,TTR          ;GET RESPONSE
3776  020200  004737  020230                    JSR     PC,..RESTORE    ;RESTORE REGISTERS
3777  020204  000207                    1$:     RTS     PC              ;RETURN TO CALLER
3778                                    ;;ROUTINE TO SAVE REGISTERS ON THE STACK
3779  020206  010546                    .SAVE:  MOV     %5,-(SP)        ;;R5 IS SAVED AT 12(SP)
3780  020210  010446                            MOV     %4,-(SP)        ;;R4 IS SAVED AT 10(SP)
3781  020212  010346                            MOV     %3,-(SP)        ;;R3 IS SAVED AT 6(SP)
3782  020214  010246                            MOV     %2,-(SP)        ;;R2 IS SAVED AT 4(SP)
3783  020216  010146                            MOV     %1,-(SP)        ;;R1 IS SAVED AT 2(SP)
3784  020220  010046                            MOV     %0,-(SP)        ;;R0 IS SAVED AT (SP)
3785  020222  016646  000014                    MOV     14(SP),-(SP)    ;;PUSH RETURN PC ON THE STACK
3786  020226  000207                            RTS     PC              ;;RETURN TO CALLER
3787
3788                                    ;;ROUTINE TO RESTORE REGISTERS SAVED ON THE STACK
3789  020230  012666  000014            .RESTORE:MOV    (SP)+,14(SP)    ;;STORE RETURN PC ON STACK
3790  020234  012600                            MOV     (SP)+,%0
3791  020236  012601                            MOV     (SP)+,%1
3792  020240  012602                            MOV     (SP)+,%2
3793  020242  012603                            MOV     (SP)+,%3
3794  020244  012604                            MOV     (SP)+,%4
3795  020246  012605                            MOV     (SP)+,%5
3796  020250  000207                            RTS     PC              ;;RETURN
3797
3798
3799                                            ;MESSAGE TABLE**********
3800
3801  020252  022445  046524  031460  MSG1:    .ASCII  /%%TM03-TU45 CONTROL LOGIC TEST- PART I (CZTUOA0)/
3802  020260  052055  032125  020065
3803  020266  047503  052116  047522
3804  020274  020114  047514  044507
3805  020302  020103  042524  052123
3806  020310  020055  040520  052122
3807  020316  044440  024040  055103
3808  020324  052524  040517  024460
3809  020332  025045  025052  051501            .ASCII  /%***ASSURE TAPE IS AT BOT***/
3810  020340  052523  042522  052040
3811  020346  050101  020105  051511
3812  020354  040440  020124  047502
3813  020362  025124  025052
3814  020366  052045  050131  020105            .ASCII  /%TYPE <CR> TO TERMINATE RESPONSE &  C TO RESTART%#/
```

```
 3815  020374  041474  037122  052040
 3816  020402  020117  042524  046522
 3817  020410  047111  052101  020105
 3818  020416  042522  050123  047117
 3819  020424  042523  023040  057040
 3820  020432  020103  047524  051040
 3821  020440  051505  040524  052122
 3822  020446  021445
 3823  020450  042045  044522  042526  MSG2:   .ASCII  /%DRIVE NUMBER OR (CR) WHEN DONE #/
 3824  020456  047040  046525  042502
 3825  020464  020122  051117  024040
 3826  020472  051103  020051  044127
 3827  020500  047105  042040  047117
 3828  020506  020105     043
 3829  020511     045  043045  051117  MSG2A:  .ASCII  /%%FOR DRIVE ADDRESS TEST;/
 3830  020516  042040  044522  042526
 3831  020524  040440  042104  042522
 3832  020532  051523  052040  051505
 3833  020540  035524
 3834  020542  020045  047105  042524          .ASCII  /% ENTER EXPT DRIVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
 3835  020550  020122  054105  052120
 3836  020556  042040  044522  042526
 3837  020564  047040  046525  042502
 3838  020572  026122  040440  046114
 3839  020600  047440  044124  051105
 3840  020606  020123  044123  052517
 3841  020614  042114  041040  020105
 3842  020622  047516  026516  054105
 3843  020630  051511  040524  052116
 3844  020636  021456
 3845  020640  047045  047117  042455  MSG3:   .ASCII  /%NON-EXIST DRIVE #/
 3846  020646  044530  052123  042040
 3847  020654  044522  042526  021440
 3848  020662  051045  020110  042504  MSG4:   .ASCII  /%RH DETECTED #/
 3849  020670  042524  052103  042105
 3850  020676  021440
 3851  020700  052045  030115  020063  MSG5:   .ASCII  /%TM03 DETECTED #/
 3852  020706  042504  042524  052103
 3853  020714  042105  021440
 3854  020720  054105  052120  047055  MSG6:   .ASCII  /EXPT-NOT RECVD#/
 3855  020726  052117  051040  041505
 3856  020734  042126     043
 3857  020737     122  053103  026504  MSG7:   .ASCII  /RCVD-NOT EXPT#/
 3858  020744  047516  020124  054105
 3859  020752  052120     043
 3860  020755     045  046123  053101  MSG8:   .ASCII  /%SLAVE NUMBER OR (CR) WHEN DONE #/
 3861  020762  020105  052516  041115
 3862  020770  051105  047440  020122
 3863  020776  041450  024522  053440
 3864  021004  042510  020116  047504
 3865  021012  042516  021440
 3866  021016  022445  047506  020122  MSG8A:  .ASCII  /%%FOR SLAVE ADDRESS TEST;/
 3867  021024  046123  053101  020105
 3868  021032  042101  051104  051505
 3869  021040  020123  042524  052123
 3870  021046     073
```

```
 3871  021047     045  042440  052116          .ASCII  /% ENTER EXPT SLAVE NUMBER, ALL OTHERS SHOULD BE NON-EXISTANT.#/
 3872  021054  051105  042440  050130
 3873  021062  020124  046123  053101
 3874  021070  020105  052516  041115
 3875  021076  051105  020054  046101
 3876  021104  020114  052117  042510
 3877  021112  051522  051440  047510
 3878  021120  046125  020104  042502
 3879  021126  047040  047117  042455
 3880  021134  044530  052123  047101
 3881  021142  027124     043
 3882  021145     045  047516  026516  MSG9:   .ASCII  /%NON-EXIST SLAVE #/
 3883  021152  054105  051511  020124
 3884  021160  046123  053101  020105
 3885  021166     043
 3886  021167     045  042522  042101  MSG10:  .ASCII  /%READ CONT BUS PAR #/
 3887  021174  041440  047117  020124
 3888  021202  052502  020123  040520
 3889  021210  020122     043
 3890  021213     045  051127  052111  MSG11:  .ASCII  /%WRITE CONT BUS PAR #/
 3891  021220  020105  047503  052116
 3892  021226  041040  051525  050040
 3893  021234  051101  021440
 3894  021240  042440  050130  020124  MSG12:  .ASCII  / EXPT #/
 3895  021246     043
 3896  021247     040  041522  042126  MSG13:  .ASCII  / RCVD #/
 3897  021254  021440
 3898  021256  046445  020122  044502  MSG14:  .ASCII  /%MR BITS 4-0#/
 3899  021264  051524  032040  030055
 3900  021272     043
 3901  021273     045  051115  041040  MSG15:  .ASCII  /%MR BITS 15-7#/
 3902  021300  052111  020123  032461
 3903  021306  033455     043
 3904  021311     045  052111  051105  MSG16:  .ASCII  /%ITER: #/
 3905  021316  020072     043
 3906  021321     045  041524  041040  MSG18:  .ASCII  /%TC BITS 12-0 #/
 3907  021326  052111  020123  031061
 3908  021334  030055  021440
 3909  021340  043045  020103  044502  MSG19:  .ASCII  /%FC BITS 15-0 #/
 3910  021346  051524  030440  026465
 3911  021354  020060     043
 3912  021357     045  052506  020116  MSG20:  .ASCII  /%FUN CODE BITS 5-1 OF C1 #/
 3913  021364  047503  042504  041040
 3914  021372  052111  020123  026465
 3915  021400  020061  043117  041440
 3916  021406  020061     043
 3917  021411     045  047507  041040  MSG21:  .ASCII  /%GO BIT NOT CORRECT AT START #/
 3918  021416  052111  047040  052117
 3919  021424  041440  051117  042522
 3920  021432  052103  040440  020124
 3921  021440  052123  051101  020124
 3922  021446     043
 3923  021447     045  047507  041040  MSG22:  .ASCII  /%GO BIT NOT SET #/
 3924  021454  052111  047040  052117
 3925  021462  051440  052105  021440
 3926  021470  043445  020117  044502  MSG23:  .ASCII  /%GO BIT NOT RESET BY INIT #/
```

```
3927  021476  020124  047516  020124
3928  021504  042522  042523  020124
3929  021512  054502  044440  044516
3930  021520  020124     043
3931  021523     045  051104  020131  MSG24:  .ASCII  /%DRY NOT SET BY INIT #/
3932  021530  047516  020124  042523
3933  021536  020124  054502  044440
3934  021544  044516  020124     043
3935  021551     045  051104  020131  MSG25:  .ASCII  /%DRY NOT RESET BY GO=1#/
3936  021556  047516  020124  042522
3937  021564  042523  020124  054502
3938  021572  043440  036517  021461
3939  021600  042045  054522  047040  MSG25A: .ASCII  /%DRY NOT SET BY GO=0#/
3940  021606  052117  051440  052105
3941  021614  041040  020131  047507
3942  021622  030075     043
3943  021625     045  047516  044440  MSG26:  .ASCII  /%NO INTERRUPT RETURNED#/
3944  021632  052116  051105  052522
3945  021640  052120  051040  052105
3946  021646  051125  042516  021504
3947  021654  041045  042101  051440  MSG27:  .ASCII  /%BAD STATUS#/
3948  021662  040524  052524  021523
3949  021670  051445  035116  021440  MSG30:  .ASCII  /%SN: #/
3950  021676  042445  051122  047040  MSG31:  .ASCII  /%ERR NOT SET #/
3951  021704  052117  051440  052105
3952  021712  021440
3953  021714  040445  040524  047040  MSG32:  .ASCII  /%ATA NOT SET #/
3954  021722  052117  051440  052105
3955  021730  021440
3956  021732  040445  020123  044502  MSG33:  .ASCII  /%AS BIT NOT SET #/
3957  021740  020124  047516  020124
3958  021746  042523  020124     043
3959  021753     045  041523  047040  MSG34:  .ASCII  /%SC NOT SET #/
3960  021760  052117  051440  052105
3961  021766  021440
3962  021770  052045  042522  047040  MSG35:  .ASCII  /%TRE NOT SET #/
3963  021776  052117  051440  052105
3964  022004  021440
3965  022006  051445  040514  047040  MSG36:  .ASCII  /%SLA NOT SET #/
3966  022014  052117  051440  052105
3967  022022  021440
3968  022024  051445  041523  047040  MSG37:  .ASCII  /%SSC NOT SET #/
3969  022032  052117  051440  052105
3970  022040  021440
3971  022042  037440  021440          MSG40:  .ASCII  / ? #/
3972  022046  022445  047105  020104  MSG41:  .ASCII  /%%END OF PASS #/
3973  022054  043117  050040  051501
3974  022062  020123     043
3975  022065     045  042504  042101  MSG42:  .ASCII  /%DEAD TRACK #/
3976  022072  052040  040522  045503
3977  022100  021440
3978  022102  022445  040515  052516  MSG43:  .ASCII  /%%MANUAL TESTS (14-17) INHIBITED: HALT%/
3979  022110  046101  052040  051505
3980  022116  051524  024040  032061
3981  022124  030455  024467  044440
3982  022132  044116  041111  052111
```

```
 3983  022140  042105  020072  040510
 3984  022146  052114     045
 3985  022151     122  051505  046105          .ASCII  /RESELECT AND PRESS CONTINUE%#/
 3986  022156  041505  020124  047101
 3987  022164  020104  051120  051505
 3988  022172  020123  047503  052116
 3989  022200  047111  042525  021445
 3990  022206  051045  043505  051511  MSG44:  .ASCII  /%REGISTER START: #/
 3991  022214  042524  020122  052123
 3992  022222  051101  035124  021440
 3993  022230  053045  041505  047524  MSG45:  .ASCII  /%VECTOR ADDRESS: #/
 3994  022236  020122  042101  051104
 3995  022244  051505  035123  021440
 3996  022252  041445  030523  020040  MSG46:  .ASCII  /%CS1    WC     BA     FC     CS2    DS     ER     AS/
 3997  022260  020040  041527  020040
 3998  022266  020040  041040  020101
 3999  022274  020040  020040  041506
 4000  022302  020040  020040  041440
 4001  022310  031123  020040  020040
 4002  022316  051504  020040  020040
 4003  022324  042440  020122  020040
 4004  022332  020040  051501
 4005  022336  020040  020040  046440          .ASCII  /     MR     TC%#/
 4006  022344  020122  020040  020040
 4007  022352  041524  021445
 4008  022356  047045  052117  051040  MSG47:  .ASCII  /%NOT RESET BY DRIVE CLEAR#/
 4009  022364  051505  052105  041040
 4010  022372  020131  051104  053111
 4011  022400  020105  046103  040505
 4012  022406  021522
 4013  022410  040445  050114  040510  MSG50:  .ASCII  /%ALPHA NOT SET#/
 4014  022416  047040  052117  051440
 4015  022424  052105     043
 4016  022427     045  047125  054105  MSG51:  .ASCII  /%UNEXPECTED ERROR BITS#/
 4017  022434  042520  052103  042105
 4018  022442  042440  051122  051117
 4019  022450  041040  052111  021523
 4020  022456  041045  042101  046040  MSG53:  .ASCII  /%BAD LRC #/
 4021  022464  041522  021440
 4022  022470  041045  042101  041440  MSG54:  .ASCII  /%BAD CK #/
 4023  022476  020113     043
 4024  022501     045  042523  052524  MSG55:  .ASCII  /%SETUP ERROR: CHECK WRAP 0 WITH TEST 50#/
 4025  022506  020120  051105  047522
 4026  022514  035122  041440  042510
 4027  022522  045503  053440  040522
 4028  022530  020120  020060  044527
 4029  022536  044124  052040  051505
 4030  022544  020124  030065     043
 4031  022551     045  052123  052101  MSG56:  .ASCII  /%STATIC TESTS ONLY: #/
 4032  022556  041511  052040  051505
 4033  022564  051524  047440  046116
 4034  022572  035131  021440
 4035  022576  052045  047515  020063  MSG57:  .ASCII  /%TM03 DRIVE: #/
 4036  022604  051104  053111  035105
 4037  022612  021440
 4038  022614  052045  032125  020065  MSG58:  .ASCII  /%TU45 SLAVE: #/
```

```
 4039  022622  046123  053101  035105
 4040  022630  021440
 4041  022632  021445                  MSG59:  .ASCII  /%#/
 4042  022634  021534                  MSG60:  .ASCII  / #/
 4043  022636  051045  046505  053117  MSG62:  .ASCII  /%REMOVE TMDP FROM SLAVE TO BE TESTED%#/
 4044  022644  020105  046524  050104
 4045  022652  043040  047522  020115
 4046  022660  046123  053101  020105
 4047  022666  047524  041040  020105
 4048  022674  042524  052123  042105
 4049  022702  021445
 4050  022704  044045  051101  053504  MSG63:  .ASCII  /%HARDWARE SWR IN USE%#/
 4051  022712  051101  020105  053523
 4052  022720  020122  047111  052440
 4053  022726  042523  021445
```

```
                                        ;TEST HEADER**********
                                        
022732  022445  047514  044507  MSLT1:  .ASCII  /%%LOGIC TEST 1: DRIVE ADDRESSING (M8939 RH)#/
022740  020103  042524  052123
022746  030440  020072  051104
022754  053111  020105  042101
022762  051104  051505  044523
022770  043516  024040  034115
022776  031471  020071  044122
023004  021451
023006  022445  047514  044507  MSLT2:  .ASCII  /%%LOGIC TEST 2: REGISTER ADDRESSING (M8939 RH)#/
023014  020103  042524  052123
023022  031040  020072  042522
023030  044507  052123  051105
023036  040440  042104  042522
023044  051523  047111  020107
023052  046450  034470  034463
023060  051040  024510     043
023065     045  046045  043517  MSLT3:  .ASCII  /%%LOGIC TEST 3: CONTROL BUS TEST (RH M8905-YB M8939)#/
023072  041511  052040  051505
023100  020124  035063  041440
023106  047117  051124  046117
023114  041040  051525  052040
023122  051505  020124  051050
023130  020110  034115  030071
023136  026465  041131  046440
023144  034470  034463  021451
023152  022445  047514  044507  MSLT4:  .ASCII  /%%LOGIC TEST 4: SLAVE ADDRESSING (M8905-YB M8933)#/
023160  020103  042524  052123
023166  032040  020072  046123
023174  053101  020105  042101
023202  051104  051505  044523
023210  043516  024040  034115
023216  030071  026465  041131
023224  046440  034470  031463
023232  021451
023234  022445  047514  044507  MSLT5:  .ASCII  /%%LOGIC TEST 5: MR BIT TEST (M8905-YB)#/
023242  020103  042524  052123
023250  032440  020072  051115
023256  041040  052111  052040
023264  051505  020124  046450
023272  034470  032460  054455
023300  024502     043
023303     045  046045  043517  MSLT6:  .ASCII  /%%LOGIC TEST 6: TC BIT TEST (M8905-YB)#/
023310  041511  052040  051505
023316  020124  035066  052040
023324  020103  044502  020124
023332  042524  052123  024040
023340  034115  030071  026465
023346  041131  021451
023352  022445  047514  044507  MSLT7:  .ASCII  /%%LOGIC TEST 7: FC BIT TEST (M8905-YB)#/
023360  020103  042524  052123
023366  033440  020072  041506
023374  041040  052111  052040
023402  051505  020124  046450
023410  034470  032460  054455
```

```
 4110  023416  024502     043
 4111  023421     045  046045  043517  MSLT10: .ASCII  /%%LOGIC TEST 10: FUNCTION BIT TEST (M8905-YB)#/
 4112  023426  041511  052040  051505
 4113  023434  020124  030061  020072
 4114  023442  052506  041516  044524
 4115  023450  047117  041040  052111
 4116  023456  052040  051505  020124
 4117  023464  046450  034470  032460
 4118  023472  054455  024502     043
 4119  023477     045  046045  043517  MSLT11: .ASCII  /%%LOGIC TEST 11: GO BIT TEST (M8939)#/
 4120  023504  041511  052040  051505
 4121  023512  020124  030461  020072
 4122  023520  047507  041040  052111
 4123  023526  052040  051505  020124
 4124  023534  046450  034470  034463
 4125  023542  021451
 4126  023544  022445  047514  044507  MSLT12: .ASCII  /%%LOGIC TEST 12: DRIVE READY BIT (M8939)#/
 4127  023552  020103  042524  052123
 4128  023560  030440  035062  042040
 4129  023566  044522  042526  051040
 4130  023574  040505  054504  041040
 4131  023602  052111  024040  034115
 4132  023610  031471  024471     043
 4133  023615     045  046045  043517  MSLT13: .ASCII  /%%LOGIC TEST 13: INTERRUPT TEST (RH)#/
 4134  023622  041511  052040  051505
 4135  023630  020124  031461  020072
 4136  023636  047111  042524  051122
 4137  023644  050125  020124  042524
 4138  023652  052123  024040  044122
 4139  023660  021451
 4140  023662  022445  047514  044507  MSLT14: .ASCII  /%%LOGIC TEST 14: MANUAL STATUS TEST 1#/
 4141  023670  020103  042524  052123
 4142  023676  030440  035064  046440
 4143  023704  047101  040525  020114
 4144  023712  052123  052101  051525
 4145  023720  052040  051505  020124
 4146  023726  021461
 4147  023730  022445  047514  044507  MSLT15: .ASCII  /%%LOGIC TEST 15: MANUAL STATUS TEST 2#/
 4148  023736  020103  042524  052123
 4149  023744  030440  035065  046440
 4150  023752  047101  040525  020114
 4151  023760  052123  052101  051525
 4152  023766  052040  051505  020124
 4153  023774  021462
 4154  023776  022445  047514  044507  MSLT16: .ASCII  /%%LOGIC TEST 16: MANUAL STATUS TEST 3#/
 4155  024004  020103  042524  052123
 4156  024012  030440  035066  046440
 4157  024020  047101  040525  020114
 4158  024026  052123  052101  051525
 4159  024034  052040  051505  020124
 4160  024042  021463
 4161  024044  022445  047514  044507  MSLT17: .ASCII  /%%LOGIC TEST 17: MANUAL STATUS TEST 4#/
 4162  024052  020103  042524  052123
 4163  024060  030440  035067  046440
 4164  024066  047101  040525  020114
 4165  024074  052123  052101  051525
```

```
 4166  024102  052040  051505  020124
 4167  024110  021464
 4168  024112  022445  047514  044507  MSLT20: .ASCII  /%%LOGIC TEST 20: ILLEGAL FUNCTION TEST (M8939)#/
 4169  024120  020103  042524  052123
 4170  024126  031040  035060  044440
 4171  024134  046114  043505  046101
 4172  024142  043040  047125  052103
 4173  024150  047511  020116  042524
 4174  024156  052123  024040  034115
 4175  024164  031471  024471     043
 4176  024171     045  046045  043517  MSLT21: .ASCII  /%%LOGIC TEST 21: RMR(M8939)#/
 4177  024176  041511  052040  051505
 4178  024204  020124  030462  020072
 4179  024212  046522  024122  034115
 4180  024220  031471  024471     043
 4181  024225     045  046045  043517  MSLT22: .ASCII  /%%LOGIC TEST 22: CPAR(M8939)#/
 4182  024232  041511  052040  051505
 4183  024240  020124  031062  020072
 4184  024246  050103  051101  046450
 4185  024254  034470  034463  021451
 4186  024262  022445  047514  044507  MSLT23: .ASCII  /%%LOGIC TEST 23: FMT(M8905-YB M8906)#/
 4187  024270  020103  042524  052123
 4188  024276  031040  035063  043040
 4189  024304  052115  046450  034470
 4190  024312  032460  054455  020102
 4191  024320  034115  030071  024466
 4192  024326     043
 4193  024327     045  046045  043517  MSLT24: .ASCII  /%%LOGIC TEST 24: DPAR(M8906 RH)#/
 4194  024334  041511  052040  051505
 4195  024342  020124  032062  020072
 4196  024350  050104  051101  046450
 4197  024356  034470  033060  051040
 4198  024364  024510     043
 4199  024367     045  046045  043517  MSLT25: .ASCII  /%%LOGIC TEST 25: NEF(M8939)#/
 4200  024374  041511  052040  051505
 4201  024402  020124  032462  020072
 4202  024410  042516  024106  034115
 4203  024416  031471  024471     043
 4204  024423     045  046045  043517  MSLT26: .ASCII  /%%LOGIC TEST 26: FCE(M8939)#/
 4205  024430  041511  052040  051505
 4206  024436  020124  033062  020072
 4207  024444  041506  024105  034115
 4208  024452  031471  024471     043
 4209  024457     045  046045  043517  MSLT27: .ASCII  /%%LOGIC TEST 27: ILR(M8939)#/
 4210  024464  041511  052040  051505
 4211  024472  020124  033462  020072
 4212  024500  046111  024122  034115
 4213  024506  031471  024471     043
 4214  024513     045  046045  043517  MSLT30: .ASCII  /%%LOGIC TEST 30:DTE(M8906 RH)#/
 4215  024520  041511  052040  051505
 4216  024526  020124  030063  042072
 4217  024534  042524  046450  034470
 4218  024542  033060  051040  024510
 4219  024550     043
 4220  024551     045  046045  043517  MSLT31: .ASCII  /%%LOGIC TEST 31: OPI(M8933)#/
 4221  024556  041511  052040  051505
```

```
 4222  024564  020124  030463  020072
 4223  024572  050117  024111  034115
 4224  024600  031471  024463     043
 4225  024605     045  046045  043517  MSLT32: .ASCII  /%%LOGIC TEST 32: UNS(M8939)#/
 4226  024612  041511  052040  051505
 4227  024620  020124  031063  020072
 4228  024626  047125  024123  034115
 4229  024634  031471  024471     043
 4230  024641     045  046045  043517  MSLT33: .ASCII  /%%LOGIC TEST 33: PIP(M8939)#/
 4231  024646  041511  052040  051505
 4232  024654  020124  031463  020072
 4233  024662  044520  024120  034115
 4234  024670  031471  024471     043
 4235  024675     045  046045  043517  MSLT34: .ASCII  /%%LOGIC TEST 34: PES(M8928)#/
 4236  024702  041511  052040  051505
 4237  024710  020124  032063  020072
 4238  024716  042520  024123  034115
 4239  024724  031071  024470     043
 4240  024731     045  046045  043517  MSLT35: .ASCII  /%%LOGIC TEST 35: SAC(M8933 M8905-YB)#/
 4241  024736  041511  052040  051505
 4242  024744  020124  032463  020072
 4243  024752  040523  024103  034115
 4244  024760  031471  020063  034115
 4245  024766  030071  026465  041131
 4246  024774  021451
 4247  024776  022445  047514  044507  MSLT36: .ASCII  /%%LOGIC TEST 36: FCS(M8933 M8905-YB)#/
 4248  025004  020103  042524  052123
 4249  025012  031440  035066  043040
 4250  025020  051503  046450  034470
 4251  025026  031463  046440  034470
 4252  025034  032460  054455  024502
 4253  025042     043
 4254  025043     045  046045  043517  MSLT37: .ASCII  /%%LOGIC TEST 37: ACCL(M8933 M8905-YB)#/
 4255  025050  041511  052040  051505
 4256  025056  020124  033463  020072
 4257  025064  041501  046103  046450
 4258  025072  034470  031463  046440
 4259  025100  034470  032460  054455
 4260  025106  024502     043
 4261  025111     045  046045  043517  MSLT40: .ASCII  /%%LOGIC TEST 40: PE TAPE MARK(M8932)#/
 4262  025116  041511  052040  051505
 4263  025124  020124  030064  020072
 4264  025132  042520  052040  050101
 4265  025140  020105  040515  045522
 4266  025146  046450  034470  031063
 4267  025154  021451
 4268  025156  022445  047514  044507  MSLT41: .ASCII  /%%LOGIC TEST 41: NRZ TAPE MARK (M8934)#/
 4269  025164  020103  042524  052123
 4270  025172  032040  035061  047040
 4271  025200  055122  052040  050101
 4272  025206  020105  040515  045522
 4273  025214  024040  034115  031471
 4274  025222  024464     043
 4275  025225     045  046045  043517  MSLT42: .ASCII  /%%LOGIC TEST 42: CRC(M8934)#/
 4276  025232  041511  052040  051505
 4277  025240  020124  031064  020072
```

```
 4278  025246  051103  024103  034115
 4279  025254  031471  024464     043
 4280  025261     045  046045  043517  MSLT43: .ASCII  /%%LOGIC TEST 43: LRC(M8934)#/
 4281  025266  041511  052040  051505
 4282  025274  020124  031464  020072
 4283  025302  051114  024103  034115
 4284  025310  031471  024464     043
 4285  025315     045  046045  043517  MSLT44: .ASCII  /%%LOGIC TEST 44: CORRECTABLE DATA (M8932 M8901)#/
 4286  025322  041511  052040  051505
 4287  025330  020124  032064  020072
 4288  025336  047503  051122  041505
 4289  025344  040524  046102  020105
 4290  025352  040504  040524  024040
 4291  025360  034115  031471  020062
 4292  025366  034115  030071  024461
 4293  025374     043
 4294  025375     045  046045  043517  MSLT45: .ASCII  /%%LOGIC TEST 45: INCORRECTABLE DATA (M8932 M8934)#/
 4295  025402  041511  052040  051505
 4296  025410  020124  032464  020072
 4297  025416  047111  047503  051122
 4298  025424  041505  040524  046102
 4299  025432  020105  040504  040524
 4300  025440  024040  034115  031471
 4301  025446  020062  034115  031471
 4302  025454  024464     043
 4303  025457     045  046045  043517  MSLT46: .ASCII  /%%LOGIC TEST 46: PEF(M8932)#/
 4304  025464  041511  052040  051505
 4305  025472  020124  033064  020072
 4306  025500  042520  024106  034115
 4307  025506  031471  024462     043
 4308  025513     045  046045  043517  MSLT47: .ASCII  /%%LOGIC TEST 47: FC OVERFLOW (M8905-YB)#/
 4309  025520  041511  052040  051505
 4310  025526  020124  033464  020072
 4311  025534  041506  047440  042526
 4312  025542  043122  047514  020127
 4313  025550  046450  034470  032460
 4314  025556  054455  024502     043
 4315  025563     045  046045  043517  MSLT50: .ASCII  /%%LOGIC TEST 50: NEF WHEN WRITE PE ON NRZ SLAVE#/
 4316  025570  041511  052040  051505
 4317  025576  020124  030065  020072
 4318  025604  042516  020106  044127
 4319  025612  047105  053440  044522
 4320  025620  042524  050040  020105
 4321  025626  047117  047040  055122
 4322  025634  051440  040514  042526
 4323  025642     043
 4324  025643     045  046045  043517  MSLT51: .ASCII  /%%LOGIC TEST 51: NEF WHEN WRITE NRZ ON PE SLAVE#/
 4325  025650  041511  052040  051505
 4326  025656  020124  030465  020072
 4327  025664  042516  020106  044127
 4328  025672  047105  053440  044522
 4329  025700  042524  047040  055122
 4330  025706  047440  020116  042520
 4331  025714  051440  040514  042526
 4332  025722     043
```

```
 4333
 4334                                        ;MANUAL INSTRUCTION**********
 4335
 4336  025723     045  054524  042520  MMSG0:  .ASCII  /%TYPE CR WHEN READY;#/
 4337  025730  041440  020122  044127
 4338  025736  047105  051040  040505
 4339  025744  054504  021473
 4340  025750  022445  047515  047125  MMSG1:  .ASCII  /%%MOUNT TAPE WITH NO WRITE RING, LOAD TO BOT, SET TO ON LINE:#/
 4341  025756  020124  040524  042520
 4342  025764  053440  052111  020110
 4343  025772  047516  053440  044522
 4344  026000  042524  051040  047111
 4345  026006  026107  046040  040517
 4346  026014  020104  047524  041040
 4347  026022  052117  020054  042523
 4348  026030  020124  047524  047440
 4349  026036  020116  044514  042516
 4350  026044  021472
 4351  026046  051445  052105  052040  MMSG2:  .ASCII  /%SET TO OFFLINE:#/
 4352  026054  020117  043117  046106
 4353  026062  047111  035105     043
 4354  026067     045  047515  042526  MMSG3:  .ASCII  /%MOVE FORWARD TO EOT, USING THE TU45 MAINTENENCE SWITCH, SET ONLINE:#/
 4355  026074  043040  051117  040527
 4356  026102  042122  052040  020117
 4357  026110  047505  026124  052440
 4358  026116  044523  043516  052040
 4359  026124  042510  052040  032125
 4360  026132  020065  040515  047111
 4361  026140  042524  042516  041516
 4362  026146  020105  053523  052111
 4363  026154  044103  020054  042523
 4364  026162  020124  047117  044514
 4365  026170  042516  021472
 4366  026174  051045  053505  047111  MMSG4:  .ASCII  /%REWIND AND UNLOAD TAPE, REINSTALL WRITE RING, AND/
 4367  026202  020104  047101  020104
 4368  026210  047125  047514  042101
 4369  026216  052040  050101  026105
 4370  026224  051040  044505  051516
 4371  026232  040524  046114  053440
 4372  026240  044522  042524  051040
 4373  026246  047111  026107  040440
 4374  026254  042116
 4375  026256  046445  053117  020105          .ASCII  /%MOVE JUST FORWARD OF BOT USING THE TU45 MAINTENENCE SWITCH,/
 4376  026264  052512  052123  043040
 4377  026272  051117  040527  042122
 4378  026300  047440  020106  047502
 4379  026306  020124  051525  047111
 4380  026314  020107  044124  020105
 4381  026322  052524  032464  046440
 4382  026330  044501  052116  047105
 4383  026336  047105  042503  051440
 4384  026344  044527  041524  026110
 4385  026352  040445  042116  051440          .ASCII  /%AND SET TO ONLINE#/
 4386  026360  052105  052040  020117
 4387  026366  047117  044514  042516
 4388  026374     043
```

```
 4389  026375     045  046445  053117  MMSG5:  .ASCII  /%%MOVE TAPE TO BOT; ON LINE#/
 4390  026402  020105  040524  042520
 4391  026410  052040  020117  047502
 4392  026416  035524  047440  020116
 4393  026424  044514  042516     043
```

```
4394
4395                                        ;TAG MESSAGE
4396
4397  026431     045  053523  020122  $MSWR:  .ASCII  /%SWR = #/
4398  026436  020075     043
4399  026441     040  042516  020127  $MNEW:  .ASCII  / NEW = #/
4400  026446  020075     043
4401  026451     045  046123  020101  TMS1:   .ASCII  /%SLA #/
4402  026456     043
4403  026457     045  047502  020124  TMS2:   .ASCII  /%BOT #/
4404  026464     043
4405  026465     045  046524  021440  TMS3:   .ASCII  /%TM #/
4406  026472  044445  041104  021440  TMS4:   .ASCII  /%IDB #/
4407  026500  051445  053504  020116  TMS5:   .ASCII  /%SDWN #/
4408  026506     043
4409  026507     045  042520  020123  TMS6:   .ASCII  /%PES #/
4410  026514     043
4411  026515     045  051523  020103  TMS7:   .ASCII  /%SSC #/
4412  026522     043
4413  026523     045  051104  020131  TMS8:   .ASCII  /%DRY #/
4414  026530     043
4415  026531     045  050104  020122  TMS9:   .ASCII  /%DPR #/
4416  026536     043
4417  026537     045  052116  020114  TMS10:  .ASCII  /%NTL #/
4418  026544     043
4419  026545     045  047505  020124  TMS11:  .ASCII  /%EOT #/
4420  026552     043
4421  026553     045  051127  020114  TMS12:  .ASCII  /%WRL #/
4422  026560     043
4423  026561     045  047515  020114  TMS13:  .ASCII  /%MOL #/
4424  026566     043
4425  026567     045  044520  020120  TMS14:  .ASCII  /%PIP #/
4426  026574     043
4427  026575     045  051105  020122  TMS15:  .ASCII  /%ERR #/
4428  026602     043
4429  026603     045  052101  020101  TMS16:  .ASCII  /%ATA #/
4430  026610     043
4431  026611     045  046111  020106  TMS17:  .ASCII  /%ILF #/
4432  026616     043
4433  026617     045  046111  020122  TMS18:  .ASCII  /%ILR #/
4434  026624     043
4435  026625     045  046522  020122  TMS19:  .ASCII  /%RMR #/
4436  026632     043
4437  026633     045  050103  051101  TMS20:  .ASCII  /%CPAR #/
4438  026640  021440
4439  026642  043045  052115  021440  TMS21:  .ASCII  /%FMT #/
4440  026650  042045  040520  020122  TMS22:  .ASCII  /%DPAR #/
4441  026656     043
4442  026657     045  047111  020103  TMS23:  .ASCII  /%INC #/
4443  026664     043
4444  026665     045  050126  020105  TMS24:  .ASCII  /%VPE #/
4445  026672     043
4446  026673     045  042520  020106  TMS25:  .ASCII  /%PEF #/
4447  026700     043
4448  026701     045  051114  020103  TMS26:  .ASCII  /%LRC #/
4449  026706     043
```

```
 4450  026707     045  051516  020107  TMS27:  .ASCII  /%NSG #/
 4451  026714     043
 4452  026715     045  041506  020105  TMS28:  .ASCII  /%FCE #/
 4453  026722     043
 4454  026723     045  051503  021440  TMS29:  .ASCII  /%CS #/
 4455  026730  044445  046524  021440  TMS30:  .ASCII  /%ITM #/
 4456  026736  047045  043105  021440  TMS31:  .ASCII  /%NEF #/
 4457  026744  042045  042524  021440  TMS32:  .ASCII  /%DTE #/
 4458  026752  047445  044520  021440  TMS33:  .ASCII  /%OPI #/
 4459  026760  053445  044522  042524  TMS33A: .ASCII  /%WRITE OPI #/
 4460  026766  047440  044520  021440
 4461  026774  051045  040505  020104  TMS33B: .ASCII  /%READ OPI #/
 4462  027002  050117  020111     043
 4463  027007     040  041517  052503  TMS33C: .ASCII  / OCCURED TO SOON%#/
 4464  027014  042522  020104  047524
 4465  027022  051440  047517  022516
 4466  027030     043
 4467  027031     040  041517  052503  TMS33D: .ASCII  / OCCURRED TO LATE%#/
 4468  027036  051122  042105  052040
 4469  027044  020117  040514  042524
 4470  027052  021445
 4471  027054  043040  044501  042514  TMS33E: .ASCII  / FAILED TO SET%#/
 4472  027062  020104  047524  051440
 4473  027070  052105  021445
 4474  027074  052445  051516  021440  TMS34:  .ASCII  /%UNS #/
 4475  027102  041445  051117  020122  TMS35:  .ASCII  /%CORR #/
 4476  027110     043
 4477  027111     045  051103  020103  TMS36:  .ASCII  /%CRC #/
 4478  027116     043
 4479  027117     045  040523  020103  TMS37:  .ASCII  /%SAC #/
 4480  027124     043
 4481  027125     045  041506  020123  TMS38:  .ASCII  /%FCS #/
 4482  027132     043
 4483  027133     045  041501  046103  TMS39:  .ASCII  /%ACCL #/
 4484  027140  021440
 4485
 4486                                          .EVEN
 4487                                          ;WRITE BUFFER
 4488
 4489  027142  000100                  WDATA:
 4490  027142  177777                          -1
 4491  027144  177777                          -1
 4492  027146  177777                          -1
 4493  027150  177777                          -1
 4494  027152  177777                          -1
 4495  027154  177777                          -1
 4496  027156  177777                          -1
 4497  027160  177777                          -1
 4498  027162  177777                          -1
 4499  027164  177777                          -1
 4500  027166  177777                          -1
 4501  027170  177777                          -1
 4502  027172  177777                          -1
 4503  027174  177777                          -1
 4504  027176  177777                          -1
 4505  027200  177777                          -1
```

```
 4506  027202  177777                    -1
 4507  027204  177777                    -1
 4508  027206  177777                    -1
 4509  027210  177777                    -1
 4510  027212  177777                    -1
 4511  027214  177777                    -1
 4512  027216  177777                    -1
 4513  027220  177777                    -1
 4514  027222  177777                    -1
 4515  027224  177777                    -1
 4516  027226  177777                    -1
 4517  027230  177777                    -1
 4518  027232  177777                    -1
 4519  027234  177777                    -1
 4520  027236  177777                    -1
 4521  027240  177777                    -1
 4522  027242  177777                    -1
 4523  027244  177777                    -1
 4524  027246  177777                    -1
 4525  027250  177777                    -1
 4526  027252  177777                    -1
 4527  027254  177777                    -1
 4528  027256  177777                    -1
 4529  027260  177777                    -1
 4530  027262  177777                    -1
 4531  027264  177777                    -1
 4532  027266  177777                    -1
 4533  027270  177777                    -1
 4534  027272  177777                    -1
 4535  027274  177777                    -1
 4536  027276  177777                    -1
 4537  027300  177777                    -1
 4538  027302  177777                    -1
 4539  027304  177777                    -1
 4540  027306  177777                    -1
 4541  027310  177777                    -1
 4542  027312  177777                    -1
 4543  027314  177777                    -1
 4544  027316  177777                    -1
 4545  027320  177777                    -1
 4546  027322  177777                    -1
 4547  027324  177777                    -1
 4548  027326  177777                    -1
 4549  027330  177777                    -1
 4550  027332  177777                    -1
 4551  027334  177777                    -1
 4552  027336  177777                    -1
 4553  027340  177777                    -1
 4554
 4555
 4556                                    ;READ BUFFER
 4557
 4558  027342  000100            RDATA:
 4559  027342  000000                    0
 4560  027344  000000                    0
 4561  027346  000000                    0
```

```
 4562  027350  000000                    0
 4563  027352  000000                    0
 4564  027354  000000                    0
 4565  027356  000000                    0
 4566  027360  000000                    0
 4567  027362  000000                    0
 4568  027364  000000                    0
 4569  027366  000000                    0
 4570  027370  000000                    0
 4571  027372  000000                    0
 4572  027374  000000                    0
 4573  027376  000000                    0
 4574  027400  000000                    0
 4575  027402  000000                    0
 4576  027404  000000                    0
 4577  027406  000000                    0
 4578  027410  000000                    0
 4579  027412  000000                    0
 4580  027414  000000                    0
 4581  027416  000000                    0
 4582  027420  000000                    0
 4583  027422  000000                    0
 4584  027424  000000                    0
 4585  027426  000000                    0
 4586  027430  000000                    0
 4587  027432  000000                    0
 4588  027434  000000                    0
 4589  027436  000000                    0
 4590  027440  000000                    0
 4591  027442  000000                    0
 4592  027444  000000                    0
 4593  027446  000000                    0
 4594  027450  000000                    0
 4595  027452  000000                    0
 4596  027454  000000                    0
 4597  027456  000000                    0
 4598  027460  000000                    0
 4599  027462  000000                    0
 4600  027464  000000                    0
 4601  027466  000000                    0
 4602  027470  000000                    0
 4603  027472  000000                    0
 4604  027474  000000                    0
 4605  027476  000000                    0
 4606  027500  000000                    0
 4607  027502  000000                    0
 4608  027504  000000                    0
 4609  027506  000000                    0
 4610  027510  000000                    0
 4611  027512  000000                    0
 4612  027514  000000                    0
 4613  027516  000000                    0
 4614  027520  000000                    0
 4615  027522  000000                    0
 4616  027524  000000                    0
 4617  027526  000000                    0
```

```
 4618  027530  000000                          0
 4619  027532  000000                          0
 4620  027534  000000                          0
 4621  027536  000000                          0
 4622  027540  000000                          0
 4623
 4624                                          ;WRAP AROUND MESSAGES******
 4625
 4626  027542  042045  052101  020101  WMSG27: .ASCII  /%DATA PAT:#/
 4627  027550  040520  035124    043
 4628  027555    045   047505  020122  WMSG31: .ASCII  /%EOR CLEAR DID NOT CLEAR GO%#/
 4629  027562  046103  040505  020122
 4630  027570  044504  020104  047516
 4631  027576  020124  046103  040505
 4632  027604  020122  047507  021445
 4633
 4634                                          .EVEN
 4635  027612  000000                  PRE:    0
 4636  027614  000000                          0
 4637  027616  000000                          0
 4638  027620  000000                          0
 4639  027622  000000                          0
 4640  027624  000000                          0
 4641  027626  000000                          0
 4642  027630  000000                          0
 4643  027632  000000                          0
 4644  027634  000000                          0
 4645  027636  000000                          0
 4646  027640  000000                          0
 4647  027642  000000                          0
 4648  027644  000000                          0
 4649  027646  000000                          0
 4650  027650  000000                          0
 4651  027652  000000                          0
 4652  027654  000000                          0
 4653  027656  000000                          0
 4654  027660  000000                          0
 4655  027662  000000                          0
 4656  027664  000000                          0
 4657  027666  000000                          0
 4658  027670  000000                          0
 4659  027672  000000                          0
 4660  027674  000000                          0
 4661  027676  000000                          0
 4662  027700  000000                          0
 4663  027702  000000                          0
 4664  027704  000000                          0
 4665  027706  000000                          0
 4666  027710  000000                          0
 4667  027712  000000                          0
 4668  027714  000000                          0
 4669  027716  000000                          0
 4670  027720  000000                          0
 4671  027722  000000                          0
 4672  027724  000000                          0
 4673  027726  000000                          0
```

```
 4674  027730  000000                          0
 4675  027732  000000                          0
 4676  027734  000000                POST:     0
 4677  027736  000000                          0
 4678  027740  000000                          0
 4679  027742  000000                          0
 4680  027744  000000                          0
 4681  027746  000000                          0
 4682  027750  000000                          0
 4683  027752  000000                          0
 4684  027754  000000                          0
 4685  027756  000000                          0
 4686  027760  000000                          0
 4687  027762  000000                          0
 4688  027764  000000                          0
 4689  027766  000000                          0
 4690  027770  000000                          0
 4691  027772  000000                          0
 4692  027774  000000                          0
 4693  027776  000000                          0
 4694  030000  000000                          0
 4695  030002  000000                          0
 4696  030004  000000                          0
 4697  030006  000000                          0
 4698  030010  000000                          0
 4699  030012  000000                          0
 4700  030014  000000                          0
 4701  030016  000000                          0
 4702  030020  000000                          0
 4703  030022  000000                          0
 4704  030024  000000                          0
 4705  030026  000000                          0
 4706  030030  000000                          0
 4707  030032  000000                          0
 4708  030034  000000                          0
 4709  030036  000000                          0
 4710  030040  000000                          0
 4711  030042  000000                          0
 4712  030044  000000                          0
 4713  030046  000000                          0
 4714  030050  000000                          0
 4715  030052  000000                          0
 4716  030054  000000                          0
 4717  030056  000000              WBUFF:      0
 4718          030470                          .=.+410
 4719  030470  000000              RBUFF:      0
 4720
 4721          000001                          .END
```

ADDFL	000746	GTSWR	020110	LT10E1	004426	LT2LP	003216	LT30C	007660
AS	000526	HDRFL	000620	LT10IT	004336	LT2X	003222	LT30D	007676
ASF	000730	HERE	002512	LT10X	004450	LT20	005662	LT30E	007726
ATAF	000720	ILFT	000544	LT11	004460	LT20A	005710	LT30IT	007476
ATTA	015620	INIT	016422	LT11B	004506	LT20B	005766	LT30X	007752
ATTB	015640	INIT1	016344	LT11C	004546	LT20C	005776	LT31	007772
ATTC	015660	INIT2	016354	LT11E1	004566	LT20IT	005676	LT31A	010150
ATTD	015666	INIT3	016372	LT11E2	004616	LT20X	006010	LT31IT	010000
ATTN	015600	INIT4	016406	LT11E3	004646	LT21	006024	LT31X	010254
ATTP	015676	INIT5	016426	LT11IT	004462	LT21A	006130	LT32	010604
ATTPA	015724	INMT	016324	LT11X	004674	LT21B	006140	LT32IT	010620
ATTX	015750	INST	016606	LT12	004704	LT21IT	006040	LT32X	010732
ATTXA	015760	ITAMT	000606	LT12B	004730	LT21XA	006150	LT32XX	010742
ATTXX	015776	ITCNT	000702	LT12C	004766	LT22	006170	LT33	010746
BA	000514	ITER	016254	LT12E1	005004	LT22A	006270	LT33IT	010762
CC	000530	ITRLP	000714	LT12E2	005026	LT22IT	006204	LT33X	011040
CHNFLG	001416	LTADD	000742	LT12E3	005050	LT22X	006300	LT34	011050
CRCNT	000774	LTGA	014722	LT12IT	004706	LT23	006314	LT34A	011110
CS	000520	LTGA1	014740	LT12X	005070	LT23A	006414	LT34A1	011070
C1	000510	LTGA2	014772	LT13	005100	LT23IT	006330	LT34B	011140
DATAD	000760	LTGB	015020	LT13A	005140	LT23X	006424	LT34C	011144
DATC	000754	LTGC	015024	LT13E1	005144	LT24	006444	LT34IT	011064
DB	000532	LTGC0	015112	LT13IT	005110	LT24B	006572	LT34X	011174
DCA	015376	LTGD	015116	LT13X	005172	LT24B0	006614	LT34XX	011200
DCA0	015412	LTGER	014704	LT14	005202	LT24C	006626	LT35	011204
DCB	015450	LTGER0	014676	LT14A	005224	LT24D	006710	LT35A	011306
DCC	015456	LTGER1	016012	LT14IT	005242	LT24IT	006460	LT35IT	011220
DCD	015352	LTGER2	016004	LT14X	005306	LT24X	006734	LT35X	011346
DCE	015466	LTGER3	014664	LT14XX	005312	LT25	006770	LT36	011356
DCEA	015516	LTGX	015146	LT15	005316	LT25A	007076	LT36IT	011372
DCEX	015536	LTGXA	015156	LT15A	005340	LT25IT	006776	LT36X	011466
DCEXA	015546	LTGXX	015174	LT15IT	005356	LT25X	007106	LT37	011476
DCEXX	015564	LTG1A	016030	LT15X	005422	LT26	007122	LT37A	011552
DERFL	000766	LTG1B	016046	LT15XX	005426	LT26IT	007130	LT37B	011604
DOUT	017704	LTG1C	016156	LT16	005432	LT26X	007322	LT37IT	011512
DOUTD	017762	LTG1T	016126	LT16A	005454	LT27	007336	LT37X	011636
DRVCLR	015344	LTG1X	016162	LT16IT	005472	LT27A	007372	LT4	003414
DRVN	000624	LTG1XX	016212	LT16X	005536	LT27B	007430	LT4A	003532
DRVTP	000604	LTG1X1	016174	LT16XX	005542	LT27IT	007362	LT4B	003574
DS	000522	LT1	002552	LT17	005546	LT27X	007440	LT4C	003602
DT	000536	LT1A	002676	LT17A	005570	LT27XX	007450	LT4D	003632
EMADDR	000622	LT1B	002744	LT17IT	005606	LT3	003232	LT4ERG	003660
EORPA	014530	LT1C	002752	LT17X	005652	LT3A	003252	LT4ER1	003642
EORPX	014656	LT1ER	002762	LT17XX	005656	LT3B	003272	LT4ER2	003652
EORP1	014564	LT1ER1	002772	LT2	003026	LT3C	003306	LT4G	003462
EORP1A	014632	LT1ER2	003000	LT2A	003066	LT3ER1	003316	LT4G0	003452
EORP2	014654	LT1G	002620	LT2B	003106	LT3ER2	003344	LT4X	003710
ER	000524	LT1G0	002610	LT2C	003122	LT3IT	003234	LT40	011646
ERADD	000672	LT1X	003022	LT2ERG	003202	LT3X	003366	LT40IT	011662
ERRF	000726	LT10	004334	LT2ER1	003132	LT3XX	003404	LT40X	011756
EXFL	000716	LT10A	004352	LT2ER2	003150	LT30	007454	LT40XX	011762
FC	000516	LT10A1	004350	LT2ER3	003166	LT30A	007556	LT41	011766
FUN	000752	LT10B	004406	LT2IT	003030	LT30B	007620	LT41IT	012002

```
LT41X   012176          LT5     003714          MSG23   021470          MSLT21  024171          RBUFF   030470
LT42    012230          LT5A    003734          MSG24   021523          MSLT22  024225          RCDP    001010
LT42A   012332          LT5B    003754          MSG25   021551          MSLT23  024262          RDAD    000762
LT42B   012346          LT5C    003764          MSG25A  021600          MSLT24  024327          RDATA   027342
LT42B1  012374          LT5D    003774          MSG26   021625          MSLT25  024367          RDRVF   001006
LT42B2  012414          LT5E    004016          MSG27   021654          MSLT26  024423          REGP    015210
LT42C   012460          LT5ER1  004030          MSG3    020640          MSLT27  024457          REGS    000612
LT42D   012474          LT5ER2  004052          MSG30   021670          MSLT3   023065          RTRN    000670
LT42E   012524          LT5IT   003722          MSG31   021676          MSLT30  024513          SAV1    000704
LT42IT  012266          LT5X    004074          MSG32   021714          MSLT31  024551          SAV2    000706
LT42X   012544          LT50    014150          MSG33   021732          MSLT32  024605          SAV3    000710
LT43    012560          LT50IT  014164          MSG34   021753          MSLT33  024641          SCF     000732
LT43C   012660          LT50X   014310          MSG35   021770          MSLT34  024675          SCOLP   000712
LT43D   012674          LT51    014324          MSG36   022006          MSLT35  024731          SCOPE   016224
LT43E   012726          LT51IT  014340          MSG37   022024          MSLT36  024776          SERFL   000772
LT43F   012760          LT51X   014464          MSG4    020662          MSLT37  025043          SERNUM  000602
LT43IT  012606          LT6     004104          MSG40   022042          MSLT4   023152          SKAT    001014
LT43X   013000          LT6A    004122          MSG41   022046          MSLT40  025111          SLAF    000722
LT44    013010          LT6A1   004120          MSG42   022065          MSLT41  025156          SLVN    000664
LT44A   013074          LT6B    004126          MSG43   022102          MSLT42  025225          SN      000540
LT44A1  013106          LT6D    004146          MSG44   022206          MSLT43  025261          SNPG    020064
LT44B   013116          LT6ER1  004166          MSG45   022230          MSLT44  025315          SNPT    020006
LT44C   013132          LT6IT   004106          MSG46   022252          MSLT45  025375          SSCF    000724
LT44D   013160          LT6X    004210          MSG47   022356          MSLT46  025457          START   001330
LT44E   013162          LT7     004220          MSG5    020700          MSLT47  025513          STATC   001012
LT44F   013222          LT7A    004236          MSG50   022410          MSLT5   023234          STATF   001004
LT44IT  013036          LT7B    004256          MSG51   022427          MSLT50  025563          STATIC  014500
LT44X   013242          LT7C    004226          MSG53   022456          MSLT51  025643          STFLG   000740
LT44XX  013252          LT7ER1  004276          MSG54   022470          MSLT6   023303          STSCD   002402
LT45    013256          LT7IT   004222          MSG55   022501          MSLT7   023352          ST0     001662
LT45A   013364          LT7X    004320          MSG56   022551          MTINT   016654          ST1     001704
LT45B   013400          MMSG0   025723          MSG57   022576          NRZOF   000662          ST2     002066
LT45D   013430          MMSG1   025750          MSG58   022614          NXTDRV  002152          SWR     000570
LT45E   013342          MMSG2   026046          MSG59   022632          NXTSLV  002224          SWREG   000176
LT45E1  013354          MMSG3   026067          MSG6    020720          OCTP    017462          TADX    001324
LT45F   013462          MMSG4   026174          MSG60   022634          OCTPE   017452          TC      000542
LT45IT  013304          MMSG5   026375          MSG62   022636          OCTPE1  017466          TEMP1   000674
LT45X   013506          MR      000534          MSG63   022704          OCTPG   017640          TEMP2   000676
LT45XX  013516          MSG1    020252          MSG7    020737          OCTPG0  017656          TEMP3   000700
LT46    013522          MSG10   021167          MSG8    020755          OCTPG1  017662          TEND    002436
LT46A   013604          MSG11   021213          MSG8A   021016          OCTP0   017506          TENDX   002534
LT46B   013652          MSG12   021240          MSG9    021145          OCTP1   017526          TEX     017450
LT46C   013666          MSG13   021247          MSLT1   022732          OCTP2   017534          TIB     000616
LT46D   013720          MSG14   021256          MSLT10  023421          OCTP3   017624          TIMER   010370
LT46IT  013550          MSG15   021273          MSLT11  023477          OFL     017702          TIMER0  010414
LT46X   013744          MSG16   021311          MSLT12  023544          PATRN   001002          TIMER1  010344
LT46XX  013754          MSG18   021321          MSLT13  023615          PCNTR   001016          TIMOK   010462
LT47    013760          MSG19   021340          MSLT14  023662          PEXFL   000736          TIMON   010276
LT47A   014050          MSG2    020450          MSLT15  023730          PFLG    000666          TIMOVF  010426
LT47B   014062          MSG2A   020511          MSLT16  023776          POST    027734          TINER   017252
LT47C   014066          MSG20   021357          MSLT17  024044          PRE     027612          TKB     000574
LT47IT  013774          MSG21   021411          MSLT2   023006          PREFL   000770          TKS     000572
LT47X   014140          MSG22   021447          MSLT20  024112          PSW     000566          TLAST   001326
```

```
TMS1    026451          TMS28   026715          TMS6    026507          TR13    000654          WCDP0   001042
TMS10   026537          TMS29   026723          TMS7    026515          TR14    000656          WCDP2   001030
TMS11   026545          TMS3    026465          TMS8    026523          TR15    000660          WCS1    001020
TMS12   026553          TMS30   026730          TMS9    026531          TSCD    002106          WCS2    001022
TMS13   026561          TMS31   026736          TOB     000614          TSCDA   002302          WDATA   027142
TMS14   026567          TMS32   026744          TOG     017434          TSCD0   002310          WDS     001024
TMS15   026575          TMS33   026752          TPB     000600          TSCD1   002316          WER     001026
TMS16   026603          TMS33A  026760          TPS     000576          TSCD2   002344          WMSG27  027542
TMS17   026611          TMS33B  026774          TREF    000734          TSCD3   002362          WMSG31  027555
TMS18   026617          TMS33C  027007          TR00    000626          TSTTBL  001054          WPGFL   001000
TMS19   026625          TMS33D  027031          TR01    000630          TTIN    017272          WTAD    000756
TMS2    026457          TMS33E  027054          TR02    000632          TTINT   016664          W2FLG   000764
TMS20   026633          TMS34   027074          TR03    000634          TTOUT   017334          $DONE   002444
TMS21   026642          TMS35   027102          TR04    000636          TTR     017012          $ENDAD  002502
TMS22   026650          TMS36   027111          TR05    000640          T24FL   000744          $MNEW   026441
TMS23   026657          TMS37   027117          TR06    000642          UDES    000776          $MSWR   026431
TMS24   026665          TMS38   027125          TR07    000644          VECT    000610          $SVPC =  000764
TMS25   026673          TMS39   027133          TR10    000646          WAM     000750          .      =  030472
TMS26   026701          TMS4    026472          TR11    000650          WBUFF   030056          .RESTO  020230
TMS27   026707          TMS5    026500          TR12    000652          WC      000512          .SAVE   020206


. ABS.  030472     000

 ERRORS DETECTED:  0

 ,CZTUOA.SEQ/SOL_CZTUOA.P11
 RUN-TIME: 34 61  4 SECONDS
 RUN-TIME RATIO: 181/99=1.8
 CORE USED:  14K  (28 PAGES)
```